ANALISIS KERENTANAN KEAMANAN WEBSITE SISTEM MANAJEMEN ABSENSI KARYAWAN GO DESIGN MENGGUNAKAN METODE OWASP

Fathoni, Adzka Fahmi Aulia Hakim, Rifko Akbar,
Muhammad Alfarizi Ramadiyansa, Moh Rizky Sinaga
Sistem Informasi, Universitas Sriwijaya
Jl. Raya Palembang - Prabumulih Km. 32 Indralaya, Ogan Ilir, Sumatera Selatan
fathoni@unsri.ac.id

ABSTRAK

Sistem Manajemen Absensi Karyawan GO!Design, yang dibangun dengan framework PHP Laravel, dirancang untuk mempermudah pengelolaan absensi karyawan dan administrasi oleh perusahaan. Namun, seiring dengan meningkatnya ancaman terhadap keamanan aplikasi web, muncul kebutuhan untuk mengevaluasi potensi kerentanannya. Tujuan dari analisis ini adalah untuk mengidentifikasi dan menilai kerentanan pada sistem tersebut guna meningkatkan keamanan. Metode yang digunakan melibatkan OWASP *Risk Rating Methodology* dan alat OWASP ZAP untuk mengumpulkan data risiko, menilai kemungkinan eksploitasi, dan mengevaluasi dampak dari setiap kerentanan yang ditemukan. Hasil pengujian mengungkapkan adanya 11 celah keamanan, termasuk masalah pada Access Control, SQL Injection, dan Cross-Site Scripting (XSS), yang memiliki tingkat keparahan mulai dari rendah hingga tinggi. Berdasarkan temuan tersebut, diharapkan untuk meningkatkan kontrol akses, mengenkripsi data sensitif, dan menerapkan header keamanan guna memperkuat perlindungan terhadap aplikasi dari potensi ancaman yang ada.

***Kata kunci**: OWASP Risk Methodology, OWASP ZAP, Vulnerability Assesment, Severity of Risk*

1. PENDAHULUAN

Sistem informasi merupakan kesatuan terintegrasi yang terdiri dari berbagai komponen seperti manusia, perangkat keras, perangkat lunak, jaringan, dan prosedur yang berfungsi untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi guna mendukung pengambilan keputusan dalam suatu organisasi [1]. Dalam penerapannya, banyak organisasi kini memanfaatkan sistem informasi berbasis web untuk meningkatkan efisiensi dan efektivitas operasional, termasuk dalam manajemen sumber daya manusia [2], [3].

GO!design merupakan perusahaan penyedia jasa desain grafis profesional yang berbasis di Denpasar, Bali. Perusahaan ini mendukung berbagai aktivitas promosi produk dan layanan dari klien-kliennya melalui desain yang menarik dan terjangkau. Untuk mendukung operasional internal, GO!design memanfaatkan sistem manajemen absensi berbasis website yang dibangun menggunakan framework PHP Laravel. Sistem ini dilengkapi dengan fitur *geolocation* untuk presensi dan absensi karyawan secara daring, serta mendukung proses manajemen cuti, pelaporan, dan rekapitulasi data kehadiran oleh pihak manajemen.

Namun, seiring dengan meningkatnya kebutuhan akan layanan digital, ancaman terhadap keamanan sistem web juga semakin kompleks [4]. Website yang terhubung langsung ke internet sangat rentan terhadap berbagai serangan seperti SQL *Injection*, *Cross-Site Scripting* (XSS), dan *Broken Authentication* jika tidak dilakukan pengujian keamanan secara berkala [5]. Dalam konteks ini, sistem manajemen absensi GO!design belum pernah dilakukan penilaian keamanan secara menyeluruh, sehingga potensi celah keamanan masih belum diketahui.

Keamanan aplikasi web menjadi salah satu aspek penting yang harus diperhatikan oleh pengembang sistem [6]. Menurut OWASP (*Open Web Application Security Project*), salah satu pendekatan untuk menganalisis risiko keamanan adalah dengan melakukan *Vulnerability Assessment* menggunakan OWASP *Risk Rating Methodology*, yang mengidentifikasi dan mengevaluasi potensi celah keamanan berdasarkan dampak dan kemungkinan eksploitasi [7].

Penelitian ini bertujuan untuk melakukan analisis kerentanan keamanan pada sistem manajemen absensi karyawan GO!design dengan menggunakan metode OWASP dan memanfaatkan OWASP ZAP sebagai alat bantu utama. Hasil analisis ini diharapkan dapat menjadi dasar dalam menyusun rekomendasi perbaikan untuk meningkatkan keamanan sistem informasi tersebut.

2. TINJAUAN PUSTAKA

Pada tahap ini dilakukan tinjauan pustaka agar mendapatkan landasan pemikiran yang mendalam untuk menunjang penelitian. Tinjauan ini tidak hanya mencakup teori-teori dasar yang digunakan dalam analisis keamanan aplikasi web, tetapi juga merujuk pada penelitian-penelitian terdahulu yang relevan dengan topik yang dibahas. Melalui referensi dari buku, artikel, jurnal, dan penelitian sebelumnya diharapkan dapat memberikan pemahaman yang lebih komperhensif tentang metode dan alat yang digunakan, serta penerapan serupa yang telah dilakukan.

2.1. *Owasp Risk Methodology*

OWASP *Risk Methodology* merupakan metode yang memungkinkan untuk memperkirakan tingkat risiko yang mungkin akan terjadi pada website dan cara menanggulanginya [3].

Dalam melakukan perhitungan risiko, terdapat beberapa tahapan diantaranya adalah[8]:

a. *Identifying Risk* : Tahap mengidentifikasi kebutuhan untuk mengumpulkan informasi dari website. Dalam kasus ini identifikasi dilakukan menggunakan tools OWASP ZAP
b. *Factors for Estimating Likelihood* : Tahap menghitung likelihood. Pada tahapan ini perhitungan dilakukan dengan menghitung Threat Agent Factor dan Vulnerability Factor
c. *Factors for Estimating Impact* : Tahap menentukan dampak yang dialami ketika serangan sukses. Terdapat dua dampak yaitu Technical Impact dan Business Impact
d. *Determining Severity of the Risk*: Tahap menentukan tingkat likelihood, impact, dan risk yang terjadi. Tahap ini dapat dilakukan dengan dua metode yaitu Informal Method atau Repeatable Method.
e. *Deciding to Fix* : Tahap menentukan bagian yang harus diperbaiki dari objek pengujian. Dalam penelitian kali tahap Deciding To Fix tidak dilakukan karena ruang lingkup penelitian hanya sebatas mendeteksi
f. Customizing the Risk Rating Model: Tahap optional, perubahan bentuk Risk Rating Model dilakukan disini.

Penelitian-penelitian terdahulu memberikan wawasan berharga mengenai penerapan metode ini. Diantaranya, penelitian oleh Riandhanu [9] melakukan analisis kerentanan pada website absensi perusahaan X menggunakan metode OWASP. Hasil pengujian menunjukkan alat pemindai kerentanan yang digunakan dapat mengetahui tingkat keamanan suatu aplikasi. Berdasarkan hasil pengujian tersebut maka perlu diiringi tingkat keamanan yang baik, sehingga diperlukan upaya untuk mengamankan website tersebut baik dari sisi infrastruktur dan aplikasi.

Selain itu, penelitian oleh Fata [10] melakukan pengujian penetrasi terhadap situs web Asosiasi Pekerja Profesional Informasi Sekolah Indonesia. Ditemukan 41 kerentanan yang dapat dieksploitasi, dengan 2 diantaranya dalam kategori kritis dan 9 dalam kategori tinggi, yang memerlukan penanganan segera.

Penelitian-penelitian tersebut menunjukkan penerapan praktis OWASP *Risk Rating Methodology* dalam menilai dan mengelola kerentanan pada aplikasi berbasis web.

2.2. *OWASP ZAP*

OWASP ZAP adalah alat gratis open-source yang membantu dalam mencari kelemahan keamanan website [11]. ZAP juga memungkinkan Anda untuk membuat skrip sendiri, seperti untuk menangani proses login saat melakukan scanning[12]. Tampilan antarmuka pengguna ZAP mudah digunakan, sehingga dapat dengan mudah memindai website dan mengatasi masalah keamanannya.

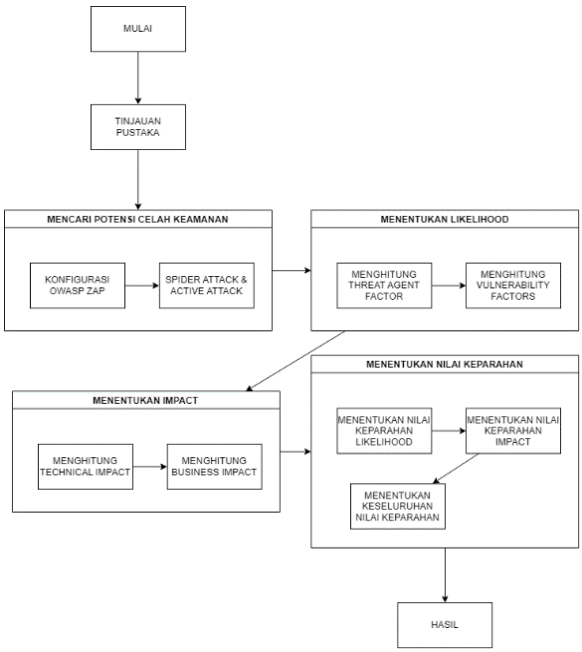
Berbagai studi sebelumnya telah menerapkan OWASP ZAP dalam menganalisis kerentanannya pada aplikasi berbasis web. Salah satunya, penelitian oleh Priambodo [4] melakukan penetration testing pada website XYZ dengan menggunakan metode black box dan alat OWASP ZAP. Hasilnya ditemukan 9 jenis kerentanan dengan tingkat keparahan bervariasi, mulai dari tinggi hingga rendah.

Penelitian lainnya oleh Kusuma [13] mengimpementasikan OWASP ZAP untuk pengujian keamanan sistem informasi akademik. Pengujian menemukan kerentanan dengan tingkat keparahan sedang, rendah, informatif, seperti HTML Form tanpa perlindungan CSRF, *Clickjacking*, dan input *password* dengan *auto-complete*.

Dengan demikian, OWASP ZAP terbukti efektif dalam membantu pengujian dan identifikasi kerentanan, serta alat yang berguna bagi pengembang dan profesional keamanan dalam meningkatkan ketahanan aplikasi berbasis web terhadap potensi ancaman.

3. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah metode *OWASP Risk Methodology*. Dengan memanfaatkan OWASP ZAP sebagai pengumpul data. Berikut adalah alur penelitian ini.



Gambar 1. Alur Penelitian

Berikut adalah penjelasan alur penelitian dari setiap tahapan berdasarkan diagram diatas:

a. Mulai : penelitian dimulai dengan tahap tinjauan pustaka, di mana peneliti melakukan pencarian referensi dan studi literatur terkait dengan topik

yang dibahas, yaitu analisis kerentanan keamanan pada website dengan menggunakan metode *OWASP Risk Rating Methodology*.

b. Mencari potensi celah keamanan :
   - Konfigurasi OWASP ZAP: pengaturan alat OWASP ZAP pada browser untuk memulai analisis.
   - *Spider Attack* & *Active Attack: Spider Attack* digunakan untuk mengumpulkan informasi tentang *website*, sementara *Active Attack* digunakan untuk mengeksplorasi lebih dalam celah-celah keamanan yang mungkin ada di *website* tersebut.

c. Menentukan *likelihood* : Likelihood merujuk pada kemungkinan terjadinya eksploitasi dari kerentanannya[12]. Perhitungan ini bertujuan untuk mengukur seberapa besar kemungkinan suatu kerentanan ditemukan dan dieksploitasi oleh penyerang[11]. Likelihood dihitung dengan mempertimbangkan dua faktor utama, yaitu Threat Agent Factors dan Vulnerability Factors.
   - *Threat Agent Factors* mengukur kemampuan kelompok penyerang dalam mengeksploitasi kerentanan[11]. Berikut rumus yang digunakan untuk menentukan nilai *threat agent :*

$$Threat\ Agent = \frac{Skill\ Level + Motive + Opportunity + Size}{4}$$

     Threat Agent Factors terdiri dari empat atribut utama yang menggambarkan karakteristik penyerang. Empat atribut ini adalah:
     - *Skill Level* : Ini mengukur tingkat keterampilan teknis yang diperlukan oleh penyerang untuk mengeksploitasi kerentanan. Nilainya bisa berkisar antara 1 (tidak perlu keterampilan teknis) hingga 9 (memerlukan keterampilan penetrasi keamanan).
     - *Motive* : Faktor ini mengukur seberapa besar motivasi penyerang untuk mengeksploitasi kerentanan, baik untuk keuntungan pribadi maupun tujuan lain. Penilaian dilakukan dengan memberi nilai antara 1 (reward rendah atau tidak ada) hingga 9 (reward tinggi).
     - *Opportunity* : Menilai seberapa besar sumber daya dan akses yang dibutuhkan penyerang untuk mengeksploitasi kerentanan. Nilai berkisar antara 0 (membutuhkan akses penuh atau sumber daya mahal) hingga 9 (tidak memerlukan akses atau sumber daya khusus).
     - *Size* : Mengukur seberapa besar lingkup kelompok penyerang yang dapat mengeksploitasi kerentanan ini. Penilaian dilakukan dengan memberikan nilai antara 2 (pengembang atau admin sistem) hingga 9 (pengguna internet anonim).
   - Vulnerability Factors mengukur kemudahan kerentanan tersebut untuk ditemukan dan dieksploitasi [11]. Berikut rumus yang digunakan untuk menghitung nilai *vulnerability factors* :

$$Vulnerability = \frac{Discovery + Exploity + Opportunity + Size}{4}$$

     Faktor-faktor yang mempengaruhi kerentanannya ini mencakup empat atribut berikut:
     - *Ease of Discovery* : Ini mengukur seberapa mudah bagi penyerang untuk menemukan kerentanannya. Penilaian dilakukan dengan nilai antara 1 (mustahil) hingga 9 (sudah tersedia alat untuk menemukannya).
     - *Ease of Exploit* : Menilai seberapa mudah penyerang dapat mengeksploitasi kerentanan tersebut setelah ditemukan. Nilai diberikan antara 1 (eksploitasi teoretis) hingga 9 (sudah tersedia alat eksploitasi).
     - *Awareness* : Menilai seberapa terkenal kerentanan ini di kalangan penyerang. Jika kerentanannya sudah menjadi pengetahuan umum, nilai diberikan 9. Sebaliknya, jika kerentanannya tidak diketahui, diberi nilai 1.
     - *Intrusion* Detection : Mengukur kemungkinan bahwa eksploitasi dari kerentanan akan terdeteksi. Jika eksploitasi terdeteksi secara aktif dalam aplikasi, diberikan nilai 2. Jika tidak terdeteksi sama sekali, diberi nilai 9.

d. Menentukan *impact* : Impact dilakukan untuk mengukur dampak yang ditimbulkan ketika suatu kerentanan ditemukan dan berhasil dieksploitasi oleh penyerang [14]. Tujuan dari perhitungan impact adalah untuk menentukan sejauh mana eksploitasi terhadap kerentanan dapat merugikan sistem secara teknis maupun bisnis[11] . Dalam konteks ini, impact dibagi menjadi dua kategori utama yaitu, *technical impact* dan *business impact*.
   - *Technical Impact* berfokus pada dampak teknis yang diakibatkan oleh eksploitasi kerentanan, yang mempengaruhi sistem atau produk itu sendiri [11]. Berikut rumus yang digunakan untuk menentukan nilai *technical impact:*

$$Technical\ Impact = \frac{confidentiality + integrity + availability + acountability}{4}$$

     Berdasarkan rumus tersebut berikut atribut dalam Technical Impact meliputi.
     - *Loss of Confidentiality* : Ini mengukur seberapa banyak data yang dapat terekspos akibat serangan, serta seberapa sensitif data tersebut. Misalnya, apakah data pribadi atau informasi penting dapat diakses oleh pihak yang tidak berwenang.
     - *Loss of Integrity* : Faktor ini mengukur sejauh mana data yang ada dapat dirusak atau dimodifikasi oleh penyerang. Dampaknya bisa berupa perubahan pada

data yang membuatnya tidak valid atau tidak dapat dipercaya.

- *Loss of Availability* : Ini mengukur seberapa banyak layanan yang terpengaruh atau hilang akibat serangan. Sebagai contoh, apakah layanan utama yang sangat penting dalam organisasi berhenti beroperasi atau tidak dapat diakses.
- *Loss of Accountability* : Faktor ini berhubungan dengan apakah tindakan yang dilakukan oleh penyerang dapat dilacak kembali ke individu yang bertanggung jawab. Kehilangan akuntabilitas dapat mempersulit pihak berwenang untuk menindaklanjuti dan mencegah ancaman lebih lanjut.

- *Business Impact* mengukur dampak yang ditimbulkan terhadap aspek bisnis organisasi yang disebabkan oleh eksploitasi kerentanan [11]. Berikut rumus untuk menghitung *business impact*.

$$Business\ impact = \frac{financial\ damage + reputation\ damage + non - compliance + p.violation}{4}$$

Berdasarkan rumus tersebut berikut faktor-faktor dalam *business impact* mencakup.

- *Financial Damage* : Faktor ini menilai kerugian finansial yang terjadi akibat serangan. Ini bisa berupa biaya yang dibutuhkan untuk perbaikan celah keamanan atau dampak terhadap pendapatan tahunan perusahaan.
- *Reputation Damage* : Kerusakan reputasi mengukur seberapa besar dampak serangan terhadap citra perusahaan. Jika serangan mempengaruhi persepsi klien atau publik, hal ini dapat menyebabkan hilangnya kepercayaan dan reputasi perusahaan yang sulit dipulihkan.
- *Non-compliance* : Faktor ini melihat sejauh mana ketidakpatuhan terhadap peraturan atau kebijakan internal perusahaan disebabkan oleh serangan. Misalnya, jika data pelanggan terekspos, itu bisa menyebabkan perusahaan melanggar hukum atau regulasi terkait perlindungan data.
- *Privacy Violation* : Mengukur seberapa banyak informasi pribadi individu yang terekspos akibat serangan. Semakin besar jumlah individu yang terpengaruh, semakin serius dampaknya terhadap bisnis dari segi reputasi dan hukum.

e. Menentukan *Severity of risk* : bertujuan untuk menilai keparahan dampak yang ditimbulkan oleh celah keamanan yang ditemukan. Setelah melakukan perhitungan Likelihood (kemungkinan eksploitasi) dan Impact (dampak yang ditimbulkan), tahap ini berfokus pada menggabungkan kedua faktor tersebut untuk menentukan tingkat keparahan risiko [11].

Tabel 1. Indikator tingkat keparahan

0 to < 3	LOW
3 to < 6	MEDIUM
6 to 9	HIGH

f. Hasil : Langkah terakhir adalah untuk menyimpulkan hasil dari seluruh analisis yang telah dilakukan, termasuk penentuan tingkat keparahan risiko dan memberikan rekomendasi perbaikan yang perlu diterapkan untuk mengurangi potensi ancaman pada sistem[15].

4. HASIL DAN PEMBAHASAN

4.1. *IDENTIFYING RISK*

Sebelum melakukan pencarian celah keamanan, penulis memanfaatkan beberapa alat untuk mendukung proses pengujian keamanan. Pada tabel 2 merupakan spesifikasi alat penelitian yang digunakan dalam pengujian.

Tabel 2. Alat Penelitian

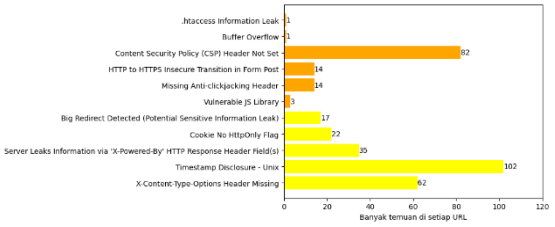
Nama Alat	Spesifikasi/Keterangan
Mozila Firefox	Tools yang digunakan sebagai web browser
OWASP ZAP	Tools yang digunakan untuk melakukan Vulnerability Assesment
FoxyProxy	Tools yang digunakan untuk mengatur proxy pada web browser
Sistem Operasi	Windows 11 Home SL 64 bit

4.2. *Konfigurasi OWASP ZAP*

Proses ini dilakukan dengan menggunakan tool OWASP ZAP yang telah dikonfigurasi pada web browser Firefox melalui FoxyProxy. Berikut adalah tampilan dari Sistem Manajemen Absensi Karyawan yang telah terhubung dengan OWASP ZAP. Setelah koneksi berhasil, peneliti melakukan Serangan Spider untuk mengumpulkan informasi mengenai sumber daya website. Selanjutnya, peneliti melaksanakan Serangan Active Scan pada setiap URL yang diperoleh dari serangan Spider dengan tujuan untuk mendeteksi potensi celah keamanan.

4.3. *Spider Attack & Active Attack*

Dari hasil *Active Scan*, ditemukan 11 peringatan atau *Alerts* mengenai kerentanannya. Sebanyak 6 peringatan memiliki tingkat risiko medium (ditandai dengan warna oranye), dan 5 peringatan lainnya berisiko rendah (ditandai dengan warna kuning).



Gambar 3. Peringatan dengan tingkat Medium dan Low.

4.4. Estimating Likelihood

Hasil yang diperoleh berdasarkan pembobotan skor Likelihood dan Impact untuk menentukan skor risiko keseluruhan dihitung dengan menggunakan Risk Rating Calculator. Dalam proses ini, skor Likelihood dihitung dengan mempertimbangkan faktor-faktor seperti Threat Agent Factors dan Vulnerability Factors, yang mengukur kemungkinan kerentanan tersebut dapat ditemukan dan dieksploitasi oleh penyerang.

4.5. Threat Agent Factors

Berikut adalah hasil perhitungan Threat Agent Factors yang menunjukkan penilaian terhadap potensi ancaman yang dapat mengeksploitasi celah keamanan dalam sistem.

Table 3. Hasil perhitungan *threat agent factors*

Index	Skill Level	Motive	Opportunity	Size	Threat Agent
.htaccess Information Leak	6.0	4.0	7.0	2.0	4.75
Buffer Overflow	9.0	1.0	9.0	6.0	6.25
Content Security Policy (CSP) Header Not Set	6.0	4.0	4.0	2.0	4.0
HTTP to HTTPS Insecure Transition in Form Post	9.0	9.0	0	9.0	6.75
Missing Anti-clickjacking Header	9.0	9.0	0	9.0	6.75
Vulnerable JS Library	6.0	1.0	4.0	2.0	3.25
Big Redirect Detected (Potential Sensitive Information Leak)	5.0	9.0	9.0	9.0	8.0
Cookie No HttpOnly Flag	9.0	4.0	4.0	2.0	4.75
Server Leaks Information via 'X-Powered-By' HTTP Response Header Field(s)	9.0	9.0	9.0	4.0	7.75
Timestamp Disclosure - Unix	6.0	1.0	9.0	2.0	4.5
X-Content-Type-Options Header Missing	6.0	4.0	4.0	6.0	5.0
Total Threat Agent Factor	7.2727...	5.0	5.3636...	4.8181.	5.6136...

4.6. Vulnerability Factors

Berikut merupakan hasil perhitungan Vulnerability Factors, yang digunakan untuk menilai kemudahan sebuah celah keamanan ditemukan dan dieksploitasi oleh penyerang.

Table 4. Hasil perhitungan *vulnerability factors*

Indeks	Ease Of Discover	Ease Of Exploit	Awareness	Intrusion Detection	Vulnerability Factor
.htaccess Information Leak	9.0	5.0	4.0	9.0	6.75
Buffer Overflow	9.0	5.0	6.0	1.0	5.25
Content Security Policy (CSP) Header Not Set	9.0	5.0	9.0	3.0	6.5
HTTP to HTTPS Insecure Transition in Form Post	9.0	3.0	4.0	9.0	6.25
Missing Anti-clickjacking Header	9.0	9.0	6.0	8.0	8.0
Vulnerable JS Library	9.0	5.0	4.0	9.0	6.75
Big Redirect Detected (Potential Sensitive Information Leak)	9.0	5.0	9.0	3.0	6.5
Cookie No HttpOnly Flag	9.0	3.0	1.0	1.0	3.5
Server Leaks Information via 'X-Powered-By' HTTP Response Header Field(s)	9.0	9.0	6.0	9.0	8.25
Timestamp Disclosure - Unix	9.0	9.0	9.0	9.0	9.0
X-Content-Type-Options Header Missing	9.0	9.0	6.0	9.0	8.25
Total Vulnerability Factor	9.0	6.0909...	5.8181...	6.3636...	6.8181...

4.7. Estimating Impact

Berikut merupakan hasil perhitungan Impact, yang digunakan untuk menilai dampak yang mungkin timbul jika celah keamanan berhasil dieksploitasi, baik dari sisi teknis (seperti kerusakan data atau gangguan layanan) maupun dampak bisnis (seperti kerugian finansial atau kerusakan reputasi).

4.8. Technical Impact

Berikut merupakan hasil perhitungan Technical Impact, yang digunakan untuk menilai sejauh mana kerentanan dapat merusak sistem teknis, seperti integritas data, ketersediaan layanan, dan kerahasiaan informasi.

Tabel 5. Hasil perhitungan Technical Impact.

Indeks	Loss Of Confidentiality	Loss Of Integrity	Loss Of Availability	Loss Of Accountability	Technical Impact
.htaccess Information Leak	7.0	1.0	9.0	9.0	6.5
Buffer Overflow	2.0	1.0	9.0	7.0	4.75
Content Security Policy (CSP) Header Not Set	7.0	7.0	1.0	9.0	6.0
HTTP to HTTPS Insecure Transition in Form Post	9.0	9.0	9.0	9.0	9.0
Missing Anti-clickjacking Header	2.0	1.0	1.0	9.0	3.25
Vulnerable JS Library	6.0	3.0	1.0	9.0	4.75
Big Redirect Detected (Potential Sensitive Information Leak)	2.0	1.0	7.0	7.0	4.25
Cookie No HttpOnly Flag	6.0	3.0	1.0	9.0	4.75
Server Leaks Information via 'X-Powered-By' HTTP Response Header Field(s)	7.0	1.0	7.0	9.0	6.0
Timestamp Disclosure – Unix	2.0	1.0	1.0	1.0	1.25
X-Content-Type-Options Header Missing	7.0	7.0	7.0	9.0	7.5
Total Technical Impact	5.1818...	3.181...	4.818...	7.9090...	5.2727...

4.9. Business Impact

Berikut merupakan hasil perhitungan Business Impact, yang digunakan untuk menilai dampak yang ditimbulkan terhadap operasional dan reputasi bisnis, termasuk kerugian finansial, kerusakan reputasi, dan pelanggaran privasi.

Table 6. Hasil perhitungan *business impact*.

Indeks	Financial Damage	Reputation Damage	Non-Compliance	Privacy Violation	Business Impact
.htaccess Information Leak	1.0	1.0	5.0	3.0	2.5
Buffer Overflow	1.0	1.0	5.0	5.0	3.0
Content Security Policy (CSP) Header Not Set	1.0	1.0	5.0	3.0	2.5
HTTP to HTTPS Insecure Transition in Form Post	1.0	4.0	2.0	3.0	2.5
Missing Anti-clickjacking Header	1.0	1.0	2.0	3.0	1.75
Vulnerable JS Library	1.0	1.0	5.0	3.0	2.5
Big Redirect Detected (Potential Sensitive Information Leak)	1.0	5.0	2.0	3.0	2.75
Cookie No HttpOnly Flag	1.0	1.0	2.0	3.0	1.75
Server Leaks Information via 'X-Powered-By' HTTP Response Header Field(s)	1.0	1.0	2.0	3.0	1.75
Timestamp Disclosure - Unix	1.0	5.0	5.0	5.0	4.0
X-Content-Type-Options Header Missing	1.0	1.0	2.0	3.0	1.75
Total Business Impact	1.0	2.0	3.3636...	3.3636...	2.431818...

4.10. DETERMINING LIKELIHOOD

Perhitungan Likelihood dilakukan dengan persamaan(2).

$$Likehood = \frac{\Sigma Threat\ Agent\ Factors + \Sigma Vulnerability\ Factors}{2}$$

Keterangan:

Likelihood : Kemungkinan kerentanan untuk menjadi dieksploitasi oleh penyerang.

Threat Agent Factors : Rata-rata dari total rata-rata setiap opsi Threat Agent Factors

Vulnerability Factors : Rata-rata dari total rata-rata setiap opsi Vulnerability Factors.

Dengan menggunakan persamaan diatas maka didapatkan nilai dari Likelihood dan tingkatnya.

Tabel 7. Hasil perhitungan nilai dan tingkat keparahan Likelihood

Threat Agent Factors				**Vulnerability Factors**			
Skill Level	Motive	Opportunity	Size	Ease Of Discover	Ease Of Exploit	Awareness	Intrusion Detection
7.2727...	5.0	5.3636...	4.8181...	9.0	6.0909...	5.8181...	6.3636...
5.6136...				6.8181...			
Likelihood = 6.21585 (HIGH)							

4.11. DETERMINING IMPACT

Perhitungan Impact dapat dilakukan memilih salah satu antara Techincal Impact atau Business Impact. Apabila informasi bisnis tersedia dan lengkap, maka Business Impact menjadi opsi utama, apabila tidak tersedia dan tidak lengkap, maka Technical Impact menjadi opsi utama. Dalam hal ini informasi bisnis tersedia tapi tidak lengkap, maka peneliti hanya menggunakan Technical Impact untuk menentukan Impact.

Tabel 8. Hasil perhitungan nilai dan tingkat keparahan Impact

Technical Impact				**Business Impact**			
Loss Of Confidentiality	Loss Of Integrity	Loss Of Availability	Loss Of Accountability	Financial Damage	Reputation Damage	Non-Compliance	Privacy Violation
5.1818...	3.181...	4.818...	7.9090...	1.0	2.0	3.3636...	3.3636...
5.2727...(MEDIUM)				2.431818...(LOW)			

4.12. DETERMINING SEVERITY

Perhitungan nilai keparahan dapat dilakukan dengan menggabungkan hasil perhitungan Likelihood dan Impact yang kemudian dihitung menggunakan matriks nilai keparahan.

Tabel 8. Matriks penentuan keseluruan nilai keparahan

***If* Impact is**	***And* Likelihood is**	***Then* Risk Severity is**
Low	Low	Note
Low	Medium	Low
Low	High	Medium
Medium	Low	Low
Medium	Medium	Medium
Medium	High	High
High	Low	Medium
High	Medium	High
High	High	Critical

Berdasarkan perhitungan sebelumnya, Likelihood memiliki tingkat keparahan yang tinggi yaitu HIGH, sedangkan Impact memiliki tingkat keparahan MEDIUM, dengan demikian keseluruhan nilai keparahannya adalah HIGH berdasarkan perhitungan menggunakan matriks diatas.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan yang telah dilakukan, didapati bahwa website Sistem Manajemen Absensi Karyawan GO!Design memiliki 11 celah keamanan website yang memiliki risiko serangan. Celah keamanan tersebut kemudian dianalisis tingkat keparahannya, didapatkan bahwa Sistem Manajemen Absensi Karyawan GO!Design memiliki Likelihood sebesar 6.21585 yang secara indikator berarti HIGH dan memiliki Impact sebesar 5.2727 yang berarti MEDIUM.

Untuk menuju simpulan, hasil analisis tingkat keparahan tersebut digabungkan dan ditentukan keseluruhan tingkat keparahannya menggunakan matriks. Hasil dari perhitungan keseluruhan tingkat keparahan adalah HIGH yang berarti Sistem Manajemen Absensi Karyawan GO!Design memiliki celah keamanan yang rentan terkena serangan dan berdampak besar.

Adapun saran untuk penelitian selanjutnya, sebaiknya melakukan analisis sampai tahap perbaikan dan sebaiknya menggunakan metode yang lain untuk melakukan analisis kerentanan keamanan. Website yang dianalisis sebaiknya sudah dihosting. Untuk penelitian selanjutnya, sebaiknya berfokus pada perbandingan framewok yang digunakan untuk mengembangkan website, dalam hal ini saran peneliti adalah perbandingan framework PHP Laravel dan PHP Codeigniter4 atau Symfony.

DAFTAR PUSTAKA

[1] M. F. A. Ramadhan and A. S. Ilmananda, "ANALISIS ANCAMAN KEAMANAN PADA SISTEM INFORMASI AKADEMIK KAMPUS MENGGUNAKAN METODE OWASP ZAP," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 4, Aug. 2024.

[2] F. Rozy, F. Amalia, and R. C. Wihandika, "Pengembangan Sistem Informasi Manajemen dan Prediksi Permintaan Pemesanan Bibit Parfum Pada Toko Blossom Perfume Berbasis Web," 2021. [Online]. Available: http://j-ptiik.ub.ac.id

[3] A. Kerentanan Keamanan, W. Menggunakan, D. Aryanti, N. Dan, and J. N. Utamajaya, "METODE OWASP (OPEN WEB APPLICATION SECURITY PROJECT) PADA DINAS TENAGA KERJA," 2021.

[4] D. F. Priambodo, A. D. Rifansyah, and M. Hasbi, "Penetration Testing Web XYZ Berdasarkan OWASP Risk Rating," *Journal of Information and Communication Technology*, vol. 12, no. 1, Mar. 2023, doi: https://doi.org/10.34148/teknika.v12i1.571.

[5] E. Nurelasari, D. Gumilang, and A. Farabi, "ANALISIS KEAMANAN SISTEM WEBSITE MENGGUNAKAN METODE OPEN WEB APPLICATION SECURITY PROJECT (OWASP) PADA SIMANTEP.ID," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 3, 2024.

[6] A. Rohim and L. Setiyani, "Analisis Celah Keamanan E-Learning Perguruan Tinggi Menggunakan Vulnerability Assessment," *Jurnal Inovasi Pengembangan Aplikasi dan Keamanan Informasi Nusantara*, vol. 1, no. 1, pp. 1–10, Oct. 2023, Accessed: Apr. 23, 2025. [Online]. Available: https://journal.edunovationresearch.org/index.php/jipakif/article/view/2

[7] T. Sutabri, A. Wijaya, M. I. Herdiansyah, and E. S. Negara, "Evaluasi Risiko Celah Keamanan Aplikasi E-Office menggunakan Metode OWASP," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 1, pp. 113–122, Jun. 2024, doi: 10.29408/edumatic.v8i1.25463.

[8] J. Williams, "OWASP Risk Rating Methodology." Accessed: May 10, 2024. [Online]. Available: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

[9] I. O. Riandhanu, "Analisis Metode Open Web Application Security Project (OWASP) Menggunakan Penetration Testing pada Keamanan Website Absensi," *Jurnal Informasi dan Teknologi (JIdT)*, vol. 4, no. 3, pp. 160–165, 2022, doi: https://doi.org/10.37034/jidt.v4i3.236.

[10] D. Fata, "Evaluasi Risiko Celah Keamanan Menggunakan Metodologi Open Web Application Security Project (Owasp) Pada Aplikasi Web Sistem Informasi Akademik (Siakad) Uin Ar-Raniry," Banda Aceh, 2023. Accessed: Apr. 23, 2025. [Online]. Available: https://repository.ar-raniry.ac.id/id/eprint/31189/

[11] D. Wijayanto and A. Firdonsyah, "Analisis Tingkat Resiko Pada Website Xyz Menggunakan Metode Owasp," *Digital Transformation Technology*, vol. 4, no. 1, pp. 644–651, Aug. 2024, doi: 10.47709/digitech.v4i1.4485.

[12] A. F. Hasibuan, Tommy, and D. Handoko, "Analisis Kerentanan Website Dengan Aplikasi Owasp Zap," *Jurnal Ilmu Komputer dan Sistem Informasi (JIRSI)*, vol. 2, no. 2, pp. 257–270, 2023.

[13] G. H. A. Kusuma, "Implementasi OWASP ZAP untuk pengujian keamanan sistem informasi akademik," *Jurnal Teknologi Informasi : Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika*, vol. 16, no. 2, pp. 178–186, 2022, doi: ttps://doi.org/10.47111/jti.v16i2.3995.

[14] F. Putra Utama, R. Muhamad, and H. Nurhadi, "Uncovering the Risk of Academic Information System Vulnerability through PTES and OWASP Method," *CommIT Journal*, vol. 18, no. 1, pp. 39–51, 2024.

[15] S. A. Nugroho and T. Rochmadi, "Analisis Keamanan Sistem Informasi Pusaka Magelang Menggunakan Open Web Application Security Project (OWASP) Dan Information Systems Security Assessment Framework (ISSAF) Security Analysis Of Magelang Pusaka Information System Using Open Web Application Security Project (OWASP) And Information Systems Security Assessment Framework (ISSAF)," 2024. [Online]. Available: https://pusaka.magelangkab.go.id