ANALISIS ANCAMAN KEAMANAN PADA SISTEM INFORMASI AKADEMIK KAMPUS MENGGUNAKAN METODE OWASP ZAP

M. Fery Afrizal Ramadhan, Asri Samsiar Ilmananda
Faktultas Teknologi Informasi, Universitas Merdeka Malang
Jalan Terusan Dieng 62-64 Klojen, Pisang Candi, Kec, Sukun, Kota Malang, Jawa Timur, Indonesia
Ramadhanfery36@gmail.com

ABSTRAK

Penggunaan internet di Indonesia terus meningkat, dengan jumlah pengguna mencapai 221.563.479 jiwa pada tahun 2024. Perkembangan ini mendorong kemajuan di bidang teknologi informasi, yang menjadi sangat penting bagi mahasiswa dalam meningkatkan efektivitas waktu dan hasil akademik termasuk di Universitas Merdeka Malang. Sistem informasi akademik (SIAKAD) yang digunakan di Universitas Merdeka Malang, adalah salah satu alat penting berbasis *website* untuk mengelola data mahasiswa dan kegiatan akademik di Universitas Merdeka Malang. Namun, *website* ini juga rentan terhadap berbagai ancaman keamanan siber seperti *SQL injection, XSS, brute force*, dan *DDoS*. Penelitian ini menggunakan OWASP ZAP, alat yang diakui untuk analisis keamanan web, untuk mengidentifikasi celah keamanan pada Sistem Informasi Akademik (SIAKAD) Universitas Merdeka Malang. Hasil pengujian menunjukkan 17 kerentanan dengan distribusi: 17,65% (3) *High Risk Level*, 17,65% (3) *Medium Risk Level*, 29,41% (5) *Low Risk Level*, dan 35,29% (6) *Informational Risk Level*. Rekomendasi diberikan khususnya pada kerentanan dalam kategori OWASP A02 (*Cryptographic Failures*), A03 (*Injection*), A04 (*Insecure Design*), dan A05 (*Security Misconfiguration*). Diharapkan hasil ini dapat membantu staf TI Universitas Merdeka Malang dalam meningkatkan keamanan dan kenyamanan akses *website* sistem informasi akademik mereka.

Kata kunci : *OWASP, Penetrasi test, Analisis Acaman, ZAP*

1. PENDAHULUAN

Menurut data Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) mengumukan jumlah pengguna internet Indonesia tahun 2024 mencapai 221.563.479 jiwa dari total populasi 278.696.200 jiwa penduduk Indonesia tahun 2023. Semakin maju Indonesia dalam bidang teknologi dan informasi semakin cepat pula kita dalam mengakses sebuah informasi dengan cepat. Hal ini yang menjadi salah satu fasilitas terutama bagi seorang mahasiswa, penggunaan teknologi informasi adalah syarat utama untuk meningkatkan efektifitas waktu dan hasil yang terbaik [1].

Dalam dunia pendidikan modern Sistem informasi akademik adalah sebuah sistem yang digunakan oleh institusi pendidikan yang dimanfaatkan untuk meningkatkan pelayanan kepada mahasiswanya [2], sistem informasi akademik kampus berbentuk sebuah *website*. *Website* adalah kumpulan komponen yang terdiri dari teks, gambar, suara animasi merupakan media informasi yang menarik dan diminati digunakan sebagai media berbagai informasi [3]. Dan hal tersebut menjadi salah satu elemen kunci yang memfasilitasi berbagai aktivitas kegiatan akademik dan administratif di kampus tersebut [2]. *Website* tersebut tidak hanya menjadi sumber informasi penting bagi mahasiswa, dosen, dan staf administrasi, tetapi juga menjadi *platform* vital untuk pengelolaan data mahasiswa, jadwal kuliah, registrasi kursus, dan informasi lainnya yang berkaitan dengan kegiatan akademik [4].

Namun dengan kelebihan tersebut, *website* sistem informasi akademik kampus juga menghadapi berbagai ancaman keamanan yang kompleks dan berkembang pesat [5]. Di Universitas Merdeka Malang, website sistem informasi akademik (SIAKAD) yang digunakan sering mengalami ancaman yang sehingga website tidak dapat diakses. Kesalahan pada penulisan kode program dalam pembuatan aplikasi berbasis website sering dimanfaatkan oleh penyerang, dalam hal ini serangan yang sering dimanfaatkan oleh penyerang diantaranya serangan siber seperti injeksi SQL, *cross-site scripting* (*XSS*), serangan *brute force*, dan serangan *DDoS* [6]. Hal tersebut dapat mengancam keamanan dan stabilitas *website*, serta mengakibatkan kebocoran data sensitif atau gangguan pada layanan penting [7].

Untuk melindungi *website* sistem informasi akademik kampus (SIAKAD) di Universitas Merdeka Malang dari serangan-serangan ini, diperlukan pendekatan proaktif yang mencakup identifikasi, evaluasi, dan penanganan potensi kerentanan keamanan. Salah satu metode yang telah diakui secara luas dalam melakukan analisis keamanan website adalah menggunakan OWASP ZAP (*Open Web Application Security Project Zed Attack Proxy*) [8]. OWAPS adalah komunitas non-profit yang bertujuan untuk meningkatkan keamanan perangkat lunak. Metode OWASP sendiri telah digunakan pada penelitian sebelumnya dengan judul "*Website Vulnerability Testing and Analysis of Internet Management Information System Using OWASP*" [9]. Salah satu *software* yang dikembangkan oleh OWASP dan digunakan untuk penelitian ini adalah ZAP. ZAP adalah *scan tool* yang digunakan untuk menemukan

celah keamanan pada aplikasi web saat pengembangan dan pengujian aplikasi web [10].

Tujuan dari penelitian ini adalah untuk mengidentifikasi celah-celah keamanan pada sistem informasi akademik kampus, serta menganalisis tingkat risiko keamanan berdasarkan klasifikasi standar OWASP TOP 10 terbaru. Studi dilakukan pada situs *website* sistem informasi akademik (SIAKAD) di Universitas Merdeka Malang. Hasil identifikasi yang dilakukan melalui pengujian keamanan kemudian digunakan untuk memberikan rekomendasi kepada institusi agar dapat melakukan peningkatan keamanan sistem. Melalui penelitian ini diharapkan dapat memberikan kontribusi positif dalam peningkatan keamanan serta kualitas pelayanan *website* sistem informasi akademik kampus, serta memperkaya pemahamanan tentang analisis keamanan website menggunakan metode OWASP ZAP.

2. TINJAUAN PUSTAKA

Penelitian ini didasari oleh beberapa artikel dan jurnal terdahulu yang menjadi dasar teori penelitian ini mencangkup tentang sistem informasi akademik, analisis ancaman keamanan, termasuk penggunaan metode OWASP TOP 10, dan ZAP.

2.1. Sistem Informasi Akademik

Sistem informasi akademik adalah sebuah sistem yang digunakan oleh institusi pendidikan yang dimanfaatkan untuk meningkatkan pelayanan kepada mahasiswanya [2]. Sistem infomasi akademik ini mempunyai banyak sekali manfaat bagi institusi dalam bidang pendidikan, baik itu dalam pengolahan data pengajaran, data nilai, dan data-data lainnya yang terkait dengan akademik pembelajaran dalam hal ini khususnya perguruan tinggi [2]. Dengan adanya sistem informasi akademik diharapkan dapat meningkatkan efisiensi dan efektivitas operasional dalam lingkungan akademik,

2.2. Analisis Ancaman Keamanan

Dalam konsep keamanan informasi, terdapat istilah ancaman (*threats*) yang merupakan setiap peristiwa yang jika terjadi, dapat menyebabkan kerusakan pada sistem dan membuat hilangnya kerahasiaan, ketersediaan, atau integritas [5]. Tujuan adanya analisis ancaman keamanan adalah untuk mengidentifikasi, mengevaluasi, dan mengatasi potensi kerentanan keamanan dalam suatu sistem atau lingkungan. Hal ini bertujuan untuk mengunrangi resiko serangan siber dan melindungi aset informasi yang berharga.

Untuk menganlisis ancaman yang terdapat pada sistem, perlu dilakukan tindakan penetration testing. Penetration testing adalah tindakan secara aktif untuk menganalisis keamanan jaringan komputer atau (*Web Server*) dengan merencanakan dan mengeksekusi semua kemungkinan serangan untuk menemukan dan mengeksploitasi kerentanan yang ada [7]. Pengujian penetrasi adalah bagian penting dari penilaian keamanan dunia maya, dengan fokus pada potensi kerentanan yang terkait sistem yang ada [11]. Proses ini melibatkan serangkaian tindakan untuk memastikan bahwa data hanya tersedia bagi pihak yang berwenang, dan tidak dapat diakses oleh pihak yang tidak berwenang [12].

2.3. OWASP TOP 10

Open Web Application Security Project atau biasa yang disebut OWAPS merupakan organisasi internasional non-profit yang berdedikasi dibidang keamanan aplikasi *website*. OWAPS membuat sebuah *website* yang gratis dan mudah diakses, sehingga memudahkan pengguna untuk meningkatkan keamanan keamanan *website* mereka. OWASP 10 adalah daftar 10 standar keamanan *website* sehingga pengembang dapat memastikan apakah *website* tersebut aman atau tidak, dengan melakukan checklist berdasarkan standar ini [10], termasuk:

- *Broken Access Control*: Kerentanan di mana sistem tidak dapat mencegah pengguna yang tidak sah dari mengakses bagian-bagian tertentu dari aplikasi atau data *website* yang seharusnya tidak mereka akses. Menyebabkan pencurian data atau penggunaan aplikasi *website* secara tidak sah oleh pihak yang tidak bertanggung jawab.
- *Cryptographic Failures*: Kesalahan dalam penerapan teknik kriptografi, seperti penggunaan algoritma yang lemah atau pengaturan sandi yang buruk. Membuat data sensitif rentan terhadap pencurian atau dimanipulasi oleh penyerang.
- *Injection*: Serangan di mana penyerang menyisipkan kode berbahaya, seperti *SQL injection* atau *XSS*, ke dalam input yang dieksekusi oleh aplikasi. Memungkinkan penyerang untuk mengambil alih kontrol atas aplikasi website atau mengakses data sensitif.
- *Insecure Design*: Kelemahan yang muncul dari desain yang tidak aman atau rentan terhadap serangan. Dikarenakan keputusan desain yang buruk atau kurangnya pertimbangan keamanan selama proses pengembangan.
- *Security Misconfiguration*: Kesalahan konfigurasi yang membuat sistem atau aplikasi *website* rentan terhadap serangan. Termasuk pengaturan default yang tidak aman, atau kegagalan dalam mengkonfigurasikan sistem keamanan dengan baik.
- *Vulnerable and Outdated Components*: Kerentanan yang muncul dari penggunaan komponen atau perangkat lunak yang sudah lama atau rentan terhadap serangan yang diketahui. Termasuk kerentanan pada *framework* yang digunakan oleh aplikasi *website*.
- *Identification and Authentication Failures*: Kegagalan dalam proses identifikasi dan autotentikasi pengguna. Ini bisa termasuk kegagalan dalam menerapkan autotentikasi dua faktor, penggunaan kata sandi yang buruk, atau kegagalan dalam proses login pengguna.

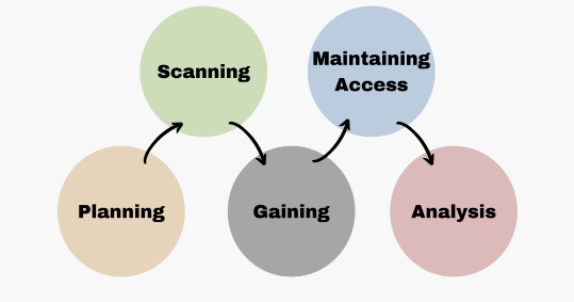
- *Software and Data Integrity Failures*: Kerentanan yang memungkinkan penyerang untuk mengubah atau merusak perangkat lunak atau data yang digunakan oleh aplikasi *website*. Hal ini dikarenakan kesalahan dalam proses validasi input atau kurangnya perlindungan pada manipulasi data.
- *Security Logging and Monitoring Failures*: Kegagalan dalam memantau dan mencatat aktivitas keamanan aplikasi *website*. Menyebabkan keterlambatan untuk mendeteksi serangan atau kesulitan dalam menelusuri serangan yang terjadi.
- *Server-Side Request Forgery*: Serangan di mana penyerang memanipulasi *server* untuk melakukan permintaan ke *database* yang tidak aman atau terlarang. Hal ini digunakan untuk mengakses data sensitif atau menyebabkan kerusakan pada sistem.

2.4. ZAP

ZAP atau biasa disebut *Zed Attack Proxy* adalah alat open-source gratis yang tersedia, untuk mengalisis celah keamanan *website*. ZAP melakukan serangan untuk menemukan kerentanan yang terdaftar dalam sepuluh kerentanan paling umum OWASP TOP 10 [8].

3. METODE PENELITIAN

Metode penelitian menggunakan pendekatan OWASP untuk mengidentifikasi celah keamanan dari *website* sistem informasi akademik kampus. Pengujian keamanan dilakukan melalui tahapan *penetration testing* dengan *flowchart* seperti yang diperlihatkan pada Gambar 1.



Gambar 1. *Flowchart penetrating test*

Uraian Tahapan dijelaskan sebagai berikut:

- *Planning*: Menentukan ruang lingkup dan tujuan dari penelitian, termasuk sistem yang akan diuji dan metode pengujian yang akan digunakan.
- *Scanning*: Memindai target dengan berbagai alat pendukung untuk menemukan. *Zed Attack Proxy* (ZAP) dari OWASP merupakan salah satu tool yang direkomendasikan karena mudah digunakan, gratis, dan mampu melakukan pemindaian otomatis pada target untuk menentukan celah dari sistem yang rentan terhadap serangan.
- *Gaining*: Upaya untuk mendapatkan akses ke dalam sistem berdasarkan hasil pemindaian pada tahap sebelumnya. Kerentanan keamanan dapat berupa *XSS, SQL injection,* dan *backdoors.*
- *Maintaining Access*: Proses pengujian apakah hak akses dari serangan sebelumnya dapat tetap terbuka atau tertutup.
- *Analysis*: Merangkum hasil dari pengujian penetrasi yang disusun ke dalam laporan, terdiri dari celah keamanan yang dapat dimanfaatkan, data yang dapat dicuri, dan waktu yang diperlukan untuk masuk ke dalam sistem.

Aplikasi ZAP akan digunakan untuk menguji keamanan dengan teknik *penetration testing*. ZAP akan mengakses sistem informasi akademik kampus untuk kemudian diuji melalui pemindaian otomatis. Agar mendapatkan hasil data yang diinginkan, proses *penetration testing* dijalankan melalui *intercepting proxy* seperti yang diperlihatkan pada Gambar 2. Teknik *intercepting proxy* memungkinkan pengguna untuk dapat memantau dan mengendalikan aktivitas lalu lintas jaringan antara *client* dengan *server* tanpa memerlukan konfigurasi langsung dari *client* sehingga bekerja dengan benar. Data yang melalui OWASP ZAP dapat disimpan dan diolah untuk tujuan penelitian ini.

Gambar 2. *Intercepting proxy*

Kemudian dilanjutkan dengan proses analisis hasil data yang diperoleh menggunakan panduan OWASP TOP 10. OWASP TOP 10 adalah daftar panduan yang diterbitkan secara umum oleh OWASP untuk mengidentifikasi sepuluh celah keamanan yang paling umum dan sering ditemui dalam aplikasi *website*. Pada Tabel 1 diuraikan daftar 10 celah keamanan berdasarkan standar OWASP TOP 10 yang merupakan versi terbaru yaitu pada tahun 2021.

Tabel 1. OWASP TOP 10 2021

Code	***Vulnerabilities***
A01	*Broken Access Control*
A02	*Cryptographic Failures*
A03	*Injection*
A04	*Insecure Design*
A05	*Security Misconfiguration*
A06	*Vulnerable and Outdated Components*
A07	*Identification and Authentication Failures*
A08	*Software and Data Integrity Failures*
A09	*Security Logging and Monitoring Failures*
A10	*Server-Side Request Forgery*

4. HASIL DAN PEMBAHASAN

Untuk proses mengidentifikasi celah keamanan dari *website* sistem informasi akademik kampus. Pengujian keamanan dilakukan melalui tahapan sesuai dengan metode yang dijelaskan sebelumnya

4.1. *Planning*

Pada proses perencanaan, peneliti menggunakan perangkat yang digunakan untuk proses pengujian keamanan. Pada Tabel 2 adalah spesifikasi *hardware* dan *software* yang digunakan untuk proses pengujian.

Tabel 2. Spesifikasi *hardware* dan *software*

Nama Alat	Spesifikasi
Komputer	OS: Windows 10 Pro 64 bit Processor: AMD Athlon 3000G quad-core 3,5GHz RAM: 8 GB DDR4 VGA: Integreted Radeon Vega Graphics SSD: 128 GB HDD: 1 TB
OWASP ZAP	Version 2.15.0
Koneksi Internet	Up to 50 Mbps
Web Browser	Google Chrome Version 124.0.6367.119

Kemudian untuk target pengujian adalah *website* sistem akademik informasi (SIAKAD) Universitas Merdeka Malang.

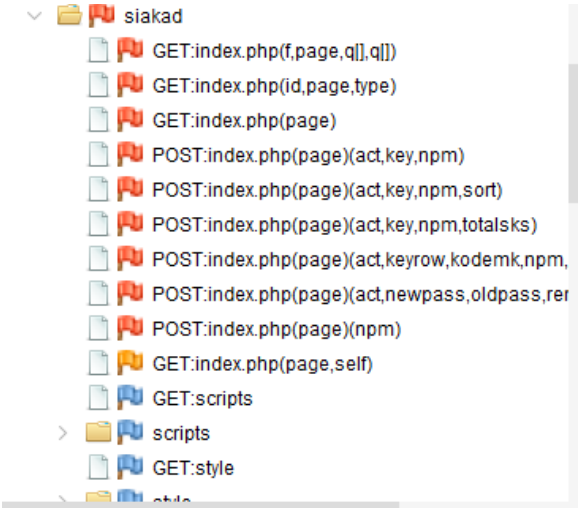
4.2. *Scanning*

Pada proses *scanning* atau pengujian, Peneliti menggunakan aplikasi OWASP ZAP untuk mendeteksi kerentanan pada *website* SIAKAD Universitas Merdeka Malang. Kemudian untuk pengaturan *browser* agar terhubung dengan OWASP ZAP dilakukan menggunakan metode *intercepting proxy* untuk mendapatkan data hasil *penetration testing*. Pada Gambar 3 diperlihatkan hasil pengujian keamanan melalui penetration testing yang dilakukan pada website SIAKAD menggunakan fitur pemindaian otomatis pada aplikasi ZAP.

Gambar 3. Hasil *scan tool* OWASP ZAP

4.3. *Gaining*

Setelah dilakukan 3 kali proses *scanning* atau pengujian, peneliti mendapatkan hasil yang sama. Hasil dari pengujian berupa *source code* yang bisa diakses oleh penyerang seperti yang ditampilkan pada Gambar 4 sehingga dapat disisipkan *source code* berbahaya seperti *SQL injection.*



Gambar 4. Data *source code* hasil pengujian

4.4. *Maintaining Access*

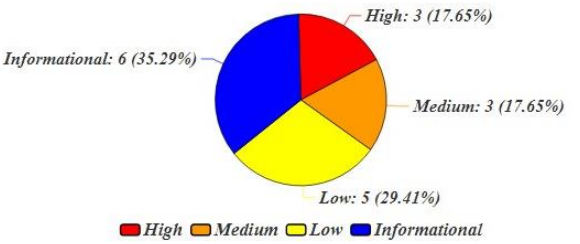
Pada proses *Maintaining Access*, peneliti menguji ulang kerentanan dan mendapatkan akses menggunakan metode *intercepting proxy* sehingga peneliti dapat memantau aktifitas yang terjadi pada *website* sistem informasi akademik (SIAKAD) Universitas Merdeka Malang. Hasil metode *intercepting proxy* ditampilkan pada Gambar 5.

ID	Source	Req. Timestamp	Method	
3.596	Proxy	19/06/2024, 00.42.35	GET	https://siakad.unr
3.597	Proxy	19/06/2024, 00.42.35	GET	https://siakad.unr
3.598	Proxy	19/06/2024, 00.42.37	GET	https://siakad.unr
3.599	Proxy	19/06/2024, 00.42.37	GET	https://siakad.unr
3.600	Proxy	19/06/2024, 00.42.37	GET	https://siakad.unr
3.601	Proxy	19/06/2024, 00.42.37	GET	https://siakad.unr
3.602	Proxy	19/06/2024, 00.42.41	GET	https://siakad.unr
3.603	Proxy	19/06/2024, 00.42.41	GET	https://siakad.unr

Gambar 5. Hasil *intercepting proxy*

4.5. *Analysis*

Proses analisi hasil pemindaian pada OWASP ZAP dilakukan secara manual agar mendapatkan hasil yang maksimal. Pada hasil gambar percobaan menggunakan scanning OWASP ZAP, didapatkan informasi jenis-jenis kerentanan beserta tingkat risiko keamanan pada kerentanan tersebut. Kemudian hasil proses dari pemindaian pada OWASP ZAP disajikan dalam bentuk presentase pada Gambar 6.



Gambar 6. Presentase tingkat resiko keamanan

Tingkat risiko keamanan disimbolkan dengan presentase tersebut, dimana warna merah menandakan risiko tingkat tinggi, warna jingga untuk risiko tingkat sedang, warna kuning untuk risiko tingkat rendah, serta warna biru untuk jenis informasional. Grafik lingkaran yang diberikan mengkategorikan kerentanan yang teridentifikasi ke dalam empat tingkat risiko: Tinggi, Sedang, Rendah, dan Informasional. Setiap segmen grafik tersebut mewakili jumlah persentase dari total kerentanan yang terdeteksi. Berikut penjelasan tingkat resiko berdasarkan hasil grafik tersebut:

- *High*: adalah cacat keamanan kritis dan berbahaya sehingga dapat dieksploitasi dengan mudah oleh perentas untuk menyebabkan kerusakan signifikan, sehingga dapat menggangu proses pada website menjadi tidak maksimal dan website tersebut menjadi tidak aman.
- *Medium*: adalah masalah keamanan signifikan sehingga dapat dieksploitasi oleh penyerang untuk mendapatkan beberapa tingkat akses tertentu atau bahkan menyebabkan kerusakan terbatas. Hal ini mungkin memerlukan tingkat keterampilan yang lebih tinggi untuk dapat dieksploitasi oleh perentas dibandingkan dengan kerentanan risiko tinggi. Hal ini menunjukkan bahwa ada kerentanan penting yang perlu diatasi untuk mengurangi ancaman keamanan.
- *Low*: adalah masalah yang kurang berbahaya yang dapat menimbulkan ancaman lebih rendah terhadap website. Hal ini mungkin lebih sulit untuk dieksploitasi oleh perentas sehingga memiliki dampak yang kurang parah jika dapat dieksploitasi.
- *Informational*: adalah temuan masalah yang tidak dapat menimbulkan ancaman keamanan secara langsung tetapi merupakan hal yang perlu ditingkatkan atau perbaikan potensial yang dapat meningkatkan keamanan website.

Kemudian berdasarkan hasil penetration testing tersebut, keterangan warna berdasarkan jenis tingkat risiko disajikan pada Gambar 7.

Risk Level	Number of Alerts
High	3
Medium	3
Low	5
Informational	6

Gambar 7. *Summary of alerts by OWAPS ZAP*

Berdasarkan hasil pengujian keamanan pada website SIAKAD, terdapat 3 kerentanan pada tingkat risiko tinggi atau *high risk level*, 3 kerentanan pada tingkat risiko sedang atau *medium risk level*, 5 kerentanan pada tingkat risiko rendah atau *low risk level*, dan 6 kerentanan yang bersifat informasional atau *informational risk level*. Kemudian jika dibuat lebih spesifik maka hasil tes kerentanan menggunakan OWASP ZAP seperti yang diperlihatkan pada Gambar 8.

Name	Risk Level	Number of Instances
Cross Site Scripting (Reflected)	High	5
SQL Injection	High	10
SQL Injection - SQLite	High	15
Content Security Policy (CSP) Header Not Set	Medium	38
Missing Anti-clickjacking Header	Medium	33
Vulnerable JS Library	Medium	2
Cookie No HttpOnly Flag	Low	3
Cookie Without Secure Flag	Low	3
Cookie without SameSite Attribute	Low	3
Cross-Domain JavaScript Source File Inclusion	Low	1
X-Content-Type-Options Header Missing	Low	43
Authentication Request Identified	Informational	2
Charset Mismatch (Header Versus Meta Content-Type Charset)	Informational	29
Information Disclosure - Suspicious Comments	Informational	6
Modern Web Application	Informational	9
Session Management Response Identified	Informational	75
User Agent Fuzzer	Informational	216

Gambar 8. *Specification alerts reports by OWAPS ZAP*

Data hasil percobaan yang didapatkan kemudian dianalisis berdasarkan standar pedoman OWASP TOP 10 untuk disesuaikan dengan *risk level*. Hasil klasifikasi kerentanan berdasarkan OWASP TOP 10 2021 ditampilkan pada Tabel 3.

Tabel 3. Klasifikasi kerentanan berdasarkan OWASP TOP 10 2021

Code	***Vulnerabilities***	***Alerts***	***Risk Level***
A01	*Broken Access Control*	*None*	*None*
A02	*Cryptographic Failures*	*Cookie No HttpOnly Flag* *Cookie Without Secure Flag* *Cookie Without SameSite Attribute*	*Low*
A03	*Injection*	*Cross Site Scripting (Reflected)* *SQL Injection* *SQL Injection - SQLite*	*High*
A04	*Insecure Design*	*Vulnerable JS Library*	*Medium*
A05	*Security Misconfiguration*	*Content Security Policy (CSP) Header not Set* *Missing Anti-clickjacking Header*	*Medium*
		X-Content-Type-Options Header Missing	*Low*
A06	*Vulnerable and Outdated Components*	*Modern Web Application*	*Informational*
A07	*Identification and Authentication Failures*	*Authentication Reques Identified* *User Agent Fuzzer*	*Informational*

Code	*Vulnerabilities*	*Alerts*	*Risk Level*
A08	*Software and Data Integrity Failures*	*Session Management Response Identified*	*Informational*
A09	*Security Logging and Monitoring Failures*	*Information Disclosure – Suspicious Comments*	*Informational*
A10	*Server-Side Request Forgery*	*Cross-Domain JavaScript Source File Inclusion*	*Informational*

Tingkat risiko keamanan tinggi atau *high* diperoleh pada kerentanan A03 yaitu *Injection*. Untuk tingkat risiko keamanan sedang atau *medium* diperoleh pada kerentanan A04 (*Insecure Design*) dan A05 (*Security Misconfiguration*). Sedangkan kerentanan A02 (*Cryptographic Failures*) dan A05 (*Security Misconfiguration*) berada pada tingkat risiko keamanan rendah atau *low* dan untuk kerentanan A06 (*Vulnerable and Outdated Components*) A07 (*Identification and Authentication Failures*) A08 (*Software and Data Integrity Failures*) A09 (*Security Logging and Monitoring Failures*) A10 (*Server-Side Request Forgery*) berada pada tingkat resiko informasi atau *Informational*.

4.6. Rekomendasi

Berdasarkan hasil penyesuaian data hasil percobaan kerentanan dengan pendoman OWASP TOP 10, untuk rekomendasi yang diberikan adalah rekomendasi dengan tingkat risiko tinggi, sedang, dan rendah. Beberapa rekomendasi untuk peningkatan keamanan pada *website* SIAKAD diuraikan pada Tabel 4.

Tabel 4. Rekomendasi berdasarkan *vulnerabilities*

Vulnerabilities	Rekomendasi
A02 - Cryptographic Failures	• Menggunakan algoritma enkripsi yang kuat seperti AES-256. • Menerapkan enkripsi untuk data sensitif saat transit dan penyimpanan data. • Menggunakan protokol HTTPS untuk semua komunikasi pada *website*. • Menerapkan *flag* HttpOnly, Secure, dan SameSite untuk cookie.
A03 - Injection	• Menggunakan ORM atau *prepared statements* untuk menghindari SQL *Injection*. • Menerapkan CSP untuk membatasi sumber daya yang dapat dimuat. • Menggunakan *library* atau *framework* yang telah teruji untuk menghindari kerentanan XSS.
A04 - Insecure Design	• Menerapkan tinjauan kode yang ketat untuk memastikan desain aman. • Membatasi penyertaan sumber daya dan hanya dari domain yang terpercaya.
A05 - Security Misconfiguration	• Mengkonfigurasi *server* dan aplikasi dengan baik dan benar. • Pastikan mekanisme otentikasi aman dan menggunakan autentikasi *multi-factor* jika memungkinkan • Pastikan bahwa semua komponen aplikasi dan *server* diperbarui secara rutin.

5. KESIMPULAN DAN SARAN

Pengujian keamanan pada *website* SIAKAD Universitas Merdeka Malang menggunakan metode OWASP ZAP dengan *intercepting proxy*. Berdasarkan hasil pengujian keamanan melalui *penetration testing*, telah berhasil ditemukan beberapa kerentanan. Dengan hasil tersebut dapat dianalisis data dan diperoleh bahwa pada *website* SIAKAD terdapat 17 kerentanan dengan tingkat kerentanan pada *High Risk Level* sebesar 17,65% atau 3 *High Risk Level*, pada *Medium Risk Level* sebesar 17,65% atau 3 *Medium Risk Level*, pada *Low Risk Level* sebesar 29,41% atau 5 *Low Risk Level*, dan untuk *Informational Risk Level* sebesar 35,29% atau 6 *Informational Risk Level*. Rekomendasi diberikan khususnya pada kerentanan pada standar OWASP A02 (*Cryptographic Failures*), A03 (*Injection*), A04 (*Insecure Design*), dan A05 (*Security Misconfiguration*), dan untuk kerentanan A06 (*Vulnerable and Outdated Components*) A07 (*Identification and Authentication Failures*) A08 (*Software and Data Integrity Failures*) A09 (*Security Logging and Monitoring Failures*) A10 (*Server-Side Request Forgery*) berada pada tingkat resiko informasi atau *Informational*, dan untuk A01 (*Broken Access Control*) tidak terdapat risk yang ditemukan.

Dengan hasil dan rekomendasi yang telah dijelaskan pada pembahasan sebelumnya, diharapkan dapat digunakan oleh pihak staf TI Universitas Merdeka Malang untuk meningkatkan keamanan dan kenyamanan dalam mengakses *website* sistem informasi akademik kampus. Diharapkan penelitian ini dapat berkontribusi dengan memberikan rekomendasi terutama pada *high risk level* dan menjadi dasar untuk meningkatkan keamanan *website* sistem informasi akademik kampus Universitas Merdeka Malang menggunakan metode OWASP ZAP yang dibandingkan dengan metode lainnya, agar hasil analisis dari hasil tersebut dapat lebih akurat dapat digabungankan dengan metode lainnya dalam menganalisis kerentanan pada *website* agar mendapatkan keakuratan yang lebih baik kedepannya.

DAFTAR PUSTAKA

[1] R. K. Ngantung and M. A. I. Pakereng, "Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 1052, 2021, doi: 10.30865/mib.v5i3.3054.

[2] A. Zakir and D. Irwan, "Perancangan Sistem Informasi Pengajuan Kerja Praktek Pada

Program Studi Sistem Informasi Menggunakan Uml," *J. Ilm. Teknol. Inf. dan Robot.*, vol. 2, no. 2, pp. 1–6, 2020, doi: 10.33005/jifti.v2i2.34.

[3] S. F. Arief and Y. Sugiarti, "Literature Review: Analisis Metode Perancangan Sistem Informasi Akademik Berbasis Web," *J. Ilm. Ilmu Komput.*, vol. 8, no. 2, pp. 87–93, 2022, doi: 10.35329/jiik.v8i2.229.

[4] P. H. Marpaung, N. Dahri, and W. Yahyan, "Perancangan Sistem Informasi Pengolahan Data Mahasiswa Magang Di Perusahaan Berbasis Web," *J. Manaj. Teknol. Inform.*, vol. 1, no. 2, pp. 109–116, 2023.

[5] M. S. Hardani and K. Ramli, "Perancangan Manajemen Risiko Keamanan Sistem Informasi Manajemen Sumber Daya dan Perangkat Pos dan Informatika (SIMS) Menggunakan Metode NIST 800-30," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 3, p. 591, 2022, doi: 10.30865/jurikom.v9i3.4181.

[6] B. Ghozali, K. Kusrini, and S. Sudarmawan, "Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) Untuk Penilaian Risk Rating," *Creat. Inf. Technol. J.*, vol. 4, no. 4, p. 264, 2019, doi: 10.24076/citec.2017v4i4.119.

[7] M. C. Ghanem and T. M. Chen, "Reinforcement learning for efficient network penetration testing," *Inf.*, vol. 11, no. 1, pp. 1–23, 2020, doi: 10.3390/info11010006.

[8] A. Jakobsson and I. Häggström, "Study of the techniques used by OWASP ZAP for analysis of vulnerabilities in web applications," p. 61, 2022, [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1675227&dswid=-4307

[9] D. Priyawati, S. Rokhmah, and I. C. Utomo, "Website Vulnerability Testing and Analysis of Internet Management Information System Using OWASP," *Int. J. Comput. Inf. Syst. Peer Rev. J.*, vol. 3, no. 3, pp. 143–147, 2022, doi: 10.29040/ijcis.v3i3.90.

[10] R. R. Daniswara, G. Made, A. Sasmita, P. Agus, and E. Pratama, "Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application)," *JITTER-Jurnal Ilm. Teknol. dan Komput.*, vol. 1, no. 1, 2020.

[11] F. Fachri, A. Fadlil, and I. Riadi, "Analisis Keamanan Webserver menggunakan Penetration Test," *J. Inform.*, vol. 8, no. 2, pp. 183–190, 2021, doi: 10.31294/ji.v8i2.10854.

[12] A. M. Ujung, M. Irwan, and P. Nasution, "Pentingnya Sistem Keamanan Database untuk melindungi data pribadi," *JISKA J. Sist. Inf. Dan Inform.*, vol. 1, no. 2, p. 44, 2023, [Online]. Available: http://jurnal.unidha.ac.id/index.php/jteksis