ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v22i3.1421

Analysis of Information System Security Using OWASP ZAP on a Web-Based Electronic Archiving System

Analisis Keamanan Sistem Informasi Menggunakan OWASP ZAP pada Sistem Kearsipan Elektronik Berbasis Website

Virda Ramadhani Putri[1], Ade Sobandi[2], Budi Santoso[3]

[1,2,3] Manajemen Perkantoran, Universitas Pendidikan Indonesia, Indonesia

[1*]virdarp@upi.edu, [2]ade@upi.edu, [3]budisantoso@upi.edu
**: Penulis korenspondensi* (*virdarp@upi.edu*)

Informasi Artikel
Received: August 2025
Revised: November 2025
Accepted: November 2025
Published: October 2025

***Abstract** (menggunakan style abstract)*

Purpose: Web-based information systems have become an essential bridge for facilitating accessibility and the use of information. However, with the convenience of access and usage, serious threats related to data security in web systems have also emerged. These threats may arise due to vulnerabilities in the web system, which can be exploited by irresponsible parties to carry out cyberattacks aimed at stealing, damaging, or altering the available information. Therefore, this research is conducted as a preventive measure against these threats through preventive actions by analyzing security vulnerabilities on websites using penetration testing, one of which utilizes the Open Web Application Security Project (OWASP).
Design/methodology/approach: Security analysis of information systems using OWASP ZAP with a penetration testing method.
Findings/result: The testing results and analys conducted on the target website of the web-based electronic archiving system, http://silancarbedas.bandungkab.go.id/, revealed 13 security vulnerabilities categorized under several OWASP ZAP 10:2021 frameworks. Based on these findings, several suggestions or recommendations have been provided to address the website vulnerabilities, which can be used by the website developers to enhance the site's security
Originality/value/state of the art: Vulnerability testing on the web-based electronic archiving information system at http://silancarbedas.bandungkab.go.id/ has not been conducted previously.

Keywords: Website Security; OWASP ZAP; Vulnerabilities; Penetration Testing
Kata kunci: Keamanan Website; OWASP ZAP; Celah Kerentanan; *Penetration Testing*

Abstrak (menggunakan style abstrak)

Tujuan: Sistem informasi berbasis web kini menjadi salah satu hal jembatan penghubung untuk memberikan kemudahan aksesibilitas dan penggunaan infomasi. Namun seiring dengan kemudahan akses dan penggunaannya, muncul juga ancaman serius terkait keamanan data pada penggunaan sistem website. Ancaman ini dapat timbul disebabkan oleh kerentanan pada sistem website yang berpotensi untuk dimanfaat oleh pihak-pihak yang tidak bertanggung jawab untuk melakukan berbagai serangan siber yang bertujuan untuk mencuri, merusak, atau mengubah informasi yang tersedia. Maka, penelitian ini dilakukan sebagai pencegahan terhadap ancaman tersebut melalui tindakan preventif dengan melakukan analisis celah kerentanan pada website dengan cara *penetration testing*, salah satunya menggunakan *Open Web Application Security Project* (OWASP).

Perancangan/metode/pendekatan: Analisis Keamanan Sistem informasi menggunakan OWASP ZAP dengan metode *penetration testing*.

Hasil: Hasil pengujian dan analisis yang telah dilakukan pada target website sistem kearsipan elektronik berbasis website http://silancarbedas.bandungkab.go.id/ menunjukkan temuan berupa 13 celah kerentanan yang dapat dikategorikan pada beberapa *framework* OWASP ZAP 10:2021. Berdasarkan temuan tersebut juga telah diberikan beberapa saran atau rekomendasi solusi untuk menangani celah kerentanan situs web yang dapat digunakan oleh pihak pengembang situs web untuk meningkatkan keamanan situs web tersebut.

Keaslian/ *state of the art*: Belum pernah dilakukan uji kerentanan sistem informasi kearsipan elektronik berbasis web pada http://silancarbedas.bandungkab.go.id/ sebelumnya.

1. Pendahuluan

Ditengah perkembangan teknologi informasi dan komunikasi yang pesat, sistem informasi berbasis web menjadi salah satu hal yang menjadi jembatan penghubung untuk memberikan kemudahan aksesibilitas dan penggunaan infomasi. Penggunaan aplikasi web kini meningkat hampir pada berbagai aspek organisasi, tak terkecuali pemerintahan di Indonesia yang sedang gencar menerapkan konsep e-*government*. Salah satu wujud implemantasi *e-government* adalah melalui penggunaan situs website di setiap institusi pemerintah, baik pusat maupun daerah. Situs website tersebut dirancang untuk melayani berbagai keperluan operasional instansi, baik yang

bersifat *Government to Government* (G2G), *Government to Business* (G2B), dan *Government to Consumers* (G2C) [1]. Penggunaan website di pemerintahan ini memberikan kemudahan dalam efisiensi operasional, sistem administrasi, penyampaian informasi, transparansi dan akuntabilitas informasi, penyampaian aspirasi masyarakat, akses pelayanan publik, dan masih banyak lagi.

Dinas Perpustakaan dan Arsip (Dispusip) bersama Dinas Komunikasi dan Informatika (Diskominfo) Kabupaten Bandung dalam rangka mewujudkan *e-government*, berkomitmen dengan meningkatkan penggunaan sistem informasi berbasis website pada pelaksanaan operasionalnya. Sistem informasi kearsipan elektronik berbasis website dengan alamat http://silancarbedas.bandungkab.go.id/, menjadi salah satu website yang dirilis untuk diterapkan oleh semua instansi pemerintahan di Kabupaten Bandung dengan tujuan memberikan kemudahan dalam pengelolaan arsip secara statis. Website tersebut memberikan kemudahan dalam pengelolaan arsip statis pada setiap instansi dengan mengurangi beban pekerjaan manual, mempermudah akses data dan informasi arsip, mempermudah penemuan kembali arsip, serta mengurangi kemungkinan kehilangan atau kerusakan dokumen.

Namun, seiring dengan kemudahan akses dan penggunaannya, muncul juga ancaman serius terkait keamanan data pada penggunaan sistem website. Ancaman ini dapat timbul dikarenakan kerentanan pada sistem website yang dapat dimanfaat oleh pihak-pihak yang tidak bertanggung jawab untuk melakukan berbagai serangan siber yang bertujuan untuk mencuri, merusak, atau mengubah informasi yang tersedia [2]. Badan Siber dan Sandi Negara (BSSN) mencatat, pada tahun 2023 Indonesia mendapatkan 279,84 juta serangan siber [3]. Tentunya hal tersebut bukanlah jumlah yang sedikit, dan di dalamnya juga termasuk serangan siber pada instansi pemerintahan. Oleh karena itu, keamanan sistem website menjadi isu yang sangat kritis untuk diperhatikan seiring dengan berbagai risiko yang dapat ditimbulkan. Salah satu solusi pengamanan web dari gangguan atau serangan kejahatam siber dapat dilakukan dengan cara *self test*, yaitu pengujian yang dilakukan terhadap server website secara legal dengan aktifitas menyerupai *hacker*. *Self test* dapat dilakukan dengan beberapa metode *penetration testing*, melalui berbagai *tools* seperti *Information Systems Security Assessment Framework* (ISSAF), *Open Web Application Security Project Zed Attack Proxy* (OWASP ZAP), atau *Open Source Security Testing Methodology Manual* (OSSTMM) [4].

Mengabaikan kerentanan-kerentanan pada sistem website dapat berakibat fatal terlebih dalam keamanan data dan integritas sistem. Dampaknya tidak hanya dirasakan bagi organisasi, tetapi juga bagi pengguna yang mengandalkan sistem tersebut. Beberapa dampak yang dapat timbul antara lain, serangan *malware*, kerugian finansial akibat biaya memperbaiki sistem, kebocoran data, penyalahgunaan informasi, kehilangan kepercayaan pengguna, kerusakan reputasi, hingga menghambat produktivitas operasional. Dengan demikian, penting untuk selalu memprioritaskan keamanan sistem website agar organisasi dapat memastikan bahwa data sensitif tetap aman dan sistem beroperasi dengan stabil. Terlebih bagi website http://silancarbedas.bandungkab.go.id/, yang berfungsi sebagai sarana pengelolaan manajemen kearsipan yang menyimpan data penting dari berbagai dokumen, menjaga keamanan sistem website menjadi hal yang sangat krusial untuk diperhatikan agar data dan informasi yang terdapat di dalamnya dapat dipastikan terlindungi dengan aman.

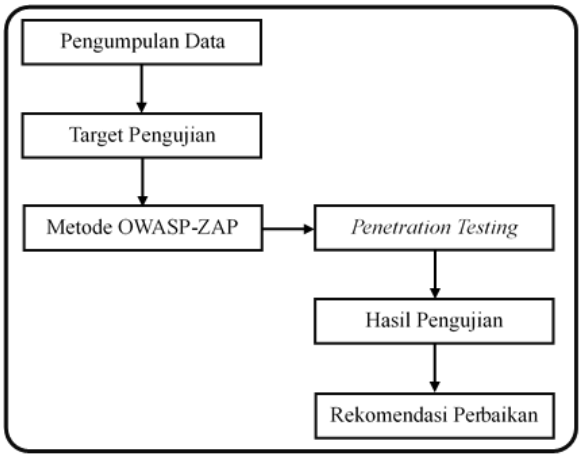
Atas dasar tersebut, peneliti melakukan analisis keamanan sistem website kearsipan elektronik http://silancarbedas.bandungkab.go.id/ dengan menggunakan metode *Application Security*

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v22i3.1421

Project Zed Attack Proxy (OWASP ZAP) untuk memastikan keamanan informasi pada web tersebut dengan menganalisis kerentanan yang ada. Melalui analisis kemanan menggunakan metode OWASP, penelitian ini tidak hanya akan mengidentifikasi potensi celah kerentanan pada website tersebut, tetapi juga memberikan rekomendasi perbaikan yang diperlukan untuk meningkatkan ketahanan dari serangan pihak yang tidak bertanggung jawab. Dengan demikian, diharapkan langkah-langkah ini dapat memperkuat perlindungan data dan mencegah risiko yang dapat mengancam integritas serta keamanan informasi arsip yang ada pada website tersebut**.**

2. Metode/Perancangan

Metode yang digunakan pada penelitian ini yaitu metode *Application Security Project Zed Attack Proxy* (OWASP ZAP) yang bertujuan untuk menguji keamanan sistem website melalui *penetration testing* pada website sistem kearsipan elektronik dengan alamat http://silancarbedas.bandungkab.go.id/. Penelitian ini menggunakan pendekatan deskriptif kualitatif karena fokus pada pemahaman mendalam tentang celah keamanan, eksplorasi potensi risiko, dan deskripsi detail dari hasil penelitian melalui identifikasi serta analisis yang mengacu pada standarisasi OWASP Top 10: 2021. Berikut merupakan kerangka alur penelitian yang dilakukan dalam melakukan analisis keamanan sistem informasi menggunakan metode OWASP ZAP.



Gambar 1. Alur Penelitian

2.1. Pengumpulan Data

Tahapan yang pertama pada penelitian ini adalah pengumpulan data melalui studi literatur dari berbagai sumber seperti jurnal ilmiah, situs internet, dan bacaan lainnya yang relevan dengan penelitian yang akan dilakukan guna memperoleh informasi yang dibutuhkan.

2.2. Target Pengujian

Penelitian dilakukan melalui pengujian pada *website* sistem kearsipan elektronik di Kabupaten Bandung yang dikembangkan oleh Dinas Perpustakaan dan Arsip Kabupaten Bandung yang berkolaborasi dengan Dinas Komunikasi dan Informatika Kabupaten Bandung dengan alamat website http://silancarbedas.bandungkab.go.id/.

2.3. Metode OWASP ZAP

Open Web Application Security Project (OWASP) adalah organisasi nonprofit yang berfokus pada peningkatan keamanan perangkat lunak. OWASP berfungsi sebagai kerangka kerja yang digunakan oleh pengembang dan ahli teknologi untuk melindungi situs web. OWASP menyediakan platform bagi pengembang sistem untuk meningkatkan keamanan sistem melalui proyek-proyek *open-source*, serta menyediakan alat-alat dalam mendukung pengujian sistem dalam rangka mengidentifikasi celah kerentanan pada sebuah aplikasi website [5]. *Zed Attack Proxy* (ZAP) merupakan alat pengujian penetrasi *open-source* gratis yang dikelola dibawah naungan Open Web Aplication Security Project (OWASP). Maka OWASP ZAP merupakan sebuah alat yang dapat digunakan untuk *penetration testing* sebagai sarana untuk menemukan celah *vulnerabilities* (kerentanan) pada suatu aplikasi website [6].

Dalam metodenya, OWASP mengembangkan daftar sepuluh kerentanan yang dikenal sebagai OWASP Top 10 untuk mengidentifikasi dan menilai risiko keamanan yang ada pada situs web. Daftar ini mencakup sepuluh risiko keamanan yang umum terjadi dalam eksploitasi aplikasi web dan terus diperbarui seiring perkembangan teknologi website. OWASP Top 10 edisi terakhir yaitu pada tahun 2021. Berikut ini merupakan daftar 10 risiko keamaan dalam web menurut OWASP Top 10 tahun 2021 [7].

1. A01 - *Broken Access Control*: Kerentanan ini terjadi ketika pengguna tidak sah memiliki kontrol akses untuk mengakses fungsi atau data yang tidak seharusnya tidak dapat mereka akses.
2. A02 - *Cryptographic Failures*: Mencakup masalah dalam pengelolaan kriptografi, seperti penggunaan algoritma yang lemah atau pengelolaan kunci yang tidak aman.
3. A03 - *Injection*: Kerentantan yang terjadi ketika penyerang dapat menyisipkan kode atau perintah berbahaya ke dalam aplikasi, seperti SQL *injection* atau *command injection*.
4. A04 - *Insecure Design*: Kerentanan ini berkaitan dengan desain arsitektur sistem yang tidak aman, yang dapat menciptakan celah bagi penyerang.
5. A05 - *Security Misconfiguration*: Kerentanan yang terjadi ketika pengaturan keamanan aplikasi terjadi kesalahan, seperti oeongaturan default yang tidak aman atau database tidak dikonfigurasi dengan benar, sehingga menciptakan potensi risiko.
6. A06 - *Vulnerable and Outdated Components*: Menggunakan komponen perangkat lunak yang rentan atau tidak diperbarui, sehingga dapat membuka celah bagi penyerang.
7. A07 - *Identification and Authentication Failures*: Kerentanan ini terkait dengan kegagalan dalam proses identifikasi dan otentikasi pengguna, seperti penggunaan kata sandi yang lemah, hal ini memungkinkan pengguna tidak sah untuk mengakses sitem.
8. A08 - *Software and Data Integrity Failures*: Mencakup masalah di mana data atau perangkat lunak dapat dimodifikasi tanpa otorisasi yang tepat karena ketidakmmapuan untuk memastikan integritas perangkat lunak dan data, termasuk penggunaan sumber yang tidak terpercaya.
9. A09 - *Security Logging and Monitoring Failures*: Kurangnya pencatatan dan pemantauan keamanan yang memadai, sehingga menyulitkan deteksi dan respon terhadap serangan yang sedang berlangsung.

10. A10 - *Server-Side Request Forgery* (SSRF): Kerentanan ini memungkinkan penyerang mengirim permintaan akses dari server ke sumber daya internal atau eksternal tanpa izin.

2.4. Penetration Testing

Penetration testing adalah pengeksplotasian keamanan sistem komputer secara sah guna membuat sistem tersebut menjadi lebih aman dengan memperbaiki kelemahan yang ada selama pengujian [8].

Sedangkan menurut modul CEH (*Certified Ethical Hacking*), *penetration testing* adalah metode untuk mengevaluasi keamanan sistem komputer atau jaringan dengan mensimulasikan serangan dari sumber berbahaya. Tujuan dari pengujian ini adalah untuk mengidentifikasi potensi serangan yang bisa terjadi akibat kelemahan dalam sistem dan merupakan bagian dari audit keamanan [9].

2.5. Hasil Pengujian

Tahap hasil pengujian adalah tahap yang berisi penjelasan hasil yang didapatkan dari pengujian keamanan *website* berdasarkan OWASP Top 10 tahun 2021.

2.6. Rekomendasi Perbaikan

Tahap rekomendasi perbaikan berisi saran atau solusi yang diberikan untuk menangani maupun memperbaiki suatu sistem ataupun permasalahan yang ada pada sistem tersebut. Rekomendasi ini dihasilkan dari analisis proses dan hasil pengujian pada target yang diuji.

3. Hasil dan Pembahasan

Berdasarkan tahap awal yang dilakukan berupa pengumpulan data, peneliti berhasil menganalisis masalah kerentanan keamanan pada situs web, khususnya pada situs sistem informasi dari berbagai sumber, termasuk jurnal penelitian, prosiding, dan literatur *online*. Dari berbagai sumber tersebut dapat disimpulkan bahwa jika pihak pengembang website mengabaikan aspek keamanan sistem pada website, maka memungkinkan untuk membuka celah yang memudahkan oknum tidak bertanggung jawab melakukan peretasan website.

Maka, untuk mengkaji lebih dalam terkait isu ini, peneliti melakukan analisis terhadap salah satu sistem informasi yang digunakan di Kabupaten Bandung yaitu berupa website http://silancarbedas.bandungkab.go.id/ yang berfungsi untuk pengelolaan manajemen kearsipan sebagai objek penelitian. Website tersebut memuat berbagai hal yang berkaitan dengan pengelolaan arsip statis, dan fitur yang paling penting adalah terkait informasi penyimpanan arsip serta arsip hasil alih medianya. Website ini dapat digunakan oleh seluruh instnasi beserta seluruh Unit Pelaksanan Teknis (UPT) yang ada di lingkup Kabupaten Bandung melalui fitur akun yang dapat dibuat oleh Adminstrator. Mempertimbangkan aspek krusial dari fungsi dan isi dari website serta tingginya tingkat akses, maka tidak dapat dihindari bahwa memungkinkan adanya celah untuk hacker masuk untuk mencuri dan menggunakan berbagai data dan informasi penting untuk tujuan tertentu.

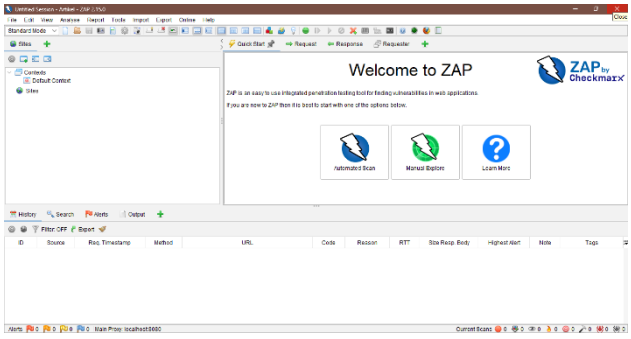
Dari permasalahan tersebut, upaya preventif pun direncanakan pada tahap selanjutnya yakni peneliti melakukan Analisis kerentanan keamanan sistem informasi berupa website http://silancarbedas.bandungkab.go.id/, untuk memastikan keamanan informasi pada web tersebut. Untuk melaksanakan penelitian ini, peneliti menggunakan aplikasi OWASP ZAP untuk melakukan penetration testing dengan *automated scanner* atau *manual explore* yang

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v22i3.1421

bertujuan untuk mendapatkan hasil kerentanan yang ada, agar hasilnya dapat digunakan sebagai bahan evaluasi dan perbaikan untuk menghindari serangan yang tidak diinginkan pada website tersebut.

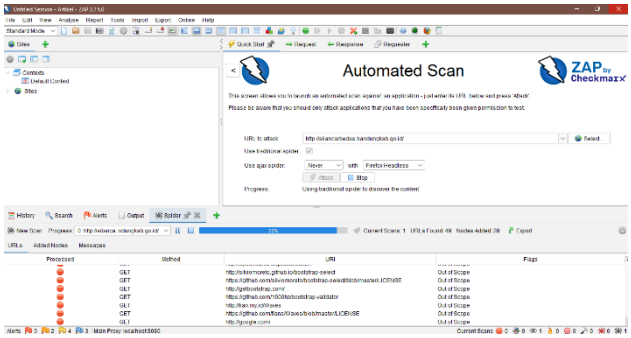
Tahap selanjutnya peneliti akan mulai melaksanakan pengujian kerentanan website menggunakan OWASP ZAP versi 2.15.0. Pada tahap ini peneliti akan melakukan *penetration testing* dengan menyeluruh pada website http://silancarbedas.bandungkab.go.id/ secara otomatis. Berikut ini langkah-langkah dalam mengidentifikasi celah kerentanan keamanan menggunakan OWASP ZAP.

Pertama buka apliaksi OWASP ZAP. Jika sudah terbuka maka akan muncul dua pilihan yang disediakan untuk melakukan *penetration testing* oleh aplikasi OWASP ZAP yaitu *Automated Scan dan Manual Explore*, dapat dilihat pada Gambar 2 dibawah ini.



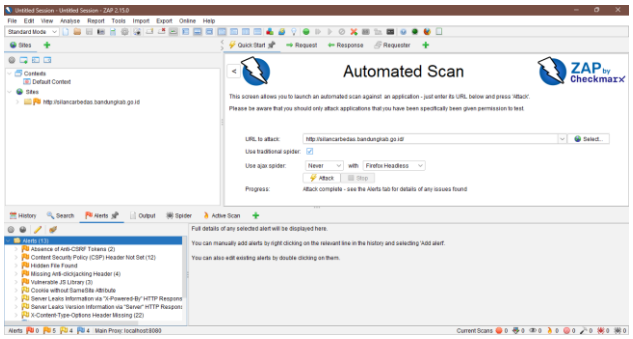
Gambar 2. Tampilan Awal OWASP ZAP

Kemudian, dalam penelitian ini peneliti akan menggunakan Automated Scan, maka klik Automated Scan, lalu masukan *link* target sistem informasi yang akan dilakukan *penetration testing*. Setelah itu klik *attack*, maka aplikasi OWASP ZAP akan langsung otomatis melakukan proses *penetration testing* dan hasil ujinya akan langsung muncul dibagian bawah pada bagian *Alerts*. Untuk proses pengujian serta hasil dari proses *penetration testing* dari website http://silancarbedas.bandungkab.go.id/ dapat dilihat pada gambar 3 dan 4 dibawah ini.



Gambar 3. Proses Penetration Testing

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v22i3.1421



Gambar 4. Hasil Kerentanan yang Ditemukan

Berdasarkan hasil proses *penetration testing* dapat diketahui website http://silancarbedas.bandungkab.go.id/ memiliki 13 keretanan yang ditemukan berdasarkan tingkat resiko, yang disajikan pada Tabel 1 berikut ini.

Tabel 1. Tabel Pengelompokan Level Kerentanan

Tingkat Resiko	Jumlah	Kerentanan
***High* (Kritis)**	0	Tidak Ditemukan
***Medium* (Menengah)** ***Low* (Rendah)**	5	*Absence of Anti-CSRF Tokens*
		Content Security Policy (CSP) Header Not Set
		Hidden File Found
		Missing Anti-clickjacking Header
		Vulnerable JS Library
***Low* (Rendah)**	4	*Cookie without SameSite Attribute*
		Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
		Server Leaks Version Information via "Server" HTTP Response Header Field
		X-Content-Type-Options Header Missing
***Informational* (Informatif)**	4	*Authentication Request Identified*
		Information Disclosure - Suspicious Comments

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v22i3.1421

	Modern Web Application
	Session Management Response Identified

Dari hasil pengujian tersebut maka setiap kerentanan yang ditemukan dikelompokkan berdasarkan *framework* OWASP Top 10:2021 untuk kemudian dianalisis. Adapun hasil pengelompokannya dapat dilihat pada tabel 2 dibawah ini.

Tabel 2. Kerentanan berdasarkan kategori OWASP Top 10:2021

OWASP Top 10	Kerentanan
A01 - *Broken Access Control*	1. *Absence of Anti-CSRF Tokens* 2. *Cookie without SameSite Attribute* 3. *Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)* 4. *Information Disclosure - Suspicious Comments*
A02 - *Cryptographic Failures*	Tidak Ditemukan
A03 - *Injection*	Tidak Ditenukan
A04 - *Insecure Design*	Tidak Ditemukan
A05 - Security Misconfiguration	1. *Content Security Policy (CSP) Header Not Set* 2. *Hidden File Found* 3. *Missing Anti-clickjacking Header* 4. *Server Leaks Version Information via "Server" HTTP Response Header Field* 5. *X-Content-Type-Options Header Missing*
A06 - Vulnerable and Outdated Components	1. *Vulnerable JS Library* 2. *Modern Web Application*
A07 - Identification and Authentication Failures	1. *Authentication Request Identified* 2. *Session Management Response Identified*

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v22i3.1421

A08 - *Software and Data Integrity Failures*	Tidak Ditemukan
A09 - *Security Logging and Monitoring Failures*	Tidak Ditemukan
A10 - *Server-Side Request Forgery* (SSRF)	Tidak Ditemukan

Adapun deskripsi lebih dari 13 kerentanan yang ditemukan dijelaskan sebagai berikut.

1. *Absence of Anti-CSRF Tokens*
   Cross-Site Request Forgery (CSRF) merupakan jenis serangan yang memanfaatkan kelemahan aplikasi untuk memaksa pengguna melakukan permintaan HTTP tertentu tanpa sepengetahuan atau persetujuan mereka. Serangan ini biasanya terjadi karena aplikasi memiliki URL atau aksi formulir yang dapat ditebak dengan mudah. CSRF bekerja dengan mengeksploitasi kepercayaan situs web terhadap pengguna, sementara *Cross-Site Scripting* (XSS) justru memanfaatkan kepercayaan pengguna terhadap situs web. Walaupun namanya mengesankan adanya interaksi antar situs, CSRF tidak selalu melibatkan lebih dari satu situs. Cross-Site Request Forgery juga dikenal dengan sebutan CSRF, XSRF, *one-click attack*, *session riding, confused deputy,* dan *sea surf.*
2. *Content Security Policy* (CSP) *Header Not Set*
   Content Security Policy (CSP) atau Kebijakan Keamanan Konten adalah lapisan keamanan tambahan yang membantu mendeteksi dan mengurangi jenis serangan tertentu, termasuk *Cross Site Scripting* (XSS) dan serangan injeksi data. Serangan-serangan ini digunakan untuk berbagai tujuan, mulai dari pencurian data hingga perusakan situs atau distribusi malware. Dengan CSP, pemilik situs dapat menentukan sumber konten apa saja yang diizinkan untuk dimuat oleh b*rowser, termasuk JavaScript, CSS, frame HTML, font*, gambar, dan objek yang dapat disematkan seperti Applet Java, ActiveX, serta file audio dan video. Pengaturan ini membantu melindungi halaman dari pemuatan konten yang tidak diinginkan.
3. *Hidden File Found*
   Terdapat file sensitif yang ditemukan dalam sistem dan dapat diakses. File ini berpotensi mengungkap informasi penting, seperti konfigurasi sistem atau kredensial, yang bisa dimanfaatkan oleh penyerang untuk menyerang sistem atau melakukan upaya rekayasa sosial.
4. *Missing Anti-clickjacking Header*
   Respon server tidak menyertakan *Content-Security-Policy* dengan direktif *'frame-ancestors'* atau *'X-Frame-Options'*. Hal ini membuat situs rentan terhadap serangan clickjacking, yaitu metode manipulasi tampilan yang dapat mengecoh pengguna.
5. *Vulnerable JS Library*
   Versi *library JavaScript* yang digunakan, seperti *Bootstrap* 3.3.6, diketahui memiliki kerentanan keamanan yang dapat dieksploitasi.

6. *Cookie without SameSite Attribute*
   Cookie yang teridentifikasi disetel tanpa atribut *SameSite* yang berpotensi dikirim dalam permintaan antar situs (*cross-site*). Hal ini meningkatkan risiko serangan seperti CSRF, XSSI, dan *timing attacks*. Atribut *SameSite* penting untuk membatasi cakupan penggunaan *cookie*, sehingga memperkuat keamanan.
7. *Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*
   Header *'X-Powered-By'* dalam respon HTTP server mengungkap informasi tentang *framework* atau teknologi yang digunakan. Informasi ini dapat digunakan oleh penyerang untuk mengidentifikasi kerentanan dalam sistem.
8. *Server Leaks Version Information via "Server" HTTP Response Header Field*
   Header *'Server'* dalam respon HTTP mengungkap versi server yang digunakan. Informasi ini dapat mempermudah penyerang dalam menemukan dan mengeksploitasi kerentanan spesifik.
9. *X-Content-Type-Options Header Missing*
   Header Anti-MIME-Sniffing X-Content-Type-Options tidak disetel ke *'nosniff'*. Hal ini memungkinkan versi lama dari *Internet Explorer* dan *Chrome* untuk melakukan *MIME-sniffing* pada respon, yang berpotensi menyebabkan tipe konten diinterpretasikan dan ditampilkan sebagai jenis konten yang salah dan membuka celah untuk serangan.
10. *Authentication Request Identified*
    Permintaan yang diberikan telah diidentifikasi sebagai permintaan otentikasi. *Field 'Other Info'* berisi sekumpulan baris *key=value* yang mengidentifikasi *field* terkait. Jika permintaan tersebut berada dalam konteks yang memiliki Metode Otentikasi yang disetel ke *'Auto-Detect'*, maka aturan ini akan mengubah otentikasi untuk menyesuaikan dengan permintaan yang diidentifikasi.
11. *Information Disclosure - Suspicious Comments*
    Komentar dalam kode atau respons yang ditemukan mungkin mengandung informasi sensitif yang dapat memberikan petunjuk kepada penyerang. Komentar ini bisa berupa catatan atau detail teknis yang tidak seharusnya terlihat oleh pihak luar.
12. *Modern Web Application*
    Aplikasi ini memiliki karakteristik sebagai aplikasi web modern, yang sering kali menggunakan teknologi seperti *Ajax*. Untuk eksplorasi otomatis, *Ajax Spider* mungkin lebih efektif dibandingkan spider tradisional.
13. *Session Management Response Identified*
    Respon yang dihasilkan mengandung token manajemen sesi. Informasi ini dapat digunakan untuk metode manajemen sesi berbasis *header*, yang memungkinkan sistem menyesuaikan pengelolaan sesi sesuai kebutuhan.

Selanjutnya dari 13 kerentanan yang teridentifikasi maka terdapat rekomendasi solusi untuk menangani keretanan yang dapat menyebabkan website terserang oleh pihak yang tidak bertanggung jawab atau kemungkinan terjadinya kebocoran data. Berikut merupakan rekomnedasi solusi yang disarankan.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v22i3.1421

1. *Absence of Anti-CSRF Tokens*
   - Fase Arsitektur dan Desain:
     Gunakan pustaka atau *framework* yang telah teruji untuk mencegah kerentanan ini. *Framework* tersebut sebaiknya memiliki fitur bawaan yang mendukung penghindaran kelemahan ini. Contohnya, gunakan paket *anti-CSRF* seperti *OWASP CSRFGuard*.
     Tambahkan *nonce* unik untuk setiap formulir. *Nonce* ini harus disisipkan dalam formulir dan diverifikasi saat formulir diterima. Pastikan *nonce* tidak dapat diprediksi (CWE-330). Identifikasi dan beri perlakuan khusus pada operasi yang dianggap berisiko tinggi. Untuk operasi semacam ini, gunakan mekanisme konfirmasi tambahan, misalnya permintaan konfirmasi terpisah. Jangan gunakan metode 'GET' untuk permintaan yang memicu perubahan status.
   - Fase Implementasi:
     Pastikan bahwa aplikasi bebas dari serangan *cross-site scripting* (XSS) atau skrip lintas situs, karena sebagian besar perlindungan CSRF dapat dilewati menggunakan skrip yang dikendalikan oleh penyerang. Pastikan juga bahwa permintaan berasal dari halaman yang sesuai. Namun, catat bahwa langkah ini mungkin memengaruhi beberapa pengguna atau *proxy* yang menonaktifkan pengiriman *'Referer'* karena alasan privasi.
2. *Content Security Policy (CSP) Header Not Set*
   Pastikan bahwa server web, server aplikasi, *load balancer*, dll. telah dikonfigurasi untuk menetapkan *header Content-Security-Policy* untuk membatasi jenis konten yang dapat dimuat.
3. *Hidden File Found*
   Evaluasi apakah file atau komponen yang ditemukan benar-benar diperlukan di lingkungan produksi. Jika tidak diperlukan, nonaktifkan file tersebut. Jika file tetap diperlukan, pastikan file tersebut dilindungi dengan autentikasi dan otorisasi yang memadai, atau batasi akses hanya pada IP tertentu atau sistem internal.
4. *Missing Anti-clickjacking Header*
   Konfigurasikan aplikasi ini untuk menggunakan *header HTTP Content-Security-Policy* atau *X-Frame-Options* pada semua halaman web. Jika halaman hanya boleh di-frame oleh server yang telah digunakan, gunakan nilai *'SAMEORIGIN'*, jika tidak diharapkan di-frame sama sekali, gunakan *'DENY'*. Sebagai opsi tambahan, gunakan direktif *frame-ancestors* dari *Content-Security-Policy.*
5. *Vulnerable JS Library*
   Perbarui pustaka *JavaScript* yang digunakan, seperti *Bootstrap*, ke versi terbaru untuk memastikan tidak ada kerentanan yang dieksploitasi.
6. *Cookie without SameSite Attribute*
   Atur atribut *SameSite* pada semua *cookie* ke *'lax*' atau, jika memungkinkan, ke *'strict'* untuk meningkatkan perlindungan terhadap serangan CSRF.

7. *Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*
   Nonaktifkan atau sembunyikan *header* "*X-Powered-By*" pada server web, server aplikasi, atau load balancer untuk mengurangi informasi yang bisa dimanfaatkan oleh penyerang. Pastikan bahwa server web, server aplikasi, load balancer, dll. dikonfigurasi untuk menekan *header "X-Powered-By*".
8. *Server Leaks Version Information via "Server" HTTP Response Header Field*
   Pastikan bahwa server web, server aplikasi, load balancer, dll. dikonfigurasi untuk menyembunyikan header "Server" atau menggantinya dengan informasi yang bersifat umum..
9. *X-Content-Type-Options Header Missing*
   Pastikan bahwa aplikasi/server web menyetel *header Content-Type* dengan tepat, dan menambhakan *header X-Content-Type-Options* ke *'nosniff'* untuk semua halaman web guna mencegah *MIME-sniffing.*
   Jika memungkinkan, pastikan bahwa pengguna akhir menggunakan peramban web modern yang mematuhi standar yang tidak melakukan *MIME-sniffing* sama sekali, atau yang dapat diarahkan oleh aplikasi/web server untuk tidak melakukan *MIME-sniffing*.
10. *Authentication Request Identified*
    Ini merupakan informasi tambahan, bukan kerentanan, sehingga tidak ada yang perlu diperbaiki.
11. *Information Disclosure - Suspicious Comments*
    Hapus semua komentar yang memberikan informasi sensitif yang dapat dimanfaatkan oleh penyerang, kmeudian tinjau dan perbaiki masalah mendasar yang dirujuk oleh komentar tersebut.
12. *Modern Web Application*
    Ini merupakan informasi tambahan, bukan kerentanan, sehingga tidak ada yang perlu diperbaiki.
13. *Session Management Response Identified*
    Ini merupakan informasi tambahan, bukan kerentanan, sehingga tidak ada yang perlu diperbaiki.

4. Kesimpulan dan Saran

Berdasarkan hasil penelitian analisis keamanan sistem informasi menggunakan OWASP ZAP pada sistem kearsipan elektronik berbasis website http://silancarbedas.bandungkab.go.id/ di Kabupaten Bandung, dapat diketahui celah kerentan yang ada pada website tersebut sebanyak 13 temuan. Hasil pengujian menunjukkan tidak terdapat celah kerentanan berkategori *high* (kritis), sebanyak 13 celah keamanan kerentanan didominasi oleh kategori *medium* (menengah), yaitu sebanyak 5 temuan celah kerentanan, sedangkan sisanya berupa 4 temuan celah kerentanan beresiko *low* (rendah) dan 4 temuan bersifat *informational* (informasi). Secara umum, berdasarkan *framework* OWASP Top 10:2021 seluruh celah keaman tersebut masuk kedalam kategori A01 - *Broken Access Control,* A05 - *Security Misconfiguration,* A06 - *Vulnerable and Outdated Components,* dan A07 - *Identification and Authentication Failures*.

Perbaikan sistem diprioritaskan pada temuan celah kerentanan yang bersifat *medium* (menengah) agar bisa memperbaiki dan menutup celah kerentanan yang ada, sehingga website akan terjaga dari serangan oleh pihak yang tidak bertanggung jawab dan kemungkinan terjadinya kebocoran data, sedangkan untuk yang bersifat *informational* (informasi), tidak menjadi keharusan untuk dilakukan perbaikan.

Daftar Pustaka

[1] P. Haryani, "PENILAIAN KUALITAS LAYANAN WEBSITE PEMERINTAH KOTA YOGYAKARTA MENGGUNAKAN METODE E-GOVQUAL," *J. Ilm. DASI*, vol. 17, no. 3, hlm. 44–50, 2016.

[2] BPPTIK Kominfo, "Jenis-Jenis Serangan Siber di Era Digital," 2023. https://bpptik.kominfo.go.id/Publikasi/detail/jenis-jenis-serangan-siber-di-era-digital.

[3] M. A. Rizaty, "Data Jumlah Serangan Siber ke Indonesia hingga 2023," 2024. https://dataindonesia.id/internet/detail/data-jumlah-serangan-siber-ke-indonesia-hingga-2023.

[4] G. H. A. Kusuma, "IMPLEMENTASI OWASP ZAP UNTUK PENGUJIAN KEAMANAN SISTEM INFORMASI AKADEMIK," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 16, no. 2, hlm. 178–186, 2022, doi: https://doi.org/10.47111/jti.v16i2.3995.

[5] A. W. Kuncoro dan F. Rahma, "Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website : Literature Review," *Pros. Autom.*, vol. 3, no. 1, 2022, [Daring]. Tersedia pada: https://journal.uii.ac.id/AUTOMATA/article/view/21893.

[6] S. A. Febriani, A. Muni, B. Rianto, M. Jalil, dan Chrismondari, "ANALISIS KERENTANAN KEAMANAN SISTEM INFORMASI AKADEMIK MENGGUNAKAN OWASP-ZAP DI UNIVERSITAS ISLAM INDRAGIRI," *J. Sist. Inf.*, vol. 2, no. 6, 2024.

[7] OWASP, "OWASP Top 10: 2021," 2021. https://owasp.org/Top10/.

[8] P. Engebretson, *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy 2nd Edition*. Rockland: Syngress, 2013.

[9] EC-Council, *Certified Ethical Hacker v8 : Module 20 Penetration Testing*. New Mexico, 2012.

[1] P. Haryani, "PENILAIAN KUALITAS LAYANAN WEBSITE PEMERINTAH KOTA YOGYAKARTA MENGGUNAKAN METODE E-GOVQUAL," *J. Ilm. DASI*, vol. 17, no. 3, hlm. 44–50, 2016.

[2] BPPTIK Kominfo, "Jenis-Jenis Serangan Siber di Era Digital," 2023. https://bpptik.kominfo.go.id/Publikasi/detail/jenis-jenis-serangan-siber-di-era-digital.

[3] M. A. Rizaty, "Data Jumlah Serangan Siber ke Indonesia hingga 2023," 2024.

ISSN: 1829-667X / E-ISSN: 2460-9021
DOI:10.31515/telematika.v22i3.1421

https://dataindonesia.id/internet/detail/data-jumlah-serangan-siber-ke-indonesia-hingga-2023.

[4] G. H. A. Kusuma, "IMPLEMENTASI OWASP ZAP UNTUK PENGUJIAN KEAMANAN SISTEM INFORMASI AKADEMIK," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 16, no. 2, hlm. 178–186, 2022, doi: https://doi.org/10.47111/jti.v16i2.3995.

[5] A. W. Kuncoro dan F. Rahma, "Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website : Literature Review," *Pros. Autom.*, vol. 3, no. 1, 2022, [Daring]. Tersedia pada: https://journal.uii.ac.id/AUTOMATA/article/view/21893.

[6] S. A. Febriani, A. Muni, B. Rianto, M. Jalil, dan Chrismondari, "ANALISIS KERENTANAN KEAMANAN SISTEM INFORMASI AKADEMIK MENGGUNAKAN OWASP-ZAP DI UNIVERSITAS ISLAM INDRAGIRI," *J. Sist. Inf.*, vol. 2, no. 6, 2024.

[7] OWASP, "OWASP Top 10: 2021," 2021. https://owasp.org/Top10/.

[8] P. Engebretson, *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy 2nd Edition*. Rockland: Syngress, 2013.

[9] EC-Council, *Certified Ethical Hacker v8 : Module 20 Penetration Testing.* New Mexico, 2012.