Journal of Information Technology and Computer Science (INTECOMS)
Volume 9 Nomor 2, Tahun 2026
e-ISSN : 2614-1574
p-ISSN : 2621-3249

ANALISIS KERENTANAN KEAMANAN APLIKASI KATALOG LAYANAN ELEKTRONIK MENGGUNAKAN *FRAMEWORK* OWASP 2021

ANALYSIS OF SECURITY VULNERABILITIES IN THE ELECTRONIC SERVICE CATALOG APPLICATION USING THE OWASP 2021 FRAMEWORK

Winda Mayasari[1], Edi Sukirman[2], Hustinawaty[3]
Universitas Gunadarma[1,2,3]
imoet.maniz@gmail.com[1], ediskm@staff.gunadarma.ac.id[2], hustina@staff.gunadarma.ac.id[3]

ABSTRACT

Ransomware attacks and data leaks in web-based applications are currently rampant, as has been the case with Indonesian Islamic banks and e-commerce applications. This occurs because web-based applications are built without adequate security systems. The Electronic Service Catalog Application is a web-based application designed to provide electronic services to stakeholders. This web-based application security vulnerability analysis study aims to identify vulnerabilities in the Electronic Service Catalog Application using the OWASP (Open Web Application Security Project) 2021 framework before they are discovered by unauthorized parties. The vulnerability analysis revealed three vulnerabilities in the Electronic Service Catalog Application: broken access control, insecure design, and security misconfiguration. The researchers provide technical recommendations to address these three vulnerabilities and strengthen the Electronic Service Catalog Application. This research is expected to make a significant contribution to strengthening web-based applications.

Keywords*: OWASP, Vulnerability, Application, Security*

ABSTRAK

Serangan ransomware dan kebocoran data aplikasi berbasis web marak terjadi saat ini sebagaimana yang terjadi pada bank Syariah Indonesia dan Aplikasi e-commerce. Hal ini terjadi karena aplikasi berbasis web dibangun tanpa adanya sistem keamanan yang memadai. Aplikasi Katalog Layanan Elektronik merupakan aplikasi berbasis web untuk memberikan layanan secara elektronik kepada *stakeholder*. Penelitian analisis kerentanan keamanan aplikasi berbasis web bertujuan untuk menemukan kerentanan pada Aplikasi Katalog Layanan Elektronik menggunakan *framework* OWASP (Open Web Application Security Project) 2021 sebelum ditemukan oleh pihak yang tidak berwenang. Hasil analisis kerentanan, ditemukan tiga kerentanan pada Aplikasi Katalog Layanan Elektronik yaitu *broken access control*, *insecure design* dan *secutiry misconfiguration*. Peneliti memberikan rekomendasi teknis untuk mengantisipasi 3 kerentanan tersebut untuk memperkuat Aplikasi Katalog Layanan Elektronik. Penelitian ini diharapkan dapat memberi kontribusi yang nyata dalam memperkuat aplikasi berbasis web.

Kata Kunci: OWASP, Kerentanan, Aplikasi, Keamanan

PENDAHULUAN

Seiring masifnya digitalisasi di berbagai sektor, ancaman terhadap keamanan siber meningkat secara signifikan. Aplikasi berbasis web menjadi salah satu target utama para pelaku kejahatan siber. Serangan tidak hanya berdampak pada kerugian finansial, tetapi juga dapat merusak reputasi dan mengancam keamanan data pengguna.

Salah satu *framework* yang diakui secara internasional dalam mendeteksi kerentanan aplikasi web adalah OWASP (*Open Web Application Security Project*). OWASP menyediakan daftar sepuluh besar (OWASP Top 10) jenis kerentanan yang paling sering terjadi dan berbahaya, seperti *Broken Access Control*, *Cryptographic Failures*, *Injection*, *Insecure Design*, dan *Security Misconfiguration*. *Framework* ini menjadi acuan utama bagi pengembang, auditor keamanan, dan peneliti untuk meningkatkan ketahanan aplikasi terhadap serangan.

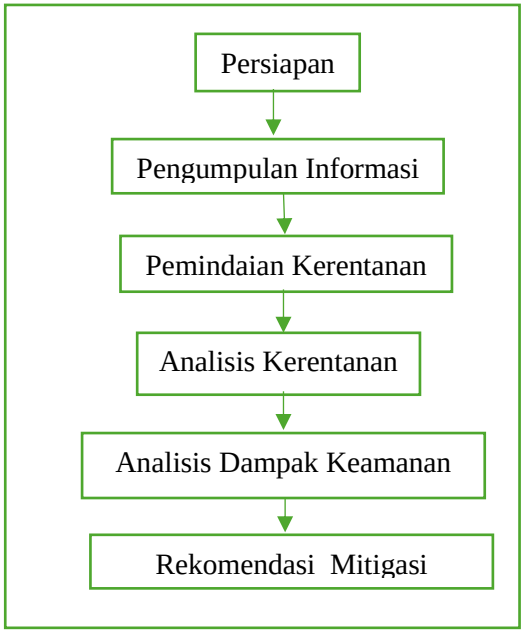
Beberapa penelitian terdahulu yang menggunakan *framework* OWASP dalam menganalisis keamanan aplikasi, diantaranya *Security analysis of web-based academic information system using OWASP framework* (Elfatiha et al, 2024),

menganalisis keamanan sistem informasi akademik Institut Bisnis Muhammadiyah Bekasi menggunakan *framework OWASP TOP 10*. Hasil analisis menunjukkan bahwa sistem rentan terhadap *Broken Authentication*, *Sensitive Data Exposure* dan *Security Misconfiguration*. Penelitian memberikan Rekomendasi Mitigasi perbaikan sistem berupa pembaruan sistem secara berkala, memperbaiki konfigurasi sistem, dan pemantauan sistem secara berkala. *Penetration Testing Web* XYZ Berdasarkan OWASP *Risk* (Priambodo et al., 2023), melaksanakan *penetration testing* dengan metode *black box* untuk mendapatkan hasil pengukuran tingkat kerentanan pada aplikasi. Hasil penelitian menemukan sebanyak 9 jenis kerentanan dengan kategori 2 tinggi, 1 sedang, dan 6 rendah. Hasil penelitian digunakan sebagai referensi pengembang aplikasi web untuk menangani kerentanan khususnya hilangnya ketersediaan layanan dan kebocoran data. *Evaluation of Common Security Vulnerabilities of State Universities and Colleges Websites Based on OWASP* (Flores & Monreal, 2024), menganalisis website menggunakan *tools* OWASP *Zed Attack Proxy* (ZAP) dan *Open Web Application Security Project* (OWASP) Top 10 untuk menganalisis kerentanan keamanan pada 17 situs web universitas dan perguruan tinggi negeri Filipina. Hasil penelitian menemukan kerentanan *injection*, *insecure design*, *outdated components*, *security misconfiguration*, dan *broken access control*. Hasil penelitian berguna untuk memastikan bahwa risiko-risiko pada situs web dapat diminimalkan. *Common Vulnerabilities and Exposures Assessment of Private Higher Educational Institutions Using Web Application Security* (Mangaoang & Monreal, 2024), penelitian menggunakan *tools* OWASP *Zed Attack Proxy* dan OWASP Top 10 untuk mengevaluasi kerentanan pada situs web pada 7 institusi pendidikan tinggi swasta. Hasil penelitian menunjukan terdapat kerentanan *broken access control*, *insecure design*, dan *software and data integrity failures*. Hasil ini membantu institusi dalam mengenali dan mencegah kerugian atau kerusakan lebih lanjut.

Berdasarkan hal tersebut di atas, Peneliti melakukan penelitian pada Aplikasi Katalog Layanan Elektronik yang baru dikembangkan Perusahaan tahun 2025 menggunakan *framework* OWASP 2021. Penelitian dilakukan dengan memindai Aplikasi Katalog Layanan Elektronik menggunakan OWASP ZAP, dan menganalisis kerentanan Aplikasi Katalog Layanan Elektronik dengan *framework* OWASP 2021 yang mencakup *Broken Access Control*, *Cryptographic Failures*, *Injection*, *Insecure Design*, *Security Misconfiguration*, *Vulnerable and Outdated Components*, dan *Identification and Authentication Failures*. Penelitian bertujuan untuk mengidentifikasi celah keamanan Aplikasi Katalog Layanan Elektronik dan memberikan Rekomendasi Mitigasi teknis guna memperkuat sistem terhadap potensi serangan siber. Hasil penelitian diharapkan dapat berkontribusi dalam membangun sistem informasi yang lebih aman dan tangguh terhadap ancaman dunia maya yang terus berkembang.

METODE

Analisis kerentanan keamanan Aplikasi Katalog Layanan Elektronik menggunakan *framework* OWASP TOP 10 2021 dengan tahapan penelitian sebagaimana pada gambar 1 sebagai berikut:



Gambar 1. Metode Penelitian

Berdasarkan gambar 1, proses penelitian dibagi ke dalam beberapa tahapan yang saling berurutan, dimulai dari tahap persiapan, yaitu penyiapan seluruh perangkat dan intrumen yang digunakan. Tahap berikutnya adalah pengumpulan informasi terkait aplikasi, baik yang diperoleh dari dokumen pendukung maupun melalui teknik enumerasi menggunakan perangkat khusus. Informasi yang terkumpul kemudian menjadi dasar untuk melakukan pemindaian kerentanan dengan memanfaatkan OWASP ZAP, yang berfungsi mengidentifikasi potensi celah keamanan dalam aplikasi. Hasil pemindaian selanjutnya dianalisis untuk menentukan jenis kerentanan yang ditemukan, kemudian dilakukan evaluasi dampak terhadap aspek kerahasiaan, integritas, dan ketersediaan sistem. Tahap akhir dari penelitian ini adalah penyusunan Rekomendasi Mitigasi perbaikan yang ditujukan kepada pengelola aplikasi agar risiko keamanan yang ditemukan dapat diminimalisasi.

HASIL DAN PEMBAHASAN

Setelah dilakukan pengujian dan analisis lebih lanjut, maka diperoleh hasil sebagai berikut:

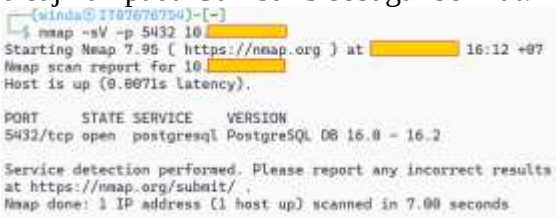
1) Pengumpulan Informasi

Peniliti melakukan *fingerprinting* WhatWeb untuk mengidentifikasi komponen penyusun situs web secara pasif. Hasil pemindaian disajikan pada Gambar 2 sebagai berikut:

Gambar 2. *Fingerprinting* WhatWeb

Berdasarkan Gambar 2, teridentifikasi bahwa aplikasi dibangun menggunakan standar HTML5 dan menggunakan beberapa *header* keamanan seperti *Strict-Transport-Security* (HSTS), *X-Frame-Options* (SAMEORIGIN), dan *X-XSS-Protection*. Informasi ini memberikan gambaran awal mengenai postur keamanan pada level HTTP *response header* yang akan dianalisis lebih mendalam pada tahap pengujian *Security Misconfiguration*.

Selain identifikasi komponen web, peneliti juga melakukan pemindaian spesifik untuk mengetahui versi basis data yang digunakan melalui perintah nmap -sV -p. Hasil pemindaian *port default* basis data disajikan pada Gambar 3 sebagai berikut:



Gambar 3. Pemindaian *port default* basis data

Berdasarkan hasil pemindaian pada Gambar 3, sistem menggunakan layanan PostgreSQL DB versi 16.0 - 16.2 yang berjalan pada *port* 5432. Informasi versi yang spesifik ini sangat penting bagi peneliti untuk melakukan pencarian pada basis data kerentanan publik (*Common Vulnerabilities and Exposures* - CVE) guna memastikan apakah versi basis data tersebut memiliki celah keamanan yang sudah diketahui atau memerlukan pembaruan (*patching*).

2) Pemindaian Kerentanan

Berdasarkan pemindaian kerentanan menggunakan OWASP ZAP diperoleh hasil sebagaimana pada gambar 4 sebagai berikut:

Gambar 4. Hasil pemindaian OWASP ZAP

Berdasarkan gambar 4, terdapat banyak kerentanan yang ditemukan hasil pemindaian otomatis. Namun Peneliti hanya melakukan analisis lebih lanjut pada kerentanan dengan tingkat risiko tinggi dan sedang yaitu *Path Traversal, SQL Injection, Spring4Shell, Absence of Anti-CSRF Tokens, CSP, Cross-Domain Misconfiguration, Format String Error, Hidden File* dan *XSLT Injection*.

3) Analisis Kerentanan

Peneliti melakukan analisis kerentanan Aplikasi Katalog Layanan Elektronik yang dikategorikan berdasarkan *framework* OWASP TOP 10 2021 dengan hasil sebagai berikut:

1. *Broken Access Control* (A01:2021)

Peneliti melakukan analisis lebih lanjut terhadap hasil pemindaian otomatis *path traversal* dan *hidden file found* yang masuk dalam kategori *broken access control*. Pengujian *path traversal* dilakukan dengan menyisipkan *payload* sekuensial ../../ yang diarahkan pada berkas sensitif sistem operasi (/etc/passwd). Langkah ini bertujuan untuk menguji apakah mekanisme *backend* melakukan proses sanitasi terhadap input string sebelum dieksekusi oleh fungsi pemanggilan berkas (*file retrieval function*) pada peladen. Hasil pengujian ditunjukan pada gambar 5 sebagai berikut:

Gambar 5. Hasil pengujian path traversal

Berdasarkan gambar 5, hasil pengujian menunjukkan bahwa sistem memberikan respons 401 Unauthorized, artinya sistem telah membatasi akses pada level direktori.

Pengujian *hidden file found*, dilakukan dengan mengakses direktorat yang menjadi temuan pemindaian otomatis (._darcs, .bzr, .hg, BitKeeper) melalui peramban. Hasil pengujian ditunjukan pada gambar 6 sebagai berikut:

Gambar 6. Hasil pengujian *hidden file found*

Berdasarkan gambar 6 web memberikan respon JSON: {"code":"02","error":"Not Found"}. Munculnya pesan *"Not Found"* menunjukkan bahwa aplikasi atau server web telah memiliki lapisan proteksi yang mampu mengenali dan menolak permintaan terhadap sumber daya yang tidak valid atau berada di luar cakupan akses publik.

Selain itu, peneliti melakukan pengujian *privilege escintrumenion*, dengan mengakses path admin pada login user sebagaimana pada gambar 7 sebagai berikut:

Gambar 7. Kerentanan *privilege escintrumenion*

Berdasarkan gambar 7 sistem memberikan izin akses ke fungsi "API Request" yang secara administratif merupakan bagian dari manajemen kredential dan identitas sistem sehingga user biasa dapat mengakses path admin /xxx/api-request dan /xxx/request-sso tanpa login sebagai admin.

2. *Injection* (A03:2021)

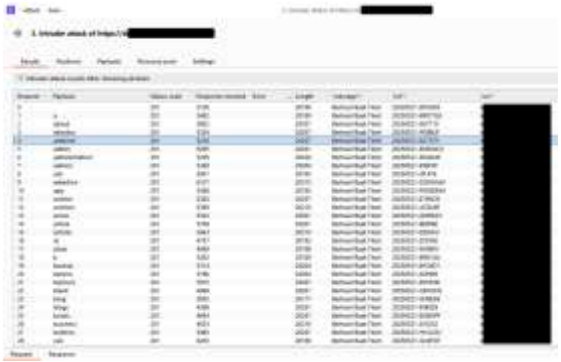
Peneliti melakukan analisis lebih lanjut terhadap hasil pemindaian otomatis *SQL Injection – SQLite* yang masuk dalam kategori *injection*. Peneliti melakukan validasi menggunakan instrumen *automated SQL injection tool* (sqlmap) untuk mengidentifikasi potensi kerentanan keamanan pada aplikasi. Berdasarkan pengujian yang dilakukan terhadap parameter POST, diperoleh hasil sebagaimana pada Gambar 8 sebagai berikut:

Gambar 8. Hasil SQLMap

Berdasarkan Gambar 8, parameter yang diuji tidak menunjukkan indikasi kerentanan terhadap SQL Injection. Sistem memberikan informasi "*all tested parameters do not appear to be injectable*", artinya tidak ada *payload* standar yang berhasil memanipulasi kueri basis data pada sisi *backend*.

3. *Insecure Design* (A04:2021)

Meskipun tidak ada temuan dari hasil pemindaian otomatis terkait kerentanan ini, namun saat Peneliti melakukan analisis lebih lanjut, ditemukan kerentanan bahwa aplikasi belum menangani pengiriman data secara masif dan berulang (*automated submission*) pada *form tiketing*. Berdasarkan pengujian menggunakan instrumen *Burp Suite Intruder*, ditemukan bahwa Aplikasi Katalog Layanan Elektronik belum menerapkan mekanisme pembatasan frekuensi pengiriman formulir (*rate limiting*) atau proteksi otomatisasi seperti CAPTCHA pada *form ticketing* sebagaimana pada Gambar 9 sebagai berikut:



Gambar 9. Hasil Intruder Burpsuite

Berdasarkan Gambar 9, melalui *intruder*, Peneliti berhasil mengirimkan puluhan permintaan (*payload*) dalam waktu singkat. Sebagaimana terdokumentasi pada kolom *"Status code"*, sistem secara konsisten memberikan respon 201 Created untuk setiap permintaan yang dikirimkan. Respon sukses yang berulang tanpa adanya blokir sementara atau tantangan keamanan menunjukkan bahwa tidak ada limitasi pada sisi pelayan (*server-side*) terhadap jumlah pengiriman data dari satu sumber dalam periode tertentu.

Analisis lebih lanjut terhadap keamanan aplikasi dilakukan pada mekanisme validasi input, khususnya terkait pembatasan jumlah karakter yang dapat dikirimkan oleh pengguna melalui formulir. Berdasarkan pengujian penetrasi menggunakan teknik *fuzzing* dan otomatisasi pengiriman data dalam volume besar, ditemukan bahwa Aplikasi Katalog Layanan Elektronik belum menerapkan batasan karakter yang ketat pada kolom input tertentu sebagaimana pada gambar 10 sebagai berikut:

Gambar 10. Belum ada pembatasan karakter

Berdasarkan Gambar 10, aplikasi menerima dan memproses seluruh data yang dikirimkan meskipun jumlah karakter yang dimasukkan melebihi batas kewajaran fungsional kolom tersebut. Hal ini menunjukkan bahwa fungsi validasi pada sisi klien (*client-side*) maupun pelayan (*server-side*) belum mencakup limitasi panjang data (*maxlength*). Ketiadaan batasan ini dapat berdampak pada efisiensi penyimpanan basis data PostgreSQL dan berisiko menyebabkan beban berlebih pada memori server saat melakukan proses kueri data tersebut. ketiadaan pembatasan jumlah karakter membuka celah bagi serangan *Buffer Overflow* (pada level sistem yang lebih rendah) atau serangan *Denial of Service* (DoS) di level aplikasi melalui pengiriman *payload* raksasa yang dirancang untuk menghabiskan sumber daya komputasi server.

Peneliti melakukan evaluasi terhadap integritas seluruh tautan (*link*) yang tersedia pada antarmuka pengguna. Pengujian ini bertujuan untuk memastikan bahwa setiap elemen navigasi mengarahkan pengguna ke sumber daya atau aksi yang valid. Berdasarkan pengujian pada modul Permohonan SSO, ditemukan adanya indikasi tautan atau aksi yang tidak berfungsi sebagaimana mestinya (*void link*) sebagaimana pada Gambar 11 sebagai berikut:

Gambar 11.

Pada Gambar 11, terlihat notifikasi di sudut kanan atas gambar, sistem memberikan respon *"FAILED:* Data gagal disimpan!*"* saat pengguna melakukan interaksi pada tombol submisi. Hal ini mengindikasikan adanya pemutusan alur kerja (*workflow*) antara antarmuka pengguna (*front-end*) dengan pelayan aplikasi (*back-end*). Temuan ini menunjukkan adanya tautan atau *endpoint* API yang bersifat *void*, di mana permintaan yang dikirimkan tidak berhasil diproses oleh basis data PostgreSQL atau gagal divalidasi oleh logika bisnis Express.js.

4. *Security Misconfiguration* (A05:2021)

Berdasarkan hasil pemindaian otomatis menggunakan instrumen OWASP ZAP, ditemukan kerentanan Absence of Anti-CSRP Tokens, yiatu kerentanan yang terjadi karena **tidak adanya mekanisme pengaman** untuk mencegah serangan ***Cross-Site Request Forgery* (CSRF).** Peneliti melakukan analisis lebih lanjut pada konfigurasi CSP Aplikasi Katalog Layanan Elektronik sebagaimana pada gambar 11 sebagai berikut:

Berdasarkan gambar 11, ditemukan bahwa kebijakan CSP yang diterapkan saat ini masih bersifat permisif pada beberapa direktif utama. Terdapat penggunaan karakter wildcard (*) pada direktif font-src, style-src, object-src, dan img-src. Penggunaan tanda bintang ini mengindikasikan bahwa aplikasi mengizinkan pemuatan aset (seperti gambar, font, dan objek) dari domain mana pun di internet. Hal ini memperluas permukaan serangan (attack surface), karena penyerang dapat memanipulasi konten visual atau memuat objek berbahaya dari server eksternal yang tidak terpercaya. ‘Unsafe-inline' mengizinkan eksekusi skrip JavaScript yang ditulis langsung di dalam elemen HTML. Hal ini secara signifikan melemahkan proteksi terhadap serangan XSS, karena peramban tidak dapat membedakan antara skrip sah milik pengembang dan skrip jahat yang diinjeksikan oleh penyerang. 'Unsafe-eval' mengizinkan penggunaan fungsi teks ke-kode seperti eval(). Ini merupakan celah

keamanan kritis yang sering dimanfaatkan untuk eksekusi kode berbahaya.

Peneliti melakukan analisis keamanan *cookie* yang diterbitkan oleh Aplikasi Katalog Layanan Elektronik. Konfigurasi ini sangat krusial dalam memitigasi risiko serangan *Session Hijacking* dan *Cross-Site Request Forgery* (CSRF). Ringkasan konfigurasi atribut *cookie* disajikan pada Gambar 12 sebagai berikut:

Berdasarkan Gambar 12, ditemukan bahwa atribut HttpOnly masih bernilai false pada sebagian besar *cookie* sesi, kondisi ini menunjukkan bahwa *cookie* tersebut dapat diakses melalui skrip di sisi klien (*client-side scripts*) seperti *JavaScript*. Ketiadaan batasan *HttpOnly* meningkatkan risiko kerentanan aplikasi terhadap serangan *Cross-Site Scripting* (XSS), di mana penyerang dapat mengekstraksi token sesi pengguna melalui injeksi skrip berbahaya. Konfigurasi pada kolom *SameSite* menunjukkan nilai None untuk seluruh *cookie* yang terdeteksi. Kebijakan SameSite=None memungkinkan peramban untuk mengirimkan cookie dalam permintaan lintas situs (*cross-site requests).*

4) Analisis Dampak Keamanan

Berdasarkan hasil analisis kerentanan di atas, Peneliti melakukan analisis terhadap dampak yang mungkin terjadi apabila kerentanan digunakan oleh orang yang tidak berwenang. Berikut adalah adalah dampak dari kerentanan yang mungkin terjadi.

Kategori OWASP	Risiko	Dampak Utama	Aset Terancam
A01: *Broken Access Control*	Tinggi	Pengambilan hak akses administratif	Database Data Pengguna
A04: *Insecure Design*	Sedang	Gangguan ketersediaan layanan (DoS)	Resource Server
A05: *Security Misconfiguration*	Sedang	Pencurian sesi dan injeksi skrip (XSS)	Sesi dan Akun Pengguna

5) Rekomendasi Mitigasi

Berdasarkan analisis dan dampak kerentanan di atas, Peneliti merekomendasikan penutupan kerentanan sebagai berikut:

Kategori OWASP	Mitigasi
A01: *Broken Access Control*	- Implementasi *Role-Based Access Control* (RBAC) yang Komprehensif - Penerapan Prinsip *Least Privilege* (Hak Akses Minimum) - Pengaktifan Audit Log dan Monitoring Aktivitas Transaksional
A04: *Insecure Design*	- Implementasi *Rate Limiting* dan Mekanisme *Anti-Automation* - Standarisasi *Comprehensive Server-Side Input Validation* - Audit Periodik dan Penataan Elemen Navigasi Antarmuka
A05: *Security Misconfiguration*	- Restrukturisasi Kebijakan *Content Security Policy* (CSP) - Implementasi *Graceful Error Handling* dan Enkapsulasi Informasi - Penguatan Kontrol Keamanan pada Atribut *Cookie Session*

SIMPULAN

Aplikasi telah menunjukkan upaya penguatan keamanan pada pilar autentikasi, yang dibuktikan dengan implementasi teknologi *Passkey* dan standarisasi kompleksitas kata sandi yang memadai. Kendati demikian, postur keamanan secara menyeluruh masih memerlukan pengerasan (*hardening*) pada lapisan logika aplikasi dan konfigurasi pelayan. Hal ini didasarkan pada identifikasi kerentanan yang persisten dalam kategori *Broken Access Control*, *Insecure Design*, dan *Security Misconfiguration* yang secara kolektif berpotensi mengompromikan stabilitas sistem.

Eksploitasi paling kritis teridentifikasi pada anomali kontrol akses melalui potensi *Privilege Escintrumenion* (A01:2021), memberikan peluang bagi subjek dengan hak akses minimal untuk melakukan *bypass* secara ilegal terhadap

fungsi-fungsi manajerial yang bersifat sensitif. ketiadaan instrumen *Rate Limiting* (A04:2021) serta malfungsi pemrosesan input yang memicu *Format String Error* (A05:2021) terkonfirmasi sebagai celah krusial yang dapat mengakibatkan instabilitas proses latar belakang (*system crash*). Kondisi ini secara sistemik dapat dimanfaatkan oleh penyerang untuk menciptakan gangguan layanan hingga mencapai kondisi *Denial of Service (DoS)*, yang melumpuhkan aksesibilitas aplikasi bagi pengguna sah.

Disarankan adanya penelitian lanjutan yang berfokus pada validasi empiris terhadap implementasi mitigasi yang telah dirumuskan dalam studi ini. Penelitian tersebut bertujuan untuk mengukur tingkat reduksi risiko secara kuantitatif serta memastikan bahwa modifikasi sistem yang dilakukan tidak menimbulkan regresi fungsional atau memicu munculnya vektor serangan baru (*new attack vectors*) yang tidak terduga.

DAFTAR PUSTAKA

Jurnal Ilmiah

M. I. A. Elfatiha, I. R. Riadi, and R. U. Umar (2024), *Security Analysis of Web-Based Academic Information System using OWASP Framework*, *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 9 (4): 353-366.

D. F. Priambodo, A. D. Rifansyah, and M. Hasbi (2023), *Penetration Testing* Web XYZ Berdasarkan OWASP *Risk Rating*, *Teknika*, 12 (1): 33-46.

C. P. Flores and R. N. Monreal (2024), *Evaluation of Common Security Vulnerabilities of State Universities and Colleges Websites Based on OWASP*, J. Electrical Systems, 20-5s: 1396-1404.

E. F. Mangaoang and R. N. Monreal (2024), *Common Vulnerabilities and Exposures Assessment of Private Higher Educational Institutions Using Web Application Security*, J. Electrical Systems, 20-5s: 668-676.

A. Gustiyono, E. I. Alwi, and S. M. Abdullah (2024), Analisa Kerentanan Website Terhadap Serangan *Cross-Site Scripting* (XSS) Metode *Penetration Testing*, CyberSecurity dan Forensik Digital, 7 (1): 25-33.

K. Isnaini, M. H. Asyari, S. F. Amrillah, and D. Suhartono (2024), *Vulnerability Assessment and Penetration Testing on Student Service Center System*, ILKOM Jurnal Ilmiah, 16 (2): 161-171.

Ika Meilina and G. R. Fernandes (2023), *Anticipate Password Security with Burp Suite Using the Brute Force Attack Method*, Jurnal E-Komtek, 7 (1): 118–127.

G. R. Fernandes and Ika Meilina (2024), *Website Penetration Testing With SQL Injection Technique Using SQLMAP on Termux*, Jurnal E-Komtek, 8 (2): 286-293.

Md. A. Masum, Md. R. Istiak Sachcha, and A. Nayem (2022), *Security Analysis of Government & Financial Websites of Bangladesh*, *I.J. Education and Management Engineering*, 12 (2): 21-29.

Ilham F. A., Leonard R. A., Nazla A. W., Siraz T. D. (2023), Analisis Celah Keamanan dan Mitigasi Website E-Learning ITERA Menggunakan OWASP Zed Attack Proxy (ZAP), Dinamika Rekayasa, 19 (1): 29-35.

Saerozi A. N. dan Tri Rochmadi (2024), Analisis Keamanan Sistem Informasi Pusaka Magelang Menggunakan Open Web Application Security Porject (OEASP) dan Information Systems Security Assessment Framework (ISSAF), VyberSecurity dan Forensik Digital, 7 (1): 56-61.

Ferzha P. U., dan R. M. Hilmi N. (2024), *Uncovering the Risk of Academic Information System Vulnerability through PTES and OWASP Method*, CommIT Journal 18 (1): 39-51.

Sabariman, Haeruddin, dan Deven Lee (2023), Analisis Kerentanan Aplikasi Akademik Berbasis Website XYZ Menggunakan OWASP, Jurnal Khatullistiwa Informatika, 11 (2): 92-102.

Agiska R. S., Imam A., Septafiansyah D. P., dan Eko S. (2024), Analisis Kerentanan Aplikasi Web E-commerce Berdasarkan Standar OWASP Top 10: Studi Kasus pada Situs Kopi Lampung Nusantara, EXPERT, 14 (2): 95-102.

Tamsir A., Hidayatul F., Taufik A., dan M. Bimo Prihandoko (2025), Implementasi OWASP untuk Analisis Kerentanan dan Keamanan pada Sistem Informasi Akademik Terintegrasi Universitas Bina Darma, STOARGE, 4 (1): 1-7.

Website

"Home - OWASP Top 10:2021" diakses pada 24 Agustus 2025, pukul 10.00 dari https://owasp.org/Top10/id/