BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis terhadap aplikasi *website* SIAKAD UPN "Veteran" Jakarta yang berfokus pada identifikasi kerentanan keamanan aplikasi web dengan mengacu pada OWASP *Top Ten* 2021, maka dapat disimpulkan bahwa:

1. OWASP ZAP berhasil digunakan untuk mengidentifikasi lima kategori kerentanan utama dengan *attack* dan *active scan*, yaitu:
    a. *Broken Access Control*, seperti keberadaan *file* tersembunyi (.DS_Store) dan penggunaan metode HTTP *GET* untuk operasi yang seharusnya menggunakan *POST.*
    b. *Injection*, seperti pustaka *Java Script* yang rentan dan potensi serangan XSS dari elemen HTML yang dapat dikendalikan pengguna.
    c. *Security Misconfiguration*, seperti tidak adanya *header* keamanan HTTP (CSP, *X-Frame-Options, X-Content-Type-Options*), konfigurasi *wildcard* yang tidak spesifik, serta kebocoran informasi versi server melalui *header.*
    d. *Identification and Authentication Failures*, seperti tidak adanya token CSRF manajemen sesi yang lemah dan tanggapan *login* yang dapat dianalisis oleh penyerang.
    e. *Server-Side Request Forgery* (SSRF), seperti permintaan server terhadap sumber eksternal tanpa validasi yang memadai.

2. Nmap, Elastic Stack, Git Secrets, Gitleaks, OWASP Dependency-Check, dan OWASP Threat Dragon digunakan untuk mengidentifikasi kerentanan yang lebih kompleks, yaitu yang berada pada lapisan infrastruktur dan desain aplikasi. Oleh karena itu, kerentanan yang ditemukan adalah sebagai berikut:

a. Nmap mendeteksi kerentanan *Cryptographic Failures*, seperti *port* 21 (FTP) yang terbuka, penggunaan HTTP tanpa enkripsi, serta *cookie* yang tidak dikonfigurasi dengan atribut keamanan, seperti *Secure, HttpOnly,* dan *SameSite.*
b. Git Secrets dan Gitleaks mengidentifikasi *Software and Data Integrity Failures*, seperti penyimpanan *password,* token, *private key,* API *key* dalam *file* konfigurasi dan *deployment.*
c. OWASP Dependency-Check menemukan *Vulnerable and Outdated Components*, seperti *library jQuery* v2.1.3, *Bootstrap* v3.3.2, dan *Chart.js* v1.0.1 yang rentan dan belum diperbarui.
d. Elastic Stack dan OWASP Threat Dragon tidak dapat digunakan secara optimal karena keterbatasan akses terhadap data log keamanan dan *Data Flow Diagram* (DFD). Oleh karena itu, dua kategori kerentanan, yaitu *Security Logging and Monitoring Failures* serta *Insecure Design*, tidak dapat dianalisis secara menyeluruh dalam penelitian ini. Namun, berdasarkan prinsip keamanan OWASP, kedua jenis kerentanan ini tetap berpotensi terjadi pada sistem apabila tidak dilakukan perancangan keamanan sejak tahap awal pengembangan dan tidak adanya pencatatan serta pemantauan yang memadai. Ketiadaan pengamanan pada desain logika aplikasi dan minimnya sistem *logging* dapat membuka celah bagi penyerang untuk mengeksploitasi aplikasi tanpa terdeteksi, yang pada akhirnya dapat berdampak serius terhadap integritas dan keamanan sistem secara keseluruhan.

3. Rekomendasi mitigasi terhadap berbagai temuan kerentanan keamanan pada *website* SIAKAD UPN Veteran Jakarta berdasarkan kategori OWASP *Top Ten*, meliputi penguatan konfigurasi sistem, pembaruan komponen rentan, penerapan validasi input dan

autentikasi berlapis, penggunaan enkripsi standar industri, serta penerapan sistem *logging* dan monitoring secara real-time. Rekomendasi ini diharapkan dapat menjadi acuan nyata dalam meningkatkan keamanan aplikasi dan mencegah potensi eksploitasi oleh pihak yang tidak bertanggung jawab. Detail rekomendasi untuk masing-masing kerentanan dapat dilihat pada Tabel 4.8.

5.2 Saran

Berdasarkan hasil pengujian dan analisis kerentanan terhadap aplikasi *website* SIAKAD UPN "Veteran" Jakarta, maka saran yang dapat diberikan untuk meningkatkan keamanan sistem adalah sebagai berikut:

1. Seluruh kerentanan yang telah teridentifikasi perlu segera ditangani dengan menerapkan mitigasi sesuai rekomendasi yang telah disusun.
2. Proses audit dan pengujian keamanan sebaiknya menjadi bagian dari *Software Development Life Cycle* (SDLC). *Tools* seperti OWASP ZAP, Dependency-Check, dan Gitleaks dapat diintegrasikan dalam CI/CD *pipeline* untuk mendeteksi kerentanan sejak dini sebelum aplikasi dirilis ke lingkungan produksi. Pemilihan ketiga *tools* tersebut karena *tools* tersebut mendukung integrasi otomatis dalam CI/CD *pipeline*.
3. Tim UPA TIK UPN "Veteran" Jakarta sebaiknya menyediakan dokumentasi berupa *Data Flow Diagram* (DFD) dan arsitektur sistem SIAKAD secara terstruktur, karena hal ini dapat mendukung proses *threat modeling* dengan lebih efektif, terutama saat menggunakan *tools* seperti OWASP Threat Dragon yang memerlukan pemahaman alur data dan arsitektur sistem untuk identifikasi potensi risiko desain.
4. Sebagai upaya penguatan aspek keamanan, disarankan agar sistem SIAKAD UPN "Veteran" Jakarta terus dikembangkan dengan mempertimbangkan penerapan mekanisme pemantauan dan pencatatan aktivitas, seperti sistem *log* terpusat dan deteksi intrusi. Implementasi fitur-fitur tersebut akan mendukung audit keamanan

yang lebih baik serta penyediaan bukti digital yang dibutuhkan apabila terjadi insiden.