Original Article

e-XXXX-XXXX

Original Article

Analisis Keamanan Aplikasi Web terhadap Kerentanan Security Misconfiguration

Aisyah Atirah ✉ [1], Pingkan Ayu Fitri [2], Rakhmadi Rahman[3]
[1,2,3]Institut Teknologi Bacharuddin Jusuf Habibie, Parepare, Indonesia,
Korespondensi Author: aisyahatirah073@gmail.com

Abstrak:

Penelitian ini bertujuan untuk menganalisis keamanan aplikasi web terhadap kerentanan Security Misconfiguration. Security Misconfiguration merupakan salah satu kerentanan yang sering ditemukan pada aplikasi web dan termasuk dalam kategori OWASP Top 10. Kerentanan ini umumnya terjadi akibat kesalahan konfigurasi pada server maupun aplikasi web, seperti penggunaan konfigurasi default dan kurangnya pengaturan keamanan. Metode penelitian yang digunakan adalah metode deskriptif dengan melakukan pengujian keamanan menggunakan tools OWASP ZAP terhadap aplikasi web contoh. Hasil penelitian diharapkan dapat mengidentifikasi potensi kerentanan serta memberikan rekomendasi perbaikan konfigurasi keamanan aplikasi web.

Keywords: Security Misconfiguration, Keamanan Aplikasi Web, OWASP Top 10, Pengujian Keamanan, OWASP ZAP

Pendahuluan

Aplikasi web saat ini banyak digunakan dalam berbagai bidang, termasuk pendidikan, pemerintahan, dan bisnis, karena kemampuannya menyediakan layanan yang mudah diakses dan efisien. Namun, peningkatan penggunaan aplikasi web juga meningkatkan potensi ancaman terhadap keamanan informasi. Serangan pada aplikasi web dapat menimbulkan kebocoran data, gangguan layanan, hingga kerugian finansial dan reputasi organisasi.

Salah satu kerentanan yang sering ditemukan pada aplikasi web adalah Security Misconfiguration, yang terjadi akibat kesalahan dalam pengaturan konfigurasi sistem, baik pada server, framework, maupun aplikasi itu sendiri. Contoh kesalahan konfigurasi meliputi penggunaan pengaturan default, direktori yang dapat diakses publik, pesan kesalahan yang terlalu rinci, serta layanan yang tidak diperlukan namun tetap aktif. Menurut OWASP Top 10, Security Misconfiguration merupakan salah satu kerentanan yang sering muncul dan memiliki tingkat risiko yang cukup tinggi.

Received: November 14, 2025 – Revised: November 29, 2025 - Accepted: Desember 28, 2025 - Published online: January 10, 2026

Oleh karena itu, analisis keamanan aplikasi web menjadi penting untuk mengidentifikasi potensi kerentanan Security Misconfiguration dan menilai tingkat risikonya. Hasil analisis ini diharapkan dapat membantu pengembang dan pengelola sistem dalam meningkatkan konfigurasi keamanan, sehingga aplikasi web menjadi lebih aman dan andal.

Tinjauan Pustaka

Keamanan Aplikasi Web

Keamanan aplikasi web merupakan upaya untuk melindungi sistem dan data dari berbagai ancaman yang dapat mengganggu kerahasiaan, integritas, dan ketersediaan informasi. Perlindungan ini mencakup penerapan konfigurasi yang tepat, pengelolaan hak akses, serta pengawasan terhadap aktivitas yang mencurigakan agar aplikasi web tetap aman dan dapat diandalkan.

Security Misconfiguration

Security Misconfiguration adalah kondisi di mana konfigurasi sistem, server, atau aplikasi web tidak diatur dengan benar, sehingga menimbulkan celah keamanan. Kesalahan konfigurasi ini dapat berupa penggunaan pengaturan default, pengaturan hak akses yang tidak tepat, direktori publik yang tidak seharusnya dapat diakses, atau layanan yang tidak diperlukan namun tetap aktif. Kerentanan ini sering menjadi target serangan karena kesalahan konfigurasi mudah dieksploitasi oleh pihak yang tidak bertanggung jawab.

OWASP Top 10

OWASP Top 10 adalah daftar sepuluh kerentanan keamanan aplikasi web yang paling kritis dan sering dijadikan acuan oleh pengembang dan auditor keamanan. Security Misconfiguration termasuk salah satu kategori utama dalam daftar ini, menekankan pentingnya konfigurasi yang tepat sebagai bagian dari praktik keamanan aplikasi web yang efektif.

Metode Penelitian

Jenis dan Pendekatan Penelitian

Penelitian ini menggunakan metode deskriptif dengan pendekatan analisis keamanan aplikasi web. Metode deskriptif dipilih karena bertujuan untuk menggambarkan kondisi keamanan aplikasi web sebagaimana adanya, khususnya terkait kerentanan Security Misconfiguration, tanpa melakukan intervensi atau perubahan langsung terhadap sistem yang diuji. Pendekatan ini memungkinkan peneliti untuk mengidentifikasi dan memetakan kesalahan konfigurasi yang berpotensi menimbulkan risiko keamanan.

Objek Penelitian

Objek penelitian adalah sebuah aplikasi web contoh yang digunakan sebagai studi kasus. Aplikasi web ini dianalisis dari sisi konfigurasi keamanan sistem, meliputi pengaturan server web, konfigurasi aplikasi, serta mekanisme keamanan dasar yang diterapkan. Pemilihan aplikasi web contoh bertujuan untuk memberikan gambaran umum mengenai kondisi konfigurasi keamanan aplikasi web yang sering ditemui dalam praktik.

Received: November 14, 2025 – Revised: November 29, 2025 - Accepted: Desember 28, 2025 - Published online: January 10, 2026

Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui tiga tahapan, yaitu:

1. Studi Literatur

Dilakukan dengan mengkaji berbagai sumber pustaka seperti buku, jurnal ilmiah, laporan penelitian, dan dokumentasi resmi OWASP Top 10 yang berkaitan dengan keamanan aplikasi web dan Security Misconfiguration. Studi literatur ini bertujuan untuk memperoleh landasan teori serta kerangka konseptual yang relevan dengan penelitian.

2. Observasi Sistem

Dilakukan dengan mengamati secara langsung konfigurasi aplikasi web dan server yang digunakan. Observasi mencakup pengaturan default sistem, struktur direktori, konfigurasi layanan, serta mekanisme pengamanan yang diterapkan.

3. Pengujian Keamanan

Dilakukan menggunakan tools OWASP ZAP untuk mendeteksi potensi kerentanan Security Misconfiguration. Pengujian ini difokuskan pada identifikasi kesalahan konfigurasi yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab.

Teknik Analisis Data

Data yang diperoleh dari observasi dan pengujian keamanan dianalisis secara kualitatif. Hasil temuan kerentanan diklasifikasikan berdasarkan jenis kesalahan konfigurasi dan tingkat risikonya. Analisis dilakukan dengan membandingkan hasil pengujian dengan standar keamanan aplikasi web yang direkomendasikan oleh OWASP, sehingga dapat memberikan gambaran yang sistematis mengenai kelemahan konfigurasi aplikasi.

Alat dan Bahan Penelitian

Alat dan bahan penelitian meliputi perangkat komputer, sistem operasi, browser web, serta tools pendukung analisis keamanan aplikasi web, yaitu OWASP ZAP. Tools ini digunakan untuk membantu proses pemindaian dan identifikasi kerentanan Security Misconfiguration pada aplikasi web yang diuji.

Tahapan Penelitian

Tahapan penelitian dimulai dengan studi literatur, dilanjutkan observasi terhadap aplikasi web contoh, kemudian dilakukan pengujian keamanan untuk mengidentifikasi kerentanan Security Misconfiguration. Tahap terakhir adalah analisis hasil pengujian serta penyusunan kesimpulan dan rekomendasi perbaikan konfigurasi keamanan aplikasi web.

Hasil dan Pembahasan

Hasil Pengujian Keamanan

Berdasarkan pengujian keamanan menggunakan tools OWASP ZAP, ditemukan beberapa potensi kerentanan Security Misconfiguration pada aplikasi web yang dianalisis. Kerentanan yang teridentifikasi meliputi penggunaan konfigurasi default, tidak diterapkannya header keamanan secara optimal, terbukanya informasi sistem yang seharusnya dibatasi, serta layanan atau direktori yang seharusnya tidak aktif namun tetap tersedia untuk publik. Hasil pemindaian menunjukkan bahwa kerentanan tersebut memiliki tingkat risiko yang berbeda-beda, mulai dari rendah hingga menengah.

Kerentanan dengan risiko rendah umumnya terkait dengan pengaturan default

Received: November 14, 2025 – Revised: November 29, 2025 - Accepted: Desember 28, 2025 - Published online: January 10, 2026

atau pesan kesalahan yang terlalu detail, yang tidak berdampak langsung pada kebocoran data namun dapat memberikan informasi tambahan bagi penyerang. Sementara itu, kerentanan dengan risiko menengah seperti header keamanan yang tidak lengkap dapat dimanfaatkan untuk melakukan serangan tertentu, misalnya cross-site scripting atau clickjacking, apabila dikombinasikan dengan kelemahan lain dalam sistem. Hasil ini menunjukkan bahwa konfigurasi keamanan yang kurang tepat, meskipun tampak sepele, tetap memiliki potensi risiko signifikan jika dimanfaatkan oleh pihak yang tidak bertanggung jawab.

Analisis Kerentanan Security Misconfiguration

Security Misconfiguration terjadi karena kesalahan dalam pengaturan konfigurasi sistem, server, maupun aplikasi web itu sendiri. Dalam kasus aplikasi web yang dianalisis, beberapa faktor utama yang menyebabkan kerentanan ini antara lain:

1. Penggunaan Konfigurasi Default

   Penggunaan username, password, atau pengaturan bawaan sistem yang tidak diubah meningkatkan kemungkinan akses tidak sah. Hal ini merupakan kesalahan konfigurasi yang paling umum dan mudah dieksploitasi.

2. Header Keamanan Tidak Optimal

   Header keamanan, seperti Content Security Policy (CSP), X-Frame-Options, dan Strict-Transport-Security, belum diterapkan dengan benar. Header ini berfungsi melindungi aplikasi web dari serangan berbasis browser, termasuk clickjacking, man-in-the-middle, dan injeksi skrip.

3. Informasi Sistem Terbuka

   Informasi terkait versi server, platform, atau framework aplikasi yang ditampilkan secara publik memudahkan pihak yang berniat menyerang untuk mencari celah spesifik. Informasi semacam ini seharusnya dibatasi agar tidak mudah diakses oleh pengguna biasa.

4. Direktori dan Layanan Tidak Diperlukan Masih Aktif

   Beberapa direktori dan layanan yang seharusnya dinonaktifkan tetap tersedia untuk publik, yang membuka peluang bagi penyerang untuk mengeksplorasi struktur aplikasi web dan menemukan titik lemah tambahan.

Implikasi Kerentanan terhadap Keamanan Aplikasi Web

Kerentanan Security Misconfiguration dapat menimbulkan beberapa risiko keamanan, antara lain kebocoran data, gangguan layanan, dan eksploitasi sebagai titik awal serangan lanjutan. Meskipun tidak semua kerentanan langsung menyebabkan kerusakan, kombinasi beberapa kesalahan konfigurasi dapat meningkatkan kerentanan sistem secara keseluruhan. Dari perspektif manajemen risiko, kondisi ini menunjukkan pentingnya penerapan prinsip secure by default dan defense in depth dalam pengelolaan aplikasi web. Secure by default berarti sistem harus dikonfigurasi secara aman sejak awal instalasi, sedangkan defense in depth menekankan perlindungan berlapis sehingga jika satu mekanisme keamanan gagal, lapisan lain tetap melindungi sistem.

Perbandingan dengan Standar OWASP

Hasil temuan juga dibandingkan dengan rekomendasi OWASP Top 10, khususnya terkait Security Misconfiguration. OWASP menekankan bahwa kerentanan ini sering muncul akibat pengaturan default yang tidak diubah, minimnya hardening pada server, serta kegagalan dalam menerapkan kebijakan keamanan dasar. Hasil pengujian pada aplikasi web contoh sejalan dengan temuan OWASP, di mana beberapa konfigurasi

Received: November 14, 2025 – Revised: November 29, 2025 - Accepted: Desember 28, 2025 - Published online: January 10, 2026

default masih digunakan, header keamanan tidak diterapkan sepenuhnya, dan informasi sensitif dapat diakses publik.

Strategi Mitigasi dan Rekomendasi

Berdasarkan hasil pengujian dan analisis, beberapa langkah mitigasi dapat diterapkan untuk mengurangi risiko Security Misconfiguration, antara lain:

1. Mengubah Pengaturan Default

   Semua username, password, dan pengaturan bawaan sistem harus segera diubah setelah instalasi untuk mencegah akses tidak sah.
2. Menerapkan Header Keamanan Secara Lengkap

   Mengaktifkan CSP, X-Frame-Options, X-XSS-Protection, dan Strict-Transport-Security untuk melindungi aplikasi dari serangan berbasis browser.
3. Membatasi Informasi Sistem yang Terbuka

   Menyembunyikan versi server, framework, dan software lain agar tidak mudah diketahui oleh pihak luar.
4. Menonaktifkan Layanan atau Direktori yang Tidak Diperlukan

   Semua layanan dan direktori yang tidak digunakan sebaiknya dinonaktifkan agar tidak menjadi titik lemah bagi penyerang.
5. Melakukan Audit dan Pemindaian Berkala

   Pengujian rutin menggunakan tools seperti OWASP ZAP atau Nessus dapat membantu mendeteksi konfigurasi yang kurang tepat sebelum dimanfaatkan oleh pihak tidak bertanggung jawab.

Diskusi

Hasil penelitian ini menunjukkan bahwa Security Misconfiguration merupakan salah satu kerentanan yang paling sering muncul pada aplikasi web, baik pada sistem yang sederhana maupun kompleks. Meskipun kerentanan ini terkesan sepele, dampaknya bisa signifikan jika dikombinasikan dengan kelemahan lain. Analisis menunjukkan bahwa banyak kesalahan konfigurasi berasal dari kurangnya kesadaran pengembang atau administrator sistem terhadap prinsip keamanan dasar.

Temuan ini konsisten dengan literatur sebelumnya yang menekankan bahwa pengelolaan konfigurasi yang benar merupakan langkah awal yang krusial dalam menjaga keamanan aplikasi web. Dengan menerapkan praktik keamanan yang tepat, risiko serangan dapat dikurangi secara signifikan, sekaligus meningkatkan kepercayaan pengguna terhadap aplikasi web.

Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa Security Misconfiguration masih menjadi salah satu kerentanan yang paling sering dijumpai pada aplikasi web dan memiliki potensi menimbulkan risiko keamanan yang cukup signifikan. Analisis yang dilakukan menggunakan tools OWASP ZAP memungkinkan identifikasi berbagai kesalahan konfigurasi, seperti penggunaan pengaturan default, terbukanya informasi sistem yang seharusnya dibatasi, serta penerapan header keamanan yang tidak optimal. Meskipun tidak semua kerentanan langsung berdampak pada kebocoran data, kondisi tersebut tetap dapat dimanfaatkan oleh pihak tidak bertanggung jawab sebagai titik awal untuk melakukan serangan yang lebih kompleks. Oleh karena itu, penerapan konfigurasi keamanan yang tepat dan terstandarisasi, pembaruan sistem secara berkala, serta pemantauan keamanan secara berkesinambungan menjadi langkah penting untuk meningkatkan ketahanan aplikasi web terhadap berbagai ancaman siber. Pendekatan ini

Received: November 14, 2025 – Revised: November 29, 2025 - Accepted: Desember 28, 2025 - Published online: January 10, 2026

tidak hanya meminimalkan risiko eksploitasi, tetapi juga mendukung terciptanya lingkungan aplikasi yang lebih aman, handal, dan mampu menjaga integritas, kerahasiaan, serta ketersediaan data secara menyeluruh.

Saran

Berdasarkan temuan penelitian, disarankan agar pengembang dan pengelola aplikasi web melakukan konfigurasi sistem sesuai dengan standar keamanan yang berlaku, memanfaatkan header keamanan secara optimal, serta menutup akses ke informasi sensitif. Selain itu, penting untuk rutin melakukan audit dan pengujian keamanan menggunakan tools seperti OWASP ZAP untuk mendeteksi kerentanan sejak dini. Langkah-langkah ini akan membantu mengurangi risiko eksploitasi, meningkatkan keamanan aplikasi web, serta memastikan kerahasiaan, integritas, dan ketersediaan data tetap terjaga.

Daftar Pustaka

WASP. (2021). OWASP Top 10 Web Application Security Risks. OWASP Foundation. Diakses dari https://owasp.org/www-project-top-ten/

Sommerville, I. (2016). Software Engineering (10th ed.). Boston: Pearson Education.

Behl, A., & Behl, K. (2017). Cybersecurity and Cyberwar. Oxford: Oxford University Press.

Kurniawan, D. (2020). Keamanan Aplikasi Web. Jurnal Teknologi Informasi.Web, 15(2), 45–55.

Sugara, V. I., & Sriyasa, I. (2024). Analisis keamanan web: pengujian kerentanan berdasarkan OWASP Top 10 pada aplikasi web. The Indonesian Journal of Computer Science (IJCS), 13(2).

Sati, D. L., Sita, D. L., & Isnaini, K. N. (2024). Identifikasi celah kerentanan keamanan pada website dengan OWASP ZAP. *Jurnal RESISTOR* (Rekayasa Sistem Komputer), 7(3), 153–161.

Arfan Dwi Madya, R. P., & Nurfiyah. (2025). Analisis kerentanan keamanan website menggunakan OWASP Top 10: studi kasus Web BNPB. *Indonesian Journal of Education and Computer Science*.

Eko Setiawan & Fachri, F. (2025). Pengujian dan mitigasi kerentanan website sistem informasi akademik menggunakan OWASP ZAP. Cyber Security dan Forensik Digital, 8(1), 25–33.

Syarifudin, M., Widyawati, L., & Asroni, O. (2025). Web security vulnerability analysis and mitigation based on OWASP Top 10. *Journal of Artificial Intelligence and Engineering Applications,* 4(3).

Received: November 14, 2025 – Revised: November 29, 2025 - Accepted: Desember 28, 2025 - Published online: January 10, 2026