Deteksi dan Pencegahan Kerentanan Cross-Site Scripting (XSS) Stored dan Broken Access Control pada Aplikasi Web kaspedia.web.id

Eko Pramono*[1], Mukhlis Nur Arif[1], Marcelinus Erix Nugroho[1], and Galih Nur Massaid[1]

1 **Universitas AMIKOM Yogyakarta**
Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta
eko.p@amikom.ac.id, mukhlisnurarif@students.amikom.ac.id;
erixnugroho@students.amikom.ac.id;galihnurm@students.amikom.ac.id

Abstrak

Keamanan siber adalah komponen krusial dalam siklus hidup pengembangan aplikasi web modern, terutama dalam menangani sejumlah besar data sensitif. Aplikasi web, meskipun didukung oleh teknologi keamanan yang terus maju, tetap menjadi target utama bagi serangan siber yang berevolusi. Penelitian ini berfokus pada analisis dan mitigasi dua ancaman keamanan paling signifikan: Cross-Site Scripting (XSS) Stored dan Broken Access Control (BAC) pada aplikasi web kaspedia.web.id. Kerentanan XSS Stored memungkinkan penyerang menyuntikkan skrip berbahaya secara permanen ke database, yang kemudian dieksekusi tanpa disadari oleh browser pengguna lain. Sementara itu, BAC timbul akibat kegagalan validasi akses yang memadai di sisi server, memungkinkan pengguna yang tidak berwenang untuk mengakses sumber daya atau fungsi yang seharusnya terbatas. Untuk mengatasi permasalahan ini, penelitian mengadopsi metodologi pengujian penetrasi yang sistematis dan analisis kode statis. Tahap deteksi melibatkan pengujian fungsionalitas aplikasi menggunakan alat bantu khusus untuk mengidentifikasi titik injeksi XSS dan celah otorisasi. Metode pencegahan yang diimplementasikan meliputi input sanitization dan output encoding yang ketat untuk menetralkan payload XSS, serta penerapan mekanisme server-side authorization yang ketat untuk memverifikasi setiap permintaan akses terhadap kebijakan hak pengguna yang ditetapkan. Hasil penelitian menunjukkan bahwa aplikasi kaspedia.web.id memiliki kerentanan XSS Stored pada modul input data dan kerentanan BAC pada beberapa endpoint manajerial. Setelah intervensi pencegahan diterapkan, kedua jenis kerentanan tersebut berhasil dieliminasi, sehingga secara signifikan meningkatkan postur pertahanan aplikasi. Kesimpulannya, deteksi proaktif dan penerapan mekanisme pencegahan yang terstruktur, yang menggabungkan validasi input dan kontrol akses berbasis server, adalah upaya yang sangat efektif dalam memperkuat keamanan aplikasi web modern terhadap ancaman XSS Stored dan Broken Access Control. Hal ini memastikan integritas dan kerahasiaan data pengguna.

Kata Kunci Keamanan Aplikasi, XSS Stored, Broken Access Control, Pengujian Penetration, Mitigasi Kerentanan

Digital Object Identifier 10.36802/jnanaloka.2025.v6-no2-87-97

1 Pendahuluan

Pengembangan aplikasi web modern melibatkan penanganan sejumlah besar data sensitif, menjadikan keamanan sebagai komponen krusial dalam siklus hidup pengembangan. Pentingnya keamanan ini ditegaskan oleh kebutuhan aplikasi web untuk memiliki mekanisme

* Corresponding author.

Jnanaloka
Jurnal Open Access
Yayasan Lentera Dua Indonesia

perlindungan yang kuat guna mendeteksi dan mencegah kerentanan [1, 2]. Meskipun kemajuan teknologi keamanan terus berkembang, aplikasi web tetap menjadi target utama serangan siber yang terus berevolusi [3]. Di antara berbagai ancaman yang ada, Cross-Site Scripting (XSS) dan Broken Access Control (BAC) secara konsisten diakui sebagai kerentanan kritis yang menduduki peringkat teratas dalam daftar risiko keamanan aplikasi web secara global [4].

Kerentanan *Cross-Site Scripting* (XSS) *Stored*, secara spesifik, memungkinkan penyerang menyisipkan skrip berbahaya ke dalam basis data aplikasi [5]. Skrip ini kemudian disajikan kepada pengguna lain secara otomatis ketika mereka mengakses halaman yang terinfeksi, sehingga dieksekusi di sisi browser pengguna [5]. Dampak dari serangan XSS Stored bisa sangat merugikan, mulai dari pencurian informasi sensitif seperti cookie sesi, pengambilalihan akun pengguna, hingga modifikasi tampilan situs (defacement) [6, 7]. Bahkan hingga tahun 2025, XSS diproyeksikan akan tetap menjadi ancaman siber yang signifikan, didukung oleh kemunculan teknik-teknik canggih dan integrasi kecerdasan buatan dalam metode serangannya [8]. Analisis kasus nyata, seperti kerentanan XSS Stored dengan tingkat keparahan tinggi yang ditemukan pada aplikasi MyCourts (CVE-2025-57424), semakin memperkuat relevansi dan urgensi penanganan kerentanan ini [9].

Di sisi lain, Broken Access Control adalah jenis kerentanan yang paling sering ditemukan, menduduki peringkat pertama dalam OWASP Top 10:2021 [10]. Kerentanan ini muncul ketika aplikasi gagal untuk menerapkan pembatasan otorisasi secara efektif, sehingga memungkinkan penyerang untuk mengakses fungsi, data, atau sumber daya yang seharusnya tidak dapat mereka capai berdasarkan hak akses mereka [10]. Konsekuensi dari broken access control bervariasi luas, mencakup pengungkapan data rahasia, manipulasi atau penghapusan informasi penting, hingga eskalasi hak istimewa yang memungkinkan penyerang untuk mengambil alih kontrol penuh atas akun atau bagian dari sistem [11]. Observasi menunjukkan bahwa mayoritas aplikasi web, yakni sekitar 94%, memiliki setidaknya satu jenis kelemahan kontrol akses [10]. Signifikansi broken access control ini juga tercermin dari posisinya yang terus menonjol dalam pembaruan OWASP Top 10 edisi 2025 [12].

Meskipun berbagai penelitian terdahulu telah membahas mekanisme deteksi dan mitigasi terhadap kerentanan XSS serta Broken Access Control, terdapat beberapa kekosongan riset (research gap) yang belum terjawab secara memadai. Sebagian besar penelitian mengenai XSS berfokus pada pengembangan model deteksi umum berbasis machine learning atau analisis payload [5, 6], sedangkan studi spesifik mengenai XSS Stored pada aplikasi web dengan arsitektur dan formulir input kompleks seperti kaspedia.web.id masih sangat terbatas. Selain itu, penelitian yang ada umumnya hanya mengevaluasi XSS dari perspektif teknis, tanpa melakukan pengujian berbasis konteks aplikasi nyata yang memiliki pola form dinamis dan multi-endpoint.

Pada aspek Broken Access Control, riset-riset sebelumnya lebih banyak menyoroti klasifikasi kerentanan, model eksploitasi, atau analisis umum kelemahan kontrol akses [10–12]. Namun, keterbatasan riset terlihat pada minimnya studi yang secara langsung mengombinasikan deteksi BAC berbasis pengujian akses antar-peran aktual (role escalation testing) dengan verifikasi kerentanan pada aplikasi web nyata, terutama aplikasi yang melibatkan lebih dari satu jenis peran seperti Operator dan Pemilik. Pendekatan tersebut sangat jarang dilakukan dalam penelitian BAC sehingga masih terdapat kekosongan dalam metodologi identifikasi BAC berbasis studi kasus langsung.

Selain itu, sangat sedikit penelitian yang mengintegrasikan pengujian manual dan otomatis secara holistik untuk mendeteksi dua jenis kerentanan sekaligus (XSS Stored dan BAC) dalam satu aplikasi web yang sama. Sebagian besar penelitian hanya fokus pada satu jenis

kerentanan atau hanya menggunakan salah satu pendekatan (manual atau otomatis), sehingga efektivitas deteksi pada konteks aplikasi nyata belum optimal [13]. Berdasarkan celah tersebut, penelitian ini menawarkan kontribusi baru berupa analisis komprehensif keamanan aplikasi web kaspedia.web.id melalui kombinasi deteksi manualotomatis, pengujian eksploitasi langsung, serta evaluasi perbaikan, yang secara khusus menargetkan dua kerentanan kritis (XSS Stored dan Broken Access Control).

2 Metode Penelitian

Penelitian ini menggunakan pendekatan eksperimental terapan (*applied experimental research*) dalam bidang keamanan aplikasi web. Pendekatan ini dipilih karena penelitian berfokus pada pengujian langsung terhadap sistem yang digunakan secara nyata. Fokus penelitian diarahkan pada dua jenis kerentanan kritis yang ditemukan pada aplikasi kaspedia.web.id, yaitu *Cross-Site Scripting* (XSS) *Stored* dan *Broken Access Control* (BAC).

Untuk memperoleh cakupan evaluasi keamanan yang menyeluruh, penelitian ini mengombinasikan dua metode pengujian, yaitu Black-box Testing dan Gray-box Testing. Black-box Testing digunakan untuk mensimulasikan serangan dari pihak eksternal yang tidak memiliki akses maupun pengetahuan internal sistem. Pendekatan ini efektif dalam mendeteksi kerentanan pada fungsi publik aplikasi, khususnya kelemahan validasi input yang dapat menyebabkan XSS Stored [10]. Model pengujian ini juga dianggap paling representatif dalam merepresentasikan pola serangan nyata yang biasanya dilakukan tanpa kredensial internal.

Di sisi lain, Gray-box Testing diterapkan untuk mengevaluasi kerentanan yang berkaitan dengan mekanisme kontrol akses internal. Pengujian terhadap BAC membutuhkan kredensial terbatas dan pemahaman mengenai struktur peran pengguna, sehingga memungkinkan identifikasi horizontal maupun vertikal privilege escalation yang tidak dapat terdeteksi melalui Black-box Testing murni [10]. Temuan penelitian sebelumnya menunjukkan bahwa sebagian besar kelemahan kontrol akses hanya dapat diungkap melalui pengujian berbasis role pengguna [11]. Dengan demikian, kombinasi Black-box dan Gray-box Testing memberikan gambaran evaluasi keamanan yang lebih akurat dan komprehensif. Masing-masing pendekatan menyasar aspek yang berbeda dari permukaan serangan aplikasi, sehingga penggunaan metode hibrida ini sejalan dengan rekomendasi OWASP untuk pengujian keamanan aplikasi web modern [10].

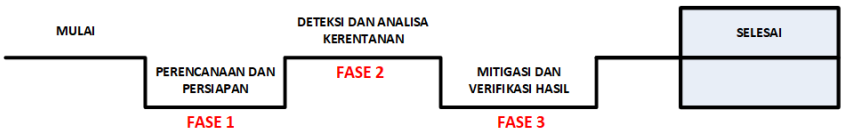
Penelitian ini juga mempertimbangkan aspek penilaian risiko untuk menentukan tingkat keparahan kerentanan. Kajian literatur menunjukkan bahwa studi terkait Cross-Site Scripting (XSS) Stored dan Broken Access Control (BAC) masih didominasi oleh penggunaan Common Vulnerability Scoring System (CVSS) sebagai metode penilaian risiko utama. Meskipun terdapat pendekatan lain seperti OWASP Risk Rating Methodology dan model DREAD, kedua metode tersebut jarang digunakan dalam penelitian keamanan aplikasi web modern [10]. Sebagian besar penelitian memilih CVSS karena bersifat terstandarisasi dan diadopsi secara luas dalam komunitas keamanan siber [14]. Kondisi ini menunjukkan bahwa belum terdapat teknik penilaian risiko alternatif yang digunakan secara konsisten selain CVSS dalam penelitian sejenis, sehingga membuka peluang bagi eksplorasi metode risk scoring yang lebih kontekstual pada aplikasi web.

Penelitian ini dilakukan pada lingkungan aplikasi kaspedia.web.id, sebuah platform berbasis web yang menyediakan layanan operasional dengan struktur akses bertingkat, meliputi peran Operator dan Pemilik. Lingkungan ini dipilih karena memiliki sejumlah fitur kritis berbasis input pengguna serta mekanisme kontrol akses yang kompleks, sehingga relevan untuk pengujian kerentanan XSS Stored dan BAC. Selain itu, aplikasi ini beroperasi pada

arsitektur web modern yang terdiri atas beberapa endpoint dinamis, form input, serta halaman berfungsi publik, yang menjadikannya representatif untuk evaluasi keamanan aplikasi web pada skala UKM hingga enterprise.

Seluruh proses pengujian keamanan pada aplikasi kaspedia.web.id dilakukan dengan mengikuti prinsip etika pengujian sistem dan memperoleh izin resmi dari pemilik aplikasi. Pengujian dilakukan dalam ruang lingkup yang telah disepakati untuk memastikan bahwa aktivitas penetration testing tidak menimbulkan gangguan operasional terhadap layanan yang berjalan. Setiap temuan kerentanan dilaporkan melalui mekanisme Responsible Disclosure, di mana informasi kerentanan disampaikan secara tertutup kepada pihak pengelola sistem untuk memastikan penerapan perbaikan sebelum dipublikasikan. Pendekatan ini sejalan dengan praktik standar industri keamanan siber yang menekankan kepatuhan terhadap aspek legal, privasi, serta perlindungan data pengguna.

Metode yang digunakan adalah metode penetration testing terstruktur berdasarkan kerangka kerja OWASP Testing Guide v4 dan OWASP Top 10:2021. Langkah-langkah penelitian dibagi menjadi tiga fase utama sebagaimana terlihat pada Gambar 1.



Gambar 1 Fase Penelitian

Fase I Perencanaan dan Persiapan

Fase ini bertujuan menyiapkan dasar penelitian agar pengujian dapat dilakukan secara sistematis dengan langkah-langkah:

1. Studi Literatur: Penelitian ini diawali dengan melakukan kajian literatur terhadap berbagai referensi keamanan siber yang membahas kerentanan Cross-Site Scripting (XSS) Stored dan Broken Access Control (BAC), serta teknik mitigasi modern yang direkomendasikan komunitas profesional. Kajian tersebut mencakup evaluasi prinsip validasi input, sanitasi data, output encoding, serta mekanisme kontrol akses berbasis peran (Role-Based Access Control atau RBAC) yang banyak digunakan dalam pengamanan aplikasi web modern. Literatur utama yang dijadikan acuan meliputi OWASP Top 10:2021, yang mengklasifikasikan XSS dan BAC sebagai kerentanan prioritas global [10], serta tinjauan komprehensif mengenai efektivitas alat dan teknik mitigasi kerentanan web sebagaimana dibahas dalam [13]. Kajian pustaka ini memberikan dasar teoritis yang kuat bagi pemilihan metode pengujian dan strategi mitigasi yang digunakan dalam penelitian ini, sekaligus memastikan bahwa pendekatan yang diterapkan sejalan dengan standar keamanan aplikasi web yang diakui secara internasional.
2. Penentuan Ruang Lingkup Pengujian:
   Pengujian difokuskan pada domain `https://kaspedia.web.id/` dengan direktori berikut: `/user/usaha/tambah`, `/user/akun`, `/user/map`, `/user/operator`, `/operator/dashboard`, dan `/user/profil`. Pemilihan area uji ini didasarkan pada hasil laporan pentest yang menunjukkan adanya potensi kerentanan tinggi pada form input dan kontrol otorisasi.
3. Persiapan Sarana dan Prasarana:
   - Perangkat Keras: Laptop dengan spesifikasi minimal prosesor Intel i7, RAM 16 GB, dan koneksi internet stabil.

- Perangkat Lunak: Burp Suite Community Edition untuk intercept dan analisis HTTP request, OWASP ZAP untuk automated scanning, Postman untuk pengujian API endpoint, Browser Developer Tools (Chrome/Firefox) untuk memantau eksekusi script dan cookie, dan Akun Pengujian: Disiapkan dua role, yaitu Operator dan Pemilik, untuk menguji kontrol akses vertikal dan horizontal.

Pemilihan kombinasi alat tersebut didasarkan pada keunggulannya dalam mendeteksi kerentanan XSS dan BAC secara efektif, serta kompatibilitasnya terhadap framework web modern [13].

Fase II Deteksi dan Analisis Kerentanan

Fase ini merupakan inti dari penelitian, di mana dilakukan identifikasi dan analisis kerentanan aktual pada aplikasi target. Pengujian dilakukan dengan pendekatan *black-box* dan *gray-box testing*, karena peneliti memiliki akses terbatas ke sistem namun menggunakan akun uji sah.

1. Deteksi Cross-Site Scripting (XSS) Stored.
   Langkah Pengujian: Menyisipkan payload berbahaya seperti: `<script>alert('XSS Vulnerability!');</script>` pada kolom input di berbagai halaman yang menerima data pengguna. Mengamati apakah payload tersimpan di basis data dan dieksekusi kembali saat halaman diakses ulang oleh pengguna lain.
2. Deteksi Broken Access Control (BAC)
   Langkah Pengujian: Melakukan login menggunakan akun Operator (role rendah). Mengakses endpoint yang seharusnya hanya diperbolehkan untuk Pemilik, misalnya `/user/profil` dan `/user/operator`. Menguji apakah sistem melakukan validasi role di sisi server.

Metode penetration testing dipilih karena, Objektif dan terukur mampu memberikan bukti konkret berupa hasil uji dan bukti eksploitasi (PoC), Sesuai tujuan penelitian mendeteksi serta memperbaiki kerentanan aktual, bukan hanya menganalisis teorinya, Fleksibel dapat dikombinasikan dengan teknik otomatis (scanner) dan manual (payload testing) untuk hasil yang lebih akurat, dan Terstandarisasi mengikuti pedoman OWASP Testing Guide yang diakui secara internasional [10].

Hasil yang diharapkan adalah tersusunnya langkah-langkah sistematis deteksi dan mitigasi kerentanan XSS Stored dan BAC pada aplikasi web, terbentuknya model keamanan berbasis manualautomated hybrid testing yang dapat diterapkan pada aplikasi sejenis dan meningkatnya tingkat keamanan aplikasi kaspedia.web.id berdasarkan pengujian ulang setelah mitigasi.

3 Hasil dan Pembahasan

Dalam penelitian ini, telah dilaksanakan pengujian keamanan pada aplikasi web kaspedia.web.id. Berdasarkan hasil uji, teridentifikasi dua jenis kerentanan utama, yaitu XSS Tersimpan dan BAC.

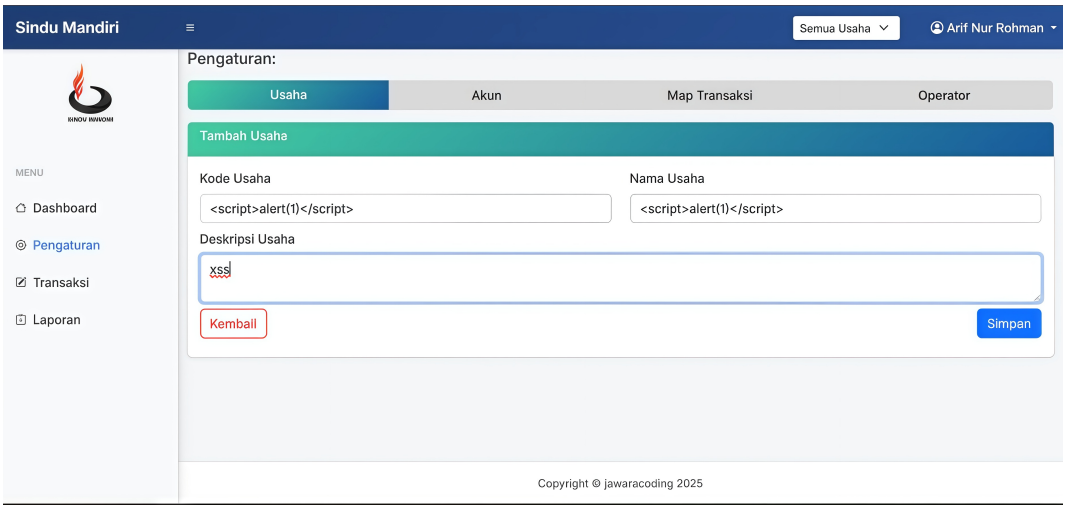
Hasil Deteksi Kerentanan Cross-Site Scripting (XSS) Stored

Pengujian XSS dilakukan dengan menyisipkan payload `<script>alert(1)</script>` pada beberapa formulir entri yang disediakan dalam aplikasi. Payload ini berhasil disimpan ke dalam database dan akan dieksekusi lagi saat halaman diakses kembali. Ini menunjukkan bahwa aplikasi tidak menerapkan mekanisme validasi input maupun pengkodean keluaran.

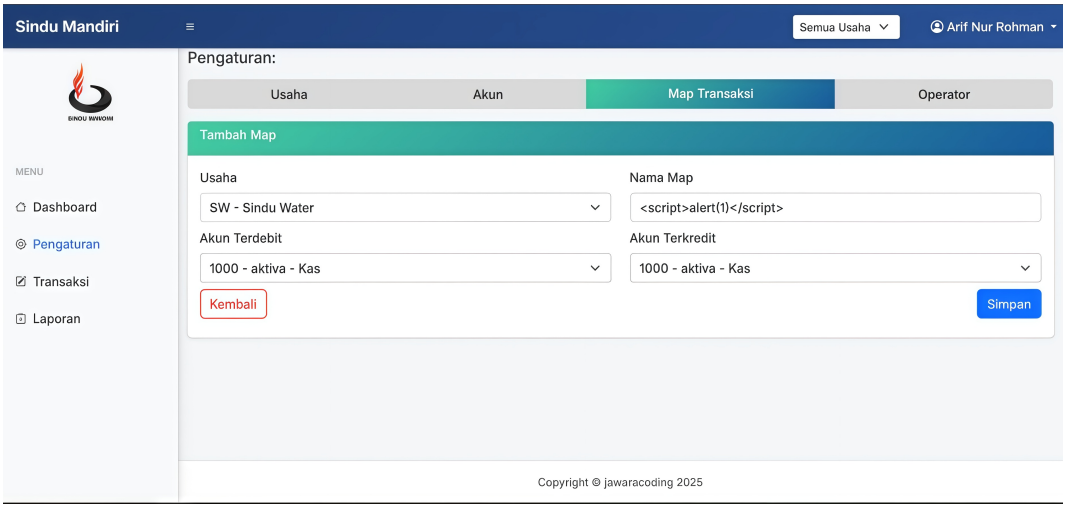
Endpoint yang terkonfirmasi rentan terhadap XSS dapat dilihat pada Tabel 1. Gambar 2, 3, 4, 5 dan 6 merupakan bukti eksekusi payload pada berbagai form.

Tabel 1 Pengujian Kerentanan Cross-Site Scripting (XSS)

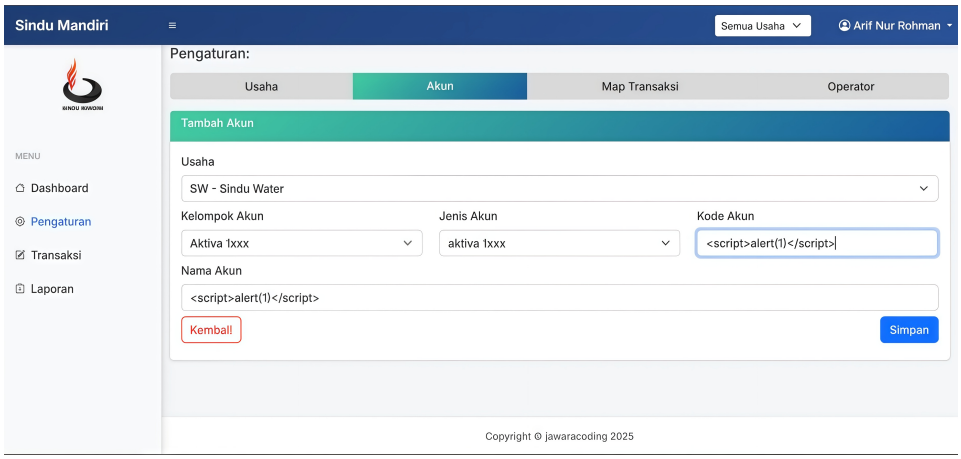
No	Endpoint	Parameter	Status	Bukti Eksekusi
1	/user/usaha/tambah	kode_usaha, nama_usaha	Rentan	Alert pop-up muncul
2	/user/map/tambah	nama_map	Rentan	Alert pop-up muncul
3	/user/akun/tambah	kode_akun, nama_akun	Rentan	Alert pop-up muncul
4	/user/operator/tambah	username, nama	Rentan	Alert pop-up muncul



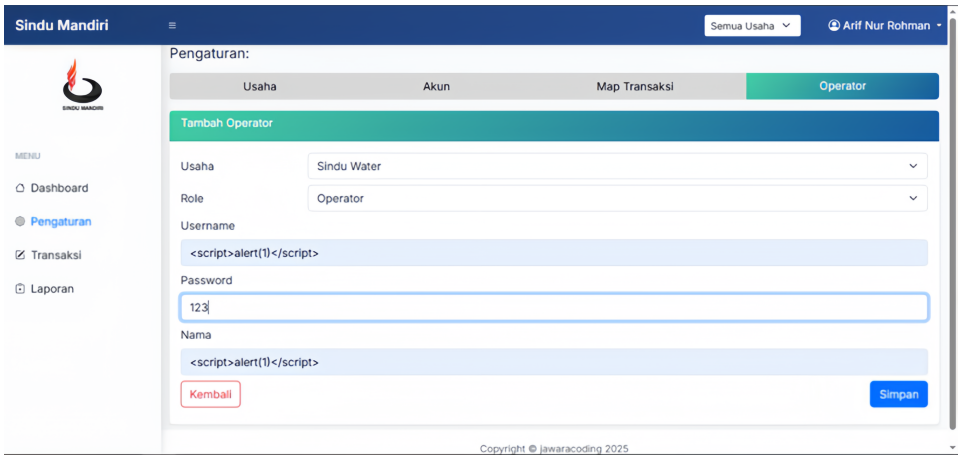
Gambar 2 Eksekusi Payload XSS Stored pada Form Tambah Usaha



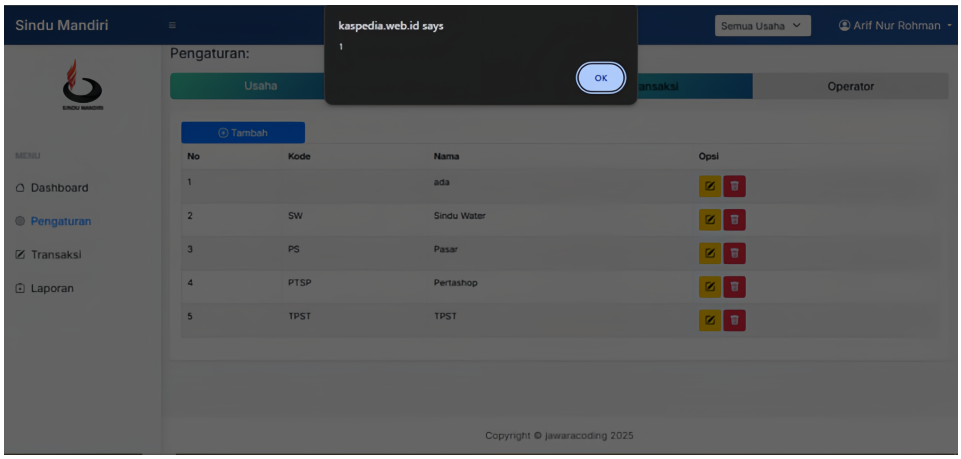
Gambar 3 Eksekusi Payload XSS Stored pada Form Tambah Map



Gambar 4 Eksekusi Payload XSS Stored pada Form Tambah Akun



Gambar 5 Eksekusi Payload XSS Stored pada Form Tambah Operator



Gambar 6 Alert Pop Up (Bukti Eksekusi)

Kemunculan alert pop-up pada setiap endpoint yang diuji menjadi bukti langsung bahwa skrip berbahaya yang disisipkan berhasil dieksekusi oleh browser. Indikator ini merupakan metode pembuktian umum dalam pengujian XSS Stored, karena menunjukkan bahwa aplikasi memproses dan menampilkan kembali input pengguna tanpa validasi atau output encoding yang memadai. Eksekusi alert pop-up tersebut memperkuat temuan bahwa payload berbahaya disimpan di dalam basis data dan dipicu kembali pada saat halaman dimuat, sehingga mengonfirmasi karakteristik XSS Stored yang bersifat persisten dan berdampak luas sebagaimana dijelaskan dalam literatur [6, 15]. Dengan demikian, keberadaan alert pop-up ini tidak hanya menunjukkan adanya kerentanan, tetapi juga menandakan bahwa potensi eksploitasi dapat terjadi pada seluruh pengguna yang mengakses halaman terkait. Berdasarkan penilaian menggunakan CVSS, tingkat keparahan kerentanan ini diklasifikasikan sebagai *High Severity* dengan skor 8.0.

Mitigasi terhadap kerentanan XSS Stored dilakukan dengan menerapkan validasi input berbasis whitelist serta output encoding pada seluruh data yang ditampilkan kembali ke pengguna untuk mencegah eksekusi skrip berbahaya. Selain itu, diperlukan penyaringan karakter berbahaya sebelum penyimpanan ke basis data serta penerapan *Content Security Policy* (CSP) sebagai lapisan proteksi tambahan. Penggunaan teknik sanitasi seperti `htmlspecialchars()` dan `strip_tags()` dapat secara signifikan menurunkan risiko eksekusi XSS pada aplikasi web. Selanjutnya seluruh perbaikan divalidasi melalui pengujian ulang guna memastikan bahwa payload tidak lagi menghasilkan eksekusi *alert pop-up* dan kerentanan telah teratasi.

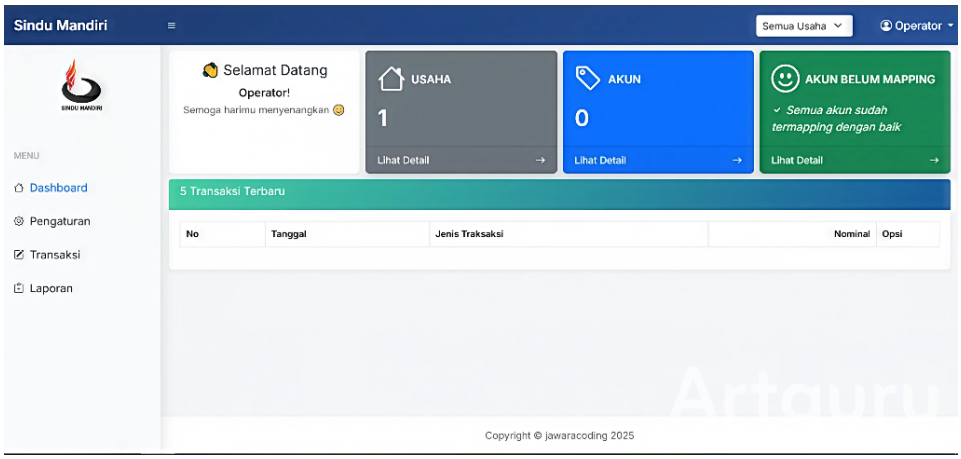
3.1 Hasil Deteksi Kerentanan Broken Access Control (BAC)

Pengujian BAC dilakukan dengan memanfaatkan peran Operator (hak rendah) dan Pemilik Usaha. Hasil temuan disajikan pada Tabel 2 dan Gambar 7 - 11.

Tabel 2 Pengujian Kerentanan Broken Access Control (BAC)

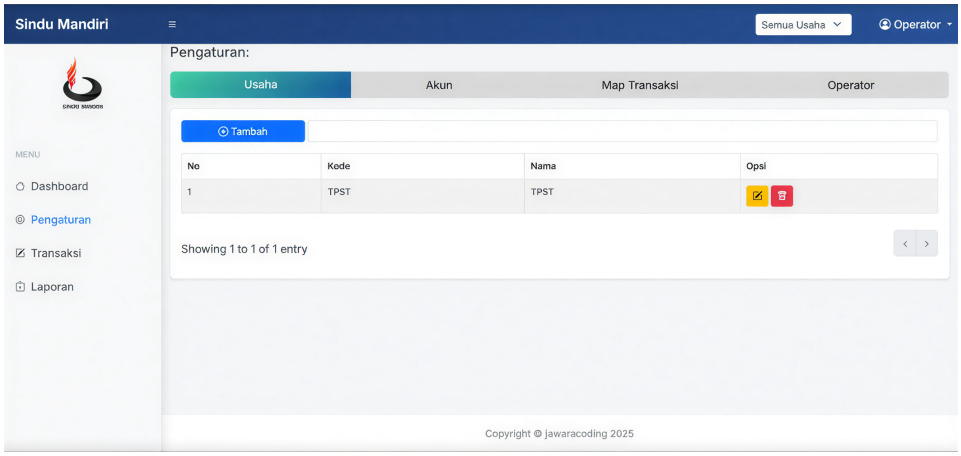
No	Role	Endpoint	Aksi Berhasil	Jenis Pelanggaran	Status
1	Operator	/user/dashboard	Akses dashboard Admin	Vertical Privilege Escalation	Rentan
2	Operator	/user/usaha	CRUD daftar usaha	Vertical & Horizontal Privilege Escalation	Rentan
3	Operator	/user/akun	Ubah data akun lain	Horizontal Privilege Escalation	Rentan
4	Operator	/user/map	Ubah/hapus data map	Vertical Privilege Escalation	Rentan
5	Operator	/user/operator	Hapus operator lain	Full Access Control Bypass	Rentan

Hasil pengujian menunjukkan bahwa peran Operator dapat mengakses dan memodifikasi data yang seharusnya hanya dapat dilakukan oleh Pemilik Usaha. Kondisi ini mengindikasikan adanya *Vertical Privilege Escalation*, yaitu ketika pengguna dengan hak akses lebih rendah memperoleh kemampuan untuk melakukan aksi yang seharusnya hanya tersedia bagi pengguna dengan hak lebih tinggi. Selain itu, ditemukan pula indikasi *Horizontal Privilege Escalation*, di mana pengguna dapat mengakses atau memodifikasi data milik pengguna lain pada tingkat peran yang setara.



Gambar 7 Operator dapat mengakses dashboard Administrator

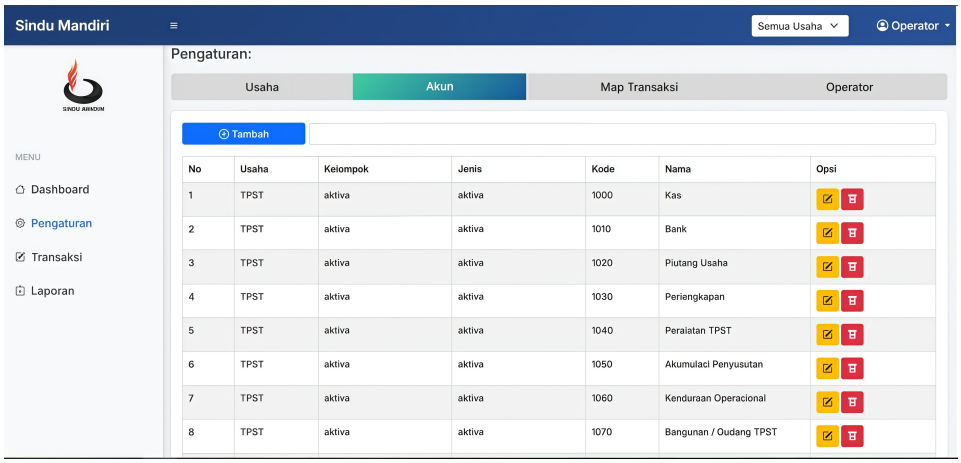
Kedua bentuk penyimpangan kontrol akses ini merupakan karakteristik utama dari kerentanan *Broken Access Control* seperti dijelaskan dalam panduan OWASP, yang menegaskan bahwa BAC termasuk salah satu kerentanan paling sering terjadi dan paling berbahaya dalam aplikasi web modern [10]. Kerentanan ini dikategorikan sebagai Critical Severity dengan skor CVSS 9.8, karena memungkinkan penyerang mendapatkan kendali penuh atas fungsi-fungsi penting dalam aplikasi, termasuk manipulasi data sensitif, eskalasi hak istimewa, dan potensi kompromi keseluruhan sistem. Literatur terbaru juga menyatakan bahwa kelemahan pada mekanisme pembatasan akses dapat menyebabkan risiko signifikan terhadap integritas dan kerahasiaan data karena memungkinkan pengguna tidak sah untuk melampaui batas otorisasinya [4, 16]. Temuan ini memperkuat urgensi perbaikan karena kegagalan kontrol akses sering kali menjadi penyebab utama kebocoran data dan penyalahgunaan sistem pada aplikasi web.



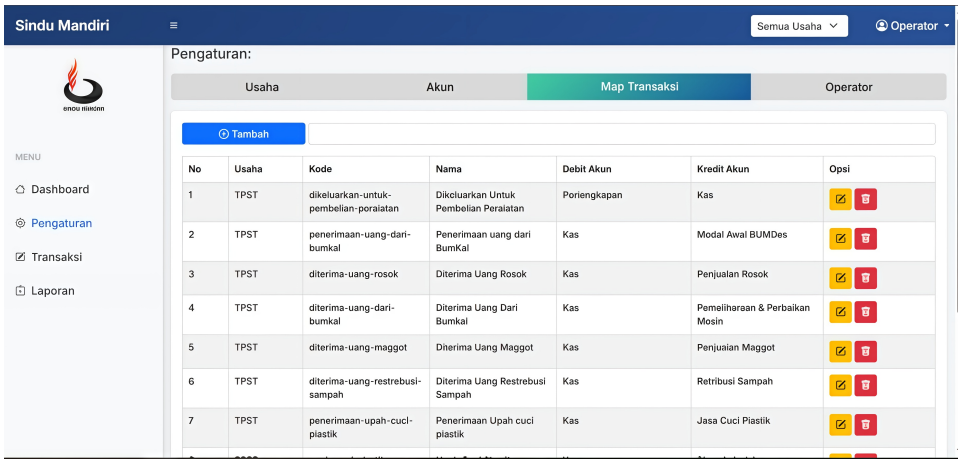
Gambar 8 Operator dapat mengakses halaman Usaha

Mitigasi terhadap kerentanan *Broken Access Control* dilakukan melalui penerapan *Role-Based Access Control* yang konsisten pada seluruh endpoint untuk memastikan setiap fungsi hanya dapat diakses oleh peran yang berwenang. Seluruh permintaan pengguna divalidasi melalui server-side authorization checks guna mencegah manipulasi akses melalui permin-

taan langsung. Selain itu, diterapkan *object-level authorization* untuk menjamin bahwa pengguna hanya dapat mengakses data yang menjadi haknya [17]. Logika kontrol akses dipusatkan pada satu modul agar kebijakan otorisasi bersifat seragam di seluruh aplikasi. Sebagai penguatan, sistem dilengkapi dengan audit logging untuk memantau aktivitas kritis dan mendeteksi indikasi penyalahgunaan hak akses [18].



Gambar 9 Operator dapat mengakses halaman Akun



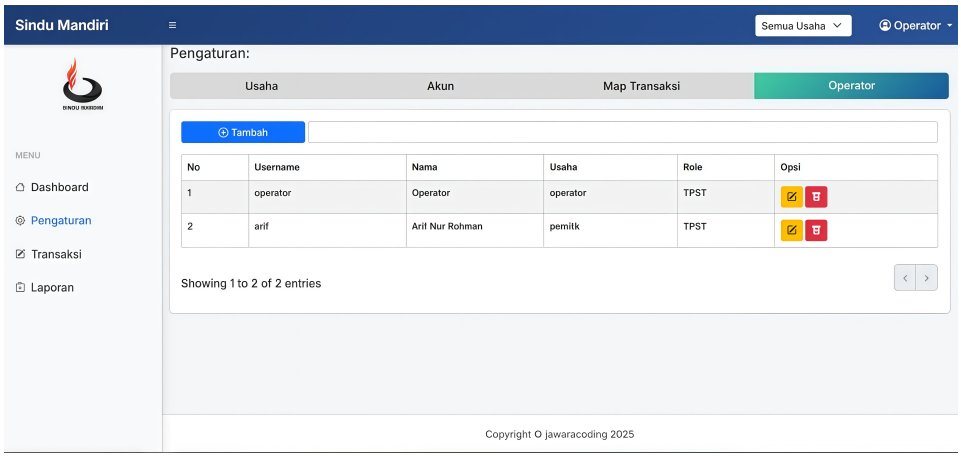
Gambar 10 Operator dapat mengakses halaman Map Transaksi

4 Kesimpulan

Penelitian ini berhasil mendeteksi kedua kerentanan utama sesuai judul, yaitu Cross-Site Scripting (XSS) Stored pada 4 endpoint (CVSS 8.0 - High) dan Broken Access Control (BAC) pada 4 endpoint (CVSS 9.8 - Critical) di aplikasi kaspedia.web.id.

BAC dengan skor CVSS 9.8 memvalidasi posisinya sebagai #1 OWASP Top 10 2021, memungkinkan penyerang mengakses data sensitif dan fungsi admin tanpa autentikasi yang memadai, membuktikan urgensi pencegahan melalui RBAC dan server-side authorization.

XSS Stored pada 4 endpoint berpotensi menyebabkan session hijacking, account takeover, dan defacement, mengkonfirmasi keberhasilan deteksi menggunakan OWASP ZAP dan



Gambar 11 Operator dapat mengakses halaman Operator

kebutuhan pencegahan melalui CSP + input sanitization.

Penelitian mengidentifikasi total 8 kerentanan kritis yang menegaskan ketidakcukupan sistem kontrol akses dan sanitasi input saat ini pada kaspedia.web.id. Kerentanan ini membuka peluang data breach massal, kerugian finansial, dan hilangnya kepercayaan pengguna, menekankan pentingnya strategi pencegahan yang komprehensif.

Temuan ini memberikan roadmap pencegahan spesifik untuk kaspedia.web.id dan menjadi referensi deteksi kerentanan bagi aplikasi web serupa di Indonesia. Penelitian selanjutnya dapat berfokus pada pengembangan dan implementasi solusi keamanan yang lebih canggih dan terintegrasi. Prioritas utama adalah membangun sistem deteksi real-time berbasis machine learning untuk XSS dan BAC, menggunakan pendekatan seperti Locate-Then-Detect untuk akurasi tinggi dan latensi rendah. Selain itu, penting untuk mengembangkan Automated Black-Box BAC Scanner yang disesuaikan dengan konteks aplikasi lokal di Indonesia, serta mengintegrasikan Integrated Security Testing Pipeline (SAST+DAST+SCA) ke dalam siklus pengembangan.

Penelitian juga dapat menguji efektivitas WAF (Web Application Firewall) versus implementasi RBAC (Role-Based Access Control) pada aplikasi web Indonesia, dan merancang Hybrid XSS Prevention Framework yang menggabungkan berbagai lapisan mitigasi seperti CSP dan deteksi anomali berbasis machine learning. Terakhir, adaptasi Kerangka Kerja DevSecOps yang komprehensif, sesuai dengan regulasi pemerintah Indonesia, akan menjadi krusial untuk menciptakan ekosistem pengembangan aplikasi yang aman.

Ucapan Terima Kasih

Penelitian ini tidak mungkin terwujud tanpa dukungan dan bimbingan yang luar biasa dari berbagai pihak. Kami mengucapkan terima kasih yang sebesar-besarnya kepada: Mukhlis Nur Arif, Marcelinus Erix Nugroho, Galih Nur Massaid dan Tim Pengembang kaspedia.web.id yang telah memberikan akses dan dukungan teknis untuk pengujian aplikasi.

Pustaka

1 B. Riskhan, M. A. U. Sheikh, M. S. Hossain, K. Hussain, Z. Zainol, dan N. Z. Jhanjh, “Major vulnerabilities of web application in real world scenarios and their prevention,”

in *2025 International Conference on Intelligent and Cloud Computing (ICoICC)*. IEEE, 2025, pp. 1–6. [Online]. Available: https://doi.org/10.1109/icoicc64033.2025.11052016

2 M. Nawrocki dan J. Kołodziej, "Vulnerabilities of web applications: Good practices and new trends," *Applied Cybersecurity & Internet Governance*, vol. 3, no. 2, pp. 122–143, 2024. [Online]. Available: https://doi.org/10.60097/acig/199521

3 V. D. Agustina, T. Ariyadi, T. S. Putra, dan A. Lega, "Teknik pengujian penetrasi http menggunakan tools burp suite pada kali linux," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 4, no. 1, pp. 16–21, 2025. [Online]. Available: https://doi.org/10.55123/storage.v4i1.4770

4 A. Anas, S. Elgamal, dan B. Youssef, "Survey on detecting and preventing web application broken access control attacks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, pp. 772–781, 2024. [Online]. Available: https://doi.org/10.11591/ijece.v14i1.pp772-781

5 P. Nagarjun dan S. A. Shaik, "Cross-site scripting research: A review," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, 2020. [Online]. Available: https://doi.org/10.14569/ijacsa.2020.0110481

6 G. Rodríguez-Galán, E. Benavides-Astudillo, D. Nuñez-Agurto, P. Puente-Ponce, S. Cárdenas-Delgado, dan M. Loachamín-Valencia, "Strategies and challenges in detecting xss vulnerabilities using an innovative cookie collector," *Future Internet*, vol. 17, no. 7, p. 284, 2025. [Online]. Available: https://doi.org/10.3390/fi17070284

7 R. Bohara, A. VV, D. J. Jaiswal, M. Nikhil, B. Pandey, U. Raghav *et al.*, "A survey paper on cross-site scripting (xss)," in *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, 2022. [Online]. Available: https://doi.org/10.2139/ssrn.4345682

8 S. Pasini, G. Maragliano, J. Kim, dan P. Tonella, "Xss adversarial attacks based on deep reinforcement learning: A replication and extension study," *arXiv preprint arXiv:2502.19095*, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2502.19095

9 CVE, "Cve-2025-57424: Xss stored in mycourts application," 2025, online. [Online]. Available: https://aardwolfsecurity.com/cve-2025-57424-stored-xss-vulnerability-in-mycourt

10 OWASP, "A01:2021 broken access control," 2021, online. [Online]. Available: https://owasp.org/Top10/A01_2021-Broken_Access_Control

11 S. E. H. Chehade, F. Hantke, dan B. Stock, "403 forbidden? ethically evaluating broken access control in the wild," in *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2025, pp. 3218–3235. [Online]. Available: https://doi.org/10.1109/sp61157.2025.00252

12 S. Qadir, E. Waheed, A. Khanum, dan S. Jehan, "Comparative evaluation of approaches & tools for effective security testing of web applications," *PeerJ Computer Science*, vol. 11, p. e2821, 2025.

13 I. Kaźmierak, "Comparison of the effectiveness of tools for testing the security of web applications," *Journal of Computer Sciences Institute*, vol. 34, pp. 36–43, 2025. [Online]. Available: https://doi.org/10.35784/jcsi.6613

14 H. Xie, "A comprehensive review on the application of cvss 4.0 and deep learning in vulnerability," *Applied and Computational Engineering*, vol. 87, no. 1, pp. 234–240, 2024.

15 J. Kaur, U. Garg, dan G. Bathla, "Detection of cross-site scripting (xss) attacks using machine learning techniques: a review," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12 725–12 769, 2023.

16 I. Dharmaadi, M. Alhanahnah, V.-T. Pham, F. Mohsen, dan F. Turkmen, "Bacfuzz: Exposing the silence on broken access control vulnerabilities in web applications," *arXiv preprint arXiv:2507.15984*, 2025. [Online]. Available: https://doi.org/10.48550/arxiv.2507.15984

17 Y. Huang, C. Shi, J. Lu, H. Li, H. Meng, dan L. Li, "Detecting broken object-level authorization vulnerabilities in database-backed applications," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 2934–2948. [Online]. Available: https://doi.org/10.1145/3658644.3690227

18 P. Kaliyaperumal, S. Periyasamy, M. Thirumalaisamy, B. Balusamy, dan F. Benedetto, "A novel hybrid unsupervised learning approach for enhanced cybersecurity in the iot," *Future Internet*, vol. 16, no. 7, p. 253, 2024.