Digital Transformation Technology (Digitech) | e-ISSN: 2807-9000
Volume 6, **Number** 2, September 2026
https://doi.org/10.47709/digitech.v6i2.9108

Komparasi Efektivitas Web Application Firewall terhadap Serangan Otomatis dan Intervensi Manual pada Simulasi Portal Akademik

Stanley Jonathan[1*]
[1]Institut Teknologi dan Bisnis Sabda Setia, Indonesia.
[1]stanley.jonathan@itbss.ac.id

open access
Histori Artikel:
Diajukan: 4 Juli 2026
Disetujui: 19 Oktober 2026
Dipublikasi: 28 Oktober 2026
Kata Kunci:
Broken Object Level Authorization; ISSAF; *Penetration Testing*; SQLmap; *Web Application Firewall*.
Digital Transformation Technology (Digitech) *is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).*

Abstrak
Lonjakan ancaman siber terhadap aplikasi berbasis *website* memaksa instansi untuk menerapkan arsitektur pelindung jaringan menggunakan *Web Application Firewall* (WAF). Namun, implementasi WAF memunculkan anomali dalam evaluasi keamanan (*penetration testing*), di mana pemindai otomatis sering kali mengalami kebuntuan akibat pemblokiran, sementara intervensi manual terbukti mampu mengeksploitasi logika aplikasi secara senyap. Penelitian ini bertujuan untuk menguji dan membandingkan secara langsung tingkat efektivitas ekstraksi data dan tingkat penghindaran (*evasion*) WAF antara serangan alat otomatis dengan intervensi manual. Mengadopsi prosedur *Information System Security Assessment Framework* (ISSAF) dan *Open Web Application Security Project* (OWASP), penelitian kuantitatif eksperimental komparatif ini diaplikasikan pada *dummy server* portal akademik SSDD-CCA (*Secure Software Design & Development*) bersistem PHP *native* dan MySQL yang dilindungi oleh WAF Cloudflare. Eksperimen dilakukan melalui 60 iterasi komparatif: 30 tembakan otomatis menggunakan SQLmap, dan 30 percobaan intervensi URL manual untuk mengeksploitasi *Broken Object Level Authorization* (BOLA). Hasil penelitian menunjukkan disparitas absolut; serangan otomatis memiliki tingkat pemblokiran 100% (skor *evasion* 0%), sedangkan serangan manual berhasil mencapai tingkat *evasion* 100% tanpa memicu sensor WAF. Temuan ini menyimpulkan bahwa WAF modern sangat tangguh menghadapi serangan berbasis pola, tetapi lemah terhadap manipulasi logika bisnis. Implikasi praktis mendesak pengembang untuk memperkuat *secure coding* pada validasi otorisasi alih-alih hanya bergantung pada pelindung eksternal.

PENDAHULUAN

Lonjakan pengguna internet di Indonesia yang bertumbuh pesat hingga mencapai 66,48% pada tahun 2022 (Sabariman et al., 2023) memaksa berbagai instansi untuk mengoptimalkan layanan digital. Peningkatan drastis ini menyebabkan meningginya risiko serangan siber terhadap aplikasi berbasis *website*, seperti eksploitasi kerentanan *SQL Injection* (SQLi) dan *Cross-Site Scripting* (XSS) yang dapat mengakibatkan manipulasi informasi hingga kebocoran data (Ramadhan et al., 2025; Wibowo & Sulaksono, 2021). Dengan demikian, pengujian keamanan sistem, seperti pengujian penetrasi, telah bergeser dari sekedar pelengkap menjadi kebutuhan dalam strategi pertahanan siber modern di setiap organisasi (Jaelani et al., 2023; Safitra et al., 2023; Yaqoob et al., 2017), sekaligus menjadi parameter untuk mengevaluasi implementasi *secure code* pada tahap pengembangan aplikasi (Ariyadi et al., 2025).

Di sektor pendidikan, institusi perguruan tinggi sangat bergantung dengan aplikasi berbasis *website* untuk mengelola data operasional dan informasi akademik mahasiswa. Sistem Informasi Terpadu pada perguruan tinggi menyimpan data krusial yang menuntut perlindungan tinggi, mulai dari nilai, presensi, hingga dokumen privasi lainnya (Sholawati et al., 2024). Untuk menjaga integritas data tersebut dari ancaman akses ilegal, pengelola sistem saat ini secara aktif menerapkan arsitektur pelindung jaringan menggunakan *Web Application Firewall* (WAF) dari layanan pihak ketiga seperti Cloudflare.

Di sisi lain, penggunaan perlindungan *firewall* modern ini menimbulkan anomali dan tantangan kritis selama pengujian penetrasi. Pada praktiknya, pengujian kerentanan konvensional yang sangat bergantung pada tembakan masif dari *tools* otomatis seperti Nmap, OWASP ZAP, Metasploit, dan SQLMap (Ramadhan et al., 2025) sering kali mengalami kebuntuan. Sebagai bukti di lapangan, eksekusi serangan eksploitasi menggunakan Metasploit Framework gagal menembus pangkalan data dan justru mengakibatkan IP Address penguji langsung diblokir secara mutlak oleh *firewall* (Sabariman et al., 2023). Serupa dengan hal tersebut, pengujian injeksi basis data otomatis menggunakan SQLMap terbukti gagal mengeksekusi parameter akibat intervensi sistem perlindungan.

Sayangnya, sebagian besar penelitian terdahulu terus melakukan pengujian yang sepenuhnya

mengandalkan alat otomatis tanpa mengeksplorasi teknik *bypassing* atau penghindaran lebih lanjut setelah *tools* tersebut diblokir. Penelitian Sabariman et al. (2023) dan Sholawati et al. (2024) memang berhasil memetakan kerentanan menggunakan standar OWASP dan ISSAF, namun eksplorasi teknis mereka terhenti dan dinyatakan "gagal" ketika *tools* otomatis diblokir oleh WAF. Belum ada analisis kritis dan mendalam yang membandingkan bagaimana intervensi manual yang tidak bising, seperti memanipulasi parameter identitas URL melalui *browser* yang justru berhasil mencuri ribuan data pribadi *user (Broken Object Level Authorization)* dapat dengan mudah melewati logika *firewall* yang selalu ditakuti oleh *tools scanner* konvensional.

Oleh karena itu, penelitian ini difokuskan untuk menganalisis *gap* yang ada di penelitian terdahulu yaitu ketergantungan eksklusif pada *tools* pengujian keamanan otomatis yang terbukti rentan digagalkan oleh sistem pelindung modern. Perangkat otomatis mampu memetakan titik jaringan tetapi proses eksekusi serangan (*exploiting*) secara masif seringkali langsung memicu alarm pertahanan yang berujung pada pemblokiran alamat IP penyerang secara mutlak oleh WAF (Sabariman et al., 2023; Sholawati et al., 2024). Di sisi lain, studi terdahulu telah membuktikan bahwa pendekatan intervensi manual yang lebih senyap mampu menembus logika keamanan sistem tanpa memicu pemblokiran jaringan. Kesenjangan (*research gap*) komparasi inilah yang menjadi inti pokok permasalahan pada penelitian ini.

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah: Bagaimana perbandingan tingkat keberhasilan (*effectiveness*) antara penetrasi keamanan menggunakan *tools* otomatis (SQLMap) dengan teknik manipulasi parameter manual pada lingkungan simulasi portal akademik perguruan tinggi yang dilindungi *Web Application Firewall* (WAF)? Sejauh mana teknik eksploitasi manual melalui *browser* dapat menghindari deteksi pemblokiran (*evasion*) oleh *firewall* dibandingkan dengan serangan otomatis yang bersifat masif pada lingkungan simulasi tersebut?

STUDI LITERATUR

Kerangka Pengujian *Cyber Security*

Berdasarkan survei Altulaihan et al. (2023), pelaksanaan pengujian aplikasi *web* saat ini sangat banyak mengandalkan alat pemindai otomatis karena efisiensi waktunya, meskipun eksekusi manual diakui mempunyai tingkat keefektifan yang lebih tinggi. Ashar (2022) menegaskan bahwa evaluasi keamanan *web* yang terbuka (*open website*) akan jauh lebih komprehensif apabila menggabungkan metode *Open Web Application Security Project* (OWASP) dengan *Information System Security Assessment Framework* (ISSAF) sebagai panduan prosedur. Hal ini sejalan dengan temuan Sanjaya et al. (2020) yang membuktikan bahwa penerapan kerangka kerja ISSAF sangat komprehensif dalam mengidentifikasi celah keamanan sekaligus mengevaluasi manajemen risiko teknologi informasi pada sebuah institusi. Pendekatan *hybrid* ini terbukti kuat dalam praktiknya sebagaimana dibuktikan oleh Andriyani et al. (2023), Tjiptabudi & Ndaumanu (2024), dan Sholawati et al. (2024). Dengan demikian, makalah ini menggabungkan ISSAF dengan standar OWASP sebagai Konsep Induk (*Grand Theory*). Variabel penetrasi keamanan diukur secara operasional dengan tahapan penetrasi ISSAF, di mana indikator pengujian dianggap "selesai/berhasil" jika eksploitasi berhasil mengambil data dari basis data simulasi.

Web Application Firewall (WAF)

Web Application Firewall adalah lapisan pertahanan aktif yang melindungi pengguna aplikasi *web* dari serangan yang berbahaya dengan cara menyaring lalu lintas HTTP/HTTPS. Pratama & Anwar (2023) dalam eksperimennya membuktikan bahwa WAF secara efektif memblokir intrusi otomatis dan menjawab dengan pesan penolakan mutlak seperti 404/403 *Forbidden*. Hal yang sama juga dialami oleh Sabariman et al. (2023) di mana eksekusi menggunakan perangkat Metasploit Framework langsung memicu pemblokiran alamat IP. Dalam tinjauan sistematis terbaru oleh Ramadhan et al. (2025), dia menyarankan pergeseran ke arah strategi adaptif yang menggabungkan teknik otomatis dengan intervensi manual untuk memastikan supaya penguji dapat menghindari deteksi oleh mesin sensor (*evasion*). Variabel *Evasion* akan diukur melalui dua indikator jaringan: (1) status penolakan koneksi (*ERR_CONNECTION_REFUSED* atau 403 *Forbidden*) sebagai respons terhadap alat otomatis, dan (2) respons HTTP 200 OK yang lolos secara normal saat parameter dimanipulasi secara manual.

Komparasi Efektivitas *Tools* Otomatis vs Intervensi Manual

Kemampuan alat otomatis memang terbukti pada sistem yang tidak terlindungi; Alanda et al. (2021) membuktikan bahwa pengujian otomatis dengan SQLMap mampu menembus basis data pada 80% sampel *website* target mereka. Tetapi, efektivitas ini menurun saat berhadapan dengan sistem pertahanan modern. Sholawati et al. (2024) mendapati bahwa pemindai otomatis (SQLMap) gagal total mengeksploitasi portal akademik ber-WAF karena bot tersebut tidak mampu menginjeksi parameter dinamis. Sebaliknya, riset Sabariman et al. (2023) memperlihatkan terobosan dengan pendekatan intervensi manual, pengubahan nilai parameter identitas secara langsung sanggup menembus logika WAF dan mengekspos kebocoran data privasi ribuan pengguna (*Broken Object Level Authorization*) tanpa memicu alarm pemblokiran. Oleh karena itu, variabel Tingkat Keberhasilan (*Effectiveness*) diukur melalui indikator biner: (1) keberhasilan menembus celah kerentanan otorisasi pada *dummy*

server, dan (2) keberhasilan mengekstraksi data spesifik sebagai *Proof of Concept* (PoC).

Kerangka Konseptual dan Hipotesis

Merujuk pada sintesis bukti terdahulu, hipotesis yang diajukan dalam eksperimen ini adalah: H1: Penetrasi keamanan menggunakan *tools* otomatis (SQLMap) memiliki tingkat keberhasilan (*effectiveness*) yang lebih rendah akibat tingginya tingkat deteksi dan pemblokiran (*error connection* / 403 *Forbidden*) oleh WAF pada lingkungan simulasi. H2: Teknik intervensi parameter secara manual memiliki tingkat penghindaran (*evasion*) yang secara signifikan lebih tinggi untuk melewati sensor WAF dibandingkan instrumen penyerang otomatis yang bersifat masif.



Gambar 1 Kerangka Konseptual

METODE

Penelitian ini menggunakan pendekatan kuantitatif-eksperimental dengan desain komparatif yang bertujuan untuk membandingkan tingkat penghindaran (*effectiveness*) dan tingkat penghindaran (*evasion*) dari dua metode penetrasi yang berbeda secara empiris. Eksperimen dilakukan pada sebuah lingkungan simulasi (*dummy server*) berupa aplikasi portal akademik berbasis PHP *native* yang dipasangkan pelindung WAF pihak ketiga, yaitu Cloudflare. Penggunaan *dummy server* ini dirancang khusus untuk mereplikasi arsitektur keamanan portal akademik di dunia nyata, sekaligus mematuhi prinsip *ethical hacking*.

Objek target dalam pengujian ini difokuskan pada antarmuka *login* dan halaman profil pengguna di dalam *dummy server* bernama SSDD-CCA (*Secure Software Design & Development*). Aplikasi sumber terbuka ini secara sengaja dirancang memiliki celah keamanan (*Vulnerable by Design*) dan memiliki fitur *Role-Based Access Control* (RBAC). Aplikasi target dijalankan pada peladen lokal yang kemudian diekspos secara aman ke internet publik menggunakan teknologi *Cloudflare Tunnel*.

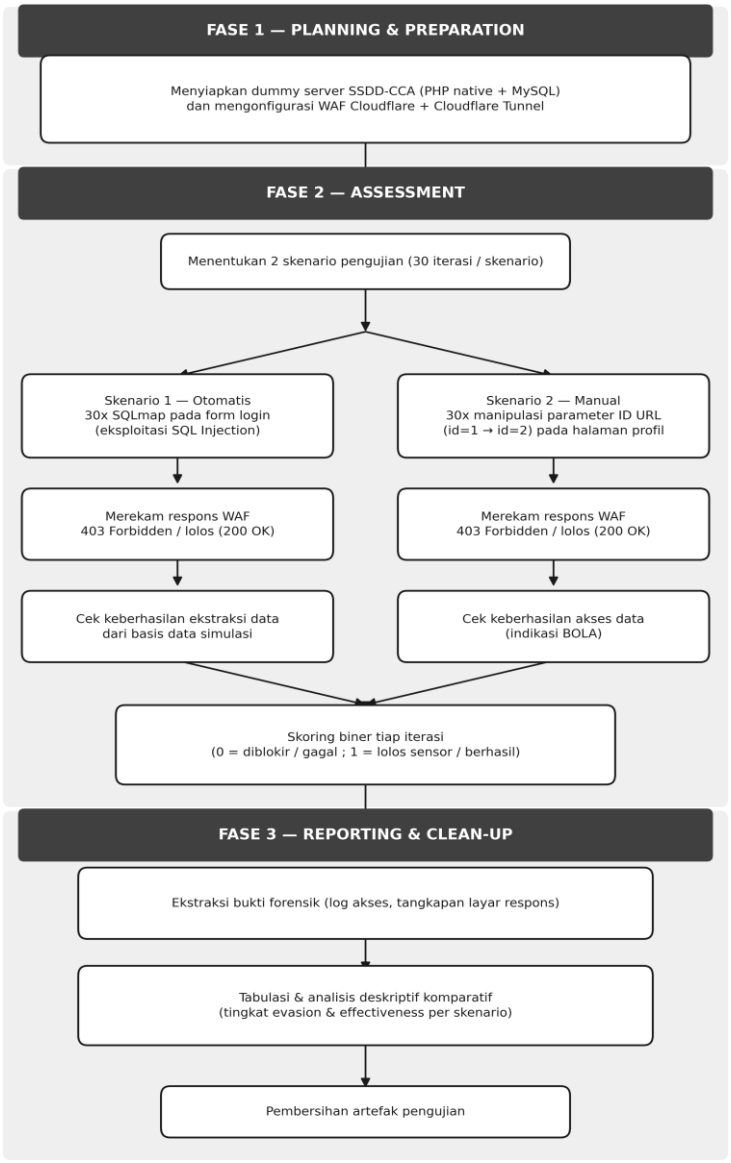
Masing-masing skenario akan diuji melalui 30 iterasi (percobaan) serangan. Skenario pengujian dibagi menjadi:

1) Skenario Serangan Otomatis: Mengirimkan 30 kali tembakan permintaan berbahaya menggunakan instrumen *scanner* otomatis (SQLmap) ke form *login* guna mengeksploitasi celah *SQL Injection* secara masif.
2) Skenario Serangan Manual: Melakukan 30 kali percobaan intervensi dengan memanipulasi parameter ID identitas (misalnya dari id=1 menjadi id=2) melalui URL *browser* untuk mengeksploitasi celah *Broken Object Level Authorization* (BOLA) pada halaman profil pengguna.

Untuk menjaga validitas data, pengujian dilakukan pada topologi jaringan, spesifikasi *server*, dan aturan WAF Cloudflare yang dikonfigurasi secara identik. Instrumen pengukuran menggunakan skala nominal biner: Skor 0 jika lalu lintas diblokir (log 403 *Forbidden*) atau tidak ada data yang diekstraksi, dan Skor 1 jika lalu lintas lolos sensor (HTTP 200 OK) atau berhasil mendapatkan data PoC.

Proses pelaksanaan mengadopsi prosedur ISSAF melalui tiga tahap utama. *Fase Planning & Preparation* meliputi persiapan integrasi *Cloudflare Tunnel*. *Fase Assessment* merupakan pelaksanaan komparasi 60 iterasi serangan antara SQLmap dan manipulasi BOLA di peramban web. *Fase Reporting & Clean-up* meliputi ekstraksi bukti forensik dan pembersihan artefak. Data dianalisis secara deskriptif komparatif dan ditabulasikan guna membuktikan hipotesis efektivitas instrumen.

Untuk memperjelas alur pelaksanaan, tahapan penelitian mulai dari persiapan hingga pelaporan divisualisasikan pada diagram alur (Gambar 2) berikut.



Gambar 2 Diagram Alur Tahapan Penelitian

Sebagaimana ditunjukkan pada Gambar 2, penelitian ini dilaksanakan melalui tiga fase ISSAF secara berurutan. Pada *Fase Planning & Preparation*, peneliti menyiapkan *dummy server* SSDD-CCA berbasis PHP *native* dan MySQL, kemudian mengonfigurasi WAF Cloudflare beserta *Cloudflare Tunnel* agar aplikasi target dapat diakses secara publik namun tetap terpantau. Pada *Fase Assessment*, pengujian bercabang menjadi dua alur paralel yang dijalankan secara independen dengan konfigurasi jaringan dan aturan WAF yang identik: (1) Skenario 1 menjalankan 30 kali tembakan otomatis menggunakan SQLmap ke form *login* untuk mengeksploitasi celah SQL *Injection*, dan (2) Skenario 2 menjalankan 30 kali intervensi manual berupa manipulasi parameter ID pada URL peramban untuk mengeksploitasi celah *Broken Object Level Authorization* (BOLA) pada halaman profil pengguna. Setiap iterasi dari kedua skenario dicatat responsnya (403 *Forbidden* atau 200 OK) dan diberi skor biner (0 jika diblokir/gagal, 1 jika lolos sensor/berhasil mengekstraksi data), sehingga kedua alur bertemu pada satu titik rekapitulasi skor sebelum masuk ke *Fase Reporting & Clean-up*, yaitu ekstraksi bukti forensik, tabulasi hasil ke dalam tabel komparatif, dan pembersihan seluruh artefak pengujian dari *dummy server*.

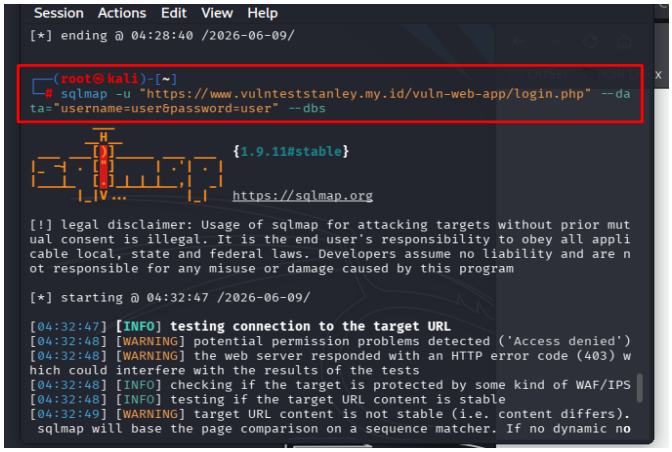
HASIL

Bagian ini menguraikan hasil analisis data komparatif yang dikumpulkan dari 60 iterasi percobaan pada masing-masing skenario serangan pada lingkungan simulasi SSDD-CCA yang dilindungi dengan WAF Cloudflare. Hasil rekapitulasi disajikan pada Tabel 1.

Tabel 1. Komparasi Tingkat Keberhasilan Penetrasi dan Evasion WAF

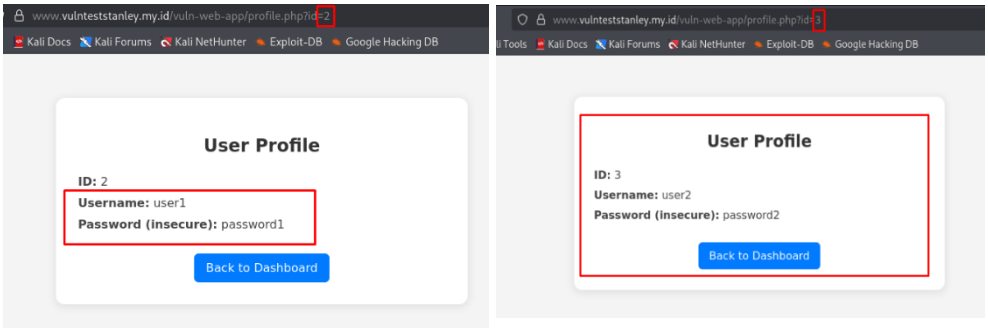
Skenario Pengujian	Instrumen	Total Iterasi	Respons WAF	Tingkat Evasion	Tingkat Effectiveness
Skenario 1 (Otomatis)	SQLmap	30	403 Forbidden	0% (0)	0% (0)
Skenario 2 (Manual)	Intervensi URL (BOLA)	30	HTTP 200 OK	100% (1)	100% (1)

Hasil uji empiris mengonfirmasi adanya perbedaan yang sangat kontras antara penggunaan alat otomatis dan metode manual. Pada Skenario 1, instrumen *SQLmap* mendapatkan tingkat pemblokiran 100%, di mana seluruh 30 iterasi penembakan langsung diputuskan jaringannya oleh WAF dengan respons *Error 403 Forbidden*. Sebaliknya, pada Skenario 2, manipulasi parameter identitas URL secara manual sukses mencapai tingkat *evasion* dan *effectiveness* 100% tanpa adanya respons pemblokiran dari sistem WAF sama sekali.
Secara teknis, Skenario 1 dijalankan dengan mengarahkan SQLmap ke endpoint form *login*, menjadikan parameter *username* dan *password* sebagai titik injeksi, dengan mode deteksi otomatis untuk mengidentifikasi payload SQL Injection yang berpotensi valid pada setiap iterasi. Dari 30 kali percobaan, seluruh permintaan (100%) konsisten dihentikan oleh WAF Cloudflare sejak permintaan pertama dikirimkan, ditandai dengan respons HTTP 403 Forbidden pada setiap log akses tanpa satu pun pengecualian. Pola ini menunjukkan bahwa WAF mengenali karakteristik payload SQLi standar—baik dari struktur *query* berbahaya maupun pola permintaan berulang dalam interval singkat—sebagai anomali lalu lintas berbasis signature, sehingga tidak satu pun payload yang sempat mencapai lapisan aplikasi maupun basis data MySQL di baliknya.



Gambar 3 Log Respons Penolakan WAF Cloudflare (HTTP 403 Forbidden) Terhadap Serangan Otomatis SQLmap.

Sementara itu, Skenario 2 dilakukan dengan *login* menggunakan satu akun pengguna sah, kemudian pada halaman profil nilai parameter identitas pada URL diubah secara manual melalui *address bar* peramban (misalnya dari id=1 menjadi id=2, id=3, dan seterusnya hingga 30 percobaan berbeda) tanpa mengubah header permintaan atau menyisipkan payload berbahaya apa pun. Karena struktur permintaan tetap berupa HTTP GET biasa yang secara sintaksis identik dengan pola akses pengguna normal, seluruh 30 percobaan (100%) direspons dengan HTTP 200 OK dan berhasil menampilkan data profil milik pengguna lain yang seharusnya tidak menjadi hak akses penguji, tanpa satu pun permintaan yang ditandai atau diblokir oleh WAF.



Gambar 4 Bukti Keberhasilan Penembusan BOLA Manual (HTTP 200 OK) dan Ekstraksi Data Profil Pengguna.

Temuan ini memperlihatkan pola yang konsisten di seluruh iterasi pada kedua skenario tanpa satu pun penyimpangan hasil, yang mengindikasikan bahwa perbedaan tingkat keberhasilan bukan disebabkan oleh faktor kebetulan, melainkan oleh karakteristik struktural dari masing-masing jenis serangan. WAF berbasis signature dan pendeteksian pola lalu lintas terbukti sangat efektif mengenali payload yang eksplisit berbahaya dan repetitif seperti pada SQLmap, namun tidak memiliki mekanisme untuk memvalidasi apakah suatu ID yang diminta memang menjadi hak akses sah dari pengguna yang sedang login—sebuah keputusan otorisasi yang semestinya menjadi tanggung jawab logika aplikasi, bukan lapisan jaringan tempat WAF beroperasi. Dengan demikian, disparitas 0% berbanding 100% pada Tabel 1 bukan sekadar selisih statistik, melainkan cerminan dari batas kemampuan WAF yang bekerja pada lapisan lalu lintas jaringan dan tidak menjangkau lapisan logika bisnis aplikasi.

PEMBAHASAN

Hasil komparasi secara empiris mengonfirmasi dan menerima hipotesis penelitian (H1 dan H2). Penetrasi menggunakan instrumen otomatis (SQLmap) terbukti memiliki tingkat efektivitas yang sangat rendah akibat tingginya angka pemblokiran WAF. Sebaliknya, teknik intervensi manual terbukti memiliki tingkat penghindaran (*evasion*) yang secara signifikan lebih superior untuk melewati sensor WAF.

Rasionalisasi di balik temuan ini adalah adanya perbedaan mendasar pada karakteristik pembentukan lalu lintas jaringan. Tingginya angka penolakan 403 pada skenario pertama sangat mungkin didasari oleh profil *scanner SQLmap* yang mengirimkan tembakan *payload* secara bising dan berulang-ulang, yang polanya sangat mudah dideteksi oleh sensor WAF modern. Jika ditelaah lebih jauh, kesuksesan mutlak serangan manual menembus pelindung terjadi karena kelemahan *Broken Object Level Authorization* (BOLA) berada pada lapisan logika bisnis aplikasi. WAF Cloudflare membaca pengubahan parameter id=1 menjadi id=2 di peramban web sebagai sebuah lalu lintas HTTP normal yang sah, sehingga WAF terkelabui dan membiarkan eksploitasi itu terjadi.

Dukungan empiris ini tidak hanya mengonfirmasi postulat kerangka kerja keamanan terkait kerentanan kontrol akses, tetapi juga memperkuat temuan Sholawati et al. (2024) yang menyatakan bahwa instrumen pemindai otomatis (*SQLmap*) seringkali gagal total dalam mengeksploitasi parameter dinamis di balik pelindung WAF. Konsisten dengan literatur terdahulu, fenomena anomali ini membuktikan bahwa pertahanan *firewall* tingkat lanjut sekalipun sangat rapuh terhadap manipulasi logika senyap seperti yang pernah dibuktikan oleh Sabariman et al. (2023).

KESIMPULAN

Penelitian ini menjawab kedua rumusan masalah dengan menyimpulkan adanya disparitas mutlak pada tingkat keberhasilan (*effectiveness*) dan penghindaran (*evasion*) antara metode penetrasi otomatis dan manual. Untuk rumusan masalah pertama, serangan otomatis menggunakan instrumen SQLmap terhadap *form login* pada lingkungan simulasi portal akademik SSDD-CCA menunjukkan tingkat keberhasilan 0% dari 30 iterasi percobaan. Seluruh permintaan secara konsisten diblokir oleh WAF Cloudflare dengan respons 403 Forbidden sebelum berhasil mengeksploitasi basis data. Sebaliknya, teknik manipulasi parameter secara manual yang menyasar celah keamanan *Broken Object Level Authorization* (BOLA) pada halaman profil pengguna sukses mencapai tingkat keberhasilan eksploitasi 100%, di mana seluruh 30 iterasi terbukti mampu mengekstraksi data pengguna lain. Menjawab rumusan masalah kedua, hasil eksperimen membuktikan bahwa manipulasi parameter ID secara manual melalui peramban menghasilkan tingkat *evasion* yang jauh lebih tinggi, mencapai 100% karena lalu lintasnya tampak seperti interaksi HTTP normal tanpa memicu pemblokiran. Hal ini sangat kontras dengan pola permintaan SQLmap yang bising, berulang, dan mudah dikenali oleh sistem WAF berbasis *signature*. Melalui temuan empiris yang menerima hipotesis H1 dan H2 ini, muncul implikasi praktis bahwa pengembang institusi pendidikan tidak boleh hanya bergantung pada perlindungan WAF eksternal. Pengelola sistem wajib memperkuat *secure coding* dan mekanisme kontrol akses pada tingkat aplikasi, mengingat WAF tidak dirancang untuk mendeteksi manipulasi logika bisnis. Menyadari keterbatasan pengujian pada *dummy server*, penelitian selanjutnya sangat disarankan untuk memperluas cakupan evaluasi menggunakan perangkat otomatis tingkat lanjut dengan kapabilitas *payload obfuscation* di lingkungan operasional yang nyata.

REFERENSI

Alanda, A., Satria, D., Ardhana, M. I., Dahlan, A. A., & Mooduto, H. A. (2021). Web Application Penetration Testing Using SQL Injection Attack. *JOIV : International Journal on Informatics Visualization*, *5*(3), 320. https://doi.org/10.30630/joiv.5.3.470

Altulaihan, E. A., Alismail, A., & Frikha, M. (2023). A Survey on Web Application Penetration Testing. *Electronics*, *12*(5), 1229. https://doi.org/10.3390/electronics12051229

Andriyani, S., Fajar Sidiq, M., & Parga Zen, B. (2023). Analisis Celah Keamanan Pada Website Dengan Menggunakan Metode Penetration Testing Dan Framework Issaf Pada Website SMK Al-Kautsar. *LEDGER: Journal Informatic and Information Technology*, *2*(1). https://doi.org/https://doi.org/10.20895/ledger.v2i1.996

Ariyadi, T., Salsabila, A. P., & Nugroho, Y. P. (2025). Implementasi Secure Code Pada Pengembangan Sistem Keamanan Website Teknik Komputer Universitas Bina Darma Menggunakan Penetration Testing dan OWASP ZAP. *JUSTER : Jurnal Sains Dan Terapan*, *4*(1), 27–30. https://doi.org/10.57218/juster.v4i1.1321

Ashar, R. (2022). Analysis of Open Website Security Using OWASP and ISSAF Methods. *Jurnal Informasi Dan Teknologi*, *4*(4), 187–194. https://doi.org/10.37034/jidt.v4i4.233

Jaelani, W. L., Yanto, Y., & Khoirunnisa, F. (2023). Penetration Testing Website Dengan Metode Black Box Testing Untuk Meningkatkan Keamanan Website Pada Instansi (Redacted). *Naratif : Jurnal Nasional Riset, Aplikasi Dan Teknik Informatika*, *5*(1), 1–8. https://doi.org/10.53580/naratif.v5i1.180

Pratama, K. D., & Anwar, N. (2023). Impact Analysis of Web Application Firewall on Website-Based Application Security (Case Study PPDB Kak Seto School Website). *Mobile and Forensics*, *5*(1), 44–58. https://doi.org/10.12928/mf.v5i1.8914

Ramadhan, L., Piranti, N. M., Pratama Martadireja, O., & Imigrasi, P. (2025). Systematic Literature Review Penetration Testing Pada Keamanan Website. *Pendas : Jurnal Ilmiah Pendidikan Dasar*, *10*(4), 460–471. https://doi.org/https://doi.org/10.23969/jp.v10i04.33554

Sabariman, S., Haeruddin, H., & Lee, D. (2023). Analisis Kerentanan Aplikasi Akademik Berbasis Website Xyz Menggunakan Owasp. *Jurnal Khatulistiwa Informatika*, *11*(2), 92–102. https://doi.org/10.31294/khatulistiwa.v11i2.10206

Safitra, M. F., Lubis, M., & Widjajarto, A. (2023). Security Vulnerability Analysis using Penetration Testing Execution Standard (PTES): Case Study of Government's Website. *Proceedings of the 2023 6th International Conference on Electronics, Communications and Control Engineering*, 139–145. https://doi.org/10.1145/3592307.3592329

Sanjaya, I. G. A. S., Sasmita, G. M. A., & Sri Arsa, D. M. (2020). Information Technology Risk Management Using ISO 31000 Based on ISSAF Framework Penetration Testing (Case Study: Election Commission of X City). *International Journal of Computer Network and Information Security*, *12*(4), 30–40. https://doi.org/10.5815/ijcnis.2020.04.03

Sholawati, A., Setyadi, H. J., & Masa, A. P. A. (2024). Implementasi Penetration Testing Pada Sistem Informasi Terpadu Layanan Prodi Menggunakan Framework Issaf. *Techno (Jurnal Fakultas Teknik, Universitas Muhammadiyah Purwokerto)*, *25*(2), 73. https://doi.org/10.30595/techno.v25i2.21140

Tjiptabudi, F. M. H., & Ndaumanu, R. I. (2024). Evaluasi Celah Keamanan Website Dana Pensiun X Melalui Penetration Testing Berdasarkan ISSAF Framework. *Jurnal Algoritma*, *21*(2), 9–17. https://doi.org/10.33364/algoritma/v.21-2.1644

Wibowo, R. M., & Sulaksono, A. (2021). Web Vulnerability Through Cross Site Scripting (XSS) Detection with OWASP Security Shepherd. *Indonesian Journal of Information Systems*, *3*(2), 149–159. https://doi.org/10.24002/ijis.v3i2.4192

Yaqoob, I., Hussain, S. A., Mamoon, S., Naseer, N., Akram, J., & Ur Rehman, A. (2017). Penetration Testing and Vulnerability Assessment. *Journal of Network Communications and Emerging Technologies (JNCET) Www.Jncet.Org*, *7*(8). www.jncet.org