Journal of Computers Sciences and Informatics
https://ejurnal.umbima.ac.id/index.php/scientific

Uji Penetrasi Injeksi SQL terhadap Celah Keamanan Website XYZ menggunakan Tools SQLMap

Muhammad Amirul Mu'min*, Zumhur Alamin, Fathir, Sahrul Ramadhan

Universitas Muhammadiyah Bima, Indonesia
* Email Korespondensi: amirulmukmin@umbima.ac.id

Abstrak: Serangan SQL Injeksi (SQLi) tetap menjadi salah satu ancaman utama dalam keamanan aplikasi web, yang memungkinkan penyerang mengakses, memodifikasi, atau menghancurkan data pada database. Latar belakang penelitian ini adalah meningkatnya kebutuhan untuk mengidentifikasi dan mengatasi potensi celah keamanan akibat serangan SQLi menggunakan metode pengujian yang efektif. Penelitian ini bertujuan untuk menguji kerentanan aplikasi web terhadap SQLi dengan menggunakan tools SQLMap melalui penyuntikan perintah SQL secara otomatis. Hasil pengujian menunjukkan bahwa *username* dan *password* berhasil ditemukan, meskipun situs web tersebut tidak memiliki kerentanan terhadap SQL Injection. Hal ini menunjukkan bahwa metode pengujian yang dilakukan efektif dalam mengidentifikasi celah keamanan. Selain itu, penelitian ini memberikan wawasan penting mengenai perlunya penerapan mekanisme keamanan yang lebih ketat untuk melindungi data pengguna, sekaligus menekankan pentingnya pengujian keamanan rutin untuk menjaga sistem tetap terlindungi dari ancaman serangan berbasis SQL.

Kata Kunci: Keamanan, Web, SQLi, SQLMap.

.

1. Pendahuluan

Seiring dengan meningkatnya penggunaan aplikasi web dalam berbagai sektor, seperti bisnis, pendidikan, pemerintahan, dan kesehatan, ancaman terhadap keamanan data yang disimpan dalam database juga menjadi semakin kompleks dan beragam [1]. Aplikasi web kini sering digunakan untuk menangani data sensitif, seperti informasi pribadi, transaksi keuangan, atau dokumen rahasia, sehingga menjadi target potensial dari berbagai jenis serangan siber [2]. Salah satu ancaman yang paling umum dan berbahaya adalah serangan SQLi [3]. Serangan ini terjadi ketika penyerang memanfaatkan celah keamanan pada input aplikasi web untuk menyisipkan perintah SQL berbahaya ke dalam sistem [4]. Jika berhasil, serangan ini dapat mengakibatkan akses tidak sah ke database, pengubahan data, atau bahkan penghancuran seluruh informasi yang tersimpan [5].

Serangan SQLi tetap menjadi salah satu ancaman keamanan aplikasi web yang paling sering terjadi di seluruh dunia. OWASP Top 10, yang secara rutin menyoroti ancaman keamanan aplikasi yang paling signifikan, terus menempatkan "Injection" sebagai salah satu ancaman utama yang perlu diwaspadai [6]. Kerentanan ini sering kali disebabkan oleh kurangnya validasi input pada sisi aplikasi, penggunaan query SQL yang tidak aman, atau implementasi kode yang tidak mengikuti standar keamanan yang baik [7]. Dampaknya dapat sangat merugikan, mulai dari pencurian data hingga kerugian finansial yang besar bagi organisasi [8].

Untuk mengatasi ancaman ini, uji penetrasi terhadap celah keamanan menjadi langkah penting dalam proses pengamanan aplikasi web [9]. Uji penetrasi tidak hanya membantu mengidentifikasi celah keamanan, tetapi juga memberikan wawasan mengenai langkah-langkah mitigasi yang dapat diambil untuk mencegah serangan di masa depan. Salah satu tools yang banyak digunakan oleh peneliti keamanan dan pengembang adalah SQLmap. SQLmap merupakan tools uji penetrasi otomatis yang dirancang khusus untuk mendeteksi dan mengeksploitasi kerentanan SQL Injection [10]. Dengan kemampuan canggihnya, SQLmap tidak hanya dapat mengidentifikasi celah keamanan, tetapi juga memberikan rekomendasi untuk memperbaiki kelemahan tersebut. Tools ini mendukung berbagai teknik injeksi SQL, termasuk union-based, error-based, dan blind SQL injection, sehingga sangat efektif dalam pengujian keamanan aplikasi web [11].

Penelitian ini bertujuan untuk melakukan uji penetrasi terhadap celah keamanan database pada aplikasi web menggunakan SQLmap. Dengan memanfaatkan SQLmap, penelitian ini akan mengeksplorasi bagaimana metode eksploitasi dilakukan, dampak yang mungkin timbul dari serangan injeksi SQL, serta rekomendasi langkah-langkah mitigasi untuk mengamankan aplikasi web dari ancaman ini. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam meningkatkan kesadaran akan pentingnya pengamanan database, sekaligus mendorong pengembangan aplikasi web yang lebih aman dan tahan terhadap serangan injeksi SQL [2].

Selain itu, penelitian ini juga diharapkan dapat menjadi acuan bagi pengembang aplikasi, administrator database, dan praktisi keamanan untuk memahami lebih dalam tentang ancaman injeksi SQL. Dengan demikian, mereka dapat mengambil langkah proaktif dalam mengidentifikasi dan menutup celah keamanan sebelum serangan terjadi. Lebih jauh, penelitian ini juga menjadi landasan untuk mengembangkan tools keamanan yang lebih canggih dan responsif terhadap evolusi ancaman siber yang terus berkembang. Dengan pendekatan ini, diharapkan aplikasi web masa depan tidak hanya fungsional, tetapi juga dapat memberikan tingkat keamanan yang tinggi bagi penggunanya [12].

2. Metode

Penelitian ini dilakukan melalui beberapa tahapan yang sistematis untuk mengidentifikasi, menguji, dan menganalisis celah keamanan database aplikasi web terhadap serangan injeksi SQL. Berikut adalah tahapan metode yang digunakan:

a. Identifikasi Target
Tahap ini dimulai dengan pemilihan aplikasi web target untuk pengujian. Target dipilih berdasarkan izin eksplisit untuk melakukan uji penetrasi, dengan memastikan bahwa pengujian dilakukan dalam lingkungan yang aman dan terkendali (legal dan etis).

b. Pengujian Kerentanan
SQLmap digunakan untuk menguji potensi kerentanan injeksi SQL.

c. Eksploitasi Celah Keamanan
Setelah kerentanan teridentifikasi, SQLMap digunakan untuk mengeksploitasi celah tersebut, seperti membaca nama tabel, kolom, atau bahkan mengekstrak data dari database. Hasil eksploitasi dicatat untuk analisis lebih lanjut.

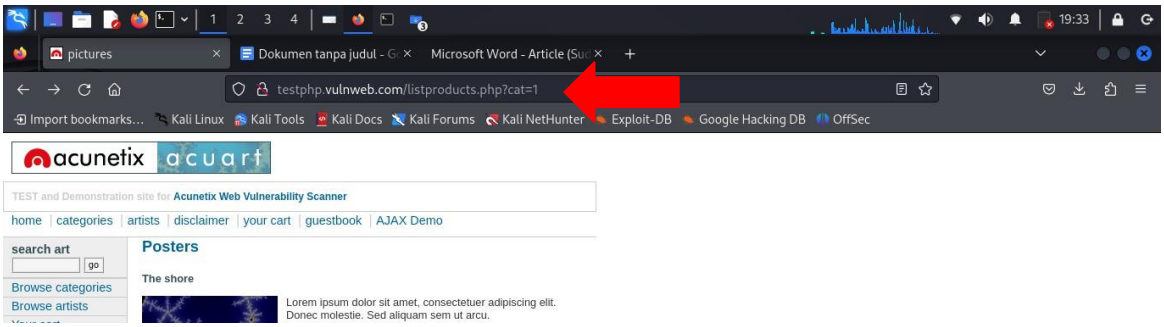
d. Analisis Dampak
Data yang berhasil diekstrak dianalisis untuk mengukur dampak potensial dari serangan. Analisis ini mencakup potensi kehilangan data, kebocoran informasi sensitif, atau kerusakan pada sistem aplikasi.

e. Rekomendasi dan Mitigasi
Berdasarkan hasil pengujian, rekomendasi diberikan untuk memperbaiki celah keamanan, seperti validasi input yang lebih ketat, penggunaan query parameterized, serta pembaruan sistem secara berkala.

Metode ini dirancang untuk memberikan pemahaman yang mendalam tentang proses pengujian penetrasi terhadap celah keamanan injeksi SQL, sekaligus memastikan bahwa pengujian dilakukan secara bertanggung jawab dan mematuhi standar etika.

3. Hasil dan Pembahasan

Pada tahapan ini dilakukan identifikasi terkait web yang memiliki celah terhadap SQLi. hasilnya dapat dilihat pada Gambar 1.

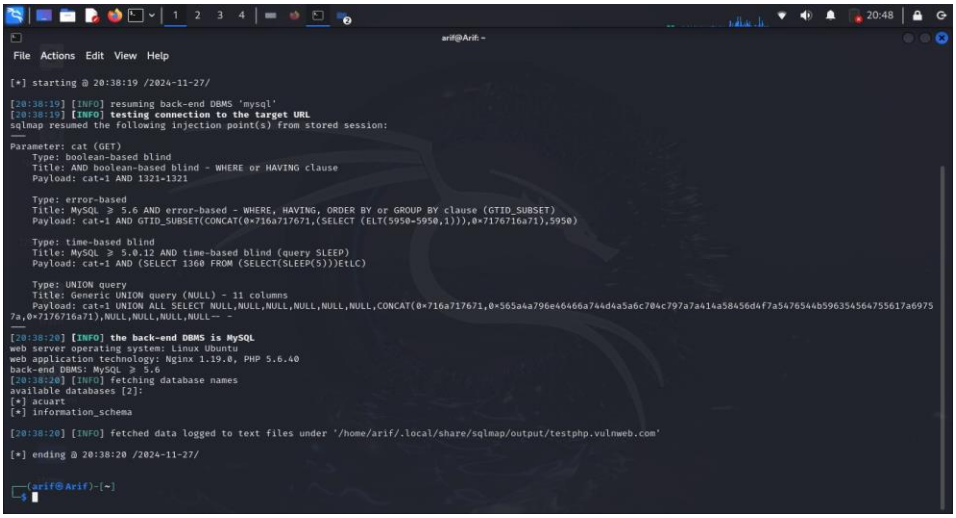


Gambar 1. Identifikasi kerentanan

Gambar 1. Menunjukkan pada url terakhir adanya tulisan php?cat=1 atau sejenisnya, sehingga ada potensi ataupun kerentanan SQLi pada web tersebut. Pada Gambar 2. Dapat dilihat langkah untuk mendapatkan database yang ada dalam web menggunakan tools SQLMap.

Gambar 2. Pencarian database dengan perintah -dbs

Pada tahap ini, Sqlmap akan mencoba menyuntikkan perintah-perintah SQL ke dalam situs web target. Setelah proses selesai, akan diperoleh berbagai perintah SQL yang dapat digunakan, serta nama-nama *database* yang terdapat di dalam situs target, yaitu acuart dan information_schema, seperti yang ditampilkan pada Gambar 3.



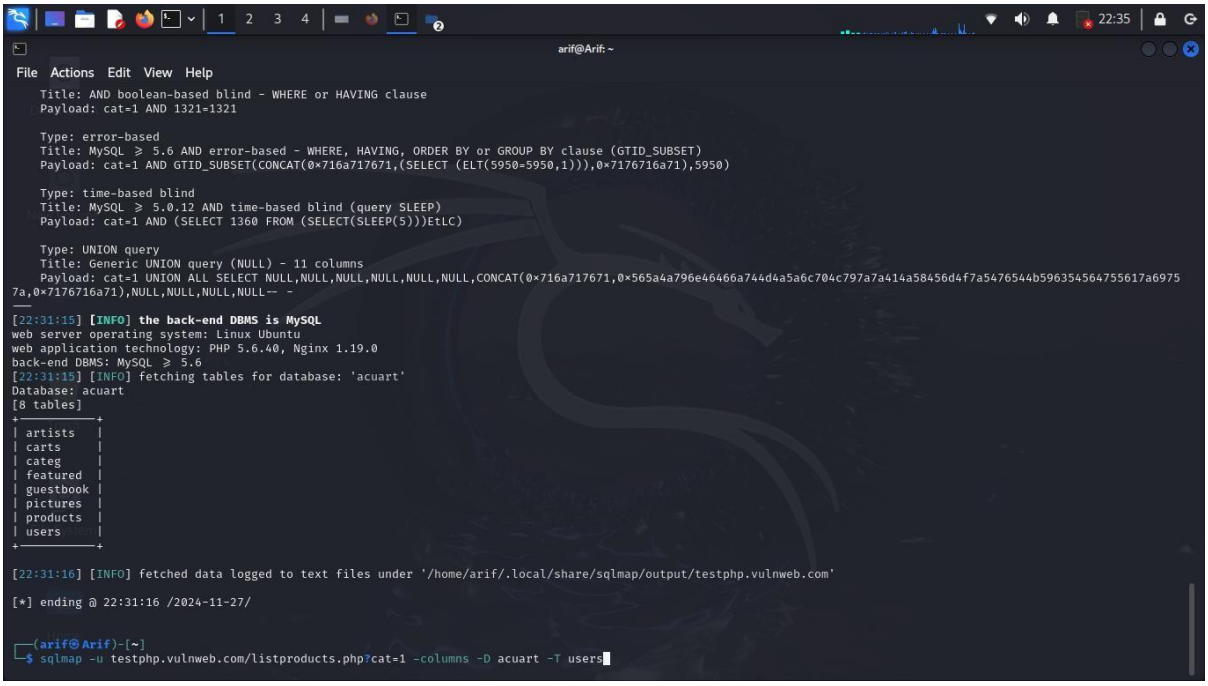
Gambar 3. Hasil database yang ditemukan

Tahap selanjutnya yaitu menggunakan Sqlmap untuk menampilkan daftar tabel dari *database* yang digunakan. Tujuannya adalah untuk memperoleh data-data sensitif yang tersimpan dalam situs web tersebut

dengan menggunakan perintah sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -tables. Seperti yang terlihat pada gambar 4.

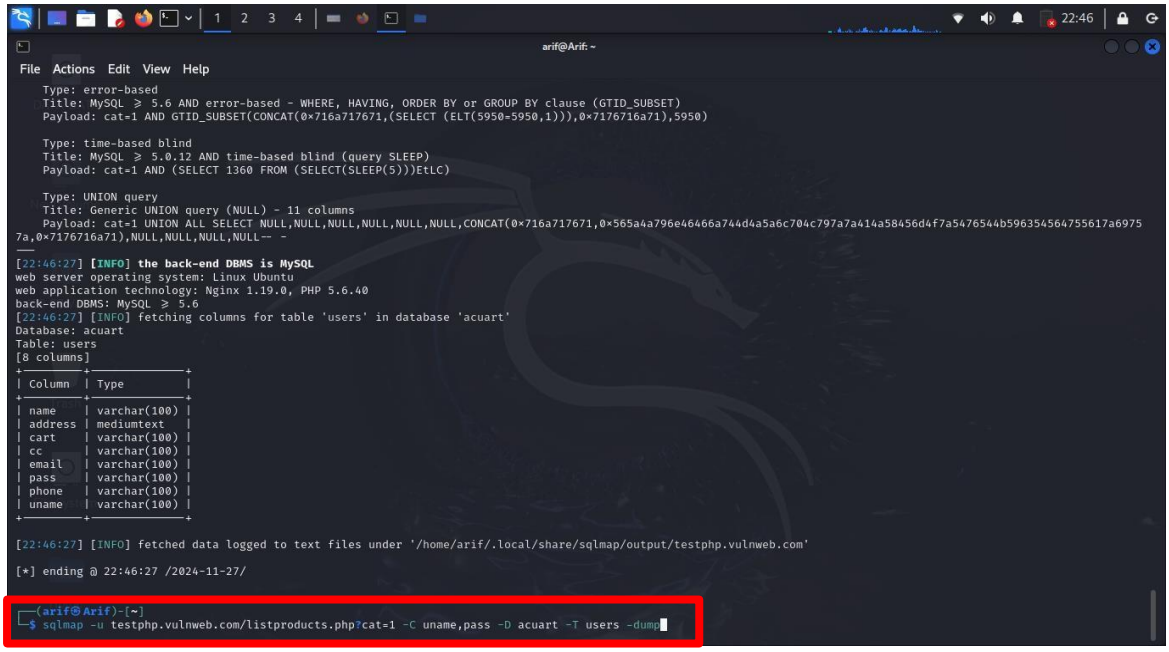
Gambar 4. Perintah Sqlmap (-tables)

Setelah proses selesai, sistem akan menampilkan tabel-tabel yang terdapat dalam database acuart dan information_schema. Kedua database ini merupakan bagian dari struktur data situs web yang dianalisis. Hasil tampilan tersebut dapat dilihat pada Gambar 5 yag merupakan proses enumerasi tabel.



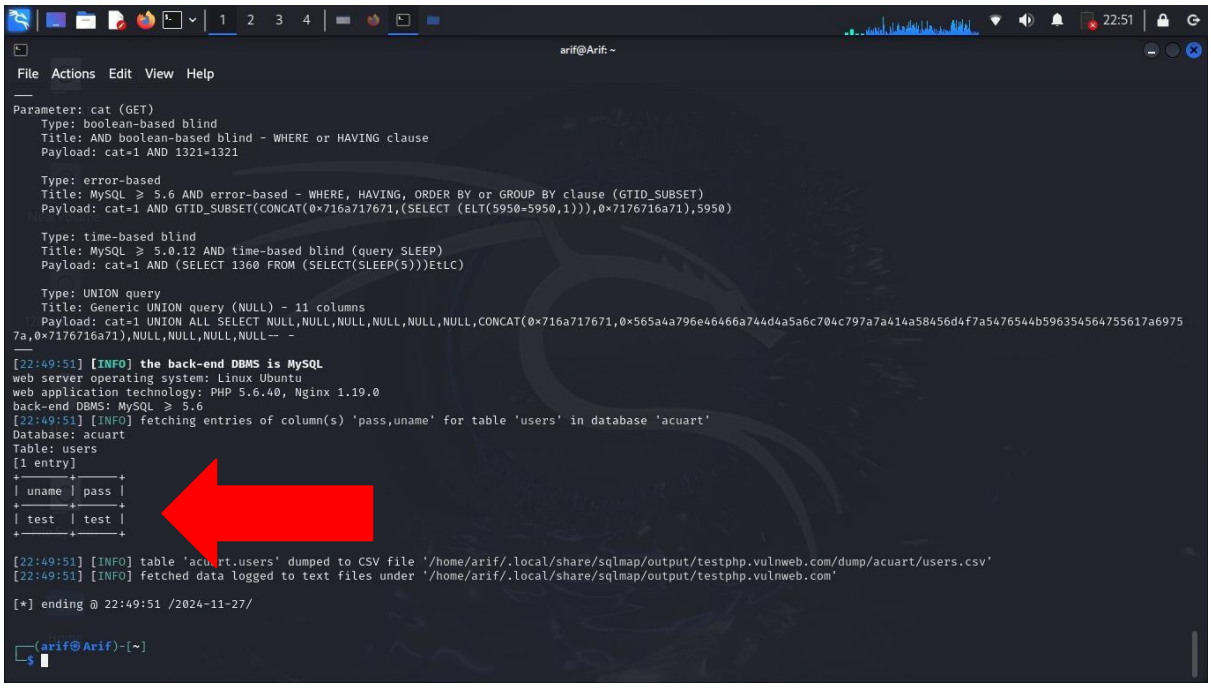
Gambar 5. Hasil Dari Enumerasi Tabel Pada *Database Acuart*

Gambar 5 menunjukkan bahwa terdapat delapan tabel yang ditemukan dalam *database* acuart. Setelah memperoleh daftar tabel tersebut, langkah selanjutnya adalah melakukan pencarian terhadap tabel yang bernama *users*. Tabel ini berpotensi menyimpan informasi penting, seperti *username* dan *password*. Hasil dari pencarian ini ditampilkan pada Gambar 6.



Gambar 6. Hasil dari tabel users

Gambar 6 merupakan hasil dari tabel *users*, di mana ditemukan delapan kolom yang terdapat dalam tabel tersebut. Salah satu kolom yang teridentifikasi adalah uname, yang kemungkinan berisi data nama pengguna. Informasi ini berpeluang untuk mengekstrak isi dari kolom uname. Hasil ekstraksi tersebut ditampilkan pada Gambar 7.



Gambar 7. Hasil columns uname

Gambar 7 menampilkan hasil ekstraksi dari kolom uname pada tabel *users*. Berdasarkan hasil tersebut, dapat disimpulkan bahwa situs web ini memiliki tingkat keamanan yang rendah terhadap kerentanan injeksi SQL. Hal ini memungkinkan penyerang untuk menyuntikkan perintah SQL secara langsung dan memperoleh data sensitif dari *database*.

Berdasarkan hasil pengujian, disarankan agar pengembang aplikasi web menerapkan beberapa langkah mitigasi untuk meningkatkan keamanan sistem dari potensi serangan SQLi. Langkah-langkah berikut dapat dilakukan:

a. Validasi Input
   Pastikan semua input dari pengguna divalidasi dengan baik untuk mencegah penyisipan perintah SQL berbahaya. Gunakan whitelist untuk memastikan hanya data yang diizinkan yang dapat diterima oleh sistem.
b. Penggunaan Parameterized Query atau Prepared Statement
   Hindari penggunaan query SQL dinamis yang langsung menggabungkan input pengguna. Sebagai gantinya, gunakan parameterized query atau prepared statement untuk memastikan bahwa input pengguna diperlakukan sebagai data, bukan perintah SQL.
c. Penerapan Web Application Firewall (WAF)
   Gunakan WAF untuk mendeteksi dan memblokir serangan SQL Injection secara real-time. Firewall ini dapat memberikan perlindungan tambahan terhadap serangan yang tidak terdeteksi oleh mekanisme validasi input.
d. Audit dan Pengujian Rutin
   Lakukan audit keamanan dan pengujian penetrasi secara rutin menggunakan tools seperti SQLMap untuk mengidentifikasi potensi kerentanan baru yang mungkin muncul seiring pembaruan aplikasi.
e. Pengelolaan Hak Akses Database
   Batasi hak akses pengguna terhadap database. Misalnya, akun yang digunakan oleh aplikasi untuk mengakses database sebaiknya hanya memiliki hak akses minimum yang diperlukan, seperti hanya membaca atau menulis data tertentu.
f. Penerapan Enkripsi Data
   Gunakan enkripsi pada data sensitif yang tersimpan di database untuk mencegah penyalahgunaan informasi jika terjadi pelanggaran keamanan.

Dengan menerapkan rekomendasi ini, aplikasi web dapat lebih terlindungi dari ancaman SQL Injection, sehingga keamanan data pengguna dan integritas sistem dapat terjaga dengan baik.

4. Kesimpulan

Berdasarkan tujuan penelitian, pengujian dilakukan menggunakan tools SQLMap dengan menyuntikkan perintah SQL. Hasil pengujian menunjukkan bahwa username dan password berhasil ditemukan pada situs web yang tidak memiliki kerentanan terhadap SQL Injection (SQLi). Hal ini mengindikasikan bahwa metode pengujian yang digunakan cukup efektif dalam mengidentifikasi potensi kelemahan atau celah keamanan pada aplikasi web. Selain itu, hasil ini memberikan wawasan penting mengenai pentingnya implementasi mekanisme keamanan yang lebih kuat untuk melindungi data pengguna. Lebih lanjut, penelitian ini juga menekankan perlunya pengujian rutin untuk memastikan bahwa sistem tetap aman dari ancaman serangan berbasis SQL.

5. References

[1] A. Dos Santos, G. S. Pereira, R. A. Syuhada, dan E. M. S. Sakti, “Uji Coba Keamanan Database Website Menggunakan Python Dan Sqlmap Melalui Command Prompt Pada Sistem Operasi Windows,” *J. Ilm. Tek. Inform.*, vol. 25, no. 1, hal. 146–153, 2024, [Daring]. Tersedia pada: https://doi.org/10.37817/tekinfo.v25i1.

[2] A. Fadlil, I. Riadi, dan M. A. Mu’Min, “Mitigation from SQL Injection Attacks on Web Server using Open Web Application Security Project Framework,” *Int. J. Eng. Trans. A Basics*, vol. 37, no. 4, hal. 635–645, 2024, doi: 10.5829/ije.2024.37.04a.06.

[3] Bangkit Wiguna, W. Adi Prabowo, dan R. Ananda, “Implementasi Web Application Firewall Dalam Mencegah Serangan SQL Injection Pada Website,” *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, hal. 245–256, 2020, doi: 10.31849/digitalzone.v11i2.4867.

[4] R. Hermawan, "Teknik Uji Penetrasi Web Server Menggunakan SQL Injection dengan SQLmap di Kalilinux," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 6, no. 2, hal. 210–216, 2021, doi: 10.30998/string.v6i2.11477.

[5] M. A. Mu'min, A. Fadlil, dan I. Riadi, "Analisis Keamanan Sistem Informasi Akademik Menggunakan Open Web Application Security Project Framework," *J. Media Inform. Budidarma*, vol. 6, no. 3, hal. 1468–1475, 2022, doi: 10.30865/mib.v6i3.4099.

[6] P. G. S. Adinata, I. P. W. P. Putra, N. P. A. I. Juliantari, dan K. D. A. Sutrisna, "Analisis Perbandingan Tools SQL Injection Menggunakan SQLmap, SQLsus dan The Mole," *Inform. J. Ilmu Komput.*, vol. 18, no. 3, hal. 286–292, 2022, doi: 10.52958/iftk.v18i3.5373.

[7] S. E. Prasetyo, H. Haeruddin, dan K. Ariesryo, "Website Security System from Denial of Service attacks, SQL Injection, Cross Site Scripting using Web Application Firewall," *Antivirus J. Ilm. Tek. Inform.*, vol. 18, no. 1, hal. 27–36, 2024, doi: 10.35457/antivirus.v18i1.3339.

[8] M. D. Purnomo, A. Chusyairi, U. B. Insani, S. Jaya, dan K. Bekasi, "Pengujian Keamanan Sistem Mengunakan Metode Penetration Testing di Website Diskominfostandi Kota Bekasi," *Sist. J. Ilm. Sist. Inf.*, vol. 1, no. 1, hal. 92–101, 2024, doi: 10.69533.

[9] M. A. Z. Risky dan Y. Yuhandri, "Optimalisasi dalam Penetrasi Testing Keamanan Website Menggunakan Teknik SQL Injection dan XSS," *J. Sistim Inf. dan Teknol.*, vol. 3, no. 4, hal. 215–220, 2021, doi: 10.37034/jsisfotek.v3i4.68.

[10] R. Rahman *et al.*, "Pengujian Penetrasi Jaringan Menggunakan OWASP Zap dan SQLM ap" *J. Ris. Sist. Inf.*, vol. 1, no. 4, hal. 9–12, 2024, doi: 10.69714/e4rhmk70.

[11] A. Riyanti, B. M. Rahmanto, D. R. Hardianto, R. D. A. Yuristiawan, dan A. Setiawan, "Uji Penetrasi Injeksi SQL terhadap Celah Keamanan Database Website menggunakan SQLmap," *J. Internet Softw. Eng.*, vol. 1, no. 4, hal. 1–9, 2024, doi: 10.47134/pjise.v1i4.2623.

[12] I. Riadi, A. Fadlil, dan M. Amirul, "OWASP Framework-Based Network Forensics to Analyze the SQLi Attacks on Web Servers," *Matrik J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 3, hal. 481–493, 2023, doi: 10.30812/matrik.v22i3.3018.