ANALISIS KERENTANAN SQL INJECTION MENGGUNAKAN SQLMAP PADA KALI LINUX

Nanda Dwi Kurniawan[1]*, Azmi Maulana Firdaus[2], Fausta Rizky Abriansah[3], Praditya Rendi Ferdian[4], Susanto[5]
[1,2,3,4,5]Sistem Informasi, Universitas Semarang
nda22kurniawan@gmail.com[1], azmimaulanaf23@gmail.com[2], semarangkiki40@gmail.com[3], praditya.rendi88@gmail,com[4], susanto@usm.ac.id[5]

Submitted December 12, 2024; Revised April 22, 2025; Accepted June 21, 2025

Abstrak

SQL *injection* merupakan salah satu ancaman serius dalam keamanan website yang dapat membahayakan integritas *database*. Penelitian ini berfokus pada penggunaan Sqlmap yang terintegrasi dalam sistem operasi Kali Linux untuk menganalisis kerentanan *database website*. Kali Linux dipilih karena merupakan distribusi Linux yang dirancang khusus untuk pengujian penetrasi dan telah dilengkapi dengan berbagai tools keamanan siber, termasuk Sqlmap sebagai *framework open-source* yang handal untuk mendeteksi dan menganalisis kerentanan SQL *injection* secara otomatis. Tujuan utama penelitian adalah menyusun metodologi sistematis dalam menganalisis kerentanan *database website* melalui teknik SQL *injection*. Metodologi mencakup tahapan instalasi dan konfigurasi Sqlmap pada Kali Linux, implementasi pengujian penetrasi, serta analisis komprehensif terhadap hasil temuan kerentanan. Penelitian ini juga menekankan pentingnya penerapan mekanisme keamanan dan strategi mitigasi untuk melindungi integritas data dari serangan SQL *injection*. Hasil penelitian ini diharapkan dapat memberikan kontribusi pada pengembangan praktik keamanan siber yang lebih baik.
Kata Kunci : SQL Injection, Website, Kali Linux, Sqlmap, Keamanan Siber

Abstract

SQL injection is one of the serious threats in website security that can jeopardize the integrity of the database. This research focuses on using Sqlmap integrated in Kali Linux operating system to analyze website database vulnerability. Kali Linux was chosen because it is a Linux distribution specifically designed for penetration testing and has been equipped with various cybersecurity tools, including Sqlmap as a reliable open-source framework to detect and analyze SQL injection vulnerabilities automatically. The main objective of the research is to develop a systematic methodology in analyzing website database vulnerabilities through SQL injection techniques. The methodology includes the installation and configuration of Sqlmap on Kali Linux, implementation of penetration testing, and comprehensive analysis of the vulnerability findings. This research also emphasizes the importance of implementing security mechanisms and mitigation strategies to protect data integrity from SQL injection attacks. The results of this research are expected to contribute to the development of better cybersecurity practice..
Keywords : *SQL Injection, Website, Kali Linux, Sqlmap, Cyber Security*

1. PENDAHULUAN

Keamanan website telah menjadi perhatian utama dalam era digital saat ini, mengingat semakin meningkatnya ancaman siber yang dapat membahayakan integritas data dan sistem [1]. Salah satu ancaman yang paling umum dan berbahaya adalah SQL *injection*, sebuah teknik eksploitasi yang memanfaatkan kerentanan pada lapisan database *website* [2].

Penelitian terdahulu yang dilakukan oleh Putranto dkk. [3] menganalisis kerentanan *website* pembelajaran elektronik universitas terhadap dua jenis serangan siber utama. Dengan metodologi penetration testing, peneliti berhasil mendemonstrasikan bahwa *website* LEADS UPNVJ telah menerapkan

mekanisme keamanan yang memadai dalam menghadapi serangan SQL injection dan *sniffing attack*. *Website* tersebut dilindungi oleh *Web Application Firewall* (WAF) yang mampu mencegah injeksi kueri SQL berbahaya dan mengimplementasikan protokol *Transport Layer Security* (TLS) untuk mengenkripsi komunikasi data antara *client* dan *server*. Penelitian ini menekankan pentingnya pembaruan berkala terhadap sistem keamanan dan sertifikat TLS, pengujian penetrasi rutin, serta selalu menggunakan versi terbaru dari *browser* untuk memastikan keamanan website tetap terjaga dalam menghadapi serangan siber yang terus berkembang.

Lebih lanjut oleh Supartini [4] membahas kerentanan aplikasi *web* terhadap serangan yang menargetkan lapisan *database.* Penelitian ini menganalisis efektivitas penerapan teknik Regular Expression sebagai metode untuk mendeteksi pola serangan SQL Injection pada *website.* Supartini mengembangkan arsitektur sistem yang terdiri dari komponen *filtering input data, Regular Expression check*, mekanisme *blocking* dan *alerting*, serta sistem *logging*. Melalui pengujian menggunakan *tools* SQLMap pada *website* yang rentan, penelitian ini membuktikan bahwa parameter *input* yang tidak dilindungi dengan baik dapat menjadi celah keamanan yang serius. Meskipun metode Regular Expression terbukti dapat digunakan untuk mendeteksi serangan SQL *Injection*, peneliti menyimpulkan bahwa penggunaannya sebaiknya dikombinasikan dengan teknik keamanan lainnya seperti *prepared statements, parameterized queries*, dan validasi input untuk memberikan perlindungan komprehensif terhadap serangan database yang semakin canggih.

Dan penelitian yang dilakukan Riyanti dkk. [5] keamanan database website merupakan hal yang krusial di era digital saat ini, karena serangan siber seperti SQL Injection terus mengalami peningkatan. Penelitian tersebut menunjukkan bahwa dengan menggunakan alat SQLmap pada sistem operasi Kali Linux, penyerang dapat dengan mudah mengeksploitasi celah keamanan untuk mengakses data penting seperti username dan password melalui skrip query SQL khusus yang ditulis dalam bahasa pemrograman Python. Hasil pengujian penetrasi memperlihatkan bahwa banyak website masih memiliki kerentanan karena kurangnya validasi input pengguna. Penelitian ini menyoroti pentingnya penerapan solusi pencegahan seperti validasi input, pembatasan hak akses database, penggunaan *Web Application Firewall* (WAF), pemindaian keamanan berkala, dan peningkatan kesadaran keamanan, serta langkah-langkah penanggulangan efektif seperti deteksi dini, isolasi sistem, pemulihan data, analisis menyeluruh, dan pelaporan insiden untuk meminimalkan dampak serangan SQL *Injection.*

Berdasarkan data yang dipublikasikan oleh *Central Intelligence Agency* (CIA), Indonesia mengalami kerugian signifikan akibat serangan dan kejahatan siber yang mencapai USD 895 miliar. Angka ini mewakili 1,20% dari total estimasi kerugian global yang tercatat sebesar USD 71,620 miliar [6]. Besarnya dampak finansial dari aktivitas kejahatan siber ini memerlukan perhatian serius dari seluruh pemangku kepentingan. Terlebih lagi, Indonesia saat ini masih belum memiliki mekanisme yang komprehensif dalam sistem pertahanan dan keamanan nasional di ranah siber (cyberspace). Kondisi ini menunjukkan adanya urgensi untuk mengembangkan dan mengimplementasikan strategi keamanan siber yang lebih terstruktur di tingkat nasional. Hal ini menunjukkan urgensitas dalam melakukan pengujian dan evaluasi keamanan *website* secara berkala untuk mengidentifikasi dan memitigasi potensi kerentanan, terutama terhadap SQL *injection.*

Kali Linux, sebagai distribusi Linux yang dirancang khusus untuk pengujian penetrasi, menyediakan berbagai tools keamanan yang terintegrasi, termasuk Sqlmap dan *Dictionary Attack* yang menggunakan *tools Hashcat* [7]. Sqlmap adalah alat penetrasi *open-source* yang dirancang khusus untuk mengotomatisasi proses deteksi dan eksploitasi kerentanan injeksi SQL pada *server web*, alat ini memungkinkan pengguna untuk secara sistematis mengidentifikasi dan mengambil alih basis data yang rentan terhadap serangan SQL *Injection* [8]. Dictionary attack adalah metode serangan siber di mana penyerang mencoba membobol akun dengan mencocokkan password menggunakan daftar kata-kata yang umum atau sudah tersedia dalam kamus (seperti rockyou.txt) [9]. Serangan ini sering menargetkan akun-akun dengan nama pengguna yang sudah diketahui (misalnya admin atau user) dan password pendek atau berbasis kata-kata populer (contoh: "mobil"), sehingga mudah ditebak. Jika berhasil, penyerang dapat menguasai sistem sepenuhnya, termasuk *Active Directory Domain Services*. Oleh karena itu, penggunaan password panjang (minimal 15 karakter) dan kompleks sangat disarankan untuk mengurangi risiko serangan ini. Hash attack adalah upaya membongkar atau memecahkan nilai *hash* (hasil enkripsi *password*) untuk mendapatkan teks aslinya (*plaintext*) dengan menggunakan teknik tertentu, seperti dictionary attack atau brute-force [10]. *Hash* sendiri merupakan hasil transformasi satu arah (one-way function) dari password menggunakan algoritma kriptografi (contoh: MD5, SHA-1). Serangan ini menjadi efektif ketika sistem menggunakan algoritma *hash* yang lemah atau tidak menerapkan teknik pengamanan tambahan seperti salting.

Penelitian ini bertujuan untuk mengembangkan metodologi sistematis dalam menganalisis kerentanan *database website* menggunakan Sqlmap pada *platform* Kali Linux. Fokus utama penelitian meliputi implementasi pengujian penetrasi SQL *injection*, analisis hasil temuan, serta rekomendasi strategi mitigasi untuk meningkatkan keamanan *website.* Metodologi yang dikembangkan diharapkan dapat memberikan kontribusi signifikan dalam praktik keamanan siber, khususnya pada konteks pengamanan *database website* dari serangan SQL *injection.*

Penelitian ini memberikan manfaat praktis dan teknologis dalam menganalisis kerentanan *database website*. Melalui penggunaan Sqlmap di platform Kali Linux, penelitian menyediakan panduan konkret untuk mengidentifikasi dan memitigasi kerentanan SQL injection, sekaligus memperkenalkan *tools* penetrasi *open-source* yang memungkinkan pengujian otomatis keamanan *website*. Pendekatan sistematis ini mendukung upaya pengurangan risiko kerentanan *database* dan meningkatkan kesadaran akan pentingnya pengujian keamanan berkelanjutan.

2. METODE PENELITIAN

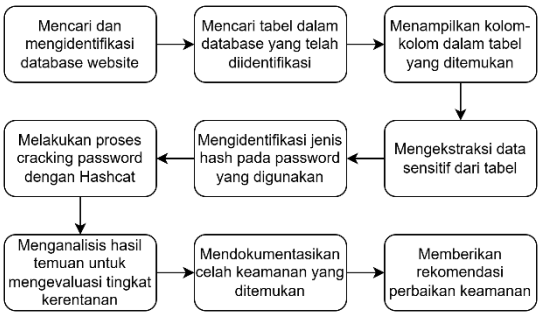
Penelitian ini menggunakan metode Deskriptif Kualitatif dengan Pendekatan Studi Penetrasi (*Penetration Testing*). Metode Deskriptif kualitatif adalah metode penelitian yang menguraikan, menggambarkan, dan merangkum berbagai kondisi berdasarkan informasi yang diperoleh melalui wawancara atau pengamatan langsung terhadap objek penelitian [11]. Karakteristik utama metode ini adalah menampilkan data secara apa adanya, tanpa melakukan manipulasi atau memberikan perlakuan tambahan pada data yang diteliti [12].

Objek penelitian ini menggunakan *website* neutralposture.com, khususnya pada halaman products.php yang memiliki parameter dinamis pada URL. *Website* ini dipilih sebagai target pengujian karena

merupakan contoh nyata dari implementasi *database web* yang dapat menunjukkan kerentanan SQL *injection* dalam lingkungan terkontrol. Perlu dicatat bahwa pengujian ini dilakukan dengan izin penuh dari pemilik *website* dan dalam lingkungan pengujian yang terisolasi untuk menghindari gangguan pada operasional website yang sebenarnya. *Website* tersebut digunakan sebagai studi kasus untuk mendemonstrasikan tahapan identifikasi dan analisis kerentanan SQL i*njection* menggunakan SQLmap pada platform Kali Linux. Pemilihan objek penelitian dilakukan melalui proses *scanning* awal menggunakan teknik fingerprinting untuk mengidentifikasi *situs* yang memiliki karakteristik yang sesuai dengan tujuan penelitian [13].

Implementasi penelitian ini menerapkan teknik *Penetration Testing* yang merupakan metode standar dalam industri keamanan informasi untuk mengevaluasi tingkat keamanan sistem [14]. Teknik ini memungkinkan peneliti untuk melakukan simulasi serangan yang terstruktur dan terkontrol guna mengidentifikasi celah keamanan potensial sebelum dapat dieksploitasi oleh pihak yang tidak bertanggung jawab [15]. Melalui pendekatan ini, tim peneliti dapat memetakan kerentanan sistem secara komprehensif dan menghasilkan rekomendasi mitigasi yang spesifik dan relevan.

Tahapan metodologi penelitian secara rinci diilustrasikan pada Gambar 1 dan diimplementasikan sebagai berikut :



Sumber : Dokumen Pribadi

Gambar 1. Flowchart Tahapan Penelitian

Mencari dan mengidentifikasi database website

Tahap awal penelitian ini melibatkan proses pencarian dan identifikasi database yang digunakan oleh website target. Peneliti mencari informasi tentang jenis database, versi, dan konfigurasi yang digunakan oleh website tersebut.

Mencari tabel dalam database yang telah diidentifikasi

Setelah database teridentifikasi, peneliti mencari tabel-tabel yang ada dalam database tersebut. Proses ini bertujuan untuk memetakan struktur database dan mengidentifikasi tabel mana yang mungkin berisi informasi sensitif.

Menampilkan kolom-kolom dalam tabel yang ditemukan

Pada tahap ini, peneliti melihat detail struktur dari tabel-tabel yang telah ditemukan, dengan fokus pada kolom-kolom (fields) yang ada dalam setiap tabel tersebut.

Mengekstraksi data sensitif dari tabel

Peneliti kemudian mengekstrak data yang dikategorikan sebagai sensitif dari tabel-tabel tersebut, seperti informasi pengguna, kredensi autentikasi, atau data pribadi lainnya.

Mengidentifikasi jenis *hash* pada *password* yang digunakan

Tahap ini berfokus pada analisis mekanisme keamanan yang digunakan untuk menyimpan *password* dalam database. Peneliti mengidentifikasi algoritma hash yang digunakan (misalnya MD5, SHA-1, bcrypt, dll).

Melakukan proses cracking password dengan Hashcat

Peneliti mencoba memecahkan (*crack*) *password* yang telah di-hash menggunakan alat khusus bernama Hashcat. Proses ini bertujuan untuk

menguji kekuatan enkripsi password yang digunakan oleh *website.*

Menganalisis hasil temuan untuk mengevaluasi tingkat kerentanan
Pada tahap ini, peneliti melakukan analisis komprehensif terhadap semua temuan untuk menilai seberapa rentan sistem keamanan website tersebut terhadap serangan.

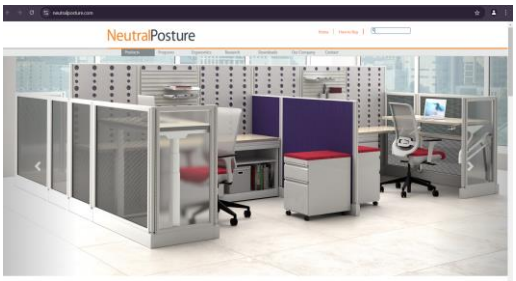
Mendokumentasikan celah keamanan yang ditemukan
Peneliti mendokumentasikan semua kerentanan yang ditemukan selama proses pengujian. Dokumentasi ini mencakup jenis kerentanan, tingkat risiko, dan potensi dampak jika kerentanan ini dieksploitasi.

Memberikan rekomendasi perbaikan keamanan
Sebagai tahap terakhir, peneliti memberikan rekomendasi spesifik untuk memperbaiki celah keamanan yang ditemukan. Rekomendasi ini bertujuan untuk meningkatkan keamanan sistem database website yang diuji.

3. HASIL DAN PEMBAHASAN

Meretas Website Menggunakan SQLMap pada Kali Linux

Dalam penelitian untuk mengidentifikasi dan menganalisis kerentanan SQL Injection pada situs *web*, tim peneliti menggunakan Kali Linux dan SQLmap sebagai alat utama. Tahap awal dimulai dengan pemilihan *website* neutralposture.com sebagai target pengujian yang memiliki parameter dinamis pada URL (http://neutralposture.com/_site/products.php?cat=04). *Website* ini telah memberikan izin untuk pengujian penetrasi dalam rangka penelitian akademis, dengan batasan dan ketentuan yang disepakati bersama. Pemilihan *website* ini didasarkan pada adanya elemen formulir input pengguna dan parameter dinamis pada URL yang sering menjadi titik rentan terhadap serangan SQL injection. Proses pengujian dilakukan dalam lingkungan terkontrol untuk mencegah dampak negatif pada operasional website yang sebenarnya, sesuai dengan etika penelitian keamanan siber.



Sumber : Website Neutral Posture

Gambar 2. Proses Mencari *Website*

Setelah target ditentukan, buka terminal dan arahkan ke direktori sqlmap yang diekstrak untuk memindai database.

Mencari *Database Website*

Langkah awal dalam mengidentifikasi kerentanan keamanan situs *web* adalah menemukan *database* yang terhubung dengan situs target. Dengan menggunakan SQLmap, proses ini menjadi lebih efisien karena alat ini mampu mendeteksi keberadaan dan jenis *database* secara otomatis melalui serangkaian query khusus. SQLmap mengirimkan permintaan yang dirancang untuk memicu respons tertentu dari *server*, memungkinkan peneliti memahami struktur, konfigurasi, dan potensi celah pada database yang digunakan. Teknik ini memberikan fondasi penting untuk tahap eksplorasi lebih lanjut dalam menguji dan menganalisis kerentanan keamanan data, menggunakan seperti ini :

```
sqlmap –u
http://neutralposture.com/_site
/products.php?cat=04 --dbs
```

Sumber : Dokumen Pribadi

Gambar 3. Proses mencari Database website

Mencari Tabel dalam Database

Setelah berhasil mendeteksi database yang terhubung dengan situs target, langkah berikutnya adalah menelusuri tabel-tabel yang terdapat di dalamnya. Proses ini merupakan tahap krusial dalam upaya identifikasi lokasi penyimpanan data sensitif, seperti *username* dan *password*. Dengan menggunakan kemampuan otomatisasi dari SQLmap, tabel-tabel dalam *database* dapat dipetakan secara menyeluruh, memungkinkan peneliti untuk memahami struktur organisasi data. Informasi ini tidak hanya membantu dalam mengevaluasi tingkat keamanan database, tetapi juga memberikan wawasan lebih mendalam tentang potensi celah yang dapat dieksploitasi untuk meningkatkan keamanan sistem. Dengan menggunakan query seperti berikut :

sqlmap –u
http://neutralposture.com/_site
/products.php?cat=04 -D 872128_npweb --
tables

Sumber : Dokumen Pribadi

Gambar 4. Proses Mencari Tabel dalam Database

Menampilkan Tabel dalam *Database*

Setelah tabel dalam *database* berhasil diidentifikasi, langkah berikutnya adalah menganalisis kolom-kolom yang terdapat di dalamnya untuk menemukan informasi penting yang berpotensi sensitif. Pada tahap ini, penelusuran lebih lanjut dilakukan terhadap tabel yang relevan, seperti tabel member, guna mengungkap kolom-kolom yang berhubungan dengan data pengguna, seperti nama, *username*, atau *password.* Proses eksplorasi ini memungkinkan pemetaan struktur data yang lebih rinci, sehingga peneliti dapat memahami pola penyimpanan informasi dalam *database* dan mengidentifikasi celah keamanan yang mungkin ada.

Sumber : Dokumen Pribadi

Gambar 5. Menampilkan Tabel dalam Database

Menampilkan Hasil Kolom yang Terdapat pada Tabel member

Setelah tabel-tabel dalam *database* berhasil diidentifikasi, langkah selanjutnya adalah menelusuri kolom-kolom yang terdapat pada tabel tertentu untuk mendapatkan gambaran lebih rinci mengenai data yang disimpan. Pada tahap ini, tabel member dipilih sebagai target eksplorasi lebih lanjut. Dengan menggunakan perintah-perintah yang disediakan oleh SQLmap, kolom-kolom dalam tabel ini dapat diungkap secara sistematis, memungkinkan peneliti untuk mengidentifikasi atribut-atribut yang menyimpan informasi penting, seperti nama pengguna, *email*, atau kata sandi. Penelusuran ini tidak hanya membantu dalam memahami struktur tabel, tetapi juga menjadi langkah awal untuk mengevaluasi risiko keamanan yang terkait dengan data sensitif di dalamnya, menggunakan query sqlmap –u
http://neutralposture.com/_site
/products.php?cat=04 -D 872128_npweb -T
member –column

Sumber : Dokumen Pribadi

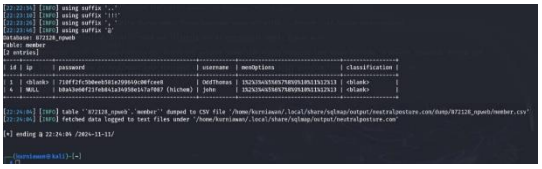
Gambar 6. Hasil Kolom Tabel member

Mengambil Nilai dari Tabel *member*

Setelah kolom-kolom dalam tabel member berhasil diidentifikasi, langkah berikutnya adalah menelusuri isi dari kolom-kolom tersebut untuk menemukan data penting, seperti *username* dan *password*. Dengan menggunakan SQLmap, peneliti dapat melakukan ekstraksi data secara spesifik dari kolom yang relevan, memanfaatkan query otomatis untuk menarik informasi langsung dari *database*. Tahap ini sangat penting untuk memahami bagaimana data sensitif disimpan dan dikelola di dalam sistem. Selain memberikan wawasan tentang potensi kerentanan, langkah ini juga membantu mengidentifikasi kelemahan dalam perlindungan data, seperti penyimpanan kata sandi yang tidak terenkripsi atau pengelolaan akses yang lemah. Dengan menggunakan query sqlmap –u http://neutralposture.com/_site /products.php?cat=04 -D 872128_npweb -T member --dump

Sumber : Dokumen Pribadi

Gambar 7. Proses Mencari Isi dari Username dan Password



Sumber : Dokumen Pribadi

Gambar 8. Hasil Username dan Password yang ada dalam Tabel Member

Identifikasi Jenis Hash pada Data Tereksfiltrasi

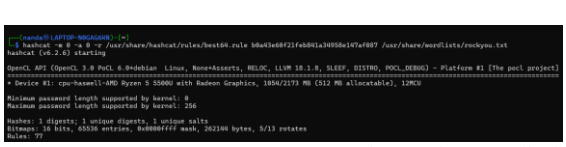
Setelah ekstraksi data *username* dan *password* berhasil dilakukan, tahap selanjutnya adalah melakukan identifikasi jenis hash yang digunakan untuk menyimpan password dalam database. Proses ini dimulai dengan menganalisis karakteristik string hash yang ditemukan, yaitu '710ff2fc5b0eeb581e299649c00fcee8' dan 'b0a43e60f21feb841a34958e147af087'. Kedua string tersebut memiliki panjang tepat 32 karakter heksadesimal, yang merupakan ciri khas dari algoritma hash MD5. Untuk memastikan jenis hash yang digunakan, dilakukan pengujian menggunakan tool hash-identifier pada Kali Linux dengan memasukkan salah satu string hash tersebut. Hasil analisis mengkonfirmasi bahwa hash yang digunakan adalah MD5, algoritma hash yang telah diketahui memiliki banyak kelemahan keamanan dan tidak lagi direkomendasikan untuk menyimpan *password.* Temuan ini memberikan informasi penting tentang tingkat keamanan sistem yang diuji, menunjukkan bahwa website target menggunakan praktik keamanan yang sudah ketinggalan zaman dan rentan terhadap berbagai teknik serangan seperti *rainbow table attack* dan *collision attack.*

Sumber : Dokumen Pribadi

Gambar 9. Hasil Identifikasi Jenis Hash

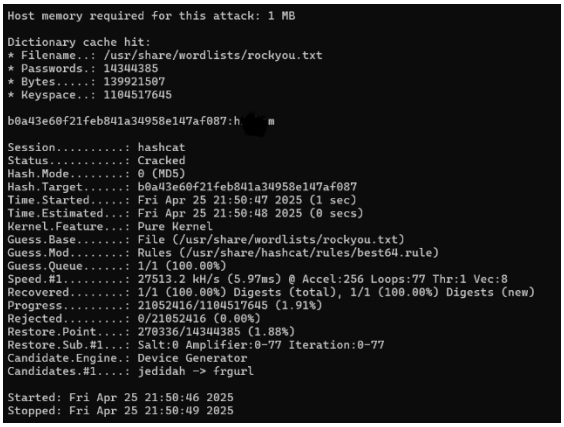
Proses Cracking Password

Setelah berhasil mengekstrak nilai username dan password dari tabel member menggunakan SQLmap, tahap selanjutnya adalah melakukan proses cracking terhadap hash MD5 yang ditemukan menggunakan tool Hashcat pada Kali Linux. Command hashcat -m 0 -a 0 -r /usr/share/hashcat/rules/best64.rule b0a43e60f21feb841a34958e147af087 /usr/share/wordlists/rockyou.txt Dijalankan untuk memecahkan hash tersebut, dengan parameter yang menentukan jenis hash, mode serangan dictionary, aturan transformasi, dan wordlist yang digunakan. Proses ini berhasil memecahkan hash 'b0a43e60f21feb841a34958e147af087' menjadi password 'hichem' dalam waktu sangat singkat (sekitar 3 detik), membuktikan kelemahan signifikan dalam sistem keamanan website yang menggunakan algoritma hash MD5 untuk menyimpan password tanpa teknik pengamanan tambahan seperti salting, sehingga mengekspos data pengguna terhadap risiko pelanggaran keamanan.



Sumber : Dokumen Pribadi

Gambar 10. Proses Cracking Password



Sumber : Dokumen Pribadi

Gambar 11. Hasil Cracking Password

Analisis Kerentanan dan Rekomendasi Mitigasi

Berdasarkan hasil pengujian penetrasi yang telah dilakukan menggunakan SQLmap pada website neutralposture.com, peneliti mengidentifikasi beberapa kerentanan signifikan yang memerlukan penanganan segera. Parameter "cat" pada URL products.php terbukti rentan terhadap serangan SQL injection yang memungkinkan akses tidak sah ke database dengan tingkat risiko kritis. Untuk memitigasi kerentanan ini, penting untuk mengimplementasikan prepared statements atau parameterized queries pada semua interaksi database, serta menerapkan validasi input yang ketat dengan whitelist karakter yang diperbolehkan. Penggunaan *stored procedures* untuk operasi database kompleks juga sangat direkomendasikan. Temuan lain yang mengkhawatirkan adalah penggunaan algoritma hash MD5 tanpa mekanisme salt untuk menyimpan password pengguna, yang dikategorikan sebagai risiko tinggi. Password 'h****m' yang berhasil di-crack dalam waktu sangat singkat membuktikan kelemahan signifikan pada sistem keamanan password. Mitigasi yang diperlukan meliputi migrasi ke algoritma hash modern seperti bcrypt, Argon2, atau PBKDF2, implementasi salt unik untuk setiap password, serta penerapan kebijakan kompleksitas password minimal 12 karakter dengan kombinasi huruf, angka, dan simbol. Penelitian juga menemukan

adanya eksposur struktur database melalui kueri SQL error yang menampilkan informasi sensitif tentang struktur *database,* dengan tingkat risiko medium. Untuk mengatasi hal ini, diperlukan aktivasi *error handling* khusus yang tidak menampilkan detail teknis database, implementasi *Web Application Firewall* (WAF) untuk mencegah serangan probe database, serta melakukan logging kesalahan ke file terpisah daripada menampilkannya kepada pengguna. Manajemen hak akses database yang lemah juga teridentifikasi sebagai kerentanan dengan tingkat risiko medium. Koneksi database menggunakan akun dengan hak akses berlebihan membuka peluang bagi penyerang untuk melakukan eskalasi hak akses. Rekomendasi mitigasi meliputi penerapan prinsip *least privilege* dengan membuat akun database yang hanya memiliki hak akses minimum yang dibutuhkan, pemisahan akun untuk operasi baca dan tulis, serta audit berkala terhadap hak akses *database*. Sebagai rekomendasi umum untuk peningkatan keamanan, disarankan untuk melakukan penetration testing secara berkala (minimal setiap 3 bulan), mengimplementasikan CAPTCHA untuk mencegah serangan otomatis, menerapkan rate limiting untuk mencegah serangan *brute force*, mengenkripsi komunikasi database menggunakan SSL/TLS, memperbarui semua komponen sistem secara teratur, serta mengimplementasikan sistem deteksi intrusi untuk mengenali pola serangan SQL injection. Dengan menerapkan rekomendasi mitigasi ini, keamanan database website dapat ditingkatkan secara signifikan. Penting untuk dicatat bahwa keamanan adalah proses berkelanjutan yang memerlukan evaluasi dan perbaikan terus-menerus.

4. SIMPULAN

Penelitian ini berhasil menganalisis kerentanan SQL *injection* pada *database website* menggunakan SQLmap pada platform Kali Linux, sesuai dengan tujuan utama yang ditetapkan. Melalui metodologi penetration testing yang sistematis, penelitian mengungkap kerentanan signifikan pada *website* target, di antaranya parameter URL yang rentan terhadap injeksi SQL, penggunaan algoritma hash MD5 yang tidak aman untuk penyimpanan *password*, eksposur struktur database, dan manajemen hak akses database yang lemah. Temuan-temuan ini menunjukkan bahwa banyak *website* yang masih belum menerapkan praktik keamanan yang memadai untuk melindungi data sensitif dari serangan SQL injection. Hasil penelitian menegaskan pentingnya pendekatan keamanan berlapis dalam pengembangan aplikasi *web*, khususnya pada level database. Implementasi teknis seperti *prepared statements, algoritma hashing* modern dengan salt, error handling yang aman, dan penerapan prinsip hak akses minimum terbukti sangat diperlukan sebagai langkah mitigasi fundamental. Panduan langkah demi langkah yang disajikan dalam penelitian ini memberikan kontribusi praktis bagi pengembang web dan administrator sistem dalam mengidentifikasi dan mengatasi kerentanan SQL *injection* pada *database website* mereka. Metodologi yang dikembangkan dalam penelitian ini dapat digunakan sebagai dasar untuk pengujian keamanan berkelanjutan, dengan penekanan pada aspek etika dan legalitas. Penting untuk menekankan bahwa pengujian penetrasi harus selalu dilakukan dengan izin resmi dari pemilik sistem dan dalam lingkungan terkontrol untuk menghindari dampak negatif pada operasional sistem yang berjalan. Penelitian ini juga menunjukkan bahwa *tools open-source* seperti SQLmap yang terintegrasi dalam Kali Linux dapat menjadi solusi efektif untuk pengujian keamanan database website tanpa memerlukan investasi besar pada tools komersial. Untuk penelitian selanjutnya, disarankan untuk memperluas cakupan

pengujian pada berbagai jenis database dan platform web yang berbeda, serta mengembangkan metode otomatisasi untuk proses deteksi dan mitigasi kerentanan SQL injection. Selain itu, pengembangan framework evaluasi keamanan yang komprehensif yang mencakup berbagai jenis serangan web lainnya seperti Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), dan Remote File Inclusion (RFI) akan memberikan gambaran yang lebih holistik tentang postur keamanan sebuah aplikasi *web.*

DAFTAR PUSTAKA

[1] A. Dos Santos, G. S. Pereira, R. A. Syuhada, and 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, pp. 146–153, 2024, [Online]. Available: https://doi.org/10.37817/tekinfo.v25i1

[2] Gusti Ayu Mas Ekayanti, Dewa Ayu Deby Cintiya, Putu Yoga Suartana, Rama Ngurah Putera Pinatih, Gede Arna Jude Saskara, and I Made Edy Listartha, "Perbandingan Tools Sql Sus, Sql Ninja, Dan the Mole Dalam Penerapan Sql Injection," *J. Inform. Teknol. dan Sains*, vol. 4, no. 4, pp. 478–482, 2022, doi: 10.51401/jinteks.v4i4.2201.

[3] D. P. Putranto, J. Jayanta, and B. Hananto, "Analisis Keamanan Website Leads UPNVJ Terhadap Serangan SQL Injection & Sniffing Attack," *Inform. J. Ilmu Komput.*, vol. 18, no. 3, p. 230, 2022, doi: 10.52958/iftk.v18i3.4690.

[4] R. Supartini and J. M. Parenreng, "Deteksi Serangan SQL Injection pada Website dengan Menggunakan Metode Reguler Expression," *Progress. Information, Secur. Comput. Embed. Syst.*, vol. 1, no. 2, pp. 107–114, 2023, doi: 10.61255/pisces.v1i2.101.

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

[6] I. Hilmy, Muhammad and N. Azmi, Rama, Halim, "Konstruksi Pertahanan Dan Keamanan Negara Terhadap Perlindungan Data Dalam Cyberspace Untuk Menghadapi Pola Kebiasaan Baru The Construction of State Defense and Security Against Data Protection in Cyberspace to Facing New Habits," *J. Lemb. Ketahanan Nas. Republik Indones.*, vol. 9, no. 1, pp. 114–124, 2021.

[7] S. P. Salsabilah, A. Al Mita, M. Zachwan Irsyad, and E. M. S. Sakti, "Implementasi Penggunaan Kali linux dengan Teknik Ddos dalam Uji coba Keamanan Website," *J. Ilm. Tek. Inform.*, vol. 25, no. 1, pp. 98–106, 2024.

[8] N. A. Prasetiyo, R. B. Huwae, and A. H. Jatmika, "Audit Dan Analisis Website Pemerintah Menggunakan Pengujian Penetrasi Sql Injection dan Cross Site Scripting (XSS)," *J. Teknol. Informasi, Komput. dan Apl.*, vol. 6, no. 2, pp. 525–533, 2024.

[9] N. Sadikin and M. S. Mahardika, "Implementasi Keamanan Server Domain Controller Active Directory Domain Services terhadap Berbagai Threat dan Attack," *J. Maklumatika*, vol. 11, no. 1, pp. 12–21, 2024, [Online]. Available: https://maklumatika.i-tech.ac.id/index.php/maklumatika/article/view/254

[10] D. P. Putra, I. W. A. P. Putra, and I. G. W. P. Sucipta, "Perbandingan Password Attack Menggunakan

Tools Barshwf, Hashcat, dan Hash Cracker Console," *JTIK (Jurnal Tek. Inform. Kaputama)*, vol. 7, no. 1, pp. 181–187, 2023.

[11] M. Ridwan, S. AM, B. Ulum, and F. Muhammad, "Pentingnya Penerapan Literature Review pada Penelitian Ilmiah," *J. Masohi*, vol. 2, no. 1, p. 42, 2021, doi: 10.36339/jmas.v2i1.427.

[12] Rusandi and Muhammad Rusli, "Merancang Penelitian Kualitatif Dasar/Deskriptif dan Studi Kasus," *Al-Ubudiyah J. Pendidik. dan Stud. Islam*, vol. 2, no. 1, pp. 48–60, 2021, doi: 10.55623/au.v2i1.18.

[13] G. A. Saputra, E. I. Alwi, and A. W. M. Gaffar, "Analisis Keamanan Website SIAKAD menggunakan Pentest Tools," *LINIER Lit. Inform. dan Komput.*, vol. 1, no. 4, pp. 379–388, 2024.

[14] F. Fachri, "Optimasi Keamanan Web Server Terhadap Serangan Brute-Force Menggunakan Penetration Testing," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 1, pp. 51–58, 2023, doi: 10.25126/jtiik.20231015872.

[15] H. Haikal Muhammad, A. Id Hadiana, and H. Ashaury, "Pengamanan Aplikasi Web Dari Serangan Sql Injection Dan Cross Site Scripting Menggunakan Web Application Firewall," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 5, pp. 3265–3273, 2024, doi: 10.36040/jati.v7i5.7320.