AUDIT DAN ANALISIS WEBSITE PEMERINTAH MENGGUNAKAN PENGUJIAN PENETRASI SQL INJECTION DAN CROSS SITE SCRIPTING (XSS)

(*Audit and Analysis of Government Websites Using SQL Injection and Cross-Site Scripting (XSS) Penetration Testing*)

Nugroho Agung Prasetiyo, Raphael Bianco Huwae, Andy Hidayat Jatmika
Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: nugrohoap18@gmail.com, [Raphael.bianco.huwae, andy]@unram.ac.id

Abstract

This study aims to analyze the security of government websites, focusing on vulnerabilities caused by SQL Injection and Cross Site Scripting (XSS) attacks. In accordance with Presidential Instruction No. 3 of 2003 on National Policy and Strategy for E-Government Development, government agencies are required to provide digital services through official websites. However, this increase in digitalization presents challenges in the context of cybersecurity. The research applies penetration testing methods to several government websites in East Java, using the OWASP Top 10 as the primary guide. The results reveal that many government websites are vulnerable to SQL Injection and XSS attacks, which could lead to data theft and information manipulation. Recommendations for enhancing security include implementing input validation techniques and regularly updating software. This research contributes to raising cybersecurity awareness in the governmental sector.

Keywords: *Cybersecurity, Penetration Testing, SQL Injection, Cross Site Scripting, OWASP Top 10*

**Corresponding Author*

1. PENDAHULUAN

Website merupakan sekumpulan web yang saling berhubungan yang dapat diakses melalui internet [1]. *Website* telah menjadi platform utama bagi lembaga pemerintah untuk memberikan layanan dan informasi publik secara digital. Mengacu pada Instruksi Presiden No. 3 Tahun 2003 mengenai Kebijakan dan Strategi Nasional Pengembangan E-*Government*, seluruh lembaga pemerintah diwajibkan untuk menyediakan layanan dan informasi digital melalui situs web resmi. Langkah ini bertujuan untuk menjamin hak masyarakat dalam mengakses informasi serta meningkatkan transparansi dan efisiensi layanan publik di era digital yang terus berkembang.

Website pemerintahan juga memungkinkan partisipasi publik dalam bentuk *feedback* dan diskusi mengenai kebijakan yang sedang berlangsung. Meskipun digitalisasi ini meningkatkan efisiensi dan aksesibilitas, tantangan signifikan muncul dalam bentuk ancaman keamanan siber. Sepanjang tahun 2021, Badan Siber dan Sandi Negara (BSSN) menerima 332 laporan mengenai insiden siber, yang mengindikasikan kebutuhan mendesak untuk memperhatikan dan mengelola risiko yang mungkin timbul seiring dengan kemajuan digitalisasi [2].

Gambar 1. Sebaran Jenis Serangan

Dari data yang telah disajikan, jenis serangan yang paling sering dilaporkan adalah SQL *Injection* dan *Cross Site Scripting* (XSS), dengan total 145 laporan yang diterima. Salah satu contoh kasusnya adalah insiden kebocoran data yang terjadi pada Agustus 2021, yang mengakibatkan terungkapnya informasi pribadi dari sekitar 1,3 juta pengguna aplikasi eHAC [3]. Insiden-

insiden ini mendorong lembaga pemerintah untuk secara rutin mengimplementasikan pengujian penetrasi sebagai langkah proaktif dalam mendeteksi dan memperbaiki kerentanan keamanan yang ada pada *website* mereka.

Penelitian ini bertujuan untuk mengukur tingkat kerentanan keamanan pada *website* pemerintah terhadap serangan SQL *Injection* dan XSS, serta mengidentifikasi langkah-langkah mitigasi yang paling efektif untuk meningkatkan keamanannya. Untuk menjalankan pengujian penetrasi secara efektif, diperlukan *framework* yang memberikan panduan dan metodologi yang jelas agar uji penetrasi dilakukan secara sistematis dan mencakup semua aspek keamanan aplikasi.

Dalam penelitian ini, digunakan *framework* dari *Open Web Application Security Project* (OWASP), yang menawarkan berbagai panduan dan alat untuk membantu dalam pengujian keamanan aplikasi. Kelebihan OWASP dibandingkan *framework* lainnya terletak pada komunitasnya yang luas dan aktif, yang terus memperbarui informasi mengenai ancaman dan teknik keamanan terbaru [4]. Salah satu panduan utama OWASP, yaitu OWASP *Top* 10, merinci sepuluh ancaman keamanan aplikasi web yang dianggap paling kritis dan sering menjadi fokus utama dalam pengujian keamanan. Penggunaan OWASP *Top* 10 dalam penelitian ini memberikan pendekatan standar yang diakui secara global untuk mengidentifikasi dan mengelompokkan kerentanan berdasarkan tingkat keparahannya, sehingga memudahkan dalam memberikan rekomendasi mitigasi yang tepat.

Keunikan penelitian ini terletak pada fokus spesifiknya terhadap *website* pemerintah, yang memiliki peran penting dalam penyediaan layanan publik. Selain itu, penelitian ini menggunakan pendekatan pengujian penetrasi yang menargetkan serangan SQL *Injection* dan XSS, dua jenis ancaman siber yang paling umum dihadapi oleh *website* pemerintah. Penelitian ini juga memberikan kontribusi praktis dengan menyediakan rekomendasi mitigasi yang dapat langsung diterapkan oleh pengelola *website* pemerintah untuk memperkuat keamanan mereka dan melindungi data publik yang sensitif.

Penelitian ini akan dilakukan melalui beberapa tahapan kunci. Dimulai dengan studi literatur dan identifikasi kebutuhan perangkat keras dan lunak untuk pengujian keamanan, penelitian ini kemudian melanjutkan dengan pengujian penetrasi pada *website* pemerintah menggunakan metode SQL *Injection* dan XSS yang berpedoman pada OWASP *Top* 10. Hasil dari pengujian tersebut akan dianalisis menggunakan CVSS untuk menilai tingkat keparahan kerentanan, yang selanjutnya akan dijadikan dasar untuk merumuskan rekomendasi mitigasi yang bertujuan meningkatkan keamanan *website* pemerintah.

2. TINJAUAN PUSTAKA

Penelitian [5] mengevaluasi keamanan pada *website* xyz menggunakan metodologi OWASP. Hasil penelitian ini menemukan 2 kerentenan tingkat *high*, 1 kerentanan tingkat *medium* dan 6 kerentanan tingkat *low*. Peneliti dapat menemukan berbagai kerentanan yang kemudian divalidasi untuk memberikan rekomendasi yang dapat diterapkan oleh pengembang aplikasi.

Penelitian [6] melakukan pengujian pada *website eform helpdesk* menggunakan metode OWASP Top 10. Hasil pengujian mengidentifikasi 6 kerentanan keamanan yang termasuk dalam OWASP *Top* 10. Peneliti memberikan rekomendasi untuk memperbaiki kerentanan-kerentanan yang ditemukan agar *website Eform Helpdesk* lebih aman dari serangan siber.

Penelitian [7] mengevaluasi keamanan *website* rental mobil menggunakan *penetration testing* XSS dan SQL *Injection*. Hasil pengujian menunjukkan bahwa pada *website* tersebut terdapat 12 celah keamanan yang rentan terhadap serangan SQL *Injection* dan XSS. Setelah identifikasi celah, dilakukan implementasi fungsi PHP yang menyaring dan menghapus semua karakter spesial yang berbahaya dari *input* pengguna.

Penelitian [8] mengevaluasi kerentanan aplikasi berbasis web menggunakan OWASP. Hasil pengujian menunjukkan bahwa hampir setiap kategori pengujian dapat menemukan kerentanan, menunjukkan pentingnya pendekatan *multi-faceted* dalam mengamankan aplikasi web. Penelitian ini memberikan dasar untuk rekomendasi lebih lanjut dalam meningkatkan keamanan aplikasi web dan bisa dijadikan standar penilaian keamanan untuk aplikasi berbasis web.

Berdasarkan yang telah diuraikan diatas, penulis melakukan penelitian penetration testing pada website lembaga pemerintah dengan metode penetration testing SQL *Injection* dan *Cross Site Scripting* (XSS). Perbedaan penelitian ini dari yang sebelumnya yaitu penelitian ini tidak hanya mengidentifikasi kerentanan, tetapi memberikan rekomendasi praktis terkait pada peningkatan keamanan website pemerintah yang melibatkan layanan publik dan memliki dampak lebih luas.

2.1. Keamanan Informasi

Keamanan Informasi mencakup berbagai praktik dan proses yang bertujuan melindungi informasi dari

ancaman dan risiko yang dapat mempengaruhi kerahasiaan, integritas, dan ketersediaan data [9]. Dalam konteks keamanan siber, perlindungan informasi digital menjadi sangat penting karena data tersebut rentan terhadap berbagai ancaman yang berasal dari aktivitas dunia maya.

2.2. Keamanan *Website*

Keamanan *website* mencakup berbagai praktik dan teknik yang diterapkan untuk melindungi situs web dari ancaman dan serangan yang berpotensi mengganggu kerahasiaan, integritas, serta ketersediaan data dan layanan yang disediakan oleh situs web tersebut [10]. Pada era digital saat ini, menjaga keamanan situs web menjadi sangat krusial karena situs web sering kali menjadi sasaran utama bagi serangan siber.

2.3. Vulnerability Assessment

Vulnerability Assessment (VA) adalah proses yang digunakan untuk mengidentifikasi, menilai, dan mengategorikan tingkat keparahan kerentanan keamanan yang ada pada jaringan komputer, sistem, aplikasi, atau komponen lain dalam ekosistem teknologi informasi [11]. Tujuan dari VA adalah untuk mengidentifikasi potensi kerentanan kritis pada situs web atau infrastruktur, sehingga organisasi dapat mengambil langkah-langkah pencegahan sebelum peretas memanfaatkan kerentanan tersebut.

2.4. *Common Vulnerability and Exposures* (CVE)

Common Vulnerability and Exposures (CVE) adalah sebuah sistem yang menyediakan metode standar untuk mengidentifikasi dan mengkategorikan kerentanan serta eksposur keamanan dalam perangkat lunak dan firmware [12]. CVE memungkinkan para pengembang, peneliti keamanan, dan profesional di bidang keamanan untuk berkomunikasi dengan istilah yang konsisten terkait kerentanan, sehingga memfasilitasi pertukaran informasi yang lebih efisien dan respon yang lebih cepat terhadap ancaman keamanan.

2.5. Burp Suite

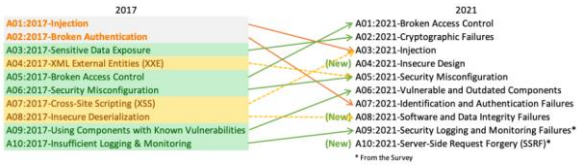
Burp Suite adalah sebuah *platform* terintegrasi yang digunakan untuk menguji keamanan aplikasi web. *Platform* ini dirancang oleh PortSwigger dan menawarkan berbagai alat untuk mendukung berbagai fase pengujian keamanan, mulai dari pemetaan dan analisis permukaan serangan hingga menemukan dan mengeksploitasi kerentanan [13].

2.6. OWASP

OWASP merupakan organisasi yang memainkan peran vital dalam peningkatan keamanan aplikasi web secara global. Melalui berbagai proyek, alat, dan sumber daya yang disediakan, OWASP membantu organisasi dan individu dalam mengidentifikasi, mengatasi, dan mencegah kerentanan keamanan aplikasi [4]. Penggunaan panduan dan alat OWASP dalam industri membantu memastikan bahwa aplikasi web dikembangkan dan diuji dengan standar keamanan yang tinggi, sehingga mengurangi risiko dan meningkatkan kepercayaan pengguna.

2.7. OWASP *Top* 10

OWASP *Top* 10 adalah dokumen yang diterbitkan oleh *Open Web Application Security Project* (OWASP) yang merangkum sepuluh ancaman keamanan aplikasi web yang paling kritis. Daftar ini diperbarui secara berkala untuk mencerminkan perkembangan terbaru dalam ancaman dan kerentanan keamanan aplikasi. OWASP *Top* 10 didasarkan pada data dari berbagai sumber, termasuk vendor keamanan, tim respons insiden, dan komunitas keamanan aplikasi [14].OWASP *TOP* 10 bertujuan untuk meningkatkan kesadaran tentang ancaman keamanan aplikasi web dan mendorong pengembang dan organisasi untuk menerapkan praktik terbaik dalam pengembangan dan pengujian keamanan aplikasi. Daftar ini berfungsi sebagai panduan penting bagi pengembang, penguji, dan manajer proyek dalam memahami dan mengatasi risiko keamanan yang paling umum dan berpotensi merusak. Ancaman pada OWASP *Top* 10 dapat dilihat pada gambar dibawah.



Gambar 2. Kerentanan OWASP *Top* 10

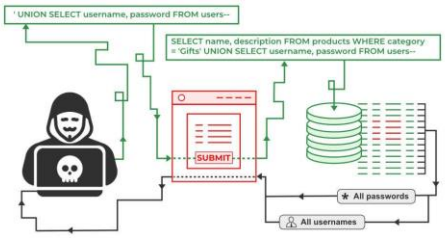
2.8. SQLmap

SQLmap adalah alat penetrasi *open source* yang dapat mengotomatiskan proses deteksi dan eksploitasi kerentanan injeksi SQL serta mengambil alih basis data server web [15]. Dengan kata lain, *SQLmap* adalah alat

yang dapat mendeteksi dan mengeksploitasi *bug* injeksi SQL secara otomatis. Melalui serangan SQL *injection*, seorang penyerang dapat menguasai dan memanipulasi *database* di dalam server web.

2.9. SQL *Injection*

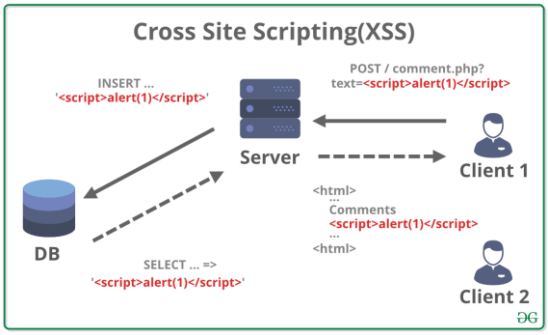
SQL *Injection* merupakan salah satu metode serangan keamanan yang paling umum dan berisiko tinggi pada aplikasi web. Serangan ini memanfaatkan kerentanan dalam lapisan basis data aplikasi web, di mana perintah SQL yang tidak diverifikasi dengan baik dapat disuntikkan (injeksi) ke dalam kueri SQL yang dijalankan oleh aplikasi [16]. Melalui SQL *Injection*, penyerang dapat mengeksekusi perintah yang tidak sah, mengakses data sensitif, memodifikasi atau menghancurkan data, dan bahkan mengambil alih kendali penuh atas sistem basis data.



Gambar 3. Ilustrasi SQL Injection

2.10. *Cross Site Scripting* (XSS)

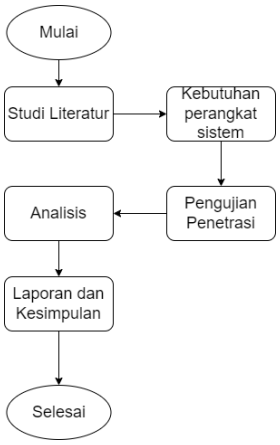
Cross-Site Scripting (XSS) merupakan salah satu bentuk kerentanan keamanan pada aplikasi web yang memungkinkan penyerang menyisipkan skrip berbahaya ke dalam halaman web yang diakses oleh pengguna lain [17]. Kerentanan XSS terjadi ketika aplikasi web menerima *input* dari pengguna tanpa melalui proses validasi atau penyaringan yang memadai, kemudian menggabungkan *input* tersebut ke dalam konten halaman web yang dikirimkan ke pengguna lain. Hal ini memberikan kesempatan bagi penyerang untuk mengeksekusi skrip berbahaya di *browser* pengguna, yang dapat mengakibatkan berbagai aksi merugikan.



Gambar 4. Ilustrasi *Cross Site Scripting*(XSS)

3. METODE PENELITIAN

Berikut merupakan rancangan metode penelitian yang akan dilakukan.



Gambar 5. Metode Penelitian

Berdasarkan Gambar 5 dapat dilihat metode penelitian yang digunakan dalam jurnal terdiri dari studi literatur, kebutuhan perangkat sistem, *penetration testing*, analisa dan rancangan dan laporan dan kesimpulan.

3.1. Studi Literatur

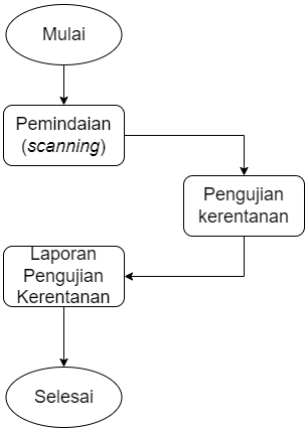
Pada tahap ini penulis melakukan pencarian berbagai referensi dan teori sebagai landasan dalam jurnal ini. Beberapa teori yang mendukung yaitu pemahanan OWASP, penggunaan *tool* dirsearch, subfinder dan httpx yang digunakan untuk melakukan *scanning*, teori *tool burp suite* yang akan digunakan untuk melakukan *penetration testing* dan sumber penelitian atau jurnal terkait terhadap jurnal ini.

3.2. Kebutuhan Perangkat Sistem

Pada tahap ini mengidentifikasi perangkat yang digunakan baik perangkat keras maupun perangkat lunak. Perangkat keras yang digunakan dalam penelitian ini yaitu laptop Lenovo Ideapad Gaming 3 dengan processor AMD Ryzen 7 4800H, Ram 16GB, dan SSD 512GB yang digunakan untuk pengujian sistem dan perangkat lunak yang digunakan yaitu *Burp Suite* untuk pengujian pada *website*.

3.3. Pengujian Penetrasi

Pengujian penetrasi memiliki tahapan seperti pada gambar 6.



Gambar 6. Tahapan Penetrasi

3.3.1. Pemindaian (*scanning*)

Pada tahap pemindaian kerentanan *website* pemerintah, penulis menggunakan beberapa alat, seperti dirsearch untuk mendeteksi direktori yang terbuka, subfinder untuk mengidentifikasi *subdomain*, dan httpx untuk memverifikasi *subdomain* yang aktif atau memiliki potensi celah keamanan yang dapat dieksploitasi.

3.3.2. Pengujian kerentanan

Pada tahap pengujian kerentanan, penulis menggunakan Burp Suite untuk menguji potensi serangan Cross Site Scripting (XSS) dan sqlmap untuk menguji kerentanan terhadap SQL Injection.

3.3.3. Laporan pengujian kerentanan

Pada tahap akhir penelitian, data yang diperoleh dari proses pemindaian dan eksploitasi akan dianalisis untuk menyusun laporan yang dapat menjadi referensi bagi pengembang dalam meningkatkan keamanan website. Data tersebut dikelompokkan berdasarkan jenis kerentanan, sumber kerentanan, dan solusi mitigasinya guna memudahkan proses analisis. Informasi yang dihasilkan selama pengujian dijelaskan secara rinci dengan menggunakan terminologi teknis yang mudah dipahami, sehingga laporan ini menjadi sumber informasi yang berguna dan relevan bagi pengelola website untuk melakukan perbaikan. Laporan penelitian ini mencakup pendekatan yang diterapkan selama proses pengujian penetrasi dan penilaian keamanan yang dilakukan.

3.4. Analisis

Selama proses *penetration testing*, kerentanan pada *website* manajemen aset akan teridentifikasi. Berdasarkan temuan ini, analisis menghitung CVE (*Common Vulnerabilities* and *Exposures*) dengan *Common Vulnerability Scoring System*(CVSS) kalkultar, selain menghitung CVE penulis juga melakukan perangkingan tingkat keretangan *website* menggunakan OWASP *top* 10, mempertimbangkan tingkat keparahan kerentangan, menjelaskan dampak yang dapat ditumbulkan dan menjelaskan mitigasi yang harus dilakukan. Hasil akhir dari proses ini akan memberikan penilaian mengenai tingkat kerentanan keamanan *website* manajemen aset tersebut.

3.5. Laporan dan Kesimpulan

Tahap terakhir yaitu membuat rangkuman analisis yang telah dilakukan ke dalam sebuah kesimpulan, perangkingan OWASP *Top* 10 dan laporan yang detail. *Output* yang dihasilkan akan berisi ringkasan tentang kerentanan keamanan di website manajemen aset, yang akan menjadi dasar untuk tindakan perbaikan.

4. HASIL DAN PEMBAHASAN

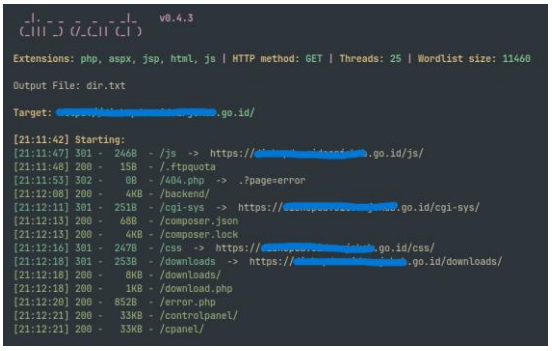
Pada tahap studi literatur, penulis mencari referensi terkait dengan pengujian pentration testing menggunakan metode Injection seperti SQL *Injection* dan *Cross Site Scripting* (XSS).

Pada tahap penelitian pengujian dilakukan pada 5 website berbeda. Daftar website bisa dilihat pada tabel I.

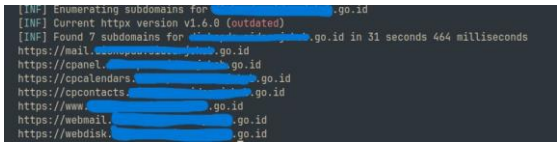
TABEL I. Daftar Website

No	Website
1	https://si***u.bo*****kab.go.id
2	https://sibumbo. bo*****kab.go.id /
3	https://damisda. bo*****kab.go.id /
4	https://bpkad. bo*****kab.go.id /
5	https://diskopda.s****kab.go.id/
6	https://www.m*****nga.com/
7	https://p*****nkab.go.id/
8	https://m*****kab.go.id
9	https://b**d.s*****kab.go.id/
10	http://b**d.p*****kab.go.id

Pada Tabel I ditampilkan website lembaga pemerintah yang akan dilakukan proses scanning untuk mengumpulkan beberapa informasi. Alat yang digunakan yaitu *Dirsearch* untuk mengumpulkan informasi dari struktur *website*, *subfinder* untuk mencari subdomin dari *webste* dan httpx digunkan untuk mengecek domain yang aktif. Seperti yang ditunjukkan pada Gambar 7 dan 8.

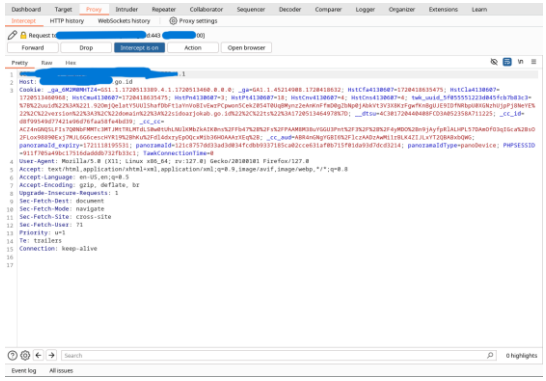


Gambar 7. Hasil *Scanning Dirsearch*



Gambar 8. Hasil *Scanning Subfinder* dan Httpx

Langkah selanjutnya yaitu melakukan pengujian *website*. Untuk pengujian SQL *Injection* dilakukan menggunakan *tool sqlmap*. Sebelum melanjutkan ke *tool sqlmap* dilakukan *scan* pada tempat yang berpotensi rentan pada aplikasi burp suite untuk mendapatkan hasil seperti gambar 9.

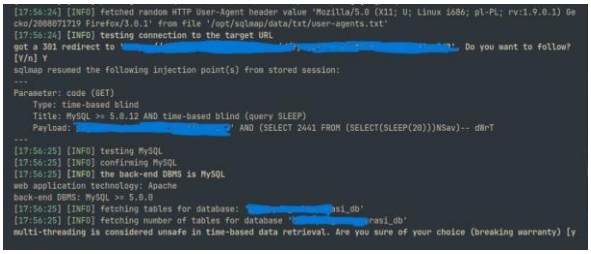


Gambar 9. Hasil Scanning Burp Suite

Setelah didapatkan seperti Gambar 9 hasil *scan* disimpan dengan nama req.txt untuk keperluan *input* pada *tool sqlmap*.

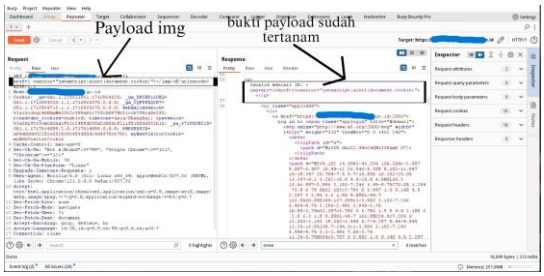
4.1. Penetration Testing

Setelah selesai melakukan *scanning* dan mendapatkan komponen yang diperlukan, selanjutnya melakukan eksekusi dengan *tool sqlmap* untuk mengetahui apakah *endpoint* tersebut rentan dengan SQL *Injection*.



Gambar 10. *Sqlmap* Mendapatkan *Database Website*

Ternyata setelah dilakukan validasi dengan sqlmap *website* tersebut rentan terhadap serangan SQL *Injection* dan didapatkan database dari *website* tersebut. Setelah melakukan pengujian SQL *Injection*, pengujian selanjutnya dilakukan pada website yang berbeda yang berfokus pada *penetration testing* XSS. Saat diuji dengan *payload* XSS menggunakan *payload* <*script*>, ternyata *website* tersbut tidak rentan tapi saat dicoba menggunakan payload XSS menggunakan <img> ternyata website tersebut rentan terhadap XSS. Dapat dilihat pada Gambar 11 bahwa *payload* XSS sudah tertanam dan untuk tampilan pada website dapat dilihat pada Gambar 12.



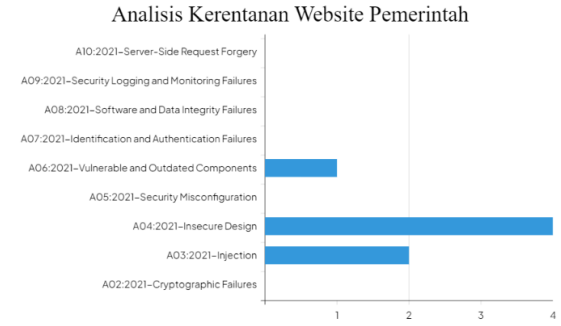
Gambar 11. *Payload* XSS Tertanam

Gambar 12. Tampilan *Website* Apabila *Script* Tertanam

Kerentanan XSS memungkinkan penyerang untuk menjalankan *script* berbahaya di dalam browser pengguna. Hal ini berpotensi menyebabkan pencurian *cookie*, mengambil alih sesi pengguna,

dan eksfiltrasi informasi pribadi. Lebih lanjut, kerentanan ini dapat dimanfaatkan untuk melaksanakan serangan *phishing*, distribusi *malware*, dan pengawasan aktivitas pengguna, yang berdampak negatif pada privasi, keamanan, dan reputasi situs web yang terkena.

Setelah pengujian keamanan selesai, selanjutnya dilakukan penelian keamanan situs pemerintahan memerlukan *freamework* yang memberikan panduan lengkap untuk mengidentifikasi kerentanan dari *website*. OWASP *Top* 10 mencantumkan 10 kerentanan utama pada *website*. *Framework* ini digunakan sebagai referensi untuk penilaian kerentanan dari *website* pemerintah. Pada gambar 13 dapat dilihat bahwa setelah dilakukan *penetration testing* pada beberapa *website*, kerentanan yang tercatat pada website pemerintah dikelompokkan berdasarkan OWASP *Top* 10. Pada gambar tersebut mencatat bahwa ditemukan celah *Injection, Cryptographic Failures* dan *Security Missconfiguration* sedangkan untuk celah lain seperti *Broken Access Control*, *Insecure Design*, *Vulnerable and Outdated*, *Identification and Authentication Failures*, *Software and* Data *Integrity Failures*, *Security Logging and Monitoring Failures* dan *Server-Side Request Forgery* (SSRF) tidak ditemukan pada website.



Gambar 13. Grafik OWASP *Top* 10 *Website* Pemerintah

Pada gambar tersebut ditampilkan pengujian analisis yang ditemukan setelah melakukan penetration testing pada 10 *website*, dimana 1 dari 4 *website* memeliki kerentanan yang serupa. Temuan ini mengidentifikasi bahwa beberapa website pemerintah kurang memperhatikan detail-detail kecil terkait keamanan website yang mana dapat membahayakan pengguna.

4.2. Laporan Penetration Testing

Setelah melakukan pengujian penetrasi pada 10 website menggunakan teknik XSS dan SQL Injection, dihasilkan sebuah laporan yang mencakup rincian kerentanan, dampak serangan, metode pencegahan dan skor CVSS untuk setiap serangan. Laporan ini diharapkan untuk digunakan oleh pengembang website terkait untuk mengevaluasi dan memperbaiki keamanan mereka, termasuk melakukan pembaruan guna mengurangi resiko serangan di masa mendatang.

TABEL II. Efek dan Pencegahan Kerentanan

No	Kerentanan	Efek Kerentanan	Pencegahan kerentanan
1	*Reflected Cross-Site Scripting* pada *address bar website*	Penyerang dapat mencuri *cookie* dari pengguna yang dapat digunakan oleh penyerang untuk mengakses semua yang tersimpan pada *browser* pengguna	Pastikan *input* dari penggunan divalidasi dengan benar, terapkan *Content Security Policy* (CSP), Hindari menampilkan URL langsung tanpa validasi dan lakukan aduit dan pengujian secara berkala
2	SQL *Injection* untuk mendapatkan *database website*	Penyerang dapat memanfaatkan data pada *database* yang bersifat pribadi atau rahasia untuk diperjual belikan/hal negatif lainnya	Menerapkan penggunan an *parameterized queries* atau *prepared statements*, validasi dan sanitasi *input* penggunan, serta membastas

No	Kerentanan	Efek Kerentanan	Pencegahan kerentanan
			i hak akses *database*
3	*Sensitive* data *exposure*	Penyerang dapat mengakses informasi sensitif seperti kunci enkripsi, token dan informasi konfigurasi server yang berisko dieksplotasi.	Perlu menerapkan pembatasan akses publik pada file konfigurasi, memastikan file terkait disimpan di luar direktori *root* publik dan menerapkan hak akses minimum agar hanya pemilik file yang dapat membaca atau menulis file tersebut.
4	*Security Misconfiguration* untuk *bypass* autentikasi	Penyerang dapat mengakses konten yang seharusnya terlindungi tanpa perlu login	Melakukan pembatasan akses terhadap *endpoint* tertentu, menerapkan validasi otentikasi yang ketat di setiap level akses.

Selain mencatat kerentanan, dampak dan metode pencegahan, penulis juga mengkategorikan level dari kerentanan yang ditemukan menggunakan CVSS kalkulator dan didapatkan hasil sebagai berikut : 1 *critical*, 2 *high*, 3 *medium* dan 1 *low*.

5. KESIMPULAN DAN SARAN

Setelah melakukan identifikasi dan analisis kerentanan pada *website* pemerintah menggunakan metode penetrasi SQL *Injection* dan *Cross Site Scripting* (XSS). Didapatkan hasil yaitu 7 dari 10 *website* memiliki kerentanan diantaranya yaitu 4 *Injection*, 2 *Cryptographic Failures* dan 1 Security missconfiguration. Temuan ini diharapkan dapat mencegah serangan siber dengan cara meningkatkan keamanan seperti validasi input, menerapkan *Content Security Policy* (CSP) agar dapat membantu membatasi jenis konten yang dapat dimuat dan dijalankan oleh browser serta menerapkan *prepared statements* agar *input* pengguna tidak diinterpretasikan sebagai bagian dari perintah SQL.

Saran untuk penelitian mendatang agar fokusnya lebih diarahkan pada pengembangan terhadap kerentanan SQL Injection dan XSS pada lingkungan website pemerintah. Penelitian selanjutnya juga sebaiknya memperluas cakupan *website* dengan cara menambahkan *website* yang diuji serta memfokuskan metode yang digunakan untuk mendapatkan hasil yang lebih representatif.

DAFTAR PUSTAKA

[1] B. Tasya Kumala Dewi and M. Andri Setiawan, "Kajian Literatur: Metode dan *Tools* Pengujian Celah Keamanan Aplikasi Berbasis Web," 2022.

[2] Cyberthreat.id, "Serangan SQL *Injection* Jadi Aduan Siber Tertinggi Selama 2021," 2022. [Daring]. Tersedia pada: https://www.cyberthreat.id/read/13925/Serangan-SQL-*Injection*-Jadi-Aduan-Siber-Tertinggi-Selama-2021. [Diakses: 30-Juni-2024].

[3] Suara.com, "Daftar Kasus Kebocoran Data di Indonesia Selama 2021, Termasuk Sertifikat Vaksin Jokowi," 2022. [Daring]. Tersedia pada: https://www.suara.com/tekno/2022/01/01/015822/daftar-kasus-kebocoran-data-di-indonesia-selama-2021-termasuk-sertifikat-vaksin-jokowi. [Diakses: 30-Juni-2024].

[4] A. W. Kuncoro, J. Informatika, F. Rahma, and M. E. Jurusan Informatika, "Analisis Metode *Open Web Application Security Project* (OWASP) pada Pengujian Keamanan *Website*: *Literature Review*," 2021. [*Online*]. Available: https://www.sciencedirect.com

[5] D. F. Priambodo, A. D. Rifansyah, and M. Hasbi, "*Penetration Testing* Web XYZ Berdasarkan OWASP *Risk Rating*," Teknika, vol. 12, no. 1, pp. 33–46, Feb. 2023, doi: 10.34148/teknika.v12i1.571.

[6] R. R. Yusuf and T. N. Suharsono, "Pengujian Keamanan Dengan Metode OWASP *Top* 10 Pada *Website Eform Helpdesk*," *2023*.

[7] Muhammad Arif Zikir Risky and Yuhandri, "Optimalisasi dalam Penetrasi *Testing* Keamanan *Website* Menggunakan Teknik SQL *Injection* dan XSS," Jurnal Sistim Informasi dan Teknologi, pp. 215–220, Aug. 2021, doi: 10.37034/jsisfotek.v3i4.68.

[8] M. Yunus, "Analisis Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi *Security Tools Project* Berdasarkan *Framework* OWASP Versi 4," *Jurnal Ilmiah Informatika Komputer*, vol. 24, no. 1, pp. 37–48, 2019, doi: 10.35760/ik.2019.v24i1.1988.

[9] A. B. Pattiradjawane, F. Setiadi, and R. G. Utomo, "Analisis Manajemen Keamanan Informasi dengan Menggunakan Kontrol ISO 27002:2013 pada Pemerintah Kota Ambon," *LOGIC:* Jurnal Penelitian Informatika, vol. 1, no. 1, p. 59, Sep. 2023, doi: 10.25124/logic.v1i1.6463.

[10] I. Riadi, A. Yudhana, and P. Korspondensi, "Analisis Keamanan *Website Open Journal System* Menggunakan Metode *Vulnerability Assessment*," vol. 7, no. 4, 2020, doi: 10.25126/jtiik.202071928.

[11] E. Z. Darojat, E. Sediyono, and I. Sembiring, "*Vulnerability Assessment Website* E-*Government* dengan NIST SP 800-115 dan OWASP Menggunakan Web *Vulnerability Scanner*," JURNAL SISTEM INFORMASI BISNIS, vol. 12, no. 1, pp. 36–44, Sep. 2022, doi: 10.21456/vol12iss1pp36-44.

[12] R. Azis and S. Yazid, "Pengujian Kerentanan *Website Wordpress* Dengan Menggunakan *Penetration Testing* Untuk Menghasilkan *Website* Yang Aman," vol. 3, no. 3, pp. 93–105, 2021, [*Online*]. *Available*: https://restikom.nusaputra.ac.id

[13] A. Subari, S. Manan, E. Ariyanto, and A. Fauzi, "Pemanfaatan Metode WAVS (*Web Application Security Scanners*) Menggunakan *Burp Suite Tools* Dalam Audit Teknis Keamanan Sistem Informasi Surat Tugas Sekolah Vokasi Undip," 2021, doi: 10.14710/gt.v21i4.46828.

[14] I. O. Riandhanu, "Analisis Metode *Open Web Application Security Project* (OWASP) Menggunakan *Penetration Testing* pada Keamanan *Website* Absensi," Jurnal Informasi dan Teknologi, Oct. 2022, doi: 10.37034/jidt.v4i3.236.

[15] R. Hermawan, "Teknik Uji Penetrasi Web Server Menggunakan SQL *Injection* Dengan *Sqlmap* Di Kalilinux," 2021.

[16] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "*Detection* of SQL *Injection Attack Using Machine Learning Techniques*: A *Systematic* Literature *Review*," *Journal of Cybersecurity and Privacy*, vol. 2, no. 4, pp. 764–777, Dec. 2022, doi: 10.3390/jcp2040039.

[17] M. I. Hany, A. Bhawiyuga, and A. Kusyanti, "Implementasi *Cross Site Scripting Vulnerability Assessment Tools* berdasarkan OWASP *Code Review*," 2021. [*Online*]. *Available*: http://j-ptiik.ub.ac.id