ANALISIS KEAMANAN WEBSITE PEMERINTAH DAERAH ABC MENGGUNAKAN METODE PENETRATION TESTING EXECUTION STANDARD (PTES)

*

1st Laras Wahyu Adiningsih
Fakultas Informatika
Universitas Telkom
Purwokerto, Indonesia
laraswahyuadiningsih@student.telkomuniversity.ac.id

2nd Iqsyahiro Kresna A
Fakultas Informatika
Universitas Telkom
Purwokerto, Indonesia
hiroka@telkomuniversity.ac.id

3rd Pradana Ananda Raharja
Fakultas Informatika
Universitas Telkom
Purwokerto, Indonesia
pradanar@telkomuniversity.ac.id

***Abstrak* — *Website* resmi Pemerintah Daerah ABC digunakan untuk menyampaikan informasi publik dan menerima pengaduan masyarakat, termasuk data pribadi seperti nama, *email*, dan nomor *handphone*. Namun, sebelumnya *website* ini pernah mengalami serangan yang menyebabkan perubahan tampilan, yang diduga merupakan serangan *XSS*. Hal ini menunjukkan adanya celah keamanan pada sistem yang seharusnya melindungi data pengguna. Di sisi lain, *website* ini juga belum memiliki sistem keamanan yang optimal, meskipun menyimpan data sensitif dan beroperasi di lingkungan publik digital. Penelitian ini bertujuan untuk menganalisis keamanan *website* Pemerintah Daerah ABC menggunakan metode *Penetration Testing Execution Standard (PTES)*. Penelitian ini berhasil mengidentifikasi beberapa kerentanan penting seperti *XSS, SQL Injection,* dan *clickjacking*, serta membuktikan bahwa beberapa serangan dapat dijalankan dengan dampak yang nyata. Sementara itu, uji *DDoS* tidak berhasil, yang menunjukkan ketahanan server terhadap serangan tersebut. Kontribusi utama dari penelitian ini adalah memberikan rekomendasi teknis untuk mitigasi kerentanan dan meningkatkan kesadaran akan pentingnya pengujian keamanan berbasis standar. Hasil ini diharapkan dapat menjadi acuan bagi instansi pemerintah dalam menjaga kualitas dan keamanan layanan berbasis web.**

Kata kunci*— *Cybersecurity, Penetration Testing Execution Standard (PTES), Website, Sistem Keamanan, Pemerintah Daerah

I. PENDAHULUAN

Kemajuan teknologi terus terjadi di seluruh dunia yang sejalan dengan perkembangan zaman. Seiring dengan perkembangan internet saat ini, penggunanya terus meningkat. Dengan demikian, kemungkinan ancaman siber pun akan meningkat[1]. *Cyber threats* merupakan tindakan ilegal yang mencuri dan merusak data yang berharga dari aplikasi atau *website*, dan dapat mengancam keamanan jaringan, *database* serta sistem komputer. Berdasarkan *Table* 1.1 di Indonesia, pada Januari hingga Juli 2019, terdapat 33.451.230 ancaman siber yang terjadi. Kemudian, pada Januari hingga Juli 2020, ancaman siber meningkat sebesar 9.35% menjadi 126.882.845. Selanjutnya, pada tahun 2021 ancaman siber meningkat hingga 1.637.973.022, menurut data yang dikumpulkan oleh Badan Siber Nasional[2]. Kemudian kejahatan siber mengalami penurunan pada tahun 2022, jumlah ancaman siber turun dari tahun sebelumnya sebanyak 976.429.996[3]. Pada tahun berikutnya yaitu 2023 ancaman siber di Indonesia mengalami penurunan sebanyak 403.990.813. Hal ini membuktikan bahwa pada tahun 2023 Indonesia telah mengalami penurunan ancaman siber yang cukup baik dari tahun sebelumnya[4].

Sebuah *website* harus dilindungi agar data tetap aman karena berfungsi sebagai media penyampaian informasi. Seorang *hacker* dapat membuat program untuk kepentingan dirinya sendiri dan merusak data apabila pemilik situs web mengabaikan keamanan. *Virus*, peretasan rekening bank, pencurian data pengguna dan pencurian *password e-mail/web server* adalah beberapa contoh kasus yang dilakukan oleh *hacker*[5]. Pengujian *penetration testing* dilakukan terhadap *website* Pemerintah Daerah ABC karena sebelumnya pernah mengalami serangan yang menyebabkan perubahan pada tampilan *font* di halaman. Dari hasil analisis, serangan ini kemungkinan besar merupakan jenis serangan *XSS (Cross-Site Scripting),* yaitu ketika kode berbahaya disisipkan oleh penyerang ke dalam *website* dan secara otomatis dijalankan oleh *browser* pengguna saat halaman diakses.

II. KAJIAN TEORI.

A. *Website*

Website dapat diartikan sebagai sarana yang menyajikan berbagai informasi secara terstruktur yang biasa diakses melalui internet. Informasi ini dapat berupa teks, gambar, film, dan suara, dan setiap orang yang memiliki koneksi internet yang memungkinkan akses tanpa batasan waktu dan tempat. Kemudian, secara teknis, *website* merupakan himpunan halaman web yang saling terhubung dan berada dalam satu domain atau subdomain tertentu[6].

B. Keamanan Jaringan

Keamanan jaringan merupakan upaya untuk mencegah serta mendeteksi akses ilegal terhadap jaringan. Tindakan pencegahan ini bertujuan agar pihak yang tidak berwenang tidak dapat mengakses komponen apa pun dalam sistem jaringan. Keamanan jaringan melindungi aktivitas jaringan komputer dari ancaman fisik dan logis[7]. Meskipun beberapa organisasi mungkin lebih memperhatikan masalah keamanan daripada masalah keamanan, keamanan jaringan harus menjadi perhatian utama. Perhatian terhadap keamanan jaringan perlu ditingkatkan guna melindungi sistem dari serangan yang semakin kompleks dan bervariasi, karena apabila sistem berhasil disusupi dan mengalami kerusakan, upaya pemulihannya akan menimbulkan masalah dan kerugian yang lebih besar[8].

C. *Penetration Testing*

Penetration testing merupakan pendekatan yang digunakan dalam menilai kerentanan dan ketahanan sistem informasi terhadap serangan untuk kelemahan teknis, kesalahan konfigurasi, dan kecacatan perangkat lunak maupun perangkat keras. *Penetration testing* membantu mengidentifikasi serta menangani celah keamanan dalam infrastruktur jaringan, sekaligus memperlihatkan tingkat kerentanan jaringan terhadap serangan berbahaya. Implementasi penggunaan *penetration testing* sebaiknya didukung oleh penerapan *framework* yang relevan. Terdapat banyak standar dan metode *penetration testing* yang dapat dijadikan acuan dalam menilai tingkat keamanan sistem. Terdapat sejumlah metode atau *framework* yang membahas tahapan dalam *penetration testing*, antara lain *Penetration Testing Execution Standard (PTES), Open Web Application Security Project (OWASP), Information System Security Assessment Framework (ISSAF), Open Source Security Testing Methodology Manual (OSSTMM),* serta standar dari *National Institute of Standards and Technology (NIST)*[9]

D. *Penetration Testing Execution Standard*

Penetration Testing Execution Standard (PTES) yang dikembangkan pada tahun 2010, dapat dimanfaatkan dalam proses audit dan analisis keamanan sistem web. Standar ini umumnya diterapkan pada prosedur keamanan, perangkat lunak, kontrol internal, maupun implementasi infrastruktur yang bertujuan untuk meningkatkan integritas serta menjaga kerahasiaan data[10]. PTES dibagi menjadi tujuh fase utama dalam implementasinya, antara lain:

1. *Pre-engagement interactions*

   Pre-engagement interactions memiliki tujuan untuk merinci instrumen dan metode yang diperlukan guna mempersiapkan langkah awal dalam kegiatan *penetration testing*[11].

2. *Intelligence Gathering*

   Data intelijen dikumpulkan di fase *Intelligence Gathering* untuk membantu mengatur setiap langkah pemeriksaan. Pada tingkat yang lebih luas, informasi tentang pekerja, fasilitas, produksi, dan perencanaan termasuk dalam data intelijensi ini[11].

3. *Threat Modeling*

   Untuk memberikan tindakan eksekusi uji penetrasi yang tepat, fase yang disebut "*Threat Modeling*" menetapkan pendekatan terhadap model ancaman. Model yang konsisten menggambarkan ancaman, kemampuan, kualitas untuk pengujian pada setiap organisasi, dan kemampuan untuk menerapkannya pada pengujian yang akan datang[11]

4. *Vulnerability Analysis*

   Vulnerability Analysis atau analisis kerentanan, berfungsi untuk mendeteksi dan menilai potensi risiko keamanan akibat adanya celah yang ditemukan dalam sistem. Proses analisis ini terdiri dari dua tahap utama, yakni identifikasi dan validasi[11].

5. *Exploitation*

   Dalam fase *eksploitation*, tujuan utama adalah mendapatkan akses ke sumber atau sistem dengan masuk secara paksa pada keamanan yang ada. Apabila tahapan menganalisis kerentanan dilaksanakan dengan kurang baik pada tahap sebelumnya[11].

6. *Post Exploitation*

   Tahap *Post Exploitation* dilakukan untuk mengukur nilai dari sistem yang telah berhasil dieksploitasi dan memastikan bahwa kontrol sistem tetap berfungsi sebagaimana mestinya[11].

7. *Reporting*

   Fase *Reporting* adalah akhir dari proses melaporkan dan mencatat momen krusial yang berlangsung selama pelaksanaan penetration testing. Tujuannya untuk memberikan informasi kepada organisasi mengenai tindakan yang telah dilakukan, potensi risiko yang dihadapi, serta langkah-langkah perbaikan yang perlu diambil untuk meningkatkan sistem organisasi[11].

E. *OWASP ZAP*

Tools OWASP ZAP merupakan sebuah fitur untuk *penetration testing* pada suatu *website*. *OWASP ZAP* adalah aplikasi yang dapat menemukan celah keamanan pada sistem web. *OWASP* terkenal karena merilis sepuluh risiko keamanan dan kerentanan teratas untuk aplikasi web, aplikasi seluler, *API*, dan lainnya, yang direvisi setiap empat tahun untuk menunjukkan ancaman dan ancaman terbaru yang memengaruhi organisasi di seluruh dunia[12].

F. *NMAP*

Network Mapper (NMAP) adalah alat berbasis *open source* yang digunakan untuk memetakan jaringan dan menilai aspek keamanannya. *Nmap* tidak hanya dapat digunakan untuk memindai jaringan dalam skala besar, namun juga mampu melakukan pemindaian terhadap satu *host* secara individual. Melalui pemanfaatan paket *IP*, *Nmap* mampu mendeteksi *host* aktif dalam jaringan, *port* yang terbuka, jenis sistem operasi yang berjalan, serta tipe *firewall* yang diterapkan[13]

G. *XRAY*

Berdasarkan *website* Pemerintah Daerah ABC dengan perkembangan penelitian saya, *tools* pemindaian kerentanan keamanan jaringan *Xray* milik *Chaitin Technology* digunakan untuk mendeteksi dan mengevaluasi keamanan aplikasi web. Memiliki fitur seperti deteksi dan pembacaan cepat, rentang pemeriksaan yang luas, kualitas kode yang tinggi, dan kustomisasi tingkat lanjut, alat ini juga aman dan tidak berbahaya.

H. *SQL Injection*

SQL injection merupakan bentuk serangan yang dilakukan dengan menyisipkan perintah *SQL* ke dalam kolom input yang disediakan untuk pengguna, sehingga input tersebut dianggap sebagai bagian dari perintah *SQL* oleh sistem. Hal ini memungkinkan penyerang menjalankan perintah langsung ke *database* dan memanipulasi data. Serangan ini dapat dimanfaatkan karena kurangnya validasi pada *input* dan memberi peluang bagi penyerang untuk mengakses, memodifikasi, atau bahkan menghapus data yang terdapat dalam *database*[14].

I. *XSS (Cross Site Scripting)*

Cross Site Scripting (XSS) merupakan bentuk serangan injeksi kode yang berlangsung di sisi pengguna, dengan cara penyerang menanamkan skrip berbahaya ke aplikasi web yang memiliki masalah dengan memeriksa *input*. Akibatnya, skrip berbahaya tersebut bisa dijalankan di *browser* pengguna tanpa sepengetahuan mereka[14]

J. *DDoS*

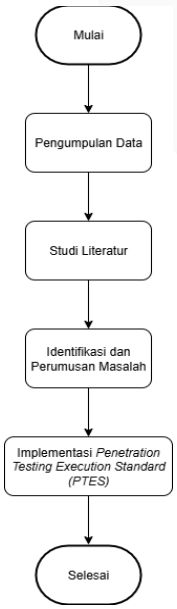
Distributed Denial of Service (DDoS) adalah bentuk serangan siber yang ditujukan untuk mengganggu akses pengguna terhadap suatu layanan atau jaringan. Serangan ini dilakukan dengan membanjiri sistem target menggunakan lalu lintas data berlebih dari banyak sumber secara bersamaan. Akibatnya, server menjadi lambat, menolak akses pengguna, bahkan dapat mengalami gangguan total atau *crash*[15].

III. METODE

Penelitian ini melihat keamanan sistem di situs web Pemerintah Daerah ABC melalui berbagai langkah, seperti pengumpulan data, merumuskan masalah, eksploitasi, dan pelaporan dan saran untuk perbaikan. Metode *Penetration Testing Execution Standard (PTES)* digunakan dalam proses ini.

A. Diagram Alir Penelitian

Untuk menguji keamanan sistem di *website* Pemerintah Daerah ABC, diagram alir berikut menggambarkan tahapan penelitian sistematis.



GAMBAR 1
Diagram Alir Penelitian

Proses penelitian dilakukan melalui langkah-langkah berikut:

1. Pengumpulan Data

   Pada tahap ini, informasi yang diperlukan dikumpulkan oleh peneliti, seperti informasi tentang serangan *cyber* di berbagai *website open access* di Indonesia. Hal ini dilakukan dengan mendapatkan informasi dari *website* resmi. Pada tahap ini juga peneliti mulai menentukan metode penelitian yang digunakan.

2. Studi Literatur

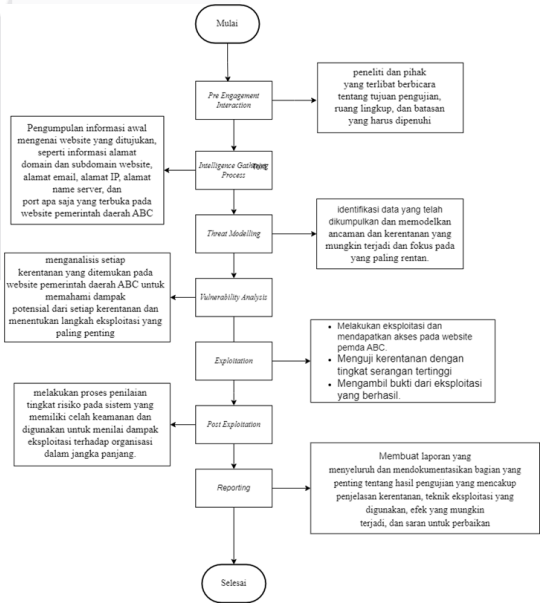
   Studi literature berguna sebagai dasar pembelajaran untuk penelitian ini. Studi Literatur berkaitan dengan tema yang sama dari penelitian sebelumnya, jurnal, atau informasi dari internet yang terkait.

3. Identifikasi dan Perumusan Masalah

   Pada tahapan ini, peneliti menentukan masalah yang terjadi dilapangan. Identifikasi permasalahan dalam penelitian ini dilakukan oleh peneliti berdasarkan hasil observasi dan wawancara yang telah dilaksanakan

4. Impelementasi *Penetration Testing Execution Standard (PTES)*

   Penetration testing merupakan salah satu metode untuk melakukan uji keamanan *website*, yang biasanya memiliki tujuan tertentu, seperti kemungkinan target diambil alih. Penelitian ini akan dilakukan uji penetrasi dan terdapat *framework* yang akan digunakan untuk pengujian yakni *Penetration Testing Execution Standard (PTES)*. Uji penetrasi yang dilakukan sesuai dengan metode pelaksanaan uji PTES yang terdiri dari 7 tahapan di antaranya *Pre-engagement Interactions, Intelligence Gathering, Threat Modeling, Vulnerability Analysis, Exploitation, Post-Exploitation, and Reporting.* Tahapan - tahapan yang dilakukan telah disusun dalam bentuk diagram tahapan *Penetration Testing Execution Standard (PTES)* berikut;



GAMBAR 2
Implementasi PTES

B. Alasan Pemilihan Metode

Penelitian ini menggunakan metode *PTES*, yaitu metode untuk memeriksa dan menganalisis keamanan sistem. Metode *PTES* memiliki tahapan yang jelas dan terperinci untuk menemukan kerentanan dan menganalisis ancaman yang mungkin terjadi. Proses evaluasinya dibuat agar mudah dimengerti oleh pengguna, baik yang berpengalaman maupun yang baru dalam pengujian penetrasi.

IV. HASIL DAN PEMBAHASAN

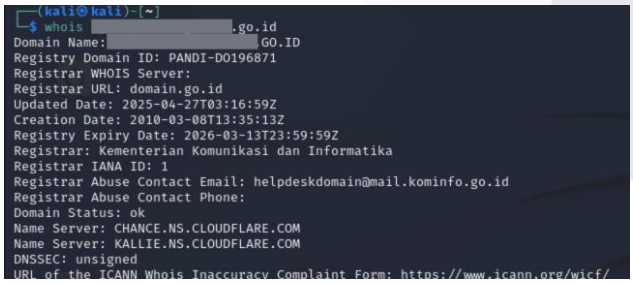
Website pemerintah daerah ABC diambil sebagai sampel untuk diuji melalui pendekatan yang mengacu pada standar *Penetration Testing Execution Standard (PTES).*

A. *Pre-engagement Interaction*

Pada tahap awal ini, seluruh proses persiapan untuk melakukan pengujian keamanan dibahas secara menyeluruh. Hal ini meliputi tujuan pengujian, ruang lingkup, batasan, persyaratan, dan perlengkapan yang diperlukan. Tahap ini mencakup kesepakatan mengenai jenis pengujian yang digunakan, di mana peneliti tidak diberikan akses internal atau informasi detail mengenai struktur sistem. Selain itu, peneliti dan pengelola *website* sepakat bahwa pengujian hanya difokuskan pada *website* utama dengan domain publik yang dapat diakses secara langsung oleh pengunjung, tanpa melibatkan server internal atau data sensitif milik instansi. Selanjutnya, peneliti melakukan persiapan terhadap peralatan dan material yang diperlukan untuk menganalisis keamanan *information system* pada situs resmi milik Pemerintah Daerah ABC.

B. *Intelligence Gathering*

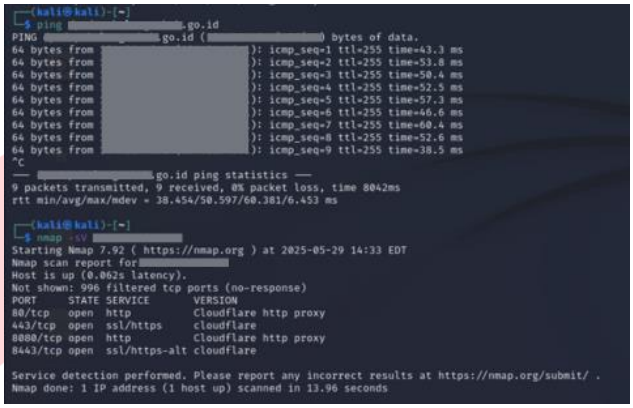
Pada tahap ini, dengan menggunakan metode *PTES* untuk menganalisis situs web Pemerintah Daerah ABC, peneliti mengumpulkan informasi yang diperlukan seperti domain dan subdomain situs dari web tersebut, alamat *IP*, *email*, dan *DNS*. Untuk melakukan analisis keamanan, peneliti menggunakan *whois* pada *tools* terminal yang ada pada *kali linux* untuk mendapatkan informasi tentang nama domain, alamat *email*, dan *DNS*. Gambar 3 menunjukkan hasil analisis *whois* dari *website* Pemerintah Daerah ABC. Ini menunjukkan hasil analisis domain dari *website* tersebut menggunakan *cloudflare.com* sebagai layanan *DNS* dan keamanan *website*.



GAMBAR 3
Hasil Whois

Website Pemerintah Daerah ABC terakhir di *update* pada tanggal 27 April 2025 pukul 03:16:59. Selain itu, lisensi domain akan berlaku hingga 13 Maret 2026. Dengan status domain OK.

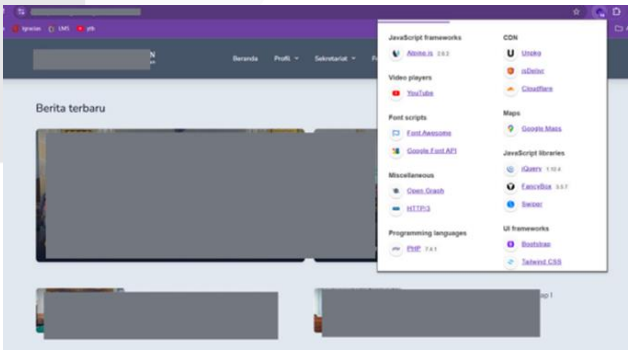
Kemudian dilanjut mengumpulkan informasi penting mengenai semua *port* pada *host* target yang terbuka atau aktif menggunakan *tools Nmap* dan melihat informasi versi layanan yang berjalan di *port* yang terbuka pada *website* Pemerintah Daerah ABC. Pada Gambar 4 terlihat bahwa ada 4 *port* yang terbuka, dimana kerentanan ini berpotensi mempengaruhi banyak server yang menggunakan skrip CGI dan memungkinkan penyerang untuk menjalankan perintah *shell* pada server yang rentan. Hal ini memberikan celah untuk akses tidak sah, pencurian data, dan manipulasi server. Seperti *port* 80 (*HTTP*) yang apabila data tidak terenkripsi maka akan berisiko tinggi mengalami serangan *man-in-the-middle (MITM), port scanning,* atau akses tidak sah ke layanan tertentu.



GAMBAR 4
Detail *port* yang terbuka dan *test ping*

Selanjutnya, menggunakan *Wappalyzer* yang dapat mengidentifikasi *stack* teknologi seperti *framework* dan bahasa pemrograman yang digunakan.

Berdasarkan Gambar 5 terlihat bahwa *website* Pemerintah Daerah ABC untuk *frameworks backend* menggunakan *Alpine.js* versi 2.8.2 dan menggunakan *Bootstrap* untuk *framework frontend*. Bahasa pemrograman yang digunakan untuk membangun bagian *backend* situs adalah *PHP* versi 7.4.1. Hal tersebut menunjukan bahwa Pemerintah Daerah ABC masih menggunakan versi lama dan termasuk yang sudah mendekati akhir masa dukungannya. *Website* Pemerintah Daerah ABC juga menggunakan *JavaScript Libraries jQuery* versi 1.12.4 yang dimana versi ini merupakan versi lama dan sangat rentan terhadap serangan *XSS (Cross-Site Scripting).*

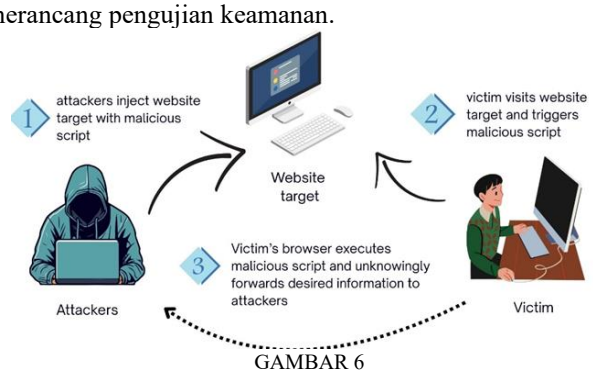


GAMBAR 5
Hasil Wappalyzer

C. *Threat Modeling*

Threat modeling berguna untuk membantu meningkatkan keamanan sistem dengan cara menemukan kelemahan, menetapkan tujuan, dan merancang cara untuk mencegah atau mengurangi dampak serangan siber. Dalam hal ini,

threat modeling digunakan sebagai langkah awal dalam merancang pengujian keamanan.



GAMBAR 6
Threat modeling Cross-Site Scripting (XSS)

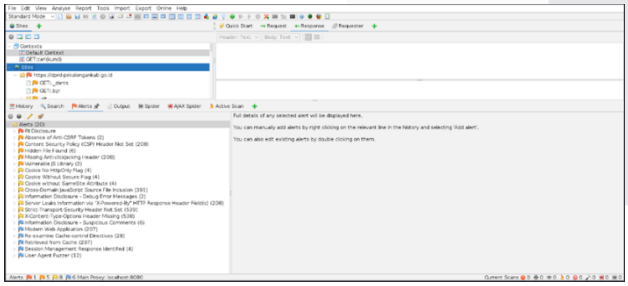
Berdasarkan Gambar 6 di atas, menggambarkan model ancaman dalam menghadapi serangan *Cross-Site Scripting (XSS)*, yang terdiri atas tiga fase utama. Fase awal dimulai ketika penyerang menyisipkan skrip berbahaya ke dalam *website* target melalui celah input yang tidak memiliki validasi atau sanitasi yang memadai, seperti pada kolom komentar, form pencarian, atau parameter *URL*. Selanjutnya, pada tahap kedua, pengguna yang mengakses halaman tersebut secara tidak sadar memicu eksekusi skrip berbahaya di dalam *browser*. Pada tahap ketiga, skrip tersebut berjalan di sisi klien dan bisa digunakan untuk mencuri informasi sensitif, seperti data pribadi, *cookie* sesi, atau bahkan mengalihkan target ke situs yang berbahaya. Informasi yang terkumpul kemudian dikirimkan secara diam-diam ke server milik penyerang.

D. *Vulnerability Analysis*

Tahapan ini bertujuan untuk menganalisis kelemahan yang ada pada *website* dengan memanfaatkan hasil dari tahap *Intelligence Gathering* dan *Threat modeling* melalui *scanning* dengan beberapa *tools* pemindai keamanan. Peneliti akan menggunakan *tools ZAP* dan *Xray* untuk melakukan *scanning*.

1. Analisis dengan *tools OWASP ZAP*

Tools ZAP merupakan *tools* pertama yang digunakan oleh peneliti untuk melakukan pemindaian kerentanan *website*. Hasil pemindaian terdiri dari 21 *alert*, 1 dengan level *high*, 5 *medium*, 9 *low*, dan 6 level *informational*.



GAMBAR 7
Hasil *Scanning* dengan *ZAP*

Dengan menggunakan *tools ZAP*, hasil analisis menghasilkan kerentanan yang tercantum pada Tabel 1.

TABEL 1
Daftar kerentanan Hasil *Scanning* dengan *ZAP*

No	Jenis Kerentanan	Level Resiko
1	*PII Disclosure*	*High*
2	*Absence of Anti-CSRF Tokens*	*Medium*
3	*Content Security Policy (CSP) Header Not Set*	*Medium*
4	*Hidden File Found*	*Medium*
5	*Missing Anti-clickjacking Header*	*Medium*
6	*Vulnerable JS Library*	*Medium*
7	*Cookie No HttpOnly Flag*	*Low*
8	*Cookie Without Secure Flag*	*Low*
9	*Cookie without SameSite Attribute*	*Low*
10	*Cross-Domain JavaScript Source File Inclusion*	*Low*
11	*Information Disclosure - Debug Error Messages*	*Low*
12	*Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*	*Low*
13	*Strict-Transport-Security Header Not Set*	*Low*
14	*X-Content-Type-Options Header Missing*	*Low*
15	*Information Disclosure - Suspicious Comments*	*Informational*
16	*Modern Web Application*	*Informational*
17	*Re-examine Cache-control Directives*	*Informational*
18	*Retrieved from Cache*	*Informational*
19	*Session Management Response Identified*	*Informational*
20	*User Agent Fuzzer*	*Informational*

2. Analisis Menggunakan *Xray*

Tools Xray merupakan *tools* kedua yang digunakan oleh peneliti untuk melakukan pemindaian kerentanan *SQL Injection* dan *XSS* terhadap situs web yang merupakan fokus pada penelitian. Gambar 8 menampilkan hasil pemindaian kerentanan *SQL Injection* yang dilakukan menggunakan *Xray*, menunjukkan bahwa situs web memiliki kerentanan *SQL Injection*. Sehingga memungkinkan mengeksploitasi *database* yang dapat mengakibatkan pencurian data sensitif dan penting pada *website*.

GAMBAR 8
Hasil *Scanning SQL Injection* dengan *Xray*

Kemudian terdapat temuan dari pemindaian terhadap kerentanan *Cross-Site Scripting (XSS)* menggunakan *Xray* yang terdapat pada gambar 9 yang memperlihatkan bahwa *website* berisiko terkena serangan *Cross-Site Scripting (XSS)*. Hal ini memungkinkan untuk memasukkan *script* berbahaya ke halaman web untuk mengeksploitasi kerentanan tersebut.

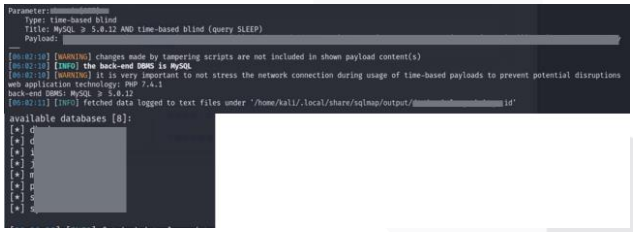
GAMBAR 9
Hasil *Scanning XSS* dengan *Xray*

E. *Exploitation*

Peneliti menyelidiki masalah yang ditemukan selama tahap analisis kerentanan pada tahap eksploitasi ini. Apakah kerentanan tersebut dapat dieksploitasi atau tidak. Beberapa kerentanan akan dieksploitasi, di antaranya sebagai berikut:

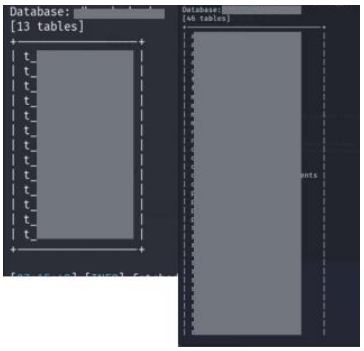
1. SQL Injection

SQL Injection adalah bentuk serangan yang terjadi saat penyerang mampu memasukkan perintah *SQL* berbahaya ke dalam *input* aplikasi, seperti *form login* atau *URL*, sehingga *query SQL* yang dijalankan oleh aplikasi dapat diubah secara tidak sah. Serangan ini dilakukan berdasarkan hasil yang didapatkan dari identifikasi potensi ancaman menggunakan *tools Xray* yang dimana telah ditemukan celah keamanan *SQL Injection* dan memanfaatkan kerentanan *PII disclosure* yang memiliki level kerentanan *high*. *Tools sqlmap* dgunakan untuk mengeksploitasi kerentanan yang ditemukan dan menguji keamanan *website*. Hasil penyerangan pada kerentanan *SQL Injection* terdapat pada Gambar 10.



GAMBAR 10
Hasil pengujian kerentanan *SQL Injection*

Gambar 11 menunjukkan bahwa kerentanan *SQL Injection* telah berhasil diuji coba. Pada pengujian ini, peneliti berhasil menemukan *database* yang berada di *website* Pemerintah Daerah ABC. Hasil pengujian yang berhasil pada kerentanan *SQL Injection* menunjukkan bahwa situs web tersebut memiliki kerentanan yang harus diperbaiki segera. Tabel yang terdapat dalam *database* di *website* Pemerintah Daerah ABC telah diperoleh oleh peneliti, seperti yang ditunjukkan pada Gambar 11.

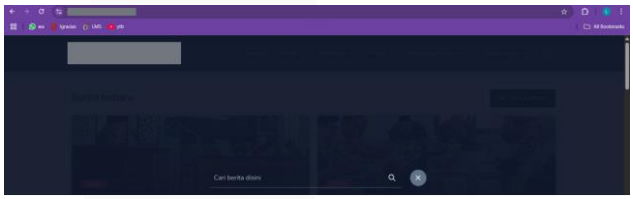


GAMBAR 11
Hasil *table* dalam *database*

Peneliti berhasil mengakses tabel-tabel yang terdapat di dalam *database* yang sebelumnya telah diperoleh. Celah keamanan ini menimbulkan risiko tinggi terhadap data sensitif yang disimpan, karena dapat dieksploitasi oleh pihak yang tidak memiliki otorisasi untuk melakukan eksploitasi. Oleh sebab itu, celah *SQL Injection* perlu segera dievaluasi dan diperbaiki untuk mencegah terjadinya kebocoran atau pencurian data oleh pihak yang tidak memiliki otoritas.

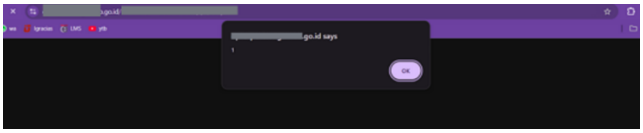
2. XSS (Cross Site Scripting)

Serangan Cross-Site Scripting (XSS) terjadi ketika kode atau script dimasukkan ke dalam form atau situs web Pemerintah Daerah ABC yang terdapat pada Gambar 12. Pelaksanaan serangan ini didasarkan pada hasil yang didapatkan dari identifikasi potensi ancaman menggunakan tools Xray yang dimana telah ditemukan celah keamanan Cross-Site Scripting (XSS) dan pada percobaan ini memanfaatkan kerentanan Content Security Policy (CSP) dan Vulnerable JS Library yang memiliki level medium.



GAMBAR 12
Form pencarian pada *website*

Kemudian peneliti menyisipkan *script payload* pada web Pemerintah Daerah ABC yaitu dengan mengetikkan *payload "<ScRiPt>alert(1)</sCrIpT>"* pada bagian *form* pencarian berita yang ada pada *website* Pemerintah Daerah ABC seperti pada gambar 13. Peneliti dapat meretas *XSS* ini dengan menggunakan *script alert*. Berikut pada Gambar 13 merupakan hasil penyerangan dengan menyisipkan *script* untuk melakukan serangan *XSS*.



GAMBAR 13
Hasil pengujian kerentanan *XSS*

Hasil dari analisis serangan *XSS* yang di exploitasi oleh peneliti menunjukkan bahwa kerentanan *Cross-Site Scripting (XSS)* yang berhasil diuji. Dalam pengujian ini, peneliti menambahkan *script alert* ke *form* pencarian di situs web Pemerintah Daerah ABC, yang menghasilkan *pop-up* seperti yang digambarkan di atas. Pengujian kerentanan *Cross-Site Scripting (XSS)* mengindikasikan bahwa situs

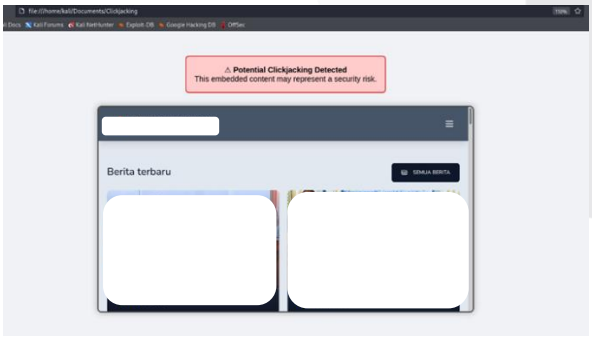
web Pemerintah Daerah ABC memiliki celah keamanan yang perlu segera ditangani guna mencegah potensi eksploitasi oleh pihak yang tidak berwenang.

3. Clickjacking

Pada kerentanan ini mencoba memanfaatkan kerentanan *Missing Anti-clickjacking Header* dengan level resiko *Medium*. Pada *website* Pemerintah Daerah ABC tidak menambahkan *X-Frame Option* sehingga *website* Pemerintah Daerah ABC dapat disematkan pada *website* lain. Percobaan serangan dilakukan peneliti dengan menyusun sebuah skrip yang dirancang untuk melakukan serangan *clickjacking* pada *website* yang teridentifikasi memiliki celah keamanan. Hal ini terdapat pada Gambar 14.

GAMBAR 14
Script Pengujian *Clickjacking*

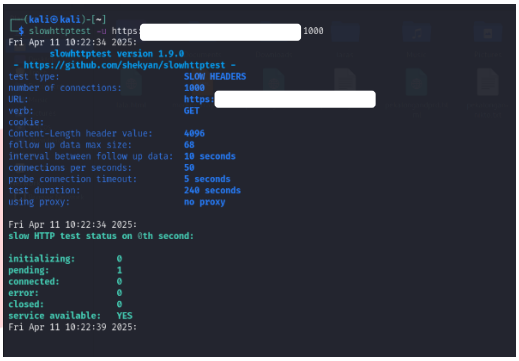
Gambar 15 menampilkan hasil dari proses pengujian kerentanan *clickjacking* pada *website* Pemerintah Daerah ABC. Pengujian dilakukan dengan memuat *website* ke dalam elemen <*iframe*> pada halaman yang dibuat peneliti. Hasilnya, tampak peringatan keamanan berupa pesan "*Potential Clickjacking Detected, This embedded content may represent a security risk*" yang menandakan bahwa situs dapat ditampilkan dalam *frame* tanpa perlindungan. Hal ini menunjukkan bahwa *website* belum menerapkan *header* keamanan *X-Frame-Options* atau *CSP* yang berfungsi guna menghindari tampilan halaman dalam *frame* dari situs lain. Dampaknya, *website* pemerintah ABC berisiko dimanipulasi secara visual oleh penyerang untuk membuat pengguna mengklik elemen yang tidak mereka sadari, seperti tombol *login* atau konfirmasi transaksi.



GAMBAR 15
Hasil Pengujian *Clickjacking*

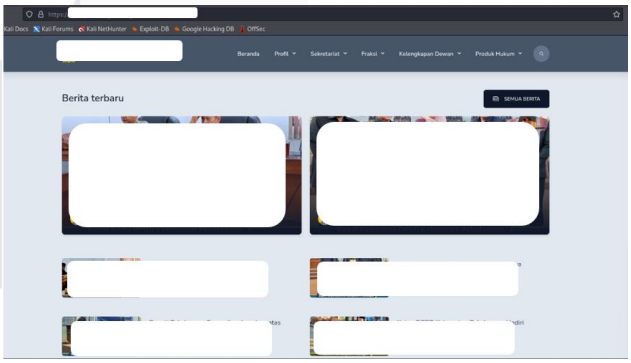
4. DdoS (Distributed Denial of Service)

Pada eksploitasi *DDoS* ini peneliti memanfaatkan hasil dari analisis *port* yang terbuka. Pada *website* pemerintah daerah ABC terdapat *port http* yang terbuka yang dimana hal ini dapat dimanfaatkan penyerang untuk melakukan serangan *DDoS*. Peneliti melakukan pengujian serangan *DDoS* jenis *Slow HTTP Attack* menggunakan *tools slowhttptest* terhadap *website* Pemerintah Daerah ABC. Pengujian ini dilakukan dengan metode *Slow Headers*, yaitu dengan mengirimkan sejumlah besar koneksi *HTTP* sebanyak 1000 koneksi yang masing-masing mengirimkan data secara perlahan. Hal ini bertujuan untuk mempertahankan koneksi tetap terbuka dan membebani server. Hasil pengujian yang terdapat pada Gambar 16 menunjukkan bahwa saat pengujian dimulai, status koneksi menunjukkan *connected*: 1, *pending*: 1, dan tidak ada *error* atau koneksi tertutup, serta status layanan tetap *service available: YES*



GAMBAR 16
Pengujian *DDoS* Menggunakan *slowhttptest*

Setelah dilakukan serangan *DDoS* pada *website*, terlihat bahwa server masih bisa menerima dan merespons permintaan dengan baik. Status "*service available: YES*" menunjukkan bahwa *website* masih dapat diakses meskipun mulai menerima banyak koneksi lambat secara bersamaan. Hal ini menunjukkan bahwa percobaan serangan *DDoS* ini belum berhasil dikarenakan server target kemungkinan telah dikonfigurasi dengan baik, seperti adanya pengaturan *timeout* koneksi dan pembatasan jumlah koneksi dari satu *IP,* sehingga dapat memutus koneksi yang mencurigakan secara otomatis serta jumlah koneksi aktif yang berhasil tersambung ke server dalam waktu pengujian mungkin belum cukup untuk membebani sistem. Hal ini terdapat pada Gambar 17.



GAMBAR 17
Respon *Website* Setelah diuji *DDoS*

F. *Post Exploitation*

Setelah mengeksploitasi kerentanan target, peneliti melanjutkan ke tahap *Post Exploitation*. Pada tahap ini, setelah pengujian yang dilakukan sebelumnya, peneliti melakukan penilaian tingkat risiko sistem dengan celah keamanan. Dengan demikian, peneliti membuat tabel 2 untuk menilai ancaman yang telah terdeteksi pada langkah sebelumnya untuk menentukan tingkat risikonya.

TABEL 2
Penilaian Tingkat Risiko

No	Jenis Serangan	Level Resiko	Tools
1	*SQL Injection*	*High*	*Xray, SQL Map*
2	*Cross-Site Scripting (XSS)*	*High*	*Xray, Script alert*
3	**C**lickjacking	*Medium*	*Script*
4	*DDoS*	*Low*	*slowhttptest*

Hasil pengujian di atas menunjukkan bahwa celah keamanan memiliki resiko yang tinggi, medium dan rendah. Kerentanan *SQL Injection* dan *XSS* adalah masalah keamanan situs web Pemerintah Daerah ABC yang memiliki tingkat resiko tinggi. Tingkat risiko tinggi ini menunjukkan bahwa sistem target memiliki celah keamanan yang signifikan yang dapat dimanfaatkan oleh penyerang. Kerentanan *SQL Injection* memungkinkan penyerang mengeksploitasi *database*, mengambil data sensitif seperti data pengguna dan memungkinkan manipulasi data. Kemudian kerentanan *Cross-Site Scripting (XSS)* memungkinkan pelaku menyisipkan kode berbahaya ke dalam *browser* pengguna tanpa diketahui pengguna, yang dapat menyebabkan peretas mencuri data dan melakukan sesuatu atas nama korban. Pada serangan *clickjacking* memiliki tingkat resiko *medium* karena pada *website* pemerintah daerah tidak memiliki fitur *login*, pengaturan akun, maupun transaksi. Walaupun *website* informasi tidak memiliki fitur *login* atau transaksi, serangan *clickjacking* tetap membawa resiko, terutama dalam hal manipulasi interaksi pengguna dan kepercayaan publik. Sehingga hal ini dapat menurunkan kredibilitas dan kepercayaan pengguna terhadap situs tersebut.

Serangan *DDoS* yang telah di ujikan tidak berhasil melumpuhkan atau mengganggu layanan *website* sistem informasi, maka resiko keamanan terhadap jenis serangan ini dapat dikategorikan sebagai rendah. Hal ini membuktikan bahwa sistem target memiliki kemampuan yang cukup baik dalam menangani beban koneksi yang tinggi. Namun, keberhasilan server dalam menahan serangan tidak sepenuhnya menghilangkan potensi ancaman di masa depan, terutama jika skala serangan meningkat atau dilakukan secara terdistribusi dari banyak sumber, jadi meskipun tingkat risikonya saat ini tergolong rendah, sistem tetap perlu dipantau dan diperkuat dengan mekanisme proteksi tambahan. Dari ke empat temuan ini menunjukkan bahwa sistem sangat rentan dan memerlukan tindakan mitigasi segera untuk mencegah eksploitasi dan pelanggaran data yang lebih luas.

G. Reporting

Terdapat tahapan terakhir yang dilaksanakan dalam penelitian ini yaitu mendokumentasikan aspek penting yang terjadi selama pentest dan membuat reporting. Peneliti menyajikan hasil penelitian memakai PTES sebagai metode testing untuk situs web Pemerintah Daerah ABC serta menyusun saran perbaikan terhadap kerentanan yang berhasil diidentifikasi. Pengujian ini disajikan pada Tabel 3.

TABEL 3
Hasil Pengujian *Penetration Testing*

No	Jenis Serangan	*Tools*	Status
1	*SQL Injection*	*SQL Map*	Berhasil
2	*Cross-Site Scripting (XSS)*	*Script alert*	Berhasil
3	***C**lickjacking*	*Script*	Berhasil
4	*DDoS*	*slowhttptest*	Gagal

Berdasarkan data pada tabel tersebut, dapat disimpulkan bahwa 3 serangan telah berhasil dieksekusi pada *website* Pemerintah Daerah ABC dan 1 serangan gagal dieksekusi. Pada serangan *SQL Injection*, peneliti berhasil mengakses *database* menggunakan *tools SQLMap*. Selanjutnya, pada serangan *XSS*, *script alert* disisipkan ke dalam kolom pencarian di situs tersebut, dan skrip tersebut berhasil dijalankan, ditandai dengan munculnya *pop-up*. Serangan *clickjacking* yang dilakukan peneliti berhasil memuat *website* ke dalam elemen <*iframe*> pada halaman yang dibuat peneliti. Kemudian pada serangan *DDoS* belum berhasil dikarenakan *website* memiliki mekanisme keamanan yang kuat, sehingga mampu mendeteksi dan menghentikan upaya akses dari penyerang saat serangan terhadap sistem dilakukan.

Dari hasil pengujian keamanan ini, saran perbaikan terhadap celah keamanan pada situs Pemerintah Daerah ABC yang terdapat pada Tabel 4.

TABEL 4
Rekomendasi Perbaikan

No	Jenis Serangan	Rekomendasi
1	*SQL Injection*	1) Membatasi hak akses *database* dengan membuat akun *MySQL* khusus untuk aplikasi dengan hak akses minimal (tidak boleh punya hak *DROP, DELETE, ALTER*, dll jika tidak perlu). 2) Menambahkan *ModSecurity* pada *Apache/Nginx* untuk mendeteksi dan memblokir serangan *SQL Injection* secara *real-time*. 3) Melakukan *update* semua versi *PHP, MySQL*, atau *framework PHP* untuk menutup celah keamanan yang diketahui. 4) Menggunakan *framework PHP* seperti *Laravel* yang mendukung

		Eloquent ORM. *ORM* secara otomatis menangani *query* parameterisasi. 5) Melakukan validasi ketat terhadap semua input pengguna. Pastikan data sesuai dengan format yang diharapkan (misalnya angka untuk ID, bukan teks), serta bersihkan karakter khusus yang berpotensi disalahgunakan. 6) Melakukan *code review, vulnerability scanning*, dan *penetration testing* secara rutin untuk mendeteksi dan menutup celah keamanan yang mungkin muncul.
2	*Cross-Site Scripting (XSS)*	1) Menggunakan *filter_input()* atau *filter_var()* di *PHP* untuk memvalidasi dan membersihkan *input*. Serta menentukan jenis *input* yang diperbolehkan (misalnya hanya huruf dan angka), terutama untuk *field* seperti nama, komentar, atau pencarian. 2) Menghindari penyimpanan data mentah langsung, yaitu dengan hanya menyimpan data yang telah disanitasi jika akan ditampilkan kembali. Apabila menyimpan data mentah, harus dipastikan dulu melakukan *escaping* sebelum ditampilkan. 3) Menggunakan *HTMLPurifier (library PHP)* untuk menyaring tag *HTML* berbahaya. 4) Menggunakan *CSP (Content Security Policy)* yang dapat mencegah *browser* mengeksekusi skrip dari sumber yang tidak sah 5) Melakukan *penetration testing* menggunakan *tools* seperti *OWASP ZAP, Xray,* atau *script XSS payload* (manual) untuk menguji apakah input masih bisa menyisipkan skrip 6) Melakukan *update Framework* dan *Library* secara berkala dan jangan gunakan *library JavaScript* atau komponen pihak ketiga yang tidak terpercaya.
3	*Clickjacking*	*Header X-Frame-Options* segera ditambahkan Menggunakan *CSP* Menggunakan *Web Application Firewall (WAF)* Melakukan peninjauan berkala terhadap konfigurasi keamanan *HTTP Header* untuk memastikan tidak terjadi kelalaian ketika melakukan pembaruan sistem atau server. Melakukan pengujian ulang untuk memastikan bahwa konten tidak dapat dimuat dalam *iframe* lainnya

V. KESIMPULAN

Berdasarkan temuan dari proses pengujian dan analisis yang dilakukan selama penelitian pada *website* Pemerintah Daerah ABC dapat disimpulkan bahwa dengan menggunakan *tools Nmap* telah ditemukan *port* terbuka seperti 80/tcp, 443/tcp, 8080/tcp, dan 8443/tcp. Pada *stack* teknologi seperti *framework* dan bahasa pemrograman yang digunakan masih menggunakan versi lama dan termasuk yang sudah mendekati akhir masa dukungannya dan ini harus segera dilakukan *update*. Dalam hasil pemindaian kerentanan *OWASP ZAP*, peneliti menemukan celah keamanan yang menjadi kelemahan keamanan *website* pemerintah daerah ABC dengan 1 jenis risiko *high*, 5 jenis risiko *medium*, 8 jenis risiko *low*, dan 6 jenis *informational*. Hasil analisis menyeluruh terhadap seluruh tahapan pengujian dengan *Penetration Testing Execution Standard (PTES)* mampu menemukan sejumlah celah. Beberapa serangan berhasil dilakukan, seperti *SQL Injection* untuk mendapatkan data dalam *database*, *XSS* yang memvalidasi kerentanan sisi klien,

serta *Clickjacking* yang berhasil memuat *website* ke dalam elemen <*iframe*> pada halaman yang dibuat peneliti.

REFERENSI

[1] M. N. Fikri, B. Parga Zen, R. Adhitama, and E. A. Firdaus, "Analisis Keamanan Sistem Informasi Website SMA Negeri 1 Sokaraja Menggunakan Metode Penetration Testing Execution Standard (PTES)," *Jurnal Informatika*, vol. 2, no. 2, 2023.

[2] S. Parulian, D. A. Pratiwi, and M. Cahya Yustina, "Ancaman dan Solusi Serangan Siber di Indonesia."

[3] "Keamanan Siber Indonesia 2022 T L P : C L E A R."

[4] "Lanskap Keamanan Siber Indonesia 2023."

[5] Y. Mulyanto, M. T. A. Zaen, Y. Yuliadi, and S. Sihab, "Analisis Keamanan Website SMA Negeri 2 Sumbawa Besar Menggunakan Metode Penetration Testing (Pentest)," *Journal of Information System Research (JOSH)*, vol. 4, no. 1, pp. 202–209, Oct. 2022, doi: 10.47065/josh.v4i1.2335.

[6] R. R. Yusuf and T. N. Suharsono, "Prosiding Seminar Sosial Politik, Bisnis, Akuntansi dan Teknik (SoBAT) ke-5 Bandung, 28 Oktober 2023 Pengujian Keamanan Dengan Metode Owasp Top 10 Pada Website Eform Helpdesk".

[7] E. J. Pranata, "Optimalisasi Keamanan Jaringan Komputer Pada Web E-Commerce Menggunakan Netfilter," 2023.

[8] M. Hasibuan and A. M. Elhanafi, "Penetration Testing Sistem Jaringan Komputer Menggunakan Kali Linux untuk Mengetahui Kerentanan Keamanan Server dengan Metode Black Box," *sudo Jurnal Teknik Informatika*, vol. 1, no. 4, pp. 171–177, Dec. 2022, doi: 10.56211/sudo.v1i4.160.

[9] G. Ary, S. Sanjaya, G. Made, A. Sasmita, D. Made, and S. Arsa, "Evaluasi Keamanan Website Lembaga X Melalui Penetration Testing Menggunakan Framework ISSAF," 2020.

[10] W. Linggih Jaelani, F. Khoirunnisa, P. Studi Teknik Informatika, S. Tinggi Teknologi Bandung, and U. Adhirajasa Reswara Sanjaya, "Penetration Testing Website Dengan Metode Black Box Testing Untuk Meningkatkan Keamanan Website Pada Instansi (Redacted)," vol. 05, 2023.

[11] S. Utoro *et al.*, "Analisis Keamanan Website E-Learning SMKN 1 Cibatu Menggunakan Metode Penetration Testing Execution Standard."

[12] T. D. H. Abdul Fattah Hasibuan, "Application Security - Analisis Keretanan Website Dengan Aplikasi OWASP ZAP," *Jurnal Ilmu Komputer dan Sistem Informasi (JIRSI)*, vol. 2, no. 2, pp. 257–270, May 2023.

[13] D. Bayu Rendro and W. Nugroho Aji, "Analisis Monitoring Sistem Keamanan Jaringan Komputer Menggunakan Software Nmap (Studi Kasus Di Smk Negeri 1 Kota Serang)," *Jurnal PROSISKO*, vol. 7, no. 2, Sep. 2020.

[14] T. Anugrah, "Penetration Testing Keamanan Website Stie Samarinda Menggunakan Teknik Sql Injection Dan Xss," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 1, Jan. 2024, doi: 10.23960/jitet.v12i1.3882.

[15] S. Joses, S. Quinevera, R. Mardianto, D. Yulvida, and A. Mazharuddin Shiddiqi, "Pendekatan Metode Ensemble Learning untuk Deteksi Serangan DDoS menggunakan Soft Voting Classifier," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 2024.