Analisis Keamanan Pada *Website* Pengadilan Negeri X Menggunakan Metode *Penetration Testing Execution Standard* (PTES)

1st Mohammad Amin Tohari
Fakultas Informatika
Universitas Telkom
Purwokerto, Indonesia
amintohari@student.telkomuniversity.ac.id

2nd Trihastuti Yuniati
Fakultas Informatika
Universitas Telkom
Purwokerto, Indonesia
trihastutiy@telkomuniversity.ac.id

***Abstrak* — Jumlah kebocoran data global pada kuartal ketiga 2022 mencapai 72,45 juta akun, dengan Indonesia berada di peringkat ketiga. Salah satu penyebab utamanya adalah lemahnya keamanan situs web, termasuk situs pemerintah. Situs web Pengadilan Negeri X menjadi objek penelitian untuk mengidentifikasi kerentanan dan meningkatkan keamanannya menggunakan metode *Penetration Testing Execution Standard* (PTES). Topik ini penting karena banyak situs pemerintah yang rawan terhadap serangan siber, seperti *Distributed Denial of Service* (DDoS) dan *Clickjacking*. Kondisi saat ini menunjukkan bahwa meskipun beberapa situs telah dilindungi *firewall*, banyak kerentanan lain seperti *header* keamanan yang tidak diatur atau tema *CMS* yang rentan terhadap eksploitasi. Solusi yang diterapkan meliputi enam tahapan *PTES*: pengumpulan data, pemodelan ancaman, analisis kerentanan, eksploitasi, post-eksploitasi, dan pelaporan. Penelitian dilakukan menggunakan alat seperti *OWASP ZAP*, *WPScan*, dan *SQL Map*. Hasil utama menunjukkan bahwa hanya serangan *Clickjacking* yang berhasil dieksploitasi, sementara empat serangan lainnya *XSS*, *SQL Injection*, *Brute Force*, dan *DDoS* belum berhasil mengeksploitasi situs web tersebut karena adanya perlindungan *firewall*. Kontribusi penelitian ini adalah mengetahui sistem keamanan dari *website* dan memberikan rekomendasi perbaikan, seperti penambahan *header X-Frame-Options* dan metode *Traffic Filtering*, untuk meningkatkan keamanan situs pemerintahan.**

Kata kunci*— Keamanan Sistem, Penetration Testing, PTES, *Website

I. PENDAHULUAN

Dalam era digital yang semakin maju, keamanan siber menjadi isu krusial, terutama dengan meningkatnya jumlah kebocoran data yang terjadi secara global. Laporan dari *Surfshark* mencatat bahwa pada kuartal ketiga tahun 2022, sebanyak 72,45 juta akun mengalami kebocoran data, dengan Indonesia menempati peringkat ketiga tertinggi, mencapai 12,74 juta akun yang diretas. *Website*, sebagai layanan informasi yang terhubung ke internet, menjadi salah satu target utama serangan siber karena masih banyaknya celah keamanan yang dapat dimanfaatkan oleh *hacker*. Laporan dari Badan Siber dan Sandi Negara (BSSN) menunjukkan bahwa pada tahun 2023, 55% dari dugaan pelanggaran database terjadi pada situs administrasi pemerintahan, menjadikannya sektor yang paling rentan terhadap serangan siber. Salah satu upaya untuk meningkatkan keamanan *website* adalah dengan melakukan pengujian penetrasi *(penetration testing)* secara berkala untuk mengidentifikasi dan menutup celah keamanan sebelum disalahgunakan oleh pihak yang tidak bertanggung jawab. Metode analisis kerentanan seperti OWASP (*Open Web Application Security Project*) dan CVSS (*Common Vulnerability Scoring System*) telah banyak digunakan untuk mengevaluasi tingkat risiko keamanan pada sistem. Namun, masih terdapat banyak *website* pemerintah yang belum menerapkan standar keamanan siber secara optimal, sehingga berpotensi mengalami peretasan yang dapat mengancam layanan publik serta kerahasiaan data. Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk menguji keamanan *website* Pengadilan Negeri X dengan melakukan analisis kerentanan dan pengujian penetrasi guna mengidentifikasi celah keamanan serta tingkat kerentanan situs, sehingga dapat memberikan solusi proaktif dalam meningkatkan keamanan sistem informasi pemerintahan.

II. KAJIAN TEORI

A. *Website*

Adalah jenis layanan informasi yang sering diakses oleh pengguna yang terhubung ke jaringan internet. Situs web harus dirancang untuk dapat menangani banyak pertanyaan pengguna agar dapat menjadi salah satu layanan informasi[1]

Salah satu jenis materi yang tersedia di internet adalah situs web. Istilah “situs web” mengacu pada kumpulan halaman web yang merupakan bagian dari sebuah domain atau subdomain di *World Wide Web* (WWW). Sebuah halaman web adalah dokumen dalam format *html* yang dapat diakses melalui *http* atau *https*. Beranda adalah URL *(Uniform Resource Locator)* yang memungkinkan seseorang mengakses halaman sebuah situs *website*. *Website* dapat diakses melalui *browser* komputer atau *smartphone*. Di antaranya adalah *mozilla, internet explorer, firefox, opera,* dan *google chrome*[2].

B. Sistem Informasi

Dalam sebuah organisasi, sebuah sistem yang mendukung transaksi harian untuk meningkatkan fungsi manajerial dalam strategi organisasi dikenal sebagai sistem informasi. Selain itu, sistem informasi dapat memberikan informasi yang diperlukan oleh pihak-pihak terkait. Sistem dalam suatu organisasi adalah sistem yang terdiri dari orang, fasilitas, teknologi, media, prosedur, dan pengendalian. Tujuan dari sistem informasi adalah untuk mengumpulkan data komunikasi yang penting, menganalisis jenis transaksi rutin, memberikan informasi kepada manajemen dan pihak lain

tentang penelitian penting yang dilakukan di dalam dan di luar organisasi, dan menyediakan data untuk pengambilan keputusan[3]

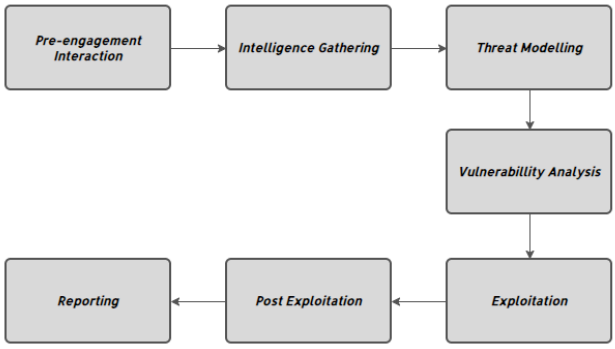
C. *Penetration Testing*

Penetration testing adalah proses yang digunakan untuk menilai keamanan sistem, memastikan bahwa kesalahan teridentifikasi, mengidentifikasi konfigurasi sistem yang tidak efektif, mendeteksi perangkat keras dan perangkat lunak, dan mengidentifikasi masalah teknis yang terkait dengan sistem informasi yang sedang diperiksa. Penetration testing memiliki manfaat yang signifikan dalam mengidentifikasi dan mengatasi kerentanan dalam infrastruktur jaringan, serta menunjukkan tingkat kelemahan yang dapat ditemui dalam sistem tersebut, sehingga memungkinkan perbaikan dan perlindungan yang lebih efektif terhadap serangan *cyber* yang berbahaya[1]

Dalam melakukan pengujian penetrasi, ada beberapa standar dan metode pengujian penetrasi yang dapat dijadikan tolak ukur dalam sistem untuk menguji keamanan sistem. Beberapa standar yang sering dipakai yaitu *Open-Source Security Testing Methodology Manual* (OSSTMM), *Open Web Application Security Project* (OWASP), *National Institute of Standards and Technology* (NIST), *Penetration Testing Execution Standards* (PTES) *dan Information System Security Assessment Framework* (ISSAF)[4].

D. *Penetration Testing Execution Standard*

Penetration Testing Execution Standard (PTES) adalah sebuah standar yang dirancang untuk memudahkan bisnis dan penyedia jasa keamanan dalam melakukan pengujian penetrasi. PTES menggunakan bahasa yang umum dan mencakup semua aspek pengujian penetrasi[1] Langkah-langkah PTES seperti berikut :



GAMBAR 1
Tahapan PTES

1. *Pre-Engagement Interaction* : Fase *Pre-engagement interactions* bertujuan menjelaskan alat dan teknik untuk tahap awal uji penetrasi. Pemilihan alat bergantung pada jenis dan kedalaman pengujian. Izin pengujian wajib dipenuhi sebelum pengujian dilakukan.
2. *Intelligence Gathering*
   Langkah ini mengumpulkan informasi untuk mengarahkan pemeriksaan, mencakup data karyawan, fasilitas, produksi, dan perencanaan, serta menyoroti temuan potensial yang relevan.
3. *Threat Modelling*
   Threat Modeling adalah fase untuk menentukan model ancaman yang tepat dalam uji penetrasi. Dalam standar PTES, tidak ada model khusus, tetapi diperlukan model yang konsisten dalam representasi ancaman, kapabilitas, kualifikasi organisasi, serta dapat diterapkan secara berulang dengan hasil yang konsisten.
4. *Vulnerability Analysis*
   Analisis kerentanan bertujuan mengidentifikasi dan menilai risiko konflik. Identifikasi kerentanan menjadi komponen utama, sementara validasi menyaringnya hingga hanya yang sah dan relevan.
5. *Exploitation*
   Fase eksploitasi dalam pengujian penetrasi bertujuan mengakses sistem dengan memanfaatkan celah keamanan. Jika analisis kerentanan kurang optimal, fase ini harus dilakukan dengan presisi tinggi untuk mengidentifikasi akses utama dan aset berharga.
6. *Post-exploitation*
   Post Exploitation bertujuan menilai nilai sistem yang disusupi berdasarkan sensitivitas data dan kinerja jaringan, serta memperkuat kontrol agar tetap berfungsi.
7. *Reporting*
   Langkah akhir uji penetrasi adalah laporan untuk menyoroti temuan, risiko, dan perbaikan sistem jangka panjang.

E. *Burp Suite*

Burp Suite adalah sebuah *software* yang lengkap dan berbasis *PortSwigger* yang digunakan untuk melakukan pengujian keamanan aplikasi *website*. *BurpSuite* berfungsi dengan menggunakan fitur *proxy* yang memungkinkan pengguna untuk memantau, mencegah, menampilkan, serta memodifikasi lalu lintas HTTP antara *browser* dan *server*[5].

F. *Nmap*

Nmap *(Network Mapper)* adalah alat yang membantu kita memindai jaringan dan membuat peta yang menunjukkan semua perangkat yang terhubung. Peta ini dibuat secara otomatis, sehingga lebih mudah digunakan daripada menggabungkan alat pemindai dan pemetaan secara manual. Namun, peta Nmap terkadang sulit dibaca, terutama pada jaringan besar dengan banyak perangkat dan koneksi. Hal ini karena banyaknya informasi yang ditampilkan dapat membuat peta menjadi rumit dan membingungkan[6].

G. *OWASP ZAP*

OWASP ZAP adalah alat bantu untuk menemukan celah keamanan di aplikasi *website*. Menurut situs web resminya, *ZAP* merupakan proyek komunitas terbuka yang memungkinkan individu atau organisasi untuk berkontribusi dalam pengembangannya. Alat ini dapat dipasang di berbagai sistem operasi, seperti *Windows*, *Linux*, dan *macOS*[2].

H. *SQL Injection*

SQL *(Structured Query Language)* adalah bahasa pemrograman sensitif yang digunakan untuk mengakses dan memanipulasi data secara relasional. *SQL* berfungsi menampilkan hasil atau memproses data yang tidak diinginkan. Sementara itu, *SQL injection* adalah teknik hacking yang menyusup ke dalam sistem untuk mengakses database sebuah situs. Serangan ini terjadi akibat kelemahan kode program dan kurangnya keamanan pengelola situs[2].

I. *Distributed Denial of Service*

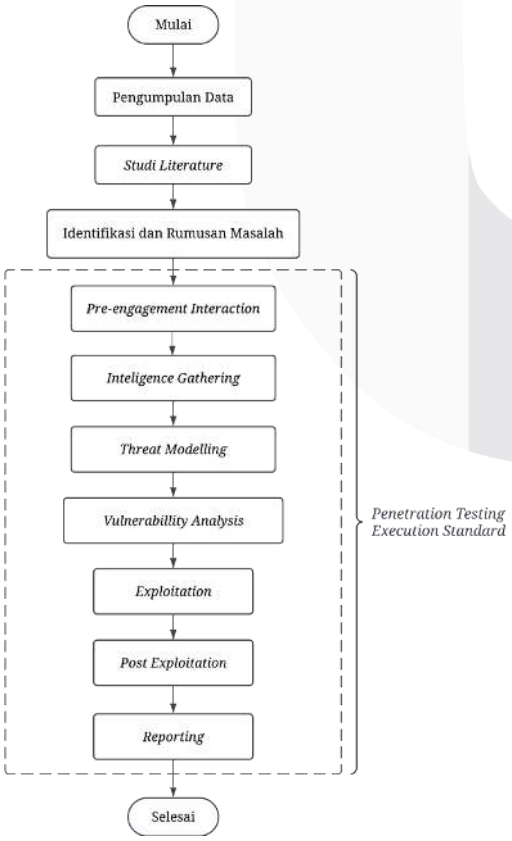
DDoS merupakan salah satu jenis serangan yang dilakukan oleh penyerang dengan tujuan membanjiri lalu lintas jaringan untuk menguras sumber daya yang dimiliki oleh komputer atau server tertentu. Ada beberapa jenis serangan yang sering digunakan diantaranya adalah *UDP Flood. UDP flood* adalah salah satu jenis serangan *DDoS* yang melibatkan pengiriman paket *User Datagram Protocol* (UDP) dalam jumlah besar ke target. Prosedur ini bertujuan untuk menentukan *port* secara acak pada *host* jarak jauh sehingga *host* secara terus menerus melakukan pengecekan untuk melihat apakah ada aplikasi yang aktif pada *port* yang dimaksud. Jika aplikasi tidak diterima, *host* akan merespons dengan mengirimkan paket *ICMP* "*Destination Unreachable*"[7].

III. METODE

Memberikan Penelitian ini menganalisis keamanan sistem di Pengadilan Negeri X melalui beberapa tahap, seperti pengumpulan informasi, identifikasi ancaman, eksploitasi, serta pelaporan dan rekomendasi perbaikan. Proses ini menggunakan metode *Penetration Testing Execution Standard* (PTES) dari *Pre-engagement Interactions* hingga *Reporting*.

A. Diagram Alir Penelitian

Penelitian ini bertujuan untuk menguji keamanan sistem di Pengadilan Negeri X menggunakan metode *Penetration Testing Execution Standard* (PTES). Proses ini mencakup pengumpulan informasi, analisis ancaman, eksploitasi, serta pelaporan dan perbaikan. Diagram alir berikut menggambarkan tahapan penelitian secara sistematis.



GAMBAR 2
Diagram Alir Penelitian

Dalam tahap penelitian, langkah-langkah yang digunakan meliputi:

1. Pengumpulan Data
   Pada tahap ini, peneliti mengumpulkan data yang relevan untuk analisis keamanan serangan, seperti data serangan siber terbaru di Indonesia. Data tersebut dikumpulkan dengan cara membaca penelitian sebelumnya yang berasal dari jurnal penelitian dan juga dengan melakukan observasi pada sumber informasi resmi melalui *website*
2. Studi Literatur
   Studi literatur dilakukan dengan memperhatikan pentingnya keberadaan sebuah studi literatur yang dapat mendukung teori dan praktik penelitian yang dilakukan. Studi literatur ini terkait dengan tema yang sama seperti penelitian sebelumnya, buku, jurnal, dan informasi dari internet yang relevan dengan penelitian ini.
3. Identifikasi dan Rumusan Masalah
   Pada tahap ini, peneliti mulai mengidentifikasi permasalahan yang terjadi di lapangan yang dinilai relevan untuk diteliti. Peneliti kemudian menemukan adanya kerentanan dalam sistem informasi *website* Pengadilan Negeri X yang belum disadari oleh admin web. Berdasarkan temuan tersebut, peneliti merumuskan masalah dan melakukan penelitian terkait analisis keamanan sistem informasi *website* Pemerintah Daerah dengan menerapkan metode PTES.
4. *Pre-Engagement Interactions*
   Pada tahap ini, peneliti melaksanakan berbagai tugas, termasuk mengidentifikasi permasalahan dalam sistem informasi pemerintah yang dijadikan sampel. Setelah itu, peneliti berkoordinasi dengan pihak terkait dengan menyusun surat izin penelitian. Salah satu dokumen krusial yang diperlukan adalah izin untuk melakukan uji penetrasi.
5. *Intelligence Gathering*
   Pada tahap ini, peneliti mengumpulkan informasi untuk menganalisis situs web Pengadilan Negeri X menggunakan metode PTES. Informasi yang dikumpulkan mencakup nama domain, subdomain, alamat IP, email, dan DNS. Peneliti memanfaatkan Whois untuk memperoleh data domain, serta Nmap untuk menganalisis status port dan platform yang digunakan. Selain itu, ekstensi Wappalyzer digunakan untuk identifikasi teknologi situs, guna memudahkan simulasi serangan.
6. *Threat Modeling*
   Pemodelan Ancaman adalah langkah dalam proses penelitian yang berfokus pada dua komponen utama: aset (aset dan proses bisnis) dan penyerang (ancaman komunitas dan kemampuan). Model ini digunakan untuk membantu peneliti agar mudah memahami kerentanan keamanan yang akan dibahas selama proses penelitian pada situs web Pengadilan Negeri X.
7. *Vulnerability Analysis*
   Pada tahap analisis kerentanan selanjutnya, peneliti memulai dengan menggunakan *tool* ZAP untuk menganalisis sistem informasi *website*. *Tool* ini dapat memberikan informasi mengenai kerentanan yang ada pada *website* Pengadilan Negeri X dan akan menunjukkan tingkat risiko dari setiap kerentanan yang teridentifikasi.

8. *Exploitation*
   Pada tahap eksploitasi, peneliti menguji celah keamanan situs web Pengadilan Negeri X berdasarkan kerentanan yang teridentifikasi dalam tahap Analisis Kerentanan. Pengujian dilakukan menggunakan teknik injeksi SQL pada sistem informasi situs web tersebut guna mengevaluasi tingkat keamanannya. Untuk melaksanakan serangan penetrasi, peneliti memanfaatkan alat SQLMap. Selain itu, peneliti juga menggunakan *tool* LOIC dalam melakukan serangan DDoS.
9. *Post-exploitation*
   Pada tahap *Post Exploitation*, dilakukan penilaian terhadap tingkat risiko pada sistem yang masih rentan terhadap ancaman keamanan setelah melewati proses uji eksploitasi.
10. *Reporting*
    Setelah pengujian selesai, peneliti melakukan pelaporan untuk menyampaikan hasil pengujian keamanan *website* Pengadilan Negeri X dengan metode PTES. Laporan ini mencakup jenis serangan, *tool*s yang digunakan, status serangan, serta rekomendasi dan solusi pascapengujian.

B. Alasan Pemilihan Metode

Dalam penelitian ini, diterapkan metode PTES, yang merupakan standar untuk audit serta analisis keamanan sistem di perusahaan. PTES menetapkan tahapan yang jelas dalam mengidentifikasi kerentanan dan menganalisis potensi ancaman. Proses evaluasi yang ditawarkan oleh PTES dirancang agar dapat dipahami dengan mudah oleh pengguna dengan berbagai tingkat keahlian dalam pengujian penetrasi.

IV. HASIL DAN PEMBAHASAN

Dalam pembahasan ini, peneliti akan menyajikan hasil penelitian yang telah dilaksanakan. Penelitian ini menggunakan *website* pemerintah daerah sebagai sampel untuk pengujian melalui pentest. Setelah pengujian pentest selesai, selanjutnya dilakukan analisis keamanan pada *website* sampel dengan menggunakan metode Penetration Testing Execution Standard (PTES).

A. *Pre-engagement Interactions*

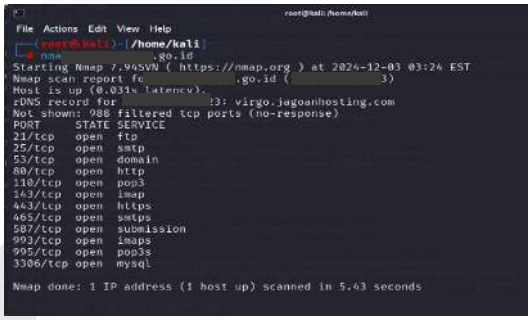
Pada tahap ini, seluruh persiapan untuk pengujian serangan dijelaskan, mencakup tujuan pengujian, lingkup, batasan, persyaratan, serta persiapan yang diperlukan. Tujuan dari serangan ini adalah untuk menguji tingkat keamanan *website* berbasis *Wordpress* terhadap serangan *XSS* dengan *payload* serta menguji *database* menggunakan teknik *SQL Injection*. Lingkup pengujian pada tahap ini meliputi contoh *website* pemerintah daerah yang memakai *Wordpress* untuk menguji ketahanannya terhadap serangan peretas. Dari berbagai kerentanan yang teridentifikasi, terdapat tingkat risiko medium, rendah, dan informasi. Penelitian ini dibatasi hanya pada pengujian kerentanan dengan tingkat risiko *medium*. Selain itu, izin penelitian telah diperoleh dan terlampir pada lembar lampiran. Selanjutnya, peneliti menyiapkan alat dan bahan yang diperlukan untuk melakukan analisis keamanan terhadap sistem informasi *website* Pemerintah Pengadilan Negeri X.

B. *Intelligence Gathering*

Pada tahap ini, peneliti mengumpulkan informasi penting seperti nama domain, *subdomain*, alamat IP, informasi domain, alamat email, dan DNS untuk menganalisis *website* Pengadilan Negeri X dengan metode PTES. Menggunakan *website* whois, peneliti memperoleh data tentang nama domain, alamat *email*, dan *DNS*, yang ditampilkan pada Gambar 3. Hasil analisis menunjukkan bahwa domain tersebut menggunakan vendor hosting *jagoanhosting.com* serta mencantumkan alamat organisasi pendaftar domain.

GAMBAR 3
Hasil *Whois Lookup*

Selanjutnya, peneliti akan memanfaatkan *tool Nmap* untuk mengumpulkan informasi penting mengenai semua port yang aktif atau terbuka, beserta banner yang menyajikan detail tentang jenis alat atau platform yang digunakan pada setiap layanan di *website* Pengadilan Negeri X.



GAMBAR 4
Hasil *Scanning Nmap*

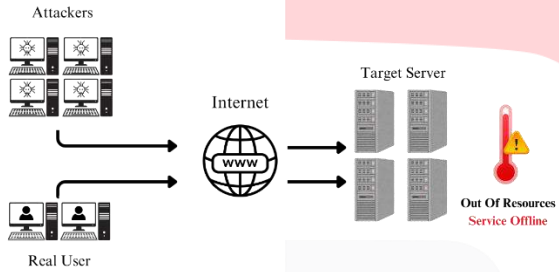
Selain menggunakan *tool Nmap*, peneliti juga memanfaatkan ekstensi browser Wappalyzer, yang dapat mendeteksi teknologi yang digunakan oleh sebuah situs web, seperti CMS, *framework*, dan bahasa pemrograman.

GAMBAR 5
Analisis *Tool Wappalyzer*

Pada gambar 5 terlihat bahwa *website* Pengadilan Negeri X untuk CMS (Content Management System) menggunakan *Wordpress* versi 6.7.1. Hal tersebut menunjukan bahwa Pengadilan Negeri X sudah menggunakan versi yang terbaru. Berbeda dengan versi sebelumnya yang memiliki kerentanan XSS (*Cross Site Scripting*), pada versi terbaru sudah diperbaiki pada kerentanan tersebut.

C. *Threat Modeling*

Threat modeling berperan dalam meningkatkan keamanan jaringan dengan mengidentifikasi kerentanan, menetapkan tujuan, serta merancang strategi mitigasi untuk mencegah atau mengurangi dampak serangan siber terhadap sistem. Dalam konteks ini, *threat modeling* diterapkan sebagai pendekatan perancangan dalam proses pengujian. Pemodelan ini bertujuan untuk membantu penguji dan perusahaan target dalam memahami potensi celah keamanan yang akan diidentifikasi selama penelitian berlangsung.



GAMBAR 6
Threat Modelling DDoS

Model ancaman dapat diilustrasikan melalui serangan *Distributed Denial of Service* (DDoS). Secara prinsip, serangan DDoS beroperasi dengan membanjiri server dengan sejumlah besar permintaan, baik pada layanan maupun jaringan, hingga melampaui kapasitas yang dapat ditangani. Akibatnya, server mengalami gangguan fungsi, yang disebut sebagai *Data Flooding*. Serangan ini umumnya dimulai dengan beberapa komputer yang dikendalikan dalam sebuah botnet melalui layanan perintah dan kontrol.

D. *Vulnerability Analysis*

Pada tahap ini peneliti akan melakukan analisis kerentanan yang dimiliki *website* dengan cara melakukan scanning menggunakan beberapa *tool*s pemindai keamanan *website*. Terdapat 2 *tool*s yang peneliti pakai dalam melakukan scanning *website*, yaitu ZAP dan WPScan.

1. Analisis Menggunakan *OWASP ZAP*

Pertama peneliti menggunakan *tool*s ZAP untuk melakukan pemindaian kerentanan *website*. Pada Tabel 1 merupakan hasil dari pemindaian kerentanan menggunakan ZAP. Terdiri dari 13 *alert*, dengan level *medium* berjumlah 2, level *low* 6 dan 5 dengan level *informational*.

TABEL 1
Kerentanan Hasil *Scanning Tool* ZAP

No	Jenis Kerentanan	Level Risiko
1	*Content Security Policy (CSP) Header Not Set*	*Medium*
2	*Missing Anti-clickjacking Header*	*Medium*
3	*Cookie No HttpOnly Flag*	*Low*
4	*Cookie without SameSite Attribute*	*Low*
5	*Secure Pages Include Mixed Content*	*Low*
6	*Strict-Transport-Security Header Not Set*	*Low*
7	*Timestamp Disclosure - Unix*	*Low*
8	*X-Content-Type-Options Header Missing*	*Low*
9	*Information Disclosure - Suspicious Comments*	*Informational*
10	*Modern Web Application*	*Informational*
11	*Re-examine Cache-control Directives*	*Informational*
12	*Session Management Response Identified*	*Informational*
13	*User Controllable HTML Element Attribute (Potential XSS)*	*Informational*

Dari hasil analisis menggunakan ZAP, menghasilkan kerentanan yang tercantum pada Tabel 1 dengan deskripsi dari masing masing kerentanan sebagai berikut:

a *Content Security Policy* (CSP) : CSP bertujuan mengatur dan mencegah serangan seperti *Cross Site Scripting* (XSS) dan intrusi. CSP menggunakan header *HTTP* standar yang memungkinkan pemilik situs menentukan sumber konten yang diperbolehkan, sehingga *browser* hanya memuat konten yang telah disetujui. Jenis konten yang diatur meliputi *JavaScript*, CSS, *HTML iframe*, *font*, gambar, serta objek seperti *Java*, *ActiveX*, audio, dan video.

b Missing Anti-clickjacking Header : Respons ini tidak memberikan perlindungan terhadap serangan *'ClickJacking'*. Oleh karena itu, perlu mencakup *Content Security Policy* yang menetapkan aturan *'frame-ancestors'* atau *X-Frame-Options*.

c *Cookie No HttpOnly Flag* : *Cookie* dikonfigurasi tanpa atribut *HttpOnly*, sehingga dapat diakses oleh *JavaScript*. Jika terdapat skrip yang berjalan dalam periode ini, *cookie* dapat diambil dan disalin ke lokasi lain. Hal ini berpotensi menyebabkan penyusupan sesi jika *cookie* tersebut digunakan.

d *Cookie without SameSite Attribute* : Karena cookie dibuat tanpa atribut *SameSite*, *cookie* tersebut dapat dianggap sebagai bagian dari permintaan "lintas situs". Fitur *SameSite* berperan sebagai mekanisme efektif untuk menangani permintaan terkait informasi lokasi, pengiriman skrip eksternal, serta kueri yang bergantung pada waktu.

e *Secure Pages Include Mixed Content* : Halaman ini mencakup konten campuran, yaitu konten yang diakses melalui *HTTP*, bukan *HTTPS*.

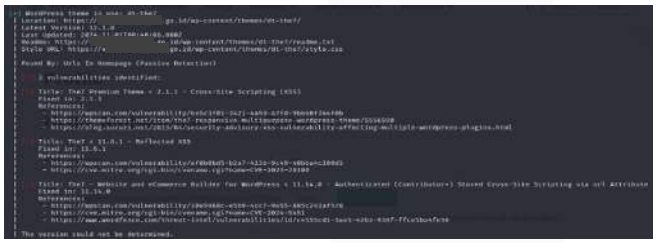
f *Strict-Transport-Security Header Not Set* : *HTTP Strict Transport Security* (HSTS) adalah mekanisme keamanan web yang diterapkan pada server untuk memastikan bahwa pengguna, seperti browser yang kompatibel, hanya berkomunikasi melalui koneksi HTTPS yang aman menggunakan *TLS/SSL*. Protokol HSTS dikembangkan oleh *IETF* dan dijelaskan dalam *RFC 6797*.

g *Timestamp Disclosure – Unix* : Stempel waktu diberikan oleh aplikasi atau server web. Produk ini mengungkapkan data rahasia kepada pihak yang tidak berwenang untuk mengakses informasi tersebut.

h *X-Content-Type-Options Header Missing* : *Header X-Content-Type-Option*s atau mekanisme *Anti-MIME-Sniffing* tidak dikonfigurasi dengan nilai "*nosniff*", sehingga memungkinkan *browser* versi lama seperti *Internet Explorer* dan *Chrome* untuk melakukan deteksi otomatis terhadap jenis MIME dalam respons. Hal ini dapat menyebabkan konten dikenali sebagai jenis yang berbeda dari yang telah ditentukan. Sementara itu, *Firefox*, termasuk versi terbaru per awal 2014 dan versi sebelumnya, akan tetap menggunakan jenis konten yang telah ditetapkan tanpa melakukan *MIME-sniffing*.

2. Analisis Menggunakan *WPScan*

Berdasarkan hasil tahap *Intelligence Gathering* yang telah dilakukan, ditemukan bahwa *website* target menggunakan *CMS Wordpress*. Oleh karena itu, peneliti memilih menggunakan *WPScan* untuk menganalisis *website* tersebut, karena *WPScan* merupakan alat yang sangat sesuai untuk mengaudit situs berbasis *Wordpress*.



GAMBAR 7
Hasil Analisis WPScan

Berdasarkan gambar 7 hasil yang peneliti dapatkan menggunakan *tools WPScan*, terindikasi bahwa *website* target menggunakan tema *The7 version* 10.0.0 memiliki 3 *themes vulnerabilities* sebagai berikut :

TABEL 2
Hasil Kerentanan *WPScan*

Title	***Vulnerability***	***Fixed in***
The7 Premium Theme	*Cross-Site Scripting (XSS)*	2.1.1
The7	*Reflected XSS*	11.6.1
The7 Website eCommerce Builder for Wordpress	*Authenticated (Contributor+) Stored Cross-Site Scripting via url Attribute*	11.14.0

Tema *The7 Premium Theme* sebelumnya memiliki kerentanan terhadap serangan XSS, yang memungkinkan penyerang menyisipkan skrip berbahaya ke dalam halaman web. Kerentanan ini telah diperbaiki pada versi 2.1.1. Selain itu, tema *The7* juga mengalami kelemahan *Reflected XSS*, di mana penyerang dapat menyuntikkan *payload* berbahaya melalui input seperti *URL* atau formulir, yang kemudian dipantulkan kembali ke pengguna tanpa disimpan di server. Masalah ini telah diperbaiki pada versi 11.6.1.

Sementara itu, pada tema *The7 Website eCommerce Builder for Wordpress*, terdapat celah keamanan terkait hak istimewa pada peran tertentu, seperti contributor di *Wordpress*. Peran ini memungkinkan penyisipan payload berbahaya yang tersimpan di server dan dieksekusi saat pengguna dengan hak akses lebih tinggi (seperti Admin) atau pengunjung membuka halaman yang terdampak.

E. *Exploitation*

Pada tahap eksploitasi ini, peneliti akan menyelidiki isu-isu yang telah ditemukan dalam fase analisis kerentanan. Tujuannya adalah untuk memastikan apakah kerentanan tersebut benar-benar dapat dimanfaatkan atau tidak. Beberapa kerentanan yang akan dieksploitasi antara lain sebagai berikut:

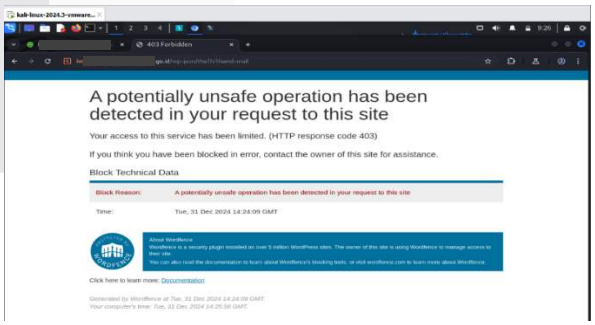
1. *Cross Site Scripting (XSS)*

Dalam kerentanan ini, peneliti akan menyisipkan skrip pada halaman *website*, tepatnya di bagian formulir komentar. Untuk melakukannya, peneliti akan menggunakan alat Burp Suite guna mencegat lalu lintas data pada situs target. Skrip yang akan disisipkan dalam kolom komentar dapat dilihat pada Tabel 3.

TABEL 3
Script Alert Kerentanan XSS

```
<input onbeforecopy=alert(1) value="XSS" autofocus>
```

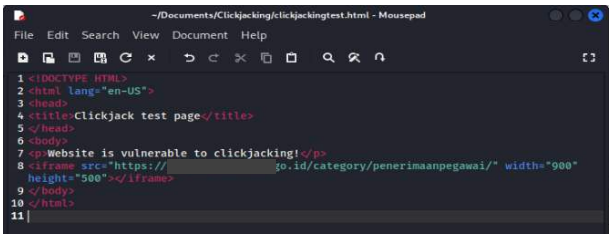
Setelah permintaan dikirim ke server *website*, *website* akan memberikan respons seperti yang ditampilkan pada Gambar 8. Respons tersebut menunjukkan bahwa *website* target dilindungi oleh firewall Wordfence, sehingga script yang disisipkan terdeteksi sebagai ancaman dan tidak dijalankan oleh server *website* tersebut.



GAMBAR 8
Respon *Website* Setelah disisipkan Script XSS

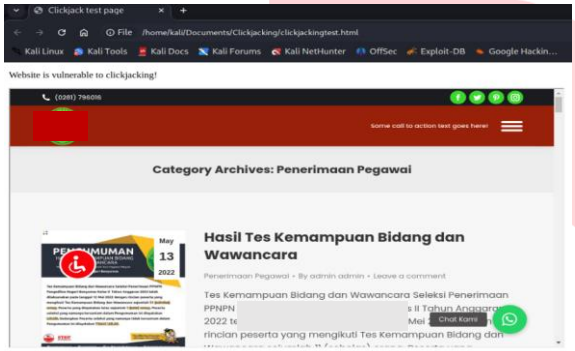
2. *Clickjacking*

Dalam kerentanan ini, peneliti akan mencoba mengeksploitasi ketiadaan X-Frame Option pada *website* tersebut. Tanpa fitur ini, *website* dapat disematkan di dalam situs lain, yang berpotensi menimbulkan risiko serangan clickjacking.



GAMBAR 9
Script Pengujian *Clickjacking*

Gambar 9 memperlihatkan sebuah percobaan yang dilakukan dengan membuat script yang bertujuan untuk melancarkan serangan clickjacking pada URL yang memiliki kerentanan.



GAMBAR 10
Hasil Percobaan Pengujian Clickjacking

Gambar 10 menunjukkan hasil percobaan untuk menguji kerentanan akibat tidak dikonfigurasinya X-Frame-Options Header. Salah satu serangan yang berpotensi terjadi adalah clickjacking, yaitu serangan yang memungkinkan penyerang menyisipkan instruksi yang dapat dijalankan di situs web lain. Kerentanan ini muncul karena X-Frame-Options Header tidak diterapkan, sehingga laman web yang rentan dapat ditampilkan dalam laman web lain. Akibatnya, serangan lain seperti phishing juga dapat terjadi.

3. *SQL Injection*

Dalam tahap analisis kerentanan, peneliti berhasil menemukan halaman login admin *Wordpress* yang dikelola oleh pemilik *website*. SQL Injection merupakan salah satu celah keamanan yang sering dimanfaatkan oleh peretas, karena memungkinkan mereka memperoleh informasi rahasia dari sebuah *website*, seperti *database*, serta melewati proses otentikasi.

GAMBAR 11
Pengujian *SQL Injection* dengan *Tool SQL Map*

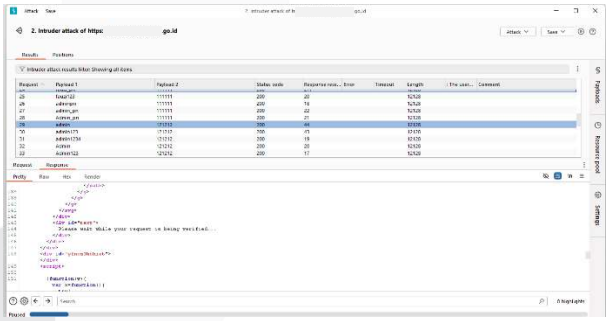
Gambar 11 menunjukkan hasil dari proses injeksi yang telah dilakukan. Dalam penelitian ini, *SQLMap* digunakan untuk mengeksploitasi celah *SQL Injection*. Namun, saat mencoba melakukan injeksi dengan perintah melalui *SQLMap*, serangan tidak berhasil menembus sistem target. Kegagalan ini disebabkan oleh tidak adanya parameter seperti *POST* atau *GET* yang dapat diuji untuk *SQL Injection*, seperti *?id=1* atau *?user=admin*. Selain itu, keberadaan *Web Application Firewall* (WAF) yang melindungi sistem juga menjadi faktor penghalang dalam serangan ini.

4. *Brute Force*

Dalam eksploitasi Brute Force ini, peneliti menguji halaman login admin *Wordpress* yang terdeteksi selama analisis menggunakan *OWASP ZAP*. Dengan memanfaatkan *payload* berisi kombinasi acak *username* dan *password*, peneliti mencoba berbagai kemungkinan untuk mengakses sistem.

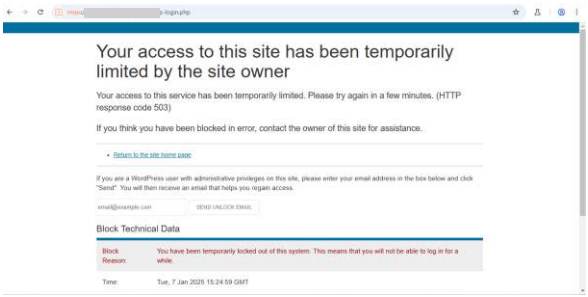
GAMBAR 12
Enumerating Users Halaman *Login*

Pada Gambar 12, peneliti melakukan enumerasi pengguna untuk memperoleh informasi tentang user yang terdaftar pada halaman login admin *Wordpress* di *website* target. Namun, saat melakukan pemindaian menggunakan *WPScan*, peneliti tidak berhasil menemukan *user* apa pun. Oleh karena itu, peneliti menggunakan *payload* yang berisi kombinasi *username* dan *password* yang diperoleh dari internet.



GAMBAR 13
Brute Force Halaman *Login* Admin

Pada Gambar 13, peneliti telah melakukan serangan *brute force* pada halaman *login* admin dengan menggunakan *tool Burp Suite*. Dalam percobaan ini, peneliti menggunakan payload yang berisi 714 kombinasi pasangan *username* dan password yang diuji pada halaman login tersebut. Namun, upaya ini belum berhasil menemukan username dan *password* yang terdaftar dalam sistem *website* tersebut.



GAMBAR 14
Respon *Website* Setelah Percobaan *Brute Force*

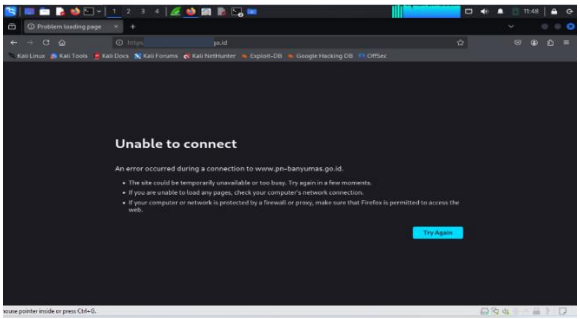
Setelah peneliti melakukan uji coba serangan *brute force* pada halaman *login*, pada percobaan ke-76, *website* memberikan respons seperti yang ditampilkan pada Gambar 14, yang menunjukkan bahwa permintaan *login* ditolak. Hal ini terjadi karena *website* telah dilindungi oleh *firewall Wordfence*, yang berfungsi membatasi jumlah percobaan *login* yang diterima, sehingga mencegah serangan *brute force* dari pihak luar yang berusaha mengakses sistem tanpa izin.

5. *Distributed Denial of Services (DDoS)*

Dalam eksploitasi serangan DDoS ini, peneliti akan mengirimkan sejumlah besar paket data ke situs web target. Menggunakan alat *LOIC*, serangan dilakukan melalui protokol UDP pada port 80.

GAMBAR 15
Pengujian *DDoS* Menggunakan *LOIC*

Pada Gambar 15, peneliti mengonfigurasi URL dan IP dari situs web yang akan diuji. Berdasarkan gambar tersebut, peneliti menetapkan jumlah *thread* yang dikirim sebanyak 1000 melalui protokol *HTTP* pada port 80. Setelah serangan *DDoS* berlangsung selama beberapa detik, peneliti mencoba kembali mengakses situs web target. Akibat serangan tersebut, situs web menjadi tidak dapat diakses dan menampilkan respons seperti yang terlihat pada Gambar 16.



GAMBAR 16
Respon *Website* Setelah Dilakukan Serangan *DDoS*

Setelah serangan DDoS dilakukan, *website* menampilkan respons "*Unable to connect*", menandakan ketidakmampuan akses. Namun, saat diakses melalui jaringan lain, *website* tetap normal tanpa gangguan. Ini menunjukkan serangan belum berhasil karena server memblokir IP penyerang, sehingga hanya pengguna dengan IP yang sama yang terblokir, sementara akses dari jaringan lain tetap lancar.

GAMBAR 17
Mengakses *Website* Menggunakan Jaringan Lain

Pada Gambar 17, peneliti mengakses *website* melalui jaringan seluler dan mendapati bahwa *website* tetap berfungsi normal. Artinya, IP di luar jaringan penyerang masih dapat mengakses *website* tanpa gangguan. Namun, jika menggunakan IP yang sama dengan penyerang, akses akan terblokir.

F. *Post Exploitation*

Pada tahap ini peneliti akan melakukan penilaian tingkat risiko terhadap sistem yang memiliki celah keamanan yang terbuka berdasarkan eksploitasi yang telah dilakukan.

1. Dampak Pada Sistem Target

   Eksploitasi menemukan kerentanan clickjacking pada *website* Pengadilan Negeri X, memungkinkan penyerang memanipulasi interaksi pengguna melalui frame tak terlihat. Akibatnya, pengguna dapat tanpa sadar memberikan informasi penting atau izin, berisiko mengalami pengambilalihan sesi atau perubahan transaksi dalam sistem.

2. Risiko Pada Aset yang Dimiliki Sistem

   Pengujian terhadap *SQL Injection, XSS*, dan *Brute Force* tidak mengungkap informasi sensitif berkat perlindungan *firewall* yang efektif. Namun, sistem masih rentan terhadap serangan *clickjacking* akibat celah pada perlindungan *header* keamanan, yang dapat mengganggu layanan pemantauan kasus dan pengaduan.

3. Skenario Pemanfaatan Celah Keamanan

   Penyerang dapat menyisipkan halaman berbahaya yang menampilkan situs Pengadilan Negeri X dalam frame tak terlihat, sehingga pengguna tanpa sadar mengaktifkan fitur pemantauan kasus atau proses pengaduan saat menekan tombol atau tautan.

G. *Reporting*

Berdasarkan seluruh proses pengujian yang telah dilakukan, peneliti menyajikan hasil penelitian dengan metode *Penetration Testing Execution* Standard pada situs web Pengadilan Negeri X, yang dituangkan dalam laporan hasil pengujian keamanan pada Tabel 4.

TABEL 4
Hasil Pengujian *Penetration Testing*

Jenis Serangan	*Tools*	Status
Cross Site Scripting (XSS)	*BurpSuite*	Gagal
Clickjacking	*Mousepad & Mozilla*	Berhasil
SQL Injection	*SQL Map*	Gagal
Brute Force	*BurpSuite*	Gagal
Distributed Denial of Service (DDos)	*Low Orbit Ion Common (LOIC)*	Gagal

Dari lima serangan pada tahap *exploitation*, hanya *Clickjacking* yang berhasil dieksploitasi. *Cross-Site Scripting* dan *Brute Force* gagal karena dilindungi *firewall Wordfence*, sementara *SQL Injection* tidak dapat dieksekusi akibat ketiadaan parameter *GET* atau *POST* di halaman *login* admin. Serangan *DDoS* juga tidak berhasil karena sistem keamanan mampu memblokir akses penyerang.

Hasil evaluasi keamanan mengidentifikasi potensi risiko pada situs Pengadilan Negeri X, dan saran perbaikannya dirangkum dalam Tabel 5.

TABEL 5
Solusi dan Rekomendasi Perbaikan

Vulnerability	Solusi
Clickjacking	• Tambahkan *header X-Frame-Options* ke dalam respon HTTP untuk menghindari pemuatan halaman dalam *iframe*. • Terapkan kebijakan CSP dengan pengaturan *frame-ancestors* untuk mengatur sumber yang diizinkan untuk memuat konten dalam bingkai. • Atur sesi *cookie* menggunakan atribut *SameSite* untuk menghalangi pengiriman *cookie* ketika halaman dibuka dalam *iframe* dari domain yang berbeda. Nilai yang direkomendasikan adalah *Strict* atau *Lax*.
Tema *wordpress* yang kadaluarsa	• Memastikan *framework website ter-update* dan tidak terdapat kesalahan konfigurasi. • Selalu *update* dan *upgrade* semua *software* yang digunakan baik pada aplikasi, perangkat jaringan maupun server.

V. KESIMPULAN

Berdasarkan pengujian dan analisis pada *website* Pengadilan Negeri X, ditemukan beberapa kerentanan. Hasil *scanning* menggunakan *Nmap* menunjukkan beberapa port terbuka, termasuk 21/TCP, 80/TCP, 443/TCP, dan 3306/TCP, serta mengidentifikasi IP domain target. Pemindaian dengan *OWASP ZAP* menemukan celah keamanan dengan risiko *medium* (2 jenis), *low* (6 jenis), dan *informational* (5 jenis). Dari lima eksploitasi yang diuji, hanya Clickjacking yang berhasil, sementara empat lainnya gagal karena perlindungan *Web Application Firewall*. *Website* telah aman dari serangan *critical*, namun masih rentan terhadap *Clickjacking*, sehingga perlu diterapkan *header X-Frame-Options* dalam respons HTTP untuk mencegah pemuatan halaman dalam *iframe*.

VI. REFERENSI

[1] G. Ary, S. Sanjaya, G. Made, A. Sasmita, D. Made, and S. Arsa, “Evaluasi Keamanan Website Lembaga X Melalui Penetration Testing Menggunakan Framework ISSAF,” Bali, Aug. 2020.

[2] S. Andriyani, M. Fajar Sidiq, and B. Parga Zen, “Analisis Celah Keamanan Pada Website Dengan Menggunakan Metode Penetration Testing Dan Framework Issaf Pada Website SMK Al-Kautsar,” 2023.

[3] A. F. Sallaby and I. Kanedi, “Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter,” Bengkulu, Feb. 2020. doi: https://doi.org/10.37676/jmi.v16i1.1121.

[4] Aakanchha Keshri, “Top 5 Penetration Testing Methodologies and Standards.” Accessed: May 28, 2024. [Online]. Available: https://www.getastra.com/blog/security-audit/penetration-testing-methodology/

[5] A. R. Irawan, A. Widjajarto, and M. Fathinuddin, “Implementasi dan Analisis Attack Tree pada Aplikasi DVWA Berdasar Metrik Time dan Probability,” 2023.

[6] Y. Mulyanto and E. Haryanti, “Analisis Keamanan Website Sman 1 Sumbawa Menggunakan Metode Vulnerability Asesement,” *JINTEKS*, vol. 3, no. 3, 2021, doi: 10.51401.

[7] G. Hendita artha Kusuma, “Sistem Firewall untuk Pencegahan DDOS Attack di Masa Pandemi Covid-19,” *Journal of Informatics and Advantage Computing (JIAC)*, vol. 3, no. 1, pp. 52–56, May 2022.