Pengujian Kerentanan Pada website X perusahaan layanan teknologi informasi dengan Pendekatan *Vulnerability Assessment* dan *Penetration Testing* Untuk Mengindentifikasi dan Mengatasi Celah Kemanan

1st Malvin Jeconia Setiawan
Sistem Informasi
Telkom University
Bandung, Indonesia
malvinjeconia@student.telkomuniversity.ac.id

2nd Muhammad Fathinuddin
Sistem Informasi
Telkom University
Bandung, Indonesia
muhammadfathinuddin@telkomuniversity.ac.id

3rd Umar Yunan Kurnia Septo Hediyanto
Sistem Informasi
Telkom University
Bandung, Indonesia
umaryunan@telkomuniversity.ac.id

Abstrak*— Dalam era digital yang semakin maju, keamanan aplikasi web menjadi isu krusial bagi perusahaan teknologi informasi. Penelitian ini bertujuan untuk mengidentifikasi dan mengevaluasi kerentanan pada website X milik perusahaan layanan teknologi informasi menggunakan pendekatan Vulnerability Assessment dan Penetration Testing (VAPT). Metodologi yang digunakan mencakup tahap information gathering, vulnerability detection dengan alat bantu seperti OWASP ZAP dan Acunetix, serta pengujian eksploitasi menggunakan Burp Suite. Hasil pengujian menunjukkan terdapat 13 jenis kerentanan dengan tingkat keparahan berbeda, di antaranya SQL Injection, Cross-Site Scripting (XSS), dan pengiriman kredensial tanpa enkripsi. Beberapa kerentanan seperti file .htaccess terbuka dan header keamanan yang hilang berhasil dimitigasi, sementara kerentanan lain seperti pengiriman data melalui HTTP tidak dapat ditangani sepenuhnya karena keterbatasan lingkungan pengujian. Evaluasi pasca mitigasi menunjukkan adanya peningkatan signifikan dalam keamanan sistem. Penelitian ini diharapkan dapat menjadi acuan bagi pengelola sistem dan pengembang web dalam meningkatkan ketahanan terhadap serangan siber.

***Kata kunci*— Keamanan, Web, VAPT, Vulnerability Assessment, Penetration Testing, Mitigasi**

I. PENDAHULUAN

Dalam era digital yang semakin maju, penggunaan Website sebagai media utama untuk mendistribusikan informasi, berkomunikasi, dan menyediakan layanan berbasis internet semakin meningkat. Website tidak hanya digunakan oleh individu atau perusahaan untuk memperluas jangkauan pasar, tetapi juga oleh organisasi pemerintah dan sektor publik untuk memberikan layanan yang lebih cepat dan efisien kepada masyarakat. Seiring dengan peningkatan pemanfaatan website, ancaman keamanan siber juga meningkat secara signifikan, terutama yang berkaitan dengan eksploitasi kerentanan yang terdapat dalam aplikasi web.

Menurut data yang dirilis oleh lembaga keamanan siber global, insiden pelanggaran keamanan data yang disebabkan oleh kerentanan aplikasi web terus mengalami peningkatan setiap tahunnya. Beberapa serangan yang umum terjadi meliputi Cross-Site Scripting (XSS), SQL Injection, dan serangan Distributed Denial of Service (DDoS). Serangan-serangan tersebut umumnya berawal dari kerentanan yang tidak terdeteksi atau tidak teratasi pada tahap pengembangan website. Oleh karena itu, metode yang efektif untuk mengidentifikasi dan menutup celah keamanan pada website sangatlah dibutuhkan.

Vulnerability Assessment dan Penetration Testing adalah dua pendekatan yang umum digunakan dalam mengidentifikasi, mengevaluasi, dan mengatasi kerentanan pada aplikasi web. Vulnerability Assessment berfungsi untuk mendeteksi dan memprioritaskan kerentanan yang ada, sementara Penetration Testing bertujuan untuk menguji seberapa jauh kerentanan tersebut dapat dieksploitasi oleh pihak tidak bertanggung jawab. Kombinasi dari kedua pendekatan ini dapat memberikan gambaran yang komprehensif mengenai status keamanan sebuah website dan langkah-langkah yang diperlukan untuk mengamankannya.

Dalam konteks ini, penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi proses Vulnerability Assessment dan Penetration Testing pada website X, serta memberikan rekomendasi keamanan berdasarkan hasil yang diperoleh. Melalui penelitian ini, diharapkan dapat memberikan kontribusi bagi pengembangan metode pengujian keamanan web yang lebih efektif dan membantu organisasi dalam mengelola risiko keamanan pada aplikasi web mereka.

II. KAJIAN TEORI

II.1 Keamanan Informasi

Keamanan informasi merupakan upaya sistematis yang bertujuan untuk melindungi informasi dari berbagai ancaman, guna menjamin kelangsungan bisnis, meminimalkan risiko, dan memaksimalkan peluang. Menurut ISO/IEC 27001, keamanan informasi adalah perlindungan terhadap kerahasiaan (confidentiality), integritas (integrity), dan ketersediaan (availability)

informasi, yang secara kolektif dikenal dengan istilah CIA Triad.

1. Kerahasiaan (*Confidentiality*)

Merujuk pada jaminan bahwa informasi hanya dapat diakses oleh pihak yang berwenang. Hal ini biasanya diwujudkan melalui mekanisme kontrol akses dan enkripsi.

2. Integritas (*Integrity*)

Menjamin bahwa informasi tetap akurat, lengkap, dan tidak diubah secara tidak sah, baik selama penyimpanan, pemrosesan, maupun transmisi.

3. Ketersediaan (*Availability*)

Menjamin bahwa informasi dan sistem yang mendukungnya tersedia bagi pihak yang berwenang saat dibutuhkan.

II.2 *Web*

Web (singkatan dari World Wide Web) merupakan sistem berbasis internet yang memungkinkan pengguna mengakses dan berinteraksi dengan informasi melalui halaman-halaman yang ditampilkan dalam bentuk website. Web berfungsi sebagai media penyampaian informasi yang dapat diakses secara global melalui jaringan internet menggunakan browser seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge. Web bekerja dengan menggunakan protokol komunikasi standar yang dikenal sebagai HTTP (Hypertext Transfer Protocol) atau versi amannya yaitu HTTPS (HTTP Secure).

II.3 Vulnerability

Vulnerability atau kerentanan adalah kelemahan atau celah pada sistem, perangkat lunak, atau jaringan yang dapat dieksploitasi oleh pihak tidak bertanggung jawab untuk mendapatkan akses atau merusak sistem. Kerentanan ini dibedakan menjadi beberapa jenis seperti software, konfigurasi, perangkat keras, dan faktor manusia (Sari & Nugroho, 2020).Ada beberapa jenis yang bersangkutan dengan Vulnerability:

a. *Software Vulnerability*

Kelemahan pada kode perangkat lunak, seperti *buffer overflow*, *SQL injection*, atau *cross-site scripting (XSS)*.

b. *Configuration Vulnerability*

Kesalahan konfigurasi, seperti pengaturan keamanan yang tidak tepat atau *default credentials* yang tidak diubah.

c. *Hardware Vulnerability*

Celah pada perangkat keras, seperti *side-channel attacks* atau kelemahan pada firmware.

d. *Human Vulnerability*

Faktor manusia, seperti penggunaan

II.4 OWASP 2021

(Open Web Application Security Project) OWASP 2021 adalah versi terbaru dari daftar *Top 10* risiko keamanan pada aplikasi web, yang dirilis oleh *OWASP*. Daftar ini berfungsi sebagai panduan utama bagi pengembang dan profesional keamanan untuk mengenali dan mengatasi ancaman paling kritis dalam pengembangan dan pengelolaan aplikasi *web* [5].

II.4.1 *OWASP ZAP*

OWASP ZAP (Zed Attack Proxy) adalah alat sumber terbuka (*open source*) yang dirancang untuk melakukan pengujian keamanan aplikasi web. Dikembangkan oleh OWASP (*Open Web Application Security Project*), ZAP digunakan untuk menemukan kerentanan keamanan dalam aplikasi web melalui simulasi serangan yang biasa dilakukan oleh penyerang. ZAP merupakan salah satu alat yang populer karena gratis, mudah digunakan, dan mendukung berbagai tingkat pengalaman, mulai dari pemula hingga profesional keamanan.

II.5 Vmware

VMware adalah salah satu perangkat lunak virtualisasi terkemuka yang memungkinkan pengguna untuk menjalankan beberapa sistem operasi secara bersamaan dalam satu perangkat fisik. *VMware* dikembangkan oleh *VMware*, Inc., dan digunakan secara luas dalam pengembangan perangkat lunak, pengujian sistem, serta implementasi infrastruktur cloud dan keamanan siber. Secara umum, *VMware* bekerja dengan menciptakan mesin virtual (*virtual machine*/VM) yang sepenuhnya terisolasi dari sistem operasi utama (host), namun memiliki kemampuan layaknya komputer fisik. Dalam satu mesin fisik, pengguna dapat menjalankan sistem operasi seperti Linux, Windows, atau lainnya secara bersamaan, tanpa perlu melakukan instalasi langsung pada perangkat keras [2].

II.5 NMAP

Nmap (*Network Mapper*) adalah sebuah alat open-source yang dirancang untuk melakukan pemindaian jaringan (network scanning) guna mengumpulkan informasi tentang host dan layanan yang berjalan pada jaringan komputer. Nmap dikembangkan oleh Gordon Lyon (alias Fyodor) dan telah menjadi salah satu tools standar yang digunakan oleh administrator jaringan, auditor keamanan, dan penetration tester di seluruh dunia [6].

II.6 *Penetration Tsting*

Penetration Testing (sering disingkat *pentest*) adalah proses evaluasi keamanan sistem informasi, jaringan, atau aplikasi dengan melakukan simulasi serangan siber untuk mengidentifikasi celah keamanan (*vulnerability*) yang dapat dieksploitasi oleh pihak tidak bertanggung jawab. Tujuannya adalah untuk menemukan dan mengatasi kelemahan sebelum digunakan oleh penyerang di dunia nyata. Menurut *NIST (National Institute of Standards and Technology*), penetration testing adalah metode pengujian keamanan yang menggunakan serangkaian langkah untuk mengevaluasi kerentanan sistem dengan mencoba mengeksploitasi kelemahan yang ada. Proses ini meniru teknik yang digunakan oleh penyerang dunia nyata untuk memberikan wawasan tentang risiko keamanan [7].

II.7 Acunetix

Acunetix adalah alat otomatisasi yang digunakan dalam proses *Vulnerability Assessment* dan *Penetration Testing* untuk mendeteksi berbagai kerentanan pada aplikasi web secara menyeluruh dan efisien. Alat ini mampu memindai lebih dari 7.000 jenis kerentanan, termasuk SQL Injection, Cross-Site Scripting (XSS), dan kelemahan pada konfigurasi keamanan seperti cookie insecure atau header yang hilang. Dengan fitur seperti

advanced crawler, *vulnerability verification*, dan *report generator*, Acunetix memudahkan proses identifikasi, analisis, dan pelaporan kerentanan secara sistematis. Selain itu, Acunetix mendukung integrasi ke dalam pipeline DevOps, sehingga cocok digunakan dalam pengujian keamanan berkelanjutan selama pengembangan aplikasi. Dalam konteks penelitian keamanan web, Acunetix sangat berguna sebagai alat pendukung untuk menemukan potensi celah sebelum dilakukan uji penetrasi manual lebih lanjut [8].

II.8 *Burp Suite*

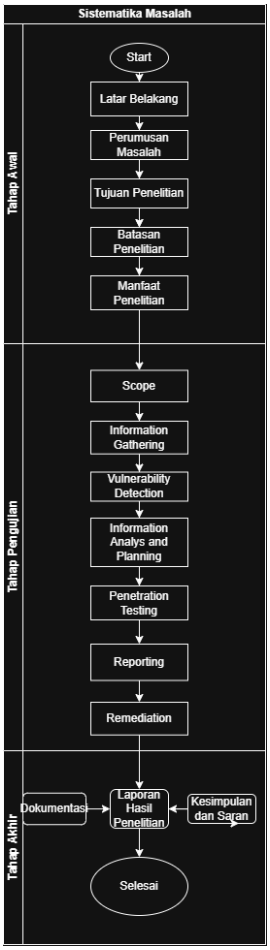
Burp Suite adalah alat profesional yang digunakan untuk pengujian keamanan aplikasi web (*web application security testing*). Dikembangkan oleh PortSwigger, Burp Suite menawarkan berbagai fitur untuk menganalisis, mengidentifikasi, dan mengeksploitasi kerentanan keamanan dalam aplikasi web. Alat ini banyak digunakan oleh *penetration tester* dan profesional keamanan karena fleksibilitas dan kemampuannya untuk menyesuaikan pengujian sesuai kebutuhan [9].

II.9 Kali Linux

Kali Linux adalah distribusi sistem operasi berbasis Linux yang dirancang khusus untuk kebutuhan keamanan informasi dan pengujian penetrasi (*penetration testing*). Kali Linux dikembangkan dan dikelola oleh Offensive Security, dan merupakan penerus dari distribusi BackTrack. Sistem operasi ini dilengkapi dengan berbagai alat keamanan canggih yang dapat digunakan untuk pengujian aplikasi web, analisis kerentanan, forensik digital, pengujian jaringan, dan eksploitasi keamanan

III. METODE

Penelitian ini menggunakan pendekatan *Vulnerability Assessment and Penetration Testing* (VAPT) untuk mengidentifikasi, mengevaluasi, dan memberikan solusi terhadap celah keamanan pada sebuah website internal milik perusahaan layanan teknologi informasi. Metodologi ini terbagi dalam beberapa tahapan utama:



Gambar 1 Sistematika penelitian

1. Tahap Awal

Penelitian dimulai dengan identifikasi kebutuhan untuk memahami atau menyelesaikan permasalahan tertentu. Langkah ini biasanya dipicu oleh adanya suatu permasalahan nyata yang terjadi di lingkungan sistem, misalnya kerentanan keamanan pada sebuah aplikasi atau website. Pada titik ini, peneliti mulai mencari alasan mengapa penelitian ini penting dilakukan dan bagaimana hasil penelitian dapat memberikan solusi.

2. Tahap Pengujian

Pada tahap ini, peniliti melakukan pengujian secara sistematis. Dengan menentukan ruang lingkup (Scope) dari pengujian yang akan dilakukan, setelah itu dilakukan pengumpulan informasi yang konkrit (Information gathering), kemudian setelah informasi telah terkumpul, selanjutnya penliti akan melakukan pendeteksian terhadap kerentanan yang ada (Vulnerability detection) terhadap objek yang dilakuakan penelitian, informasi yang telah didapatkan setelah itu dianalisis dan melakuakan persiapan terhadap langkah-langkah yang akan dilakukan berikutnya (Information analysis and planning). Proses yang akan dilakukan berikutnya adalah pengujian terhadap penetrasi dari website itu sendiri (Penetration Testing), dilakukan simulasi serangan untuk melakukan evaluasi terhadap keamanan sistem yang di uji, hasil dari pengujian tersebut didokumentasikan dalam sebuah laporan (Reporting), dan

setelah itu langkah yang terakhir adalah dilakukan perbaikan terhadap kerentanan yang ada (Remediation).

3. Tahap Akhir

Pada tahap akhir ini lebih berfokus pada hasil dari penelitian yang dilakukan dan penyusunan laporan yang berisikan proses dan temuan yang didapatkan selama dilakukan penelitian tersebut. Mengacu terhadap laporan tersebut, peneliti memberikan kesimpulan dan saran terhadap penelitian yang dilakukan sebelumnya untuk dilakukan peningkatan di masa yang akan datang. setelah semua proses selesai penelitian juga dianggap selesai.

IV. HASIL DAN PEMBAHASAN

Bagian ini menampilkan hasil dan proses yang dilakukan pada *Vulnerablity Testing and Penetration Testing* yang dilakukan terhadap *website X* perusahaan layanan teknologi informasi:

IV.1 *Scope*

Tahapan awal untuk melakukan *Vulnerability Assessment and Penetration Testing* (VAPT) yang di dalamnya berisikan penentuan dari ruang lingkup yang dibutukan untuk dilakukan penelitian, berbagai tools yang digunakan dalam perancangan pengujian dan pengujian dari *website* X perusahaan layanan teknologi informasi. Berbagai *tools* yang akan digunakan untuk melakukan pengujian ini adalah Nmap, Nessus dan Acunetix dengan menggunakan teknik grey box testing.

IV.2 Perancangan Pengujian

Dalam melakukan pengujian dari *website* X perusahaan layanan teknologi informasi ini ada berbagai *Software* dan *Hardware* yang dibutuhkan untuk mendukung kelancaran dalam melakukan proses pengujian ini. Berikut adalah berbagai spesifikasi yang digunakan untuk melakukan pengujian terhadap website X ini yang di jelaskan pada Tabel 1 dan Tabel 2.

Tabel 1 Perangkat Hardware

Nama Perangkat	Spesifikasi	
PC Rakit	Processor	I5-9400F (6 core 6 Thread)
	RAM	16 GB DDR4 2666 Mhz
	Storage	1 TB SSD
Lenovo Ideapad Gaming 3	Processor	Ryzen 5 5600H
	RAM	16 GB DDR4 2666Mhz
	Storage	512 GB SSD

Tabel 2 Perangkat Software

Nama Software	Versi	Fungsionalitas
Kali Linux	2025.1c	Sebagai Operating System yang digunakan untuk melakukan Vulnerability Assessment dan Penetration Testing
OWASP ZAP	2.16.0	Vulnerability Scan dan Penetration Testing
Nessus	10.8.4	Melakukan Vulnerability scanning
Windows	11 Pro	Main OS pada perangkat hardware yang digunakan
Virtual Box	7.1.8	Membuat perangkat yang sedang di Virtualisasikan
Acunetix	25.4.0	Melkaukan Vulnerability Scanning terhadap website
BurpSuite Community	2025.2.4	Melakukan Exploit penetration testing terhadap website

IV.3 Information Gathering

Tahapan ini menjelaskan proses pengumpulan informasi yang akan digunakan dalam kegiatan pengujian. Informasi dari web praktikum Fakultas Rekayasa Industri Telkom University akan dikumpulkan dengan bantuan tools untuk mempermudah dalam memperoleh data seperti nama domain, alamat IP, dan port yang digunakan. Alat bantu yang dimanfaatkan dalam proses ini adalah NMAP, hasil dari information gatherting bisa dilihat pada Tabel 3 dibawah ini.

Tabel 3 Hasil Information Gathering

No	Spesifikasi	Keterangan
1.	IP Address	192.168.0.110:3000
2.	Port	22,25,80,143,587,993,3000,10050,10051

IV.4 Vulnerability Detection dan Analysis

Tahap *Vulnerability Detection* merupakan bagian krusial dalam proses *Vulnerability Assessment* dan *Penetration Testing* karena pada tahap inilah dilakukan pencarian dan identifikasi berbagai celah keamanan (*security vulnerabilities*) yang mungkin terdapat dalam sistem target. Tujuan dari tahap ini adalah untuk mengumpulkan data terkait potensi kerentanan yang dapat dimanfaatkan oleh penyerang untuk mengakses atau merusak sistem, baik dari sisi konfigurasi, versi perangkat lunak, layanan yang berjalan, maupun celah-celah yang terdapat pada aplikasi web.

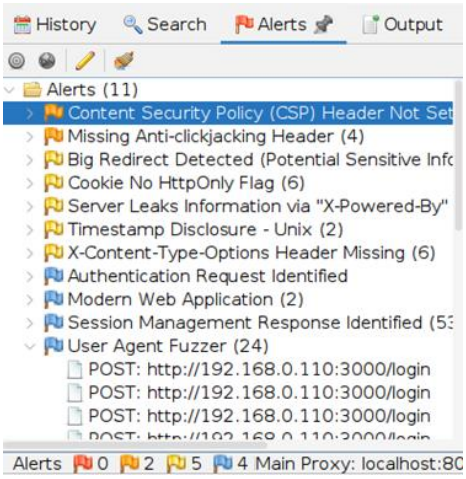
a) Pengujian menggunakan *OWASP ZAP*

OWASP ZAP digunakan untuk melakukan pemindaian terhadap sisi aplikasi web. ZAP merupakan

tools open-source yang dikembangkan oleh OWASP dan dirancang khusus untuk mendeteksi kerentanan pada aplikasi web, seperti *Cross-Site Scripting (XSS)*, *SQL Injection*, *Broken Authentication*, dan berbagai *OWASP Top 10 vulnerabilities* lainnya.

Tabel 4 Hasil Scanning *OWASP ZAP*

No	Severity	Judul Kerentanan
1	*High*	*SQL Injection*
2	*Medium*	*Content Security Policy (CSP) Header Not Set*
3	*Medium*	*Missing Anti-clickjacking Header*
4	*Low*	*Big Redirect Detected (Potential Sensitive Information Leak*
5	*Low*	*Cookie No HttpOnly Flag*
6	*Low*	*Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*
7	*Low*	*Timestamp Disclosure - Unix*
8	*Low*	*X-Content-Type-Options Header Missing*



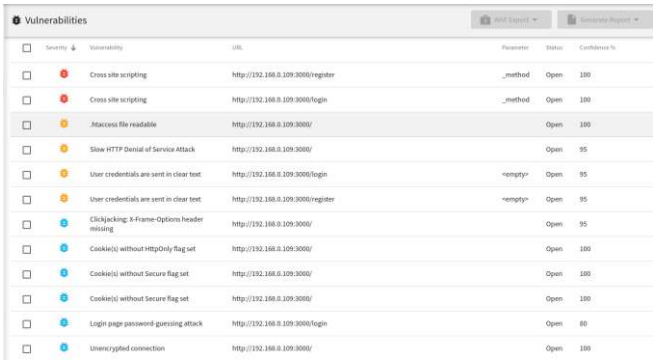
Gambar 2 Hasil Scanning OWASP ZAP

b) Pengujian menggunakan Acunetix

Acunetix merupakan salah satu alat pemindai kerentanan (Vulnerability Scanner) otomatis yang digunakan untuk mendeteksi berbagai celah keamanan pada aplikasi web. Dalam penelitian ini, acunetix digunakan untuk membantu proses pengujian kemanan pada website X perusahaan layanan teknologi informasi dengan menggunakan pemindaan secara menyeluruh terhadap potensi kerentanan yang ada. Acunetix memiliki berbagai tes pengujian yang ada di dalamnya, Seperti SQL Injection, Cross-Site Scripting (XSS), Remote file Inclusino (RFI), Directory Traversal, dan kelemahan dalam melakukan implementasi HTTPS/SSL selain itu juga, Acunetix dapat melakukan pendeteksian misconfigurasi server, kerentanan CMS (seperti WordPress, Joomla, dan Drupal). Di bawah ini adalah hasil *Vulnerability Scanning* yang dilakukan.

Tabel 5 Hasil Pengujian *Acunetix*

No	*Severity*	Judul Kerentanan
1	*High*	*Cross site scripting*
2	*High*	*Cross site scripting*
3	*Medium*	*.htaccess file readable*
4	*Medium*	*Slow HTTP denial of service attack*
5	*Medium*	*User credentials are sent in clear text*
6	*Medium*	*User credentials are sent in clear*
7	*Low*	*Clickjakcing: X-Frame-Options header missing*
8	*Low*	*Cookie(s) without HttpOnly flag set*
9	*Low*	*Cookie(s) without secure flag set*



Gambar 3 Hasil Scanning *Acunetix*

IV.5 *Attack and penetration Testing*

Suatu metode pengujian keamanan sistem informasi dengan cara melakukan simulasi serangan layaknya yang dilakukan oleh pihak yang tidak bertanggung jawab. Tujuannya adalah untuk mengidentifikasi celah keamanan (Vulnerabilities) yang terdapat pada sebuah sistem, aplikasi, atau jaringan, sehingga pihak pengelola dapat mengetahui sejauh mana sistem tersebut mampu bertahap dari potensi serangan siber. Berdasarkan hasil celah keamanan yang ditemukan menggunakan *tools OWASP ZAP* dan *Acunetix*, hasil dari temuan kedua *tools* tersebut akan dilakukan analiss terhadap *Evidence* yang ditemukan pada kedua *tools* tersebut pada pada *website*[12].

Tabel 5 Penggabungan Kerentanan

No	*Severity*	*Vulnerability*
1	*High*	*SQL Injection (Time-based)*
2	*High*	*Cross Site Scripting (XSS)*
3	*Medium*	*User Credentials Sent in Clear Text (HTTP)*
4	*Medium*	*.htaccess File Readable*
5	*Medium*	*Slow HTTP DoS (Slowloris Attack)*
6	*Medium*	*Missing Content Security Policy (CSP)*
7	*Medium*	*Missing Anti-clickjacking Header (X-Frame-Options)*
8	*Low*	*Cookie(s) without secure flag set*

9	*Low*	*Big Redirect Detected (Potential Sensitive Information Leak)*
10	*Low*	*Cookie No HttpOnly Flag*
11	*Low*	*Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*
12	*Low*	*Timestamp Disclosure - Unix*
13	*Low*	*X-Content-Type-Options Header Missing*

Berikut adalah penjelasan secara rinci dari masing-masing kerentanan yang akan diuji dalam tahap penetration testing:

a) SQL Injection (Time-based)
mengeksploitasi celah pada query SQL dengan menambahkan perintah tambahan yang membuat server menunggu dalam waktu tertentu (delay).

b) *Cross Site Scripting (XSS)*
XSS terjadi ketika aplikasi web tidak menyaring input dari pengguna secara benar, sehingga memungkinkan penyerang menyisipkan skrip berbahaya ke dalam halaman web yang dilihat oleh pengguna lain. Dampaknya bisa mencakup pencurian cookie, redirect, atau bahkan pengambilalihan akun.

c) *User Credentials Sent in Clear Text (HTTP)*
Kredensial pengguna (username dan password) dikirimkan tanpa enkripsi melalui protokol HTTP. Hal ini membuat data tersebut mudah disadap oleh pihak ketiga melalui teknik seperti man-in-the-middle (MITM) attack..

d) *.htaccess File Readable*
File .htaccess yang seharusnya hanya dibaca oleh server dapat diakses oleh pengguna dari browser. File ini biasanya berisi aturan keamanan penting, dan keterbukaannya dapat memberikan informasi sensitif kepada penyerang.

e) *Slow HTTP DoS (Slowloris Attack)*
Website rentan terhadap serangan Slowloris, yaitu jenis serangan DoS yang mengirimkan permintaan HTTP secara perlahan dan tidak selesai, membuat server tetap membuka koneksi dan akhirnya kehabisan sumber daya untuk pengguna sah.

f) *Missing Content Security Policy (CSP)*
Tidak adanya header CSP menyebabkan browser tidak memiliki petunjuk dalam membatasi sumber daya yang dapat dimuat. Hal ini membuat aplikasi lebih rentan terhadap serangan XSS dan injeksi konten dari sumber yang tidak dipercaya.

h) *Missing Anti-clickjacking Header (X-Frame-Options)*
Absennya header *X-Frame-Options* memungkinkan halaman web ditampilkan di dalam *iframe*, sehingga membuka celah bagi serangan *clickjacking*, di mana pengguna diarahkan untuk mengklik elemen yang tidak mereka sadari.

i) *Cookie(s) without secure flag set*
Cookie dikirim tanpa flag Secure, sehingga tetap dikirim meskipun koneksi tidak aman (HTTP). Hal ini membuka risiko pencurian cookie saat lalu lintas data disadap.

j) *Big Redirect Detected (Potential Sensitive Information Leak)*
Terdapat redirect besar (dalam jumlah data atau jumlah hop) yang dapat dimanipulasi untuk mengarahkan pengguna atau pihak ketiga ke situs tertentu. Jika tidak dikontrol, ini bisa menyebabkan kebocoran informasi atau phishing.

k) *Cookie No HttpOnly Flag*
Cookie yang tidak memiliki flag *HttpOnly* bisa diakses melalui *JavaScript* di *browser*. Hal ini memungkinkan skrip berbahaya (seperti XSS) mencuri informasi *cookie*, termasuk token sesi.

l) *Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*
Header X-Powered-By mengungkapkan teknologi yang digunakan oleh server (misalnya PHP, ASP.NET). Informasi ini bisa dimanfaatkan penyerang untuk menyusun serangan yang sesuai dengan teknologi tersebut.

m) *Timestamps Disclosure – Unix*
Server mengungkapkan timestamp dalam format Unix dalam respons HTTP. Meskipun berdampak kecil, informasi ini bisa dimanfaatkan untuk fingerprinting atau mengetahui waktu pembuatan dan modifikasi aplikasi.

n) *X-Content-Type-Options Header Missing*
Tidak adanya header *X-Content-Type-Options: nosniff* memungkinkan browser mencoba menebak tipe konten dari sebuah respons, yang bisa menyebabkan eksekusi file yang seharusnya tidak dijalankan. Ini bisa membuka celah XSS atau serangan konten berbahaya.

Sebelum dilakukan mitigasi kita harus menguji terlebih dahulu apakah kerentanan yang ditemukan di sana benar-benar terletak pada website yang dilakukan *testing* atau bisa jadi kerentanan tersebut adalah false positif*Tabel* 6.

Tabel 6 Hasil Penetration Testing

Jenis kerentanan	Hasil Penetration Testing
SQL Injection (Time Based)	Jenis kerentanan keamanan ini merupakan jenis kerentantan yang paling umum ditemukan pada sebuah website dan cukup berbahaya. kerentanan ini ditemukan oleh tools OWASP ZAP, dapat dieksploitasi dengan menyisipkan perintah SQL berbahaya pada input formulir yang tidak divalidasi dengan baik, seperti ' OR IF(SLEEP(5),1,0)--, yang menyebabkan server menunggu beberapa detik sebelum merespons, menandakan adanya

	kerentanan. Untuk kasus di sini dengan website X perusahaan layanan tekonologi informasi, setelah dilakukan exploit terhadap website tersebut tidak menandakan adanya SQL Injection (Time Based. bisa di simpulkan hasil evidence yang diberikan oleh OWASP ZAP adalah false positif.
Cross-Site-Scripting (XSS)	Temuan ini ditemukan dengan tools Acunetix, ditemukan bahwa website ini tidak memiliki proteksi terhadap Input/Output pengguna, Cross Site Scripting (XSS) dieksploitasi dengan menyisipkan skrip JavaScript berbahaya ke dalam input pengguna, seperti <script>alert(1)</script>, yang kemudian dijalankan di browser korban ketika halaman dimua. Tetapi di temukan di sini bahwa response yang ada tidak reflected, sehingga tidak bisa dilakukan Cross-Site-Scripting (XSS).
User Credentials Sent in Clear Text (HTTP)	Temuan ini ditemukan dengan tools Acunetix, pada tools tersebut Menunjukan bahwa website tidak menerapkan lapisan enkripsi seperti HTTPS/SSL/TLS, serta tidak memiliki pengaturan keamanan untuk memastikan bahwa informasi sensitif seperti username dan password dikirimkan secara aman. Hal ini membuat kredensial pengguna sangat rentan terhadap penyerangan dan pencurian data pada website tersebut. Di bawah ini terbukti bahwa vulnerability ini benar-benar ada di website X perusahaan layanan teknologi informasi.
.htaccess	File konfigurasi yang digunakan pada server web Apache untuk mengatur berbagai aturan secara lokal pada direktori tertentu. Kerentanan ini ditemukan pada website ini melalui Acunetix, .htaccess File Readable memungkinkan penyerang mengakses file konfigurasi .htaccess melalui browser, seperti dengan membuka http://login/.htaccess, yang bisa mengungkap aturan keamanan atau jalur direktori sensitif. Di sini kita dapat melihat dapat dilakukan penglihatan terhadap file .htaccess setelah dilakukan exploit.
Slow HTTP Dos (Slowris Attack)	Kerentanan ini ditemukan dengan tools Acunetix, di sini ditemukan evidence server web ini tidak memiliki pengaturan timeout yang tepat, pembatasan koneksi per IP, serta tidak dilengkapi dengan zsistem proteksi seperti WAF, reverse proxy, atau modul keamanan server. Penyerang dapat mengeksploitasi sumber daya server secara perlahan hingga layanan menjadi tidak responsif (DoS) Denial of Service. Di bawah ini adalah hasil dari eksploitasi yang dilakukan menggunalan slowhttptest dan di sini terbukti bahwa adanya kerentana dari Slow HTTP Dos.
Missing Content Security Policy (CSP)	Temuan ini ditemukan dengan tools OWASP ZAP,pada temuanya menunjuka bahwa website tidak memiliki mekanisme pembatasan sumber daya dan eksekusi skrip di sisi browser, yang seharusnya mencegah serangan seperti XSS. Hal ini tidak berarti tidak adanya header keamanan yang penting, serta tidak ada kontrol atas asal dan jenis konten yang dijalankan oleh halaman web. Kita dapat lihat di bawah ini bahwa cookie yang ada memang tidak memiliki Content-Security-Policy pada cookie yang tersedia sehingga website ini tidak membatasi sumber daya apa yang boleh dimuat oleh browser. Gambar di bawah adalah bukti Content-Security-Policy
Missing-Content-Security-Policy	Berdasarkan hasil pengujian menggunakan devtools terhadap situs fripraktikum.my.id, ditemukan bahwa *header* keamanan *Content-Security-Policy* (CSP) tidak diterapkan. Header ini

	berfungsi untuk mencegah browser memuat dan mengeksekusi konten dari sumber yang tidak dipercaya. Tanpa \
Missing Anti-clickjacking Header (X-Frame-Options)	Temuan ini ditemukank oleh kedua tools tersebut, keduanya menemukan bahwa website tidak memiliki mekanisme proteksi terhadap serangan clickjacking, karena tidak membatasi apakah konten web boleh dimuat dalam iframe oleh pihak luar. Hal ini bisa menyebabkan pengguna ditipu untuk melakukan tindakan yang tidak disengaja melalui antarmuka yang dimanipulasi oleh penyerang. Di bawah ini adalah hasil dari Exploit dari kerentanan ini, yang menunjukan bahwa website ini dapat dilakukan clickjacking.
Cookie(s) without secure flag set	Temuan ini ditemukan oleh OWASP ZAP, ditemukan bahwa website ini tidak memiliki cookie yang dikirim tidak melalui koneksi yang tidak ter-enkripsi. Karena di website ini tidak menggunakan HTTPS sudah dapat dipastikan bahwa website ini tidak memiliki Secure flag set pada cookienya.
Big Redirect Detected (Potential Sensitive Information Leak)	Hasil kerentanan ini yang ditemukan dengan OWASP ZAP ini adalah sebuah kerentanan yang dapat memberitahu informasi yang krusial melalui URL atau Referer header
Cookie No HttpOnly Flag	Kerentanan ini ditemukan dengan OWASP ZAP, Cookie No HttpOnly Flag adalah kerentanan yang terjadi ketika cookie yang dikirim oleh server tidak memiliki atribut HttpOnly, sehingga cookie tersebut dapat diakses melalui JavaScript di sisi klien. Hal ini berbahaya karena jika terjadi serangan Cross-Site Scripting (XSS), penyerang dapat dengan mudah membaca nilai cookie, termasuk cookie sesi (session cookie), lalu menggunakannya untuk mengambil alih akun korban (session hijacking).
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Server Leaks Information via "X-Powered-By" HTTP Response Header adalah kerentanan di mana server web mengungkapkan informasi teknologi atau framework yang digunakan, seperti versi PHP, ASP.NET, atau Express.js, melalui header HTTP X-Powered-By.
Timestamp Disclosure – Unix	Kerentanan ini ditemukan oleh tools OWASP ZAP, Timestamp Disclosure adalah kerentanan yang tidak terlalu berbahaya. Biasanya penyerang akan menggunakan informasi ini untuk waktu pembuatan akun, sesi login, waktu kompilasi sistem, atau aktivitas lainnya.
X-Content-Type-Options Header Missing	Kerentanan yang ada di sini adalah kerentanan yang ditemukan oleh tools oleh OWASP ZAP, Vulnerability X-Content-Type-Options Header Missing adalah celah keamanan yang terjadi ketika suatu website tidak menyertakan header HTTP X-Content-Type-Options: nosniff pada responsnya. Tanpa header ini, browser dapat melakukan MIME sniffing, yaitu menebak tipe konten dari isi file, bukan berdasarkan header Content-Type yang diberikan server. Hal ini dapat dimanfaatkan oleh penyerang untuk menyisipkan kode berbahaya (seperti JavaScript) dalam file yang terlihat aman, misalnya .txt atau .csv, sehingga memungkinkan terjadinya serangan Cross-Site Scripting (XSS) saat file tersebut diakses oleh pengguna. Untuk mencegahnya, server harus selalu menyertakan header X-Content-Type-Options: nosniff dalam setiap respons HTTP.

IV.6 *Remediation*

Pada tahap ini dilakukan perancangan strategi mitigasi terhadap kerentanan yang telah teridentifikasi pada tahap sebelumnya. Setelah itu, implementasi mitigasi diterapkan pada sistem, disertai dengan pengujian ulang guna memastikan bahwa kerentanan tersebut telah berhasil dimitigasi, akan dilakukan analisis lanjutan untuk mengidentifikasi penyebab serta kemungkinan solusi yang dapat diterapkan [10].

a) Perancangan Mitigasi

Dari hasil pengujian yang dilakukan di atas terdapat beberapa kerentanan yang tidak ditemukan di dalam website yang ada, atau bisa di sebut *false positif* , sehingga beberapa kerentanan tersebut tidak perlu dilakukan mitigasi, ada beberapa rekomendasi yang disarankan untuk beberapa kerentanan yang di tulis pada tabel 7.

Tabel 7 Rekomendasi Mitigasi Kerentanan

Jenis Kerentanan	Rekomendasi
User Credentials Sent in Clear Text (HTTP)	Pada kerentanan User Credentials Sent in Clear Text (HTTP), kerentanan ini tidak dapat dilakukan mitigasi karena website intranet ini tidak menggunakan HTTPS sebagai protocol securitynya, sehingga Vulnerability ini tidak bisa dilakukan mitigasi.
.htaccess	Pada kerentanan .htaccess, di sini dilakukan konfigurasi terhadap file .htaccess yang berisikan seperti di bawah ini untuk melakukan blocking terhadap penyerang yang ingin masuk ke dalam file .htaccess, tambahkan kode ini <FilesMatch "\.(env|json|log|git|ini|phps|bak|old|sql)$"> Order allow,deny Deny from all </FilesMatch> di dalam file .htaccess.
Slow HTTP DoS (Slowloris Attack)	mengaktifkan SSL/TLS agar situs dapat menggunakan protokol HTTPS yang aman dan terenkripsi. Tanpa SSL, data seperti *username* dan *password* akan dikirim secara jelas melalui jaringan, sehingga mudah disadap. Setelah SSL terpasang dengan benar, sangat penting untuk mengatur *redirect* otomatis dari HTTP ke HTTPS menggunakan *file* pengaturan cpanel dan konfigurasi pada *.env.*
Missing Content Security Policy (CSP)	Melakukan penambahan file ContenctSecurityPolicy.php untuk menambakan header menggunakan middlerware, lalu di daftarkan pada kernel.php.
Missing Anti-clickjacking Header (X-Frame-Options)	Melakukan penambahan file XframeOptions.php pada middleware untuk dapat melakukan anti clickjacking secara otomatis
Cookie(s) without secure flag set	Melakukan konfigurasi pada session.php untuk agar menjaga cookie yang ada tetap secure dengan menambahkan command 'true'
Big Redirect Detected (Potential Sensitive Information Leak)	Yang pertama Gunakan session atau post data pada setiap controller untuk menghindari mengirimkan informasi sensitif melalui URL, Batasi dan validasi URL redirect (Open Redirect protection), Jangan expose token dalam URL setelah login atau reset password, terakhir gunakan middleware atau eader untuk cegah redirect
Cookie No HttpOnly Flag	Sama seperti vulnerability Cookie(s) no secure flag, melakukan konfigurasi terhadap session.php untuk keamanan lebih dapat melakukan konfigurasi pada http_only dan same_site
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Konfigurasi file .htaccess Pada header X-Powered-By lalu lakukakn unset pada X-Powered-By
BREACH Attack Potential	Menambahkan middleware khusus yang menyisipkan random padding pada respons HTML. Middleware ini bekerja dengan menambahkan karakter acak di bagian akhir respons HTML untuk membuat ukuran respons menjadi tidak konsisten. Teknik ini bertujuan mengacaukan analisis ukuran data yang digunakan dalam serangan BREACH, karena attacker tidak lagi bisa membandingkan ukuran response untuk menebak isi data sensitif.
Timestamp Disclosure - Unix	Melakukan konfigurasi terhadap file .htaccess agar web dapat menyembunyikan waktu modifikasi, dan menghapus informasi Etag yang juga bisa memuat waktu modifikasi file
X-Content-Type-Options Header Missing	Melakukan konfigurasi terhadap .htaccess lalu konfigurasi terhadap header untuk X-Content-Type-Options:; public function handle($request, Closure $next) { $response = $next($request); $response->headers->set('X-Content-Type-Options', 'nosniff'); return $response; }

Dari setiap rekomendasi dan mitigasi yang dilakukan pada Tabel 7. Akan dilakukan pengujian ulang setelah mitigasi untuk dapat melihat bagaimana kerentanan tertutup atau tidak setelah dilakukan mitigasi.

b) Pengujian Ulang Pasca Mitigasi

Setelah dilakukan rekomendasi dan mitigasi di tahap sebelumnya, dilakaukan pengetesan ulang pasca mitigasi, di bawah ini adalah hasil dari pengetesan ulang yang dilakukan setelah mitigasi.

Tabel 8 Hasil Pengujian Pasca Mitigasi

No	*Vulnerability Scanning* Sebelum Mitigasi	*Vulnerability Scanning* Setelah Mitigasi	Keterangan
1	*User Credentials*	*User Credentials*	Belum berhasil

	Sent in Clear Text (HTTP)	*Sent in Clear Text (HTTP)*	tertutup dan diberikan rekomendasi kepada *developer* terkait. untuk menggunakan protokol HTTPS
2	*.htaccess File Readable*	-	Kerentanan Berhasil tertutup
4	*Missing Content Security Policy (CSP)*	-	Kerentanan Berhasil tertutup
5	*Missing Anti-clickjacking Header (X-Frame-Options)*	-	Kerentanan Berhasil Tertutup
6	*Cookie(s) without secure flag set*	*Cookie(s) without secure flag set*	Belum berhasil tertutup karena harus menggunakan protokol HTTPS pada server yang dijalankan
7	*Big Redirect Detected (Potential Sensitive Information Leak)*	*Big Redirect Detected (Potential Sensitive Information Leak)*	Belum berhasil tertutup dan diberikan rekomendasi kepada *developer* terkait
8	*Cookie No HttpOnly Flag*	-	Belum berhasil tertutup karena harus menggunakan protokol HTTPS pada server yang dijalankan
9	*Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*	-	Kerentanan Berhasil Tertutup
10	*Timestamp Disclosure - Unix*	*Timestamp Disclosure - Unix*	Belum berhasil tertutup karena harus melakukan konfigurasi terhadap session pada kodingan yang dimiliki oleh websiter tersebut
11	*X-Content-Type-Options Header Missing*	-	Kerentanan Berhasil tertutup

Berdasarkan hasil pengujian setelah dilakukan mitigasi, terlihat ada beberapa kerentanan yang berhasil tertutup dan ada beberapa kerentanan yang tidak berhasil dilakukan penutupan. Sehingga hanya bisa dilakukan rekomendasi terhadap *developer* yang akan melakukan perbaikan terhadap websiter tersebut.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa pendekatan Vulnerability Assessment dan Penetration Testing (VAPT) secara efektif mampu mengidentifikasi dan mengevaluasi berbagai jenis kerentanan pada website X milik perusahaan layanan teknologi informasi. Berdasarkan hasil pengujian menggunakan OWASP ZAP dan Acunetix, ditemukan 13 jenis kerentanan dengan tingkat keparahan yang bervariasi, mulai dari SQL Injection, Cross-Site Scripting (XSS), hingga kelemahan konfigurasi seperti tidak adanya header keamanan dan pengiriman kredensial melalui HTTP.

Setelah dilakukan proses mitigasi, beberapa kerentanan berhasil ditutup, seperti .htaccess File Readable, Missing CSP Header, Missing X-Frame-Options, X-Powered-By Disclosure, dan X-Content-Type-Options Missing. Namun, masih terdapat beberapa kerentanan yang belum dapat ditutup, terutama karena keterbatasan lingkungan pengujian, seperti penggunaan HTTP tanpa dukungan

REFERENSI

[1] ISO/IEC 27001, *Information technology — Security techniques — Information security management systems — Requirements*, ISO, 2013.
[2] Mendy, "Pengertian Website: Apa itu Web, Manfaat, Jenis, dan Contoh," *Kampus IT*, 7 Aug. 2023. [Online]. Available: https://kampusit.id/pengertian-website
[3] D. Sari and R. Nugroho, *Keamanan Sistem Informasi: Teori dan Implementasi*, Yogyakarta, Indonesia: Deepublish, 2020.
[4] OWASP Foundation, "OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks," *Open Web Application Security Project*, 2021. [Online]. Available: https://owasp.org/Top10/
[5] S. K. Rakshit, *Ethical Hacker's Penetration Testing Guide*, New Delhi, India: BPB Publications, 2022.

[6] A. Orebaugh and B. Pinkard, *Nmap in the Enterprise: Your Guide to Network Scanning*, Rockland, MA: Syngress Publishing, 2008.
[7] R. Rogers, *Nessus Network Auditing*, Rockland, MA: Syngress Publishing, 2011.
[8] G. Kusuma, "Implementasi OWASP ZAP untuk Pengujian Keamanan Sistem Informasi Akademik," *Jurnal Teknologi Informasi*, vol. 9, no. 1, pp. 15–22, 2022.
[9] T. Muhyidin and U. S. Utama, "Perbandingan Tingkat Keamanan Website Menggunakan Nmap dan Nikto dengan Metode Ethical Hacking," *Jurnal Teknik Logika Matematika*, vol. 5, no. 2, pp. 55–63, 2022.
[10] Y. Khera, D. Kumar, Sujay, and N. Garg, "Analysis and Impact of Vulnerability Assessment and Penetration Testing," in *Proc. IEEE Int. Conf. Computing, Communication, and Automation*, Greater Noida, India, 2019, pp. 100–104.
[11] D. R. Mulyawan and J. Benjamin, "Penetration Testing and Vulnerability Scanning of Web Application Using Burp Suite," in *Natl. Conf. Emerging Computing Applications*, 2021.
[12] J. I. Firmansyah, R. Anggraini, and M. E. A. W. Kuncoro, "Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website: Literature Review," *AUTOMATA: Jurnal Teknik dan Sistem Komputer*, vol. 3, no. 1, pp. 34–41, 2021.