

# Pengujian Keamanan Situs *Web* Praktikum Fakultas Rekayasa Industri Pada Telkom University Dengan Pendekatan *Vulnerability dan Penetration Testing*

1<sup>st</sup> Dio Alif Ananda  
Sistem Informasi  
Telkom University  
Bandung, Indonesia  
liffdio@student.telkomuniversity.ac.id

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

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

**Abstrak**—*Web telah menjadi sarana penting dalam penyampaian informasi bagi organisasi, termasuk institusi pendidikan. Namun, kemudahan akses ini disertai dengan risiko kerentanan keamanan yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab. Penelitian ini bertujuan untuk menganalisis tingkat keamanan pada website praktikum Fakultas Rekayasa Industri Telkom University dengan pendekatan Vulnerability Assessment and Penetration Testing (VAPT). Metode ini terdiri dari tahap information gathering, vulnerability detection, penetration testing, dan remediation. Pengujian dilakukan menggunakan beberapa tools keamanan seperti NMAP, Nessus, OWASP ZAP, dan Nikto. Hasil pengujian menunjukkan sejumlah kerentanan pada sistem, seperti tidak adanya header keamanan, konfigurasi cookie yang tidak aman, dan potensi serangan clickjacking. Rekomendasi mitigasi kemudian diterapkan untuk menutup celah-celah tersebut, diikuti dengan pengujian ulang mengvaluasi efektifitas perbaikan. Penelitian ini menghasilkan rekomendasi peningkatan sistem keamanan yang dapat digunakan sebagai acuan dalam pengelolaan keamanan website berbasis akademik dan institusional*

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

## I. PENDAHULUAN

Pada zaman sekarang ini, segala informasi harus memiliki akses yang lebih cepat untuk di infokan ke semua orang. Perusahaan maupun suatu organisasi menghadirkan layanan informasi secara online yang mana bisa lebih cepat di akses semua orang. Salah satu layanan informasi secara online yaitu adalah Web. Web merupakan suatu halaman yang mana didalamnya berisikan informasi penting suatu Perusahaan, perorangan maupun organisai. Dengan penggunaan web ini mempermudah perusahaan maupun organisasi untuk menjalankan bisnis mereka, meningkatkan citra dan kredibilitas, kemudahan dalam pembaruan informasi dan keunggulan kompetitif [1]. Dengan manfaat yang banyak didapatkan melalui penggunaan web, ada kekurangan yang muncul dalam pemakaian web ini. Sering terjadi pembobolan atau mencuri data oleh orang yang tidak bertanggung jawab. Oleh sebab itu, akan muncul kerentanan terhadap web itu sendiri dan akan merugikan yang punya web tersebut .

*Vulnerability* merupakan kerentanan dalam sebuah sistem atau jaringan yang akan menyebabkan pihak yang tidak berwenang mendapatkan akses tanpa izin. Celah keamanan ini dapat timbul akibat pengelolaan keamanan yang kurang optimal dari pihak pengembang situs atau adanya bug yang

belum terdeteksi dan diperbaiki. Apabila kerentanan pada situs tidak segera ditangani, maka risiko terkena serangan siber meningkat, yang dapat menyebabkan kerugian bagi perusahaan atau organisasi .

Dalam melakukan *testing* pada *web* untuk melihat keamanannya maka diperlukan metode *Vulnerability Assesment and Penetration testing* (VAPT). Dengan menggunakan VAPT sebagai teknologi pertahanan *siber*, maka dapat menghapus kerentanan pada sistem dan mengurangi kemungkinan serangan *siber* [2]. Setelah melakukan pengujian, akan diperoleh laporan mengenai tingkat kerentanan pada *web* tersebut, yang kemudian menjadi dasar untuk menentukan langkah mitigasi dalam menutup kerentanan yang teridentifikasi.

Metode VAPT ini akan diujikan pada *web* praktikum Fakultas Rekayasa Industri untuk mengetahui bagaimana keamanan dari data yang disimpan dalam *web* tersebut, untuk menemukan dimana letak kerentanan *web* praktikum Fakultas Rekayasa Industri itu sendiri. Setelah menemukan masalahnya, mitigasi dapat dilakukan untuk menutup celah keamanan yang ada dan menguji ketahanan keamanan *website* terhadap serangan terkait kerentanan yang telah ditemukan.

*Web* telah menjadi alat penting bagi perusahaan dan organisasi untuk menyampaikan informasi dengan cepat, meski rentan terhadap ancaman seperti pembobolan data akibat celah keamanan. Data dari BSSN menunjukkan tingginya kasus peretasan di Indonesia, menekankan perlunya langkah pencegahan [3]. VAPT menjadi solusi efektif untuk mengidentifikasi dan menutup celah keamanan, sehingga risiko serangan dapat diminimalkan. Pengujian ini akan diterapkan pada *web* praktikum Fakultas Rekayasa Industri untuk menganalisis kerentanan, memberikan rekomendasi mitigasi, dan meningkatkan ketahanan sistem terhadap ancaman siber.

## II. KAJIAN TEORI

### II.1 Keamanan Informasi

Keamanan informasi adalah suatu mekanisme yang tepat digunakan untuk memastikan perlindungan data [4]. Perlindungan terhadap data dari ancaman seperti pencurian, manipulasi, atau perusakan sangat penting untuk menjaga integritas, kerahasiaan, dan ketersediaan informasi. Didalam keamanan Informasi juga dikenal sebagai *CIA* yaitu *Confidentiality* (kerahasiaan), *Integrity* (integritas), dan *Availability* (ketersediaan), Unsur-unsur dari tiga serangkai

tersebut dianggap tiga komponen yang paling penting dari system keamanan [4].

## II.2 Web

*Web* adalah kumpulan halaman yang di dalamnya memuat suatu informasi dan dapat diakses melalui jaringan *internet* oleh siapa saja, kapan saja, dan di mana saja [1]. Banyak sekali manfaat yang dihasilkan dari *web* itu sendiri yaitu sebagai platform bisnis, sarana promosi dan untuk menambah wawasan

## II.3 Vulnerability

Proses sistematis untuk mengidentifikasi, mengukur, dan memprioritaskan kelemahan yang ada dalam sistem, jaringan, atau aplikasi. Tujuan utama dari *assessment* ini adalah untuk menemukan potensi ancaman keamanan sebelum mereka dapat dimanfaatkan oleh aktor jahat. Proses ini biasanya menghasilkan laporan lengkap yang memuat semua kerentanan yang ditemukan, lengkap dengan tingkat keparahan dan rekomendasi mitigasi [5].

## II.4 Penetration

Simulasi serangan yang dilakukan oleh seorang *ethical hacker* untuk mengevaluasi keamanan sistem atau aplikasi secara mendalam. *Penetration testing* tidak hanya bertujuan untuk menemukan celah keamanan, tetapi juga untuk mengeksploitasi kelemahan tersebut, dengan tujuan memahami seberapa besar risiko yang bisa ditimbulkan dari serangan yang sebenarnya. Proses ini lebih praktis dan mendalam daripada *Vulnerability assessment*, karena bertujuan untuk memvalidasi seberapa efektif pertahanan yang ada [5].

## II.5 Vulnerability assessment and Penetration testing (VAPT)

*Vulnerability assessment dan Penetration testing (VAPT)* adalah metode yang digunakan untuk mengidentifikasi dan menguji kerentanannya dalam sistem informasi dengan tujuan meningkatkan keamanan dengan mengeksploitasi celah-celah yang ada dan memberikan rekomendasi mitigasi [2].

## II.5 NMAP

*NMAP (Network Mapper)* adalah alat sumber terbuka yang digunakan oleh administrator jaringan dan profesional keamanan *IT* untuk memindai jaringan korporat. Fungsinya meliputi identifikasi host aktif, layanan yang berjalan, dan sistem operasi yang digunakan dalam jaringan. Keunggulan *NMAP* terletak pada kemampuannya untuk membuat paket *IP* dari awal dan mengirimkannya menggunakan metode unik untuk melakukan berbagai jenis pemindaian [6].

## II.6 Nessus

*Nessus* didefinisikan sebagai alat penilaian kerentanan sumber terbuka terkemuka yang digunakan untuk mengidentifikasi kelemahan keamanan dalam sistem dan jaringan komputer. *Nessus* memungkinkan pengguna untuk melakukan pemindaian terhadap berbagai perangkat dan layanan, mendeteksi kerentanan seperti kesalahan konfigurasi, patch keamanan yang hilang, dan kelemahan lainnya yang dapat dieksploitasi oleh penyerang [7].

## II.7 OWASP ZAP

*Zed Attack Proxy (ZAP)* adalah aplikasi untuk melakukan *pentest* untuk menemukan *vulnerabilities* dalam suatu *web applications* dengan cara mudah, *ZAP* menyediakan *scanner* otomatis sebaik bila kita menggunakan *tools* untuk menemukan *vulnerabilities* secara manual. Ketika digunakan sebagai *server proxy*, ini memungkinkan pengguna untuk memanipulasi semua lalu

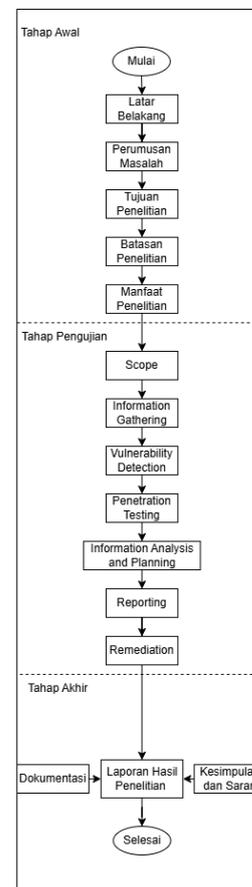
lintas yang melewatinya, termasuk lalu lintas menggunakan *https*, itu juga dapat berjalan dalam mode daemon yang kemudian dikontrol melalui *REST API*. *ZAP* telah ditambahkan ke dalam Radar Teknologi *ThoughtWorks* pada 30 Mei 2015 di cincin Percobaan. *ZAP* awalnya bercabang dari Paros, proxy pentesting lainnya. Simon Bennetts, pemimpin proyek, menyatakan pada tahun 2014 bahwa hanya 20% dari kode sumber *ZAP* masih dari Paros [8].

## II.8 Nikto

*Nikto* adalah sebuah webserver dan sekaligus alat untuk penilaian aplikasi web untuk menemukan kerentanan yang ada pada website tersebut, yang mana dapat mengidentifikasi ancaman sebelum adanya risiko peretasan dari serangan siber yang tidak bertanggung jawab [9].

## III. METODE

Penelitian ini menggunakan pendekatan yang terstruktur dan sistematis untuk menjelaskan tahapan-tahapan dalam menyelesaikan penelitian. Tahapan tersebut mencakup langkah-langkah yang mendeskripsikan proses pelaksanaan VAPT, yang terdiri dari tiga tahap utama, yaitu tahap awal, tahap pengujian, dan tahap akhir seperti pada Gambar 1 dibawah ini:



Gambar 1 Sistematika penelitian

### 1. Tahap Awal

Pada tahap awal ini dimulai dengan proses mendefinisikan latar belakang penelitian untuk memberikan konteks dan alasan dilakukannya penelitian. Selanjutnya, dilakukan perumusan masalah yang ingin diselesaikan, diikuti oleh penentuan tujuan penelitian. Setelah itu, batasan penelitian ditentukan untuk memperjelas ruang lingkup studi yang

dilakukan. Tahap ini diakhiri dengan mengidentifikasi manfaat yang dapat diperoleh dari penelitian yang dilakukan.

## 2. Tahap Pengujian

Pada tahap pengujian ini, proses pengujian dilakukan secara sistematis. Dimulai dengan menentukan ruang lingkup (*scope*) pengujian, kemudian dilanjutkan dengan pengumpulan informasi (*information gathering*) yang relevan. Setelah informasi terkumpul, dilakukan pendeteksian kerentanan (*vulnerability detection*) pada objek yang diteliti. Informasi tersebut kemudian dianalisis dan direncanakan untuk langkah-langkah yang akan dilakukan berikutnya (*information analysis and planning*). Proses berikutnya adalah pengujian penetrasi (*penetration testing*), dimana simulasi serangan dilakukan untuk mengevaluasi keamanan sistem. Hasil pengujian tersebut didokumentasikan dalam laporan (*reporting*), dan langkah perbaikan (*remediation*) diambil untuk menangani kerentanan yang ditemukan

## 3. Tahap Akhir

Tahap akhir ini difokuskan pada dokumentasi hasil penelitian dan penyusunan laporan lengkap tentang proses dan temuan penelitian. Berdasarkan laporan tersebut, peneliti memberikan kesimpulan dan saran sebagai rekomendasi untuk peningkatan di masa mendatang. Penelitian dianggap selesai setelah seluruh proses ini tuntas.

# IV. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari penelitian yang telah dilakukan, sekaligus membahasnya secara menyeluruh. Penjelasan bisa dilengkapi dengan gambar dan tabel, atau elemen lain yang membantu pembaca lebih mudah memahami isi penelitian ini. Jika pembahasannya cukup panjang, penulis bisa membaginya ke dalam sub-subjudul agar lebih terstruktur, seperti contoh berikut:

### IV.1 Scope

*Scope* merupakan suatu tahapan yang mana digunakan sebelum melakukan vulnerability assessment [10]. Pada tahapan ini juga melakukan rancangan pengujian untuk mendapatkan informasi kerentanan yang ada pada web praktikum Fakultas Rekayasa Industri dengan menggunakan tools OWASP ZAP, Nessus dan Nikto pada port 443 yaitu HTTPS.

### IV.2 Perancangan Sistem

Dalam proses pengujian terhadap web praktikum milik Fakultas Rekayasa Industri, dibutuhkan perangkat keras dan perangkat lunak yang memadai untuk mendukung kelancaran kegiatan pengujian. Oleh karena itu, berikut ini disajikan spesifikasi hardware dan software yang digunakan selama pelaksanaan pengujian tersebut pada Tabel 1 dan Tabel 2.

Tabel 1 Perangkat Hardware

Nama Perangkat	Spesifikasi
Asus ROG Strix G513IH	Processor AMD Ryzen 7 4800H 4.2 GHz

	RAM	16 GB RAM 3200 DDR4
	Storage	SSD 512 GB

Tabel 2 Perangkat Software

Nama software	Versi	Fungsionalitas
Windows	10 Home	Main OS pada perangkat Hardware
VMware	16.1.2	Membuat perangkat virtualisasi
Rocky Linux	8.10	Main OS pada perangkat virtual
Nessus	10.8.4	Vulnerability scan
OWASP ZAP	2.16.1	Vulnerability scan dan Penetration Testing
NMAP	7.92	Information gathering
Nikto	2.5.0	Vulnerability scan

### 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 gathering bisa dilihat pada Tabel 3 dibawah ini.

Tabel 3 Hasil Information Gathering

No	Spesifikasi	Keterangan
1	Nama Domain	Fripraktikum.my.id
2	Alamat IP	95.216.36.164
3	Port yang digunakan	21,53,80,110,143,443,465, 587,993,995,3306

### IV.4 Vulnerability Detection dan Analysis

*Vulnerability detection* adalah proses untuk menemukan celah keamanan pada sistem [11]. Pada penelitian ini menggunakan tools OWASP ZAP, Nessus dan Nikto.

#### a) Pengujian menggunakan Nessus

Nessus adalah suatu alat yang digunakan untuk uji tes kerentanan pada suatu *web*. Pada *domain* fripraktikum.my.id setelah melakukan uji kerentanan terdapat tiga belas kerentanan yang terdapat pada *web* ini, yang mana terdiri dari *medium*, *low* dan *information*.

Tabel 4 Hasil Scanning Nessus

Severity	Jenis Kerentanan
----------	------------------

Medium	Web Application Potentially Vulnerable to Clickjacking
Medium	HSTS Missing From HTTPS Server (RFC 6797)
Low	Web Server Allows Password Auto-Completion
Low	Web Server Transmits Cleartext Credentials
Informational	Web Application Cookies Not Marked HttpOnly
Informational	Web Application Cookies Not Marked Secure
Informational	Missing or Permissive X-Frame-Options HTTP Response Header
Informational	Web Application Sitemap
Informational	Nessus SYN scanner
Informational	Web Server No 404 Error Code Check
Informational	Web Server Uses Basic Authentication over HTTPS
Informational	Web mirroring
Informational	HTTP Server Type and Version

b) Pengujian menggunakan OWASP ZAP

OWASP ZAP adalah salah satu alat yang cukup populer untuk melakukan proses pemindaian (*scanning*). Dalam tahap ini, OWASP ZAP digunakan untuk melakukan *scanning* secara otomatis. Langkah-langkah penggunaannya dimulai dengan membuat pemindaian otomatis dan memasukkan *domain* target pada pengujian ini, *domain* yang digunakan adalah *fripraktikum.my.id*. Setelah itu, pengguna dapat memilih menu *attack* dan menunggu hasil pemindaian. Di bawah ini akan dijelaskan hasil dari *vulnerability scanning* menggunakan *OWASP ZAP*.

ZAP by Checkmarx Scanning Report

Generated with  The ZAP logo ZAP on Fri 16 May 2025, at 10:56:30  
 ZAP Version: 2.16.1  
 ZAP by [Checkmarx](#)

**Contents**

- About this report
  - Report parameters

**About this report**

**Report parameters**

**Contexts**

The following contexts were selected to be included:

- Default Context

**Sites**

The following sites were included:

- <http://fripraktikum.my.id>

(If no sites were selected, all sites were included by default.)

An included site must also be within one of the included contexts for its data to be included in the report.

**Risk levels**

Included: High, Medium, Low, Informational

Excluded: None

**Confidence levels**

Included: User Confirmed, High, Medium, Low

Excluded: User Confirmed, High, Medium, Low, False Positive

No alerts were found within the report parameters.

Gambar 2 Hasil Scanning OWASP ZAP

c) Pengujian menggunakan Nikto

Nikto adalah suatu program *open source* yang bisa melakukan berbagai tes terhadap *web* untuk melihat kerentanan yang ada pada *web* tersebut. Pada *domain*

*fripraktikum.my.id* setelah melakukan uji tes kerentanan terdapat dua belas kerentanan yang ditemukan, diantaranya terdiri dari *high*, *medium*, *low* dan *information*. Di bawah ini akan dijelaskan hasil dari *vulnerability scanning* menggunakan *Nikto*.

Risk Level	Jenis Kerentanan
High	Cookie XSRF-TOKEN tanpa HttpOnly
High	Cookie guest user tanpa HttpOnly
High	Missing Strict-Transport-Security
Medium	Missing X-Content-Type-Options
High	Missing-Content-Security-Policy
Low	Missing Permissions-Policy
Medium	Missing Referrer-Policy
Medium	BREACH Attack Potential
Low	Wildcard Certificate
Info	WordPress Detection
Info	HTTP/3 Support
Info	WordPress API Endpoint

IV.5 Attack and penetration Testing

Berdasarkan hasil dari pengujian kerentanan yang telah dilakukan menggunakan tools OWASP ZAP, Nessus dan Nikto didapatkan kerentanan dari *web* target. Selanjutnya kerentanan dari semua hasil *scanning* menggunakan *tools* tersebut digabungkan dan akan dilanjutkan ke tahap mitigasi. Berikut list kerentanan yang akan dilakukan ke tahap mitigasi [12].

Tabel 5 Penggabungan Kerentanan

Severity	Jenis Kerentanan
Medium	Web Application Potentially Vulnerable to Clickjacking
Low	Web Server Allows Password Auto-Completion
Low	Web Server Transmits Cleartext Credentials
High	Cookie XSRF-TOKEN tanpa HttpOnly
High	Cookie guest user tanpa HttpOnly
High	Missing Strict-Transport-Security
High	Missing-Content-Security-Policy
Medium	Missing X-Content-Type-Options
Medium	Missing Referrer-Policy
Medium	BREACH Attack Potential
Low	Missing Permissions-Policy
Low	Wildcard Certificate

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

- Web Application Potentially Vulnerable to Clickjacking*  
Website tidak punya proteksi *clickjacking*.
- Web Server Allows Password Auto-Completion*  
*Form password* di *website* tidak mematkan fitur *autocomplete* di *browser* (atribut *autocomplete="off"* tidak ada).
- Web Server Transmits Cleartext Credentials*  
Website mengirim data *login* (*username & password*) dalam bentuk *teks* biasa (*cleartext*) tanpa enkripsi (HTTP).
- Cookie XSRF-TOKEN tanpa HttpOnly*  
*Cookie XSRF-TOKEN* dapat diakses via *JavaScript* (*document.cookie*). Jika terjadi *XSS attack*, *attacker* bisa

mencuri token ini dan melakukan *CSRF attack* dengan *bypassing protection*.

- e) *Cookie guest\_user tanpa HttpOnly*  
*Cookie guest\_user* rentan pencurian via XSS. *Attacker* bisa *impersonate guest user* atau mengakses *session information* yang tidak seharusnya.
- f) *Missing Strict-Transport-Security*  
*Website* tidak memaksa koneksi HTTPS. *Browser* bisa di-downgrade ke HTTP, memungkinkan *MITM attack* untuk *intercept traffic* atau *inject malicious content*.
- g) *Missing-Content-Security-Policy*  
Tidak ada pembatasan sumber konten yang bisa dimuat. *Attacker* bisa *inject malicious script, iframe*, atau *resource* dari domain lain untuk *XSS attack*.
- h) *Missing X-Content-Type-Options*  
*Browser* bisa melakukan *MIME type sniffing*, mengeksekusi *file* sebagai *JavaScript* meski bukan. Ini memungkinkan XSS via *file upload* atau *content injection*.
- i) *Missing Referrer-Policy*  
URL dengan parameter sensitif (*session ID*, token) bisa bocor ke situs *external via referrer header* saat *user click link eksternal*
- j) *BREACH Attack Potential*  
Kompresi *deflate + HTTPS* bisa dieksploitasi untuk *BREACH attack*. *Attacker* bisa *extract secret data* (CSRF token, *session*) dengan menganalisa *compressed response size*.
- k) *Missing Permissions-Policy*  
Tidak ada kontrol fitur *browser* seperti *camera, microphone, geolocation*. *Malicious script* bisa mengakses fitur sensitif tanpa *user consent* yang eksplisit
- l) *Wildcard Certificate*  
Satu *private key* mengamankan semua *subdomain*. Jika *key compromised*, seluruh *subdomain* terpengaruh. *Risk* lebih tinggi untuk *certificate management*.

Sebelum masuk ke tahap mitigasi, akan dilakukan penetration tesring yang mana akan memastikan kerentanan yang ada setelah hasil scanning itu benar ada. Hasil dari penetration testing dapat dilihat Tabel 6.

Tabel 6 Hasil Penetration Testing

Jenis kerentanan	Hasil Penetration Testing
<i>Web Application Potentially Vulnerable to Clickjacking</i>	Dari hasil pemindaian menggunakan <i>tools online</i> yaitu <i>securityheaders</i> , terbukti bahwa header <i>X-Frame-Options</i> memang tidak disetel. Oleh karena itu, dapat disimpulkan bahwa <i>website</i> fripraktikum.my.id memiliki potensi kerentanan terhadap <i>clickjacking</i> .
<i>Web Server Allows</i>	Pada halaman <i>login</i> yang dapat dibuka melalui <i>devtools</i> , ditemukan <i>input</i>

<i>Password Auto-Completion</i>	<i>password</i> yang tidak menggunakan atribut <i>autocomplete="off"</i> . Tanpa atribut ini, browser bisa menyimpan password yang dimasukkan dan secara otomatis menampilkannya kembali saat pengguna mengakses <i>form</i> yang sama di lain waktu.
<i>Web Server Transmits Cleartext Credentials</i>	Hasil dari <i>header respons</i> saat mengakses halaman <i>login (/loginSSO)</i> di situs fripraktikum.my.id. Terlihat bahwa koneksi menggunakan protokol HTTPS dan <i>cookie</i> sudah disetel dengan atribut keamanan seperti <i>Secure</i> dan <i>HttpOnly</i> . Hal ini memperkuat temuan bahwa kredensial tidak dikirim dalam bentuk <i>teks</i> biasa, sehingga kerentanan <i>Web Server Transmits Cleartext Credentials</i> tidak dapat divalidasi,
<i>Cookie XSRF-TOKEN tanpa HttpOnly</i>	Pada <i>devtools</i> , Ditemukannya kemampuan untuk menjalankan <i>JavaScript</i> di sisi klien dapat dilihat pada gambar digabungkan dengan fakta bahwa <i>cookie XSRF-TOKEN</i> dapat diakses oleh <i>JavaScript</i> karena tidak memiliki atribut <i>HttpOnly</i> , membuka potensi serangan yang lebih besar seperti pencurian token dan penyalahgunaan token tersebut untuk melakukan serangan <i>CSRF</i> .
<i>Cookie guest_user tanpa HttpOnly</i>	Pada pengujian menggunakan <i>devtools</i> , menunjukkan bahwa <i>cookie</i> dengan nama <i>XSRF-TOKEN</i> dapat ditampilkan secara langsung di konsol. Hal ini mengindikasikan bahwa <i>cookie</i> tersebut tidak dilindungi oleh atribut <i>HttpOnly</i> , sehingga memungkinkan untuk diakses melalui skrip di sisi klien.
<i>Missing Strict-Transport-Security</i>	Berdasarkan hasil pemeriksaan, bahwa <i>header Strict Transport Security</i> tidak ditemukan pada response header server. Ini menunjukkan bahwa website ini tidak mengaktifkan <i>HTTP Strict Transport Security</i> , yang merupakan bagian dari mekanisme perlindungan terhadap kerentanan berbasis <i>transport layer</i> . Jadi membuka celah bagi penyerang untuk menurunkan tingkat keamanan koneksi
<i>Missing-Content-Security-Policy</i>	Berdasarkan hasil pengujian menggunakan <i>devtools</i> terhadap situs fripraktikum.my.id, ditemukan bahwa <i>header</i> keamanan <i>Content-Security-Policy</i> (CSP) tidak diterapkan. Header ini berfungsi untuk mencegah browser memuat dan mengeksekusi konten dari sumber yang tidak dipercaya. Tanpa adanya CSP, <i>browser</i> tetap mengizinkan eksekusi <i>JavaScript</i> dari sumber atau input yang tidak divalidasi, sehingga dapat membuka celah untuk

	serangan seperti <i>Cross-Site Scripting (XSS)</i> .
<i>Missing X-Content-Type-Options</i>	Berdasarkan hasil pemeriksaan menggunakan perintah “curl -I “, ditemukan bahwasanya server tidak mengirimkan <i>header X Content Type Options</i> . Hal ini menunjukkan adanya kerentanan yang berpotensi serangan <i>MIME sniffing</i> yang dapat menyebabkan eksekusi skrip berbahaya.
<i>Missing Referrer-Policy</i>	Eksplorasi diawali dengan pembuatan URL unik melalui <i>Webhook.site</i> , kemudian situs <i>fripraktikum.my.id</i> dibuka di <i>browser</i> dan perintah <i>JavaScript window.location.href = "https://webhook.site/44c14d11-ecb1-4784-8549-05f209b1066a"</i> ; dijalankan melalui fitur <i>Developer Tools</i> . Hasilnya, <i>Webhook.site</i> berhasil menerima permintaan dari browser yang memuat <i>header Referer</i> dengan nilai <i>https://fripraktikum.my.id/</i> . Hal ini menunjukkan bahwa tidak terdapat pengaturan kebijakan <i>Referrer-Policy</i> pada situs tersebut. Ketiadaan pengaturan ini berpotensi dimanfaatkan oleh pihak tidak bertanggung jawab untuk melakukan pelacakan atau penyadapan informasi internal.
<i>BREACH Attack Potential</i>	Kode HTML ini diambil melalui <i>Developer Tools</i> pada browser dan menunjukkan bahwa situs <i>fripraktikum.my.id</i> menyisipkan token CSRF langsung ke dalam isi respons HTML. Penyisipan token semacam ini merupakan indikator umum dari potensi kerentanan <i>BREACH</i> , karena nilai token yang bersifat statis dapat dikompresi bersama konten lainnya. Jika server menggunakan kompresi HTTP seperti <i>gzip</i> , maka penyerang yang dapat mengontrol sebagian input dalam permintaan dan mengamati ukuran respons terkompresi berpotensi menebak nilai token.
<i>Missing Permissions-Policy</i>	Situs <i>web</i> <i>fripraktikum.my.id</i> diketahui tidak mengimplementasikan <i>header Permissions-Policy</i> , yang bertujuan untuk membatasi akses terhadap fitur-fitur sensitif yang disediakan oleh browser seperti kamera, mikrofon, dan geolokasi. Untuk menguji kerentanan ini, dilakukan simulasi eksploitasi melalui <i>DevTools Console</i> pada halaman login situs. Dengan mengeksekusi skrip <i>JavaScript navigator.mediaDevices.getUserMedia</i> , terlihat bahwa browser langsung memunculkan permintaan izin akses terhadap kamera dan mikrofon kepada pengguna.

<i>Wildcard Certificate</i>	Berdasarkan hasil pencarian pada situs <i>crt.sh</i> , diketahui bahwa domain <i>fripraktikum.my.id</i> menggunakan <i>wildcard certificate</i> yang aktif, yaitu <i>*.fripraktikum.my.id</i> . Sertifikat ini memungkinkan semua <i>subdomain</i> dari <i>domain</i> tersebut menggunakan satu sertifikat yang sama, sehingga penyerang berpotensi menyalahgunakan sertifikat <i>wildcard</i> untuk menyamar sebagai <i>subdomain</i> lain yang lebih sensitif, seperti <i>admin.fripraktikum.my.id</i> atau <i>mail.fripraktikum.my.id</i> . Hal ini bisa dimanfaatkan untuk serangan seperti <i>Man-in-the-Middle</i> atau penipuan digital.
-----------------------------	---

#### 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

Berdasarkan hasil pengujian dan analisis terhadap jenis kerentanan pada *web* praktikum Fakultas Rekayasa Industri, diperlukan rekomendasi perbaikan yang bertujuan untuk meminimalkan kerentanan yang ditemukan. Rekomendasi ini berfungsi untuk meningkatkan keamanan dari *website* ini untuk mencegah risiko serangan yang dapat menimbulkan kerugian. Dibawah ini, dapat dilihat perancangan mitigasi melalui tabel 7.

Tabel 7 Rekomendasi Mitigasi Kerentanan

Jenis Kerentanan	Rekomendasi
<i>Web Application Potentially Vulnerable to Clickjacking</i>	Konfigurasi header keamanan melalui file <i>middleware</i> merupakan langkah preventif yang esensial dalam mitigasi kerentanan <i>clickjacking</i> pada aplikasi <i>web</i> . Tujuan utama dari konfigurasi ini adalah untuk mencegah halaman <i>web</i> dimuat di dalam elemen <i>&lt;iframe&gt;</i> oleh <i>domain eksternal</i> yang tidak memiliki otorisasi, yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk melakukan serangan manipulasi antarmuka pengguna ( <i>User Interface Redressing</i> ). Implementasi konfigurasi dilakukan dengan menambahkan instruksi <i>Header always set</i>

	<i>X-Frame-Options "DENY"</i> dan <i>Header always set Content-Security-Policy "frame-ancestors 'none';"</i> .
<i>Web Server Allows Password Auto-Completion</i>	Menambahkan atribut <i>autocomplete="off"</i> pada tag <i>&lt;form&gt;</i> dan <i>autocomplete="new-password"</i> pada <i>input password</i> di <i>file login.blade.php</i> .
<i>Web Server Transmits Cleartext Credentials</i>	mengaktifkan SSL/TLS agar situs dapat menggunakan protokol HTTPS yang aman dan terenkripsi. Tanpa SSL, data seperti <i>username</i> dan <i>password</i> akan dikirim secara jelas melalui jaringan, sehingga mudah disadap. Setelah SSL terpasang dengan benar, sangat penting untuk mengatur <i>redirect</i> otomatis dari HTTP ke HTTPS menggunakan <i>file</i> pengaturan <i>cpanel</i> dan konfigurasi pada <i>.env</i> .
<i>Cookie XSRF-TOKEN tanpa HttpOnly</i>	Mitigasi yang tepat terhadap potensi eksploitasi melalui <i>cookie XSRF-TOKEN</i> adalah menambahkan <i>HttpOnly</i> pada <i>cookie XSRF-TOKEN</i> di <i>Laravel</i> .
<i>Cookie guest_user tanpa HttpOnly</i>	Menambahkan atribut <i>HttpOnly</i> pada <i>cookie guest_user</i> , sehingga tidak bisa diakses via <i>JavaScript</i> .
<i>Missing Strict-Transport-Security</i>	Menambahkan <i>header Strict-Transport-Security</i> dengan konfigurasi yang tepat untuk memaksa HTTPS dan meningkatkan keamanan web
<i>Missing-Content-Security-Policy</i>	Menerapkan <i>header Content-Security-Policy</i> berguna untuk membatasi sumber daya <i>eksternal</i> yang dapat dimuat oleh <i>browser</i> . CSP berfungsi sebagai pertahanan terhadap serangan seperti <i>Cross-Site Scripting (XSS)</i> dengan hanya mengizinkan konten dari sumber yang tepercaya. Implementasi dapat dilakukan melalui <i>Laravel Middleware</i> dengan menambahkan <i>header CSP</i> secara

	langsung dalam respon HTTP pada <i>file middleware</i> .
<i>Missing X-Content-Type-Options</i>	Menambahkan <i>header X-Content-Type-Options: nosniff</i> guna mencegah <i>browser</i> melakukan <i>MIME type sniffing</i> yang dapat dimanfaatkan untuk serangan eksekusi skrip berbahaya. Rekomendasi terbaik adalah menambahkan <i>header</i> ini melalui <i>file middleware</i> .
<i>Missing Referrer-Policy</i>	Menambahkan <i>header Referrer-Policy</i> melalui <i>file middleware</i> yang berada di <i>folder public</i> . Rekomendasi nilai kebijakan yang digunakan adalah <i>strict-origin-when-cross-origin</i> , karena memberikan perlindungan privasi yang seimbang tanpa mengganggu fungsi normal situs.
<i>BREACH Attack Potential</i>	Menambahkan <i>middleware</i> khusus yang menyisipkan <i>random padding</i> pada respon <i>HTML</i> . <i>Middleware</i> ini bekerja dengan menambahkan karakter acak di bagian akhir respon <i>HTML</i> untuk membuat ukuran respon menjadi tidak konsisten. Teknik ini bertujuan mengacaukan analisis ukuran data yang digunakan dalam serangan <i>BREACH</i> , karena <i>attacker</i> tidak lagi bisa membandingkan ukuran <i>response</i> untuk menebak isi data sensitif.
<i>Missing Permissions-Policy</i>	Menambahkan <i>header Permissions-Policy</i> secara eksplisit, dengan penerapan yang tepat, <i>website</i> menjadi lebih aman karena <i>fitur browser</i> yang sensitif tidak dapat diakses oleh pihak yang tidak berwenang, sehingga mengurangi risiko kebocoran data dan serangan berbasis <i>browser</i> .
<i>Wildcard Certificate</i>	Aktifkan <i>HTTP Strict Transport Security (HSTS)</i> melalui <i>file middleware</i> untuk memperkuat

	keamanan komunikasi HTTPS.
--	----------------------------

Berdasarkan rekomendasi pada tabel 7 yang telah dilakukannya tahap mitigasi akan dilakukan pengujian ulang pada web praktikum fakultas rekayasa industri setelah dilakukan mitigasi. Pengujian ini dilakukan untuk membandingkan hasil pengujian sebelum mitigasi dan setelah dilakukannya mitigasi.

b) Pengujian Ulang Pasca Mitigasi

Setelah dilakukannya mitigasi sebelumnya, dilakukan kembali pengujian ulang setelah mitigasi pada web praktikum Fakultas Rekayasa Industri dengan Nessus dan Nikto. Tujuan dilakukannya pengujian ulang ini untuk mengetahui kerentanan sebelumnya apakah sudah berhasil atau masih ada kerentanan tersebut. Berikut hasil dari pengujian ulang setelah dilakukan mitigasi.

Tabel 8 Hasil Pengujian Pasca Mitigasi

Pra Mitigasi	Pasca Mitigasi	Keterangan
<i>Cookie XSRF-TOKEN tanpa HttpOnly</i>	<i>Cookie XSRF-TOKEN tanpa HttpOnly</i>	Sudah menambahkan HttpOnly pada website tetapi kerentanan masih ada dan perlu pengecekan lebih lanjut oleh developer dari konfigurasi pada web tersebut.
<i>Missing X-Content-Type-Options</i>	<i>Missing X-Content-Type-Options</i>	Sudah menambahkan header <i>X-Content-Type-Options: nosniff</i> tetapi kerentanan masih ada dan perlu pengecekan lebih lanjut oleh developer dari segi kode dan konfigurasi web tersebut.
<i>Missing Referrer-Policy</i>	<i>Missing Referrer-Policy</i>	Sudah Menambahkan header <i>Referrer-Policy</i> tetapi kerentanan masih ada dan perlu pengecekan lebih lanjut oleh developer dari segi kode dan konfigurasi web tersebut.
<i>BREACH Attack Potential</i>	<i>BREACH Attack Potential</i>	Setelah menambahkan random padding tetapi kerentanan masih ada dan

		perlu pengecekan lebih lanjut oleh developer dari segi kode dan konfigurasi web tersebut.
<i>Web Application Potentially Vulnerable to Clickjacking</i>	-	Kerentanan berhasil ditutup
<i>Web Server Allows Password Auto-Completion</i>	-	Kerentanan berhasil ditutup
<i>Web Server Transmits Cleartext Credentials</i>	-	Kerentanan berhasil ditutup
<i>Cookie guest_user tanpa HttpOnly</i>	-	Kerentanan berhasil ditutup
<i>Missing Strict-Transport-Security</i>	-	Kerentanan berhasil ditutup
<i>Missing-Content-Security-Policy</i>	-	Kerentanan berhasil ditutup
<i>Missing Permissions-Policy</i>	-	Kerentanan berhasil ditutup
<i>Wildcard Certificate</i>	-	Kerentanan berhasil ditutup

Berdasarkan hasil pengujian setelah dilakukan mitigasi, terdapat beberapa kerentanan yang berhasil diatasi, namun masih ada juga kerentanan yang belum dapat dihilangkan sepenuhnya. Kerentanan yang masih ada memerlukan perbaikan lebih lanjut, baik dari sisi kode maupun konfigurasi oleh tim pengembang web praktikum Fakultas Rekayasa Industri. Meskipun begitu, sejumlah kerentanan yang berhasil dimitigasi telah memberikan kontribusi positif dalam meningkatkan keamanan web tersebut

V. KESIMPULAN

Pengujian keamanan terhadap web praktikum Fakultas Rekayasa Industri menggunakan metode Vulnerability Assessment and Penetration Testing (VAPT) dengan tools OWASP ZAP, Nikto, dan Nessus menunjukkan adanya sejumlah kerentanan dengan tingkat risiko yang bervariasi. Tools Nessus mendeteksi 13 kerentanan (tingkat medium, low, dan informasi), Nikto menemukan 12 kerentanan (tingkat high, medium, low, dan informasi), sedangkan OWASP ZAP tidak menemukan kerentanan yang signifikan. Langkah mitigasi dilakukan berdasarkan temuan tersebut, di antaranya dengan menambahkan berbagai header keamanan (seperti X-Frame-Options, Content-Security-Policy, Strict-Transport-Security), pengaturan atribut keamanan pada form login dan cookie, serta aktivasi SSL/TLS. Mitigasi ini bertujuan untuk memperkuat lapisan keamanan web dan mengurangi risiko eksploitasi terhadap sistem. Dengan diterapkannya mitigasi ini, diharapkan web praktikum menjadi lebih aman dan dapat memberikan perlindungan yang lebih baik terhadap potensi serangan siber.

## REFERENSI

- [1] Mendy, "Pengertian Website: Apa itu Web, Manfaat, Jenis, dan Contoh," Kampus IT, 7 August 2023. [Online]. Available: <https://kampusit.id/pengertian-website>.
- [2] R. A. C. E. V. A. G. K. M. C. M. J. V. T. Kit Arvin R. Cadiente, "APPLYING VULNERABILITY ASSESSMENT AND PENETRATION TESTING (VAPT) AND NETWORK ENHANCEMENT ON THE NETWORK INFRASTRUCTURE OF JOURNEY TECH INC," *INNOVATUS*, vol. 3, no. 1, 2020.
- [3] D. B. O. K. S. D. Sandi, "LAPORAN BULANAN PUBLIK," 2023.
- [4] B. Raharjo, KEAMANAN SISTEM INFORMASI, 2021.
- [5] S. K. Rakshit, Ethical Hacker's Penetration Testing Guide, new delhi, 2022.
- [6] B. P. Angela Orebaugh, Nmap in the Enterprise: Your Guide to Network Scanning, 2008.
- [7] R. Rogers, Nessus network auditing, 2011.
- [8] G. Kusuma, "IMPLEMENTASI OWASP ZAP UNTUK PENGUJIAN KEAMANAN SISTEM INFORMASI AKADEMIK," *Teknologi Informasi*, 2022.
- [9] T. & U. Muhyidin, "Perbandingan Tingkat Keamanan Website Menggunakan Nmap dan Nikto dengan Metode Ethical Hacking," *Teknik Logika Matematika*, 2022.
- [10] Y. Khera, D. Kumar, Sujay and N. Garg, "Analysis and Impact of Vulnerability Assessment and Penetration Testing," *IEEE*, 2019.
- [11] D. R. M. a. J. Benjamin, "Penetration Testing and Vulnerability Scanning of Web Application Using Burp Suite," *Natl. Conf. Emerg. Comput. Appl*, 2021.
- [12] J. I. F. R. a. M. E. A. W. Kuncoro, "Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website: Literature Review," *AUTOMATA*, 2021.