ANALISIS KEAMANAN WEBSITE E-LEARNING POLITEKNIK BHAKTI SEMESTA BERBASIS *VULNERABILITY ASSESSMENT*

Rivort Pormes[1], Puspa Ira Dewi Candra Wulan[2], Danis Putra Perdana[3], Rofiq Fauzi[4], Alfian Yuda Syahputra[5]

[1,2,3,4,5] Rekayasa Keamanan Siber, Politeknik Bhakti Semesta, Salatiga
Jl. Argoluwih No.15, Ledok, Kec. Argomulyo, Kota Salatiga, Jawa Tengah 50732
Email: rivort@bhaktisemesta.ac.id[1],puspa@bhaktisemesta.ac.id[2], danis@bhaktisemesta.ac.id[3], rf@bhaktisemesta.ac.id[4], alfianyudasyahputra@gmail.com[5]

Abstract

This research analyzes the security of the e-learning website at Politeknik Bhakti Semesta using the Vulnerability Assessment and Penetration Testing (VAPT) approach. This research identified vulnerabilities in the system during the Information Gathering phase, utilizing Nslookup, Nmap, and Testssl.sh. Vulnerability Scanning using Nikto and OWASP ZAP found weaknesses in TLS/SSL encryption, security header configuration, and potential attacks such as Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF). Vulnerability Exploitation involved injecting XSS scripts to test the robustness of the system. TheBurp Suite tool was used to access specific paths, while Arjun helped identify queries on URL endpoints. The test results confirmed the presence of security risks, emphasizing the need for further security enhancements in the system.

Keywords: *VAPT, E-learning security, Vulnerability Scanning, Information Gathering, Vulnerability Exploitation*

Abstrak

Penelitian ini menganalisis keamanan situs *web e-learning* di Politeknik Bhakti Semesta menggunakan pendekatan *Vulnerability Assessment and Penetration Testing (VAPT)*. Penelitian ini mengidentifikasi kerentanan dalam sistem selama fase Pengumpulan Informasi *(Information Gathering)*, dengan memanfaatkan *Nslookup, Nmap,* dan *Testssl.sh*. Pada Pemindaian Kerentanan *(Vulnerability Scanning)* menggunakan *Nikto* dan *OWASP ZAP*, ditemukan kelemahan dalam enkripsi *TLS/SSL*, konfigurasi *header* keamanan, serta potensi serangan seperti *Cross-Site Scripting (XSS)* dan *Cross-Site Request Forgery (CSRF)*. Eksploitasi Kerentanan *(Vulnerability Exploitation*) melibatkan penyuntikan skrip *XSS* untuk menguji ketahanan sistem. *Tools Burp Suite* digunakan guna mengakses jalur tertentu, sementara *Arjun* membantu mengidentifikasi *query* pada titik akhir *URL*. Hasil pengujian mengkonfirmasikan adanya risiko keamanan, sehingga menekankan perlunya peningkatan keamanan lebih lanjut dalam sistem.

Kata kunci: *Pengujian Kerentanan dan Penetrasi, Keamanan E-learning, Pemindaian Kerentanan, Pengumpulan Informasi, Eksploitasi Kerentanan*

Pendahuluan

Dalam pelaksanaan pendidikan tinggi, setiap institusi diharapkan untuk menyediakan fasilitas pembelajaran yang sesuai dengan standar yang ditetapkan oleh pemerintah. Salah satu syaratnya adalah penyediaan fasilitas pembelajar secara darinng *(e-learning)* [1]. *E-learning* juga dapat mempersingkat waktu pembelajaran, dan juga menghemat waktu biaya yang harus dikeluarkan oleh sebuah program studi ataupun Fakultas. Dengan *e-learning*, dosen dan instruktur dapat memutakhirkan materi pemebelajaran lebih muda, mengontrol kegiatan belajar, serta menyesuaikan bahan ajar dengan perkembangan ilmu pengetahuan. *Website e-learning* memungkinkan proses belajar mengajar dapat dilakukan secara *online*, memudahkan akses dan interaksi antara guru dan siswa. Namun, penggunaan teknologi ini juga membawa tantangan baru, yaitu isu keamanan. Keamanan *website e-learning* menjadi sangat penting mengingat data yang ditangani oleh sistem ini sangat sensitif, seperti data pribadi siswa, data nilai, dan materi pelajaran. Karena itu, perlu dilakukan analisis keamanan untuk mengetahui sejauh mana *website e-learning* aman dari serangan siber[2]

Isu keamanan dalam sistem *e-learning* berkaitan dengan prinsip dasar keamanan aplikasi berbasis *web application*. Semisal kemungkinan manipulasi yang dilakukan oleh orang luar ataupun mahasiswa untuk memasuki sistem yang ada. Hal ini bersesuaian dengan isu keamanan *confidentiality* dan *authentification*. Kerentanan pada aplikasi berbasis *web* bisa beragam, tergantung dari *module, library, CMS,* dan *database* yang dipakai. Sehingga aplikasi berbasis *web* mempunyai banyak sisi untuk diserang. Berdasarkan data dari *Id-SIRTII (Indonesian Security Incident Response Team on Internet Infrastructure)* tahun 2018, tercatat 10 jenis serangan di *internet*, seperti *trojan*, serangan *DoS*, upaya mendapatkan hak administrator, dan pelanggaran kebijakan. Insiden siber yang tercatat pada laporan tahunan Pusopskamsinas BSSN tahun 2019 masih banyak terjadi karena instansi pemerintah banyak yang tidak mengimplementasikan *Web Application Firewall (WAF)*, tidak ada pembatasan akses pada *login administrator*, penggunaan *password* yang relatif lemah, kurangnya pengawasan terhadap akun *website* dan *access log* serta menggunakan aplikasi atau *framework* yang *out of update*. Salah satu bentuk upaya dalam hal tersebut yakni dengan melakukan penilaian kerentanan *(vulnerability assessment)* terhadap situs atau aplikasi yang menyimpan informasi atau data yang bersifat sensitif atau rahasia [3].

Terdapat sebuah standar yang digunakan sebagai panduan dalam melakukan proses analisa keamanan pada sebuah *website*, panduan tersebut adalah *OWASP (Open Web Application Security Project) TOP 10*. *OWASP* merupakan sebuah organisasi nirlaba yang mempunyai visi untuk menjaga keamanan *website* dengan banyak menyediakan *resource* atau sumber daya. Penilaian kerentanan dilakukan dengan mengacu kepada daftar kerentanan yang tersedia di dalam *OWASP Top 10*, sehingga penilaian kerentanan ini akan membantu praktisi TI terkait masalah keamanan pada aplikasi *web*, karena hasil penilaian akan menggambarkan celah keamanan terkini. Dalam melakukan penilaian kerentanan, digunakan beberapa *tools* untuk melakukan penemuan, pengujian, analisis, dan pelaporan sistem serta kerentanan[4].

Penelitian ini secara spesifik mengkaji kefektifitasan dari sistem keamanan *e-learning* yang telah digunakan. Dengan pendekatan *Vulnerability Assessment* serta penggunaan *tool*s, seperti *nslook*, *nmap* dan *testssl.sh* untuk mendapatkan *information gathering*, dan *vulnerability scanning* dengan menggunakan *nikto* dan *OWASP ZAP* serta *vulnerability exploiting* dengan *Cross-Site Scripting (XSS)* menggunakan dua pendekatan, yaitu secara manual dan otomatis sebagai bagian dari proses validasi dalam pengujiannya. Analisis pendekatan *Vulnerability Assessment* dengan tujuan untuk mengidentifikasi, mengklasifikasikan, dan membantu memperbaiki kerentanan yang ada dalam sistem. Dengan demikian, diharapkan dapat meningkatkan tingkat keamanan *website e-learning.*

Landasan Teori

A. E-learning

Penggunaan media *e-learning* diatur dalam UU Nomor 12 Tahun 2012 tentang Pendidikan Tinggi Pasal 31, yang menyebutkan bahwa pembelajaran jarak jauh dilakukan melalui berbagai media komunikasi. Salah satu upaya yang bisa dilakukan oleh pendidik yaitu menggunakan *e-learning* sebagai alat pendukung. *E-learning* sendiri adalah media pembelajaran berbasis elektronik yang memanfaatkan komputer, laptop, ataupun handphone yang dapat terhubung dengan jaringan *internet*.

E-learning dapat disajikan dalam berbagai bentuk media interaktif yang dapat menarik perhatian peserta didik, seperti video, animasi, dan game edukasi. Efektivitas *e-learning* terletak pada fleksibilitasnya sebagai alat pendukung pembelajaran yang memungkinkan siswa belajar kapan saja dan di mana saja.

B. Vunerability Assessment

Vulnerability assessment merupakan proses penting untuk menyelidiki kerentanan, kelemahan, serta kekurangan dalam sebuah sistem. Dengan melakukan *vulnerability assessment* membantu lembaga, organisasi, maupun individu mengidentifikasi dan mengatasi masalah keamanan sebelum dieksploitasi oleh peretas.

Menurut *ISACA (The Information Systems Audit and Control Association)*, Terdapat empat jenis *vulnerability assessment* yaitu:

1) Network Based Scans

Network Based Scans digunakan untuk mengidentifikasi kemungkinan serangan keamanan jaringan. Dikarenakan pada pemindaian ini dapat menyebutkan layanan yang berjalan, memindai berbagai *port TCP*, memeriksa *banner* sistem atau menggunakan sejumlah teknik lain untuk menentukan jenis dan versi *host* atau perangkat.

2) Host Based Scans
Host Based Scans digunakan untuk menemukan dan mengidentifikasi kerentanan pada *server*. Pemindaian ini dapat memberikan visibilitas yang lebih besar kepengaturan konfigurasi sistem dan detail tambahan, sambil mencakup *port* dan layanan juga terlihat oleh *network based scans*, tetapi menawarkan visibilitas yang lebih besar ke pengaturan konfigurasi dan *patch history* dan sistem *scan*.

3) Wireless Network Scans
Wireless Network Scans dari jaringan WI-FI organisasi biasanya berfokus pada titik-titik serangan dalam infrastruktur jaringan nirkabel. Selain pemindaian jaringan nirkabel juga dapat memvalidasi bahwa jaringan perusahaan atau organisasi dikonfigurasi dengan aman

4) Application Scans
Application Scans dapat digunakan untuk menguji situs *web* dalam mendeteksi kerentanan perangkat lunak yang diketahui dan konfigurasi yang salah dalam aplikasi jaringan atau *web*[5].

Penelitian ini akan menggunakan beberapa *tools* dalam proses pengujian. Penggunaan *tools* tersebut diharapkan dapat memberikan gambarana tentang *vulnerability assessment* yang ada pada sistem *e-learning* Politeknik Bhakti Semesta.

Jenis pendekatan yang digunakan berdasarkan pada *ISACA (The Information Systems Audit and Control Association)* meliputi, *Network Based Scans, Host Based Scans,* dan *Application Scans*.

a) Nslook
Nslookup adalah alat yang berguna untuk mengetahui alamat *IP* suatu *domain*. Selain itu, ini juga berguna untuk mendiagnosis masalah jaringan *DNS* [6]

b) Nmap (Network Mapper)
Nmap (Network Mapper) adalah sebuah *tool open source* untuk eksplorasi dan audit keamanan jaringan. *Nmap* menggunakan paket *IP raw* untuk mendeteksi *host* yang terhubung dengan jaringan dilengkapi dengan layanan (nama aplikasi dan versi) yang diberikan, sistem operasi (dan versi), jenis *firewall/filter* paket yang digunakan, dan sejumlah karakteristik lainnya [6]

c) Tessl.sh
Tessl.sh adalah alat baris perintah gratis dan *open source*, kaya fitur yang digunakan untuk memeriksa layanan yang mendukung enkripsi *TLS/SSL* untuk sandi, protokol, dan beberapa kelemahan kriptografi yang didukung, di *server Linux/BSD* [7]

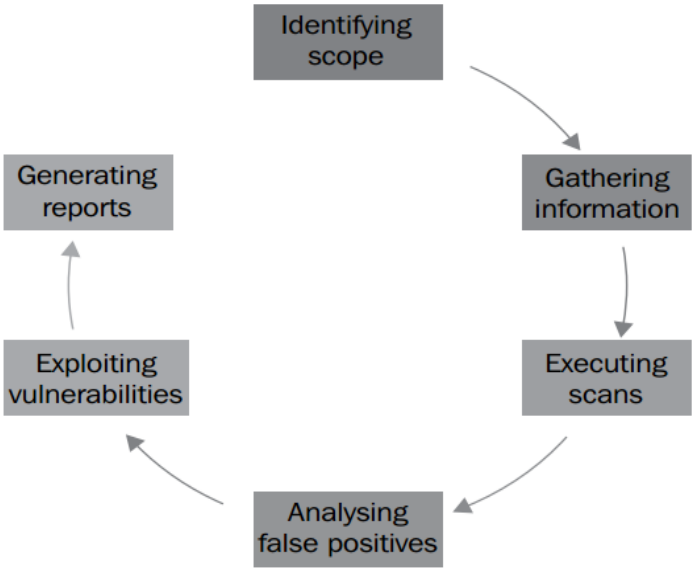
d) Nikto
Nikto adalah pemindai *server web open source (GPL)* yang melakukan pengujian komprehensif terhadap *server web* untuk beberapa item, termasuk lebih dari 6700 *file/program* yang berpotensi berbahaya, memeriksa versi usang lebih dari 1250 *server*, dan masalah khusus di lebih dari 270 *server*. *Nikto2* juga dapat melakukan pemeriksaan terhadap *file indeks* pada *web server* serta mencari beberapa opsi pada *HTTP*, *Nikto2* juga memiliki kemampuan untuk mengetahui tentang *server web* dan apa saja *software* yang di pasang. *Nikto2* memiliki fitur-fitur diantaranya yaitu, mendukung *SSL*, proksi *HTTP*, pemeriksaan *server* yang kadaluarsa, *multiple server* dan *port scanning*, *host authentication*, dapat melakukan hasil *report* dalam *plain text, XML, HTL,* dan *NBE/CSV format* [8]

e) OWASP ZAP
OWASP ZAP merupakan sebuah aplikasi yang digunakan dalam menemukan kerentanan pada suatu Aplikasi *web*. *OWASP ZAP* menyediakan *scanner* secara otomatis. Dalam penggunaan *OWASP ZAP*, *scanner* dapat digunakan untuk menguji *server*, jaringan, perangkat dan *end points*. Saat melakukan *scanning*, tahap-tahap yang dilakukan adalah *Explore*, *Attack* dan *Report*. [9]

f) Penetration Testing
Penetration testing dapat didefinisikan sebagai kerentanan jaringan komputer atau lubang keamanan, dan kerentanan keamanan dipahami sebagai kelemahan program/infrastruktur yang memungkinkan sistem untuk dieksploitasi. Kerentanan ini disebabkan oleh kesalahan desain, pembuatan, atau implementasi sistem. Kerentanan ini digunakan sebagai sarana bagi peretas untuk mendapatkan akses tidak sah ke sistem sehingga secara leluasa mengeksploitasi kerentanan yang ditemukan [10]

Metode Penelitian

Metode penelitian di uraikan dengan menggunakan pendekatan *vulnerability assessment* dan *penetration testing life cycle (VAPT Life Cycle)* yang merupakan bagian yang menjelaskan fase-fase utama dalam *Vulnerability Assesment* dan *Penetration Testing* [5]



Gambar 1. *VAPT Life Cycle*

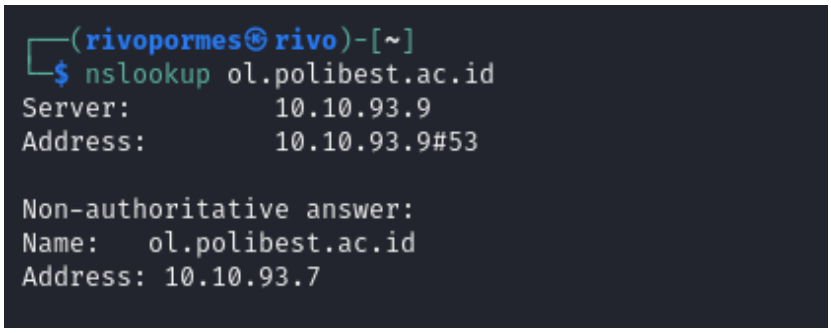
1) *Identifying Scope*
   Tahap pertama mencangkup ruang lingkup penelitian yang akan diteliti, penelitian ini berfokus pada website *e-learning* Politeknik Bhakti semesta sebagai objek penelitian
2) *Information Gathering*
   Tahap kedua melibatkan pengumpulan informasi sistem target menggunakan beberapa alat, seperti Nslookup, Nmap, dan Testssl.sh.
3) *Vulnerability Scanning*
   Tahap ketiga dilakukan pencarian kerentanan pada *website e-learning* Politeknik Bhakti Semesta dengan menggunakan *tools nikto* dan *OWASP ZAP*
4) *False Positive Analysis*
   Tahap keempat, mengumpulkan daftar kerentanan dari *Webseite e-learning* Politeknik Bhakti Semesta. Salah satu kegiatan utama yang harus dilakukan dengan output yang menjadi *false positive analysis* yaitu menghilangkan atau memastikan bahwasanya kerentanan yang ditemukan bukan kerentanan yang salah.
5) *Vulnerability Exploitation*
   Tahap kelima, merupakan tahapan yang bertujuan untuk menargetkan sistem *e-learning* Politkenik Bhakti Semesta berdasarkan eksploitasi terhadap hasil yang teridentifikasi atau hasil eksploitasi kerentanan yang tersedia secara publik.
6) *Generating Report*
   Tahap keenam merupakan tahapan pembuatan laporan tentang kerentanan pada *website e-learning* Politeknik Bhakti Semesta, berserta dampaknya serta memberikan rekomendasi untuk perbaikan kedepannya.

Hasil dan Pembahasan

1) *Identifying Scope*
   Tahap awal *Identyfing Scope* bertujuan mempersempit ruang lingkup penelitian yang hanya berfokus pada sistem *e-learning* Politeknik Bhakti Semesta.
2) *Information Gathering*
   Tahap kedua dilakukan proses *vulnerability assessment* terhadap sistem *e-learning* Politeknik Bhakti Semesta berupa *information gathering* dengan menggunakan *tools nslookup, nmap,* dan *testssl.sh*.

a. *Nslookup*

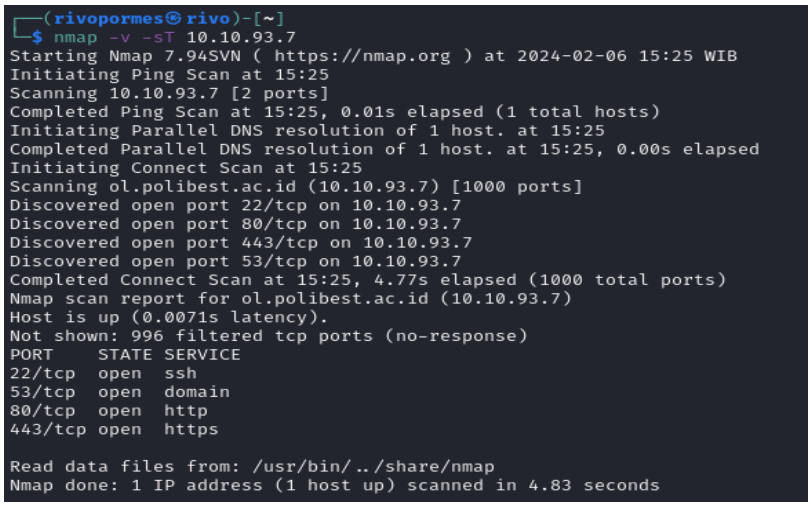
Tahap awal *information gathering* digunakan *tools nslookup* pada *domain ol.polibest.ac.id* untuk mengetahui informasi dari *IP* dan *DNS* yang digunakan pada *e-learning* Politeknik Bhakti Semesta. *IP* dari *o.polibest.ac.id* adalah 10.10.93.7, serta *IP DNS server* menggunakan 10.10.93.9.



Gambar 2. *Scanning* dengan *Nslookup*

b. *Nmap*

Tahap selanjutnya pencarian *information gathering* menggunakan *tools* pemindaian *nmap*, ke alamat *IP ol.polibest.ac.id* yang merupakan alamat dari *e-learning* Politeknik Bhakti Semesta. Perintah yang digunakan adalah *nmap -v -sT 10.10.93.7* dan hasil pemindaian ditampilkan



Gambar 3. *Scanning* dengan *Nmap*

Pemindaian dengan *Nmap* menghasilkan informasi penting, seperti *port* yang terbuka *(22/TCP, 53/TCP, 80/TCP, dan 443/TCP)* dan layanan yang aktif *(SSH, domain, HTTP, dan HTTPS).*

c. *Testssl.sh*

Tahap ketiga dari *information gathering* adalah melakukan proses pemindaian enkripsi *TLS/SSL* untuk sandi, protokol, dan beberapa kelemahan kriptografi serta informasi mengenai jenis *testing vulnerability*.

Hasil menjalankan *tools testssl.sh* didapatkan kerentanan pada *Breach (CVE-2013-3587)* dengan status *potentially NOT ok, "gzip" HTTP compression detected*, artinya memungkinkan penyerang menemukan rahasia yang dibungkus dalam kompresi *HTTP* di dalam *SSL*. Dengan menyuntikkan *plaintext* ke dalam permintaan *HTTPS*, penyerang dapat mempelajari informasi tentang *response HTTPS* yang sesuai dengan mengukur ukurannya.

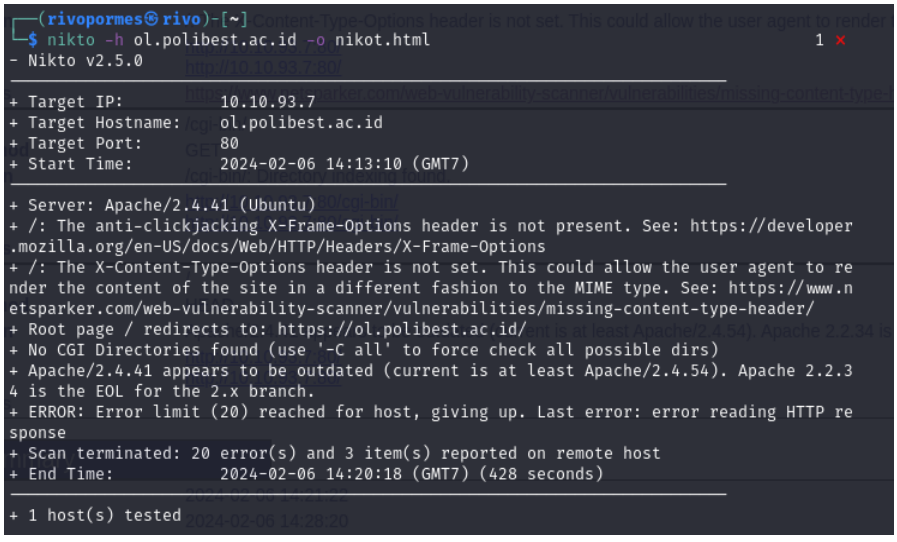
Gambar 4. Hasil *scanning testssl.sh*

Tindakan ini bergantung pada kemampuan penyerang untuk mengamati ukuran *ciphertext* yang diterima oleh *browser* sambil memicu sejumlah permintaan yang dibuat secara strategis ke situs target. *Lucky13 (CVE-2013-0169)* dengan status *potentially VULNEARBLE*, *use cipher block chaining (CBC) chiphers with TLS*, merupakan kerentanan yang memungkinkan serangan pada pengaturan waktu yang bertujuan untuk memulihkan *plain text* ketika sandi *mode CBC* sedang digunakan. Pemindai keamanan sebagian besar melaporkan kerentanan ketika *cipher mode CBC* sedang digunakan

3) Vulnerability Scanning

Tahap ketiga menggunakan *vulnerability assessment* terhadap sistem *e-learning* Politeknik Bhakti Semesta berupa *vulnerability scanning* dengan menggunakan *tools nikto,* dan *OWASP ZAP*.

a. *Nikto*



Gambar 5. Hasil *Scanning Nikto*

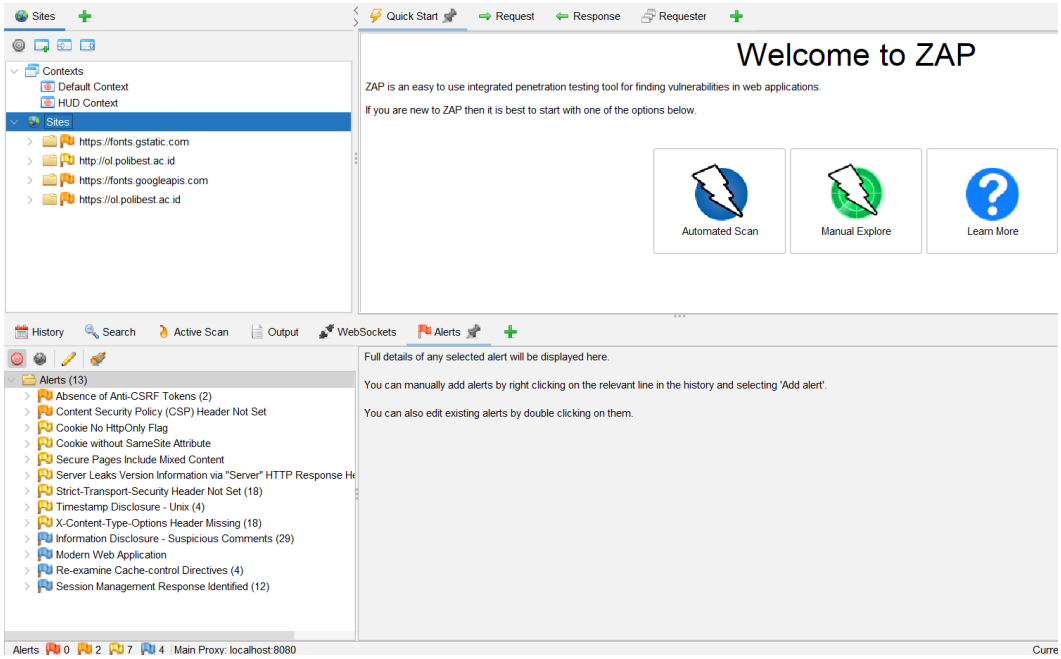
Pemindaian menggunakan *tools nikto* mendapatkan beberapa informasi kerentanan mengenai halaman *domain/website ol.polibest.ac.id* dengan *IP Address 10.10.93.7*. Hasil pengujian di tampilkan pada gambar 3 dan beberapa kerentanan yang ditemukan, antara lain:

Tabel 1. Hasil *Vulnerability Scanning Nikto*

Kerentanan	Keterangan
The anti-clickjacking X-Frame-Option Header is not present	Keberadaan *header* ini membantu melindungi situs *web* dari serangan *clickjacking*. *website e-learning* politeknik bhakti semesta telah menambahkan perintah: *Header always append X-Frame-Options SAMEORIGIN* untuk membatasi halaman *website* agar hanya dapat dimuat dalam *frame* yang berasal dari domain yang sama. Termasuk *false positif*
The X-Content-Type-Options header is not set	*Cookies XSRF-Token* yang dibuat tidak menggunakan *secure flag* serta beresiko terkena serangan *sniffing MIME*. *Server website e-learning* politeknik bhakti semesta harus dikonfigurasi untuk menyertakan *header sniffing* anti *MIME.* Contoh: *X-Content-Type-Option=nonsniff*. Termasuk *true positive* karna belum diimplementasikan pada *e-learning* Politeknik Bhakti Semesta.
Uncommon header 'x-direct-by' found, with contents: Moodle	*Header direct* dari perangkat lunak *website e-learning* Politeknik Bhakti Semesta terekspos dan menggunakan basis *Moodle*. Termasuk *true positive* karna framework yang digunakan sebagai *e-learning* Politeknik Bhakti Semesta berbasis pada *platform moodle*.

b. OWASP ZAP

Tahap selanjutnya *vulnerability scanning* dengan menggunakan *tools OWASP ZAP* untuk mencari kerentenan pada *e-learning* Politeknik Bhakti Semesta. Dari hasil *vulnerability scanning*, terdapat 7 topik kerentanan yang terbagi menjadi 2 tingkatan kerentanan yaitu *medium* dan *low*.



Gambar 6. Hasil *Scanning OWSP ZAP*

Tabel 2. *Vulnerability Scanning OWASP ZAP*

Vulnerability	Description	Risk	Jumlah
Absence of Anti-CSRF Tokens	*Token Anti-CSRF* tidak ditemukan dalam formulir pengiriman *HTML*. Termasuk *false positif* karna telah diimplementasikan *logintoken <input type="hidden" name="logintoken" value="eTTo9CxXQgTOdbpTAuiObXPJmx2i8eS9">* pada *website e-learning* Politeknik Bhakti Semesta.	Medium	2
Content Security Policy (CSP) Header Not Set	*CSP header* tidak tersedia, memungkinkan terjadi serangan *Cross Site Scripting (XSS)* dan serangan injeksi data. Termasuk kedalam *true positive* karna belum terimplementasi	Medium	1
Cookie No HttpOnly Flag	*Cookie* telah di atur tanpa *HttpOnly flag*. Artinya apabila skrip berbahaya dapat dijalankan menggunakan *JavaScript* maka *cookie* dapat diakses dan dikirimkan ke situs lain. Kemungkinan *hijacking* dapat dilakukan. Termasuk *True positive* karna belum terimplementasi	Medium	1
Cookie without SameSite Attribute	Terdapat *cookie* yang dikirimkan melalui permintaan *'cross-site'*. Artinya berpotensi terhadap pemalsuan permintaan *cross-site script* dan *timing attack*. Termasuk *true positive* karna belum terimplementasi.	Low	1
Secure Pages Include Mixed Content	Terdapat halaman berisi konten campuran yang diakses melalui *HTTP* dan bukan *HTTPS*. Termasuk *true positive* karna *website e-learning* belum mengganti sepenuhnya konten untuk diakses langsung dari *HTTPS*	Low	1
Server Leaks Version	*Server web*/aplikasi mengekspos informasi versi *header* melalui *HTTP server*. Informasi tersebut	Low	18

Information via "Server" HTTP Response Header Field	memudahkan penyerang mengidentifikasi kerentanan lain pada *server* web/aplikasi. Termasuk *true positive* karna *server* terekspos menggunakan *Apache/ 2.4.41 (Ubuntu)* sehingga dapat memberikan petunjuk kepada penyerang mengenai perangkat lunak yang digunakan. Konfigurasi *server* untuk tidak mengirimkan informasi versi dengan cara *ServerTokens* dan *ServerSignature* untuk menyembunyikan informasi yang dikirim. Contoh: *ServerToken Prod* *ServerSignature Off*		
Strict-Transport-Security Header Not Set	*Hilangnya header Strict-Transport-Security* menyebabkan komunikasi melalui *HTTP* diizinkan ke domain yang ditentukan. Hal ini membuat situs *web* rentan terhadap serangan *man-in-the-middle*, sehingga laman *login* palsu menjadi salah satu opsinya. Termasuk *true positive* karna *website e-learning* belum sepenuhnya diakses menggunakan *HTTPS*. Dibutuhkan penambah *header HSTS* pada konfigurasi *Server*. Contoh: *Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"*	Low	18

4) Vulnerability Exploitation

Tahap terakhir adalah *Vulnerability Exploitation* dimana *pentester* secara aktif mencoba memanfaatkan kerentanan yang telah diidentifikasi dalam sistem atau aplikasi. Tujuan utamanya adalah untuk memvalidasi apakah kerentanan tersebut dapat dieksploitasi oleh penyerang nyata, menilai dampaknya, dan memberikan rekomendasi untuk mitigasi risiko.

Tahap ini penting karena membantu organisasi memahami risiko keamanan secara konkret dan memungkinkan mereka mengambil langkah pencegahan yang tepat.

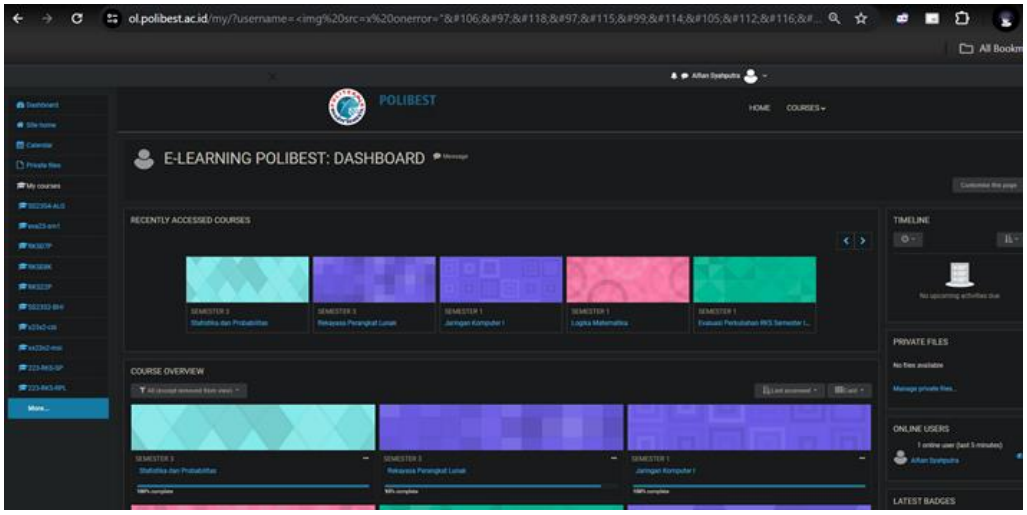
Tujuan dari *vulnerability exploitation*:

- Validasi Kerentanan
  Proses ini bertujuan untuk memastikan bahwa kerentanan yang diidentifikasi memang bisa dieksploitasi oleh penyerang potensial. Validasi ini penting untuk membedakan antara kerentanan yang nyata dan yang hanya bersifat teoretis, sehingga organisasi bisa fokus pada ancaman yang benar-benar berisiko.
- Penilaian Risiko
  Eksploitasi kerentanan memberikan gambaran jelas tentang risiko nyata yang dihadapi oleh organisasi jika kerentanan tersebut tidak segera diatasi. Dengan mengetahui potensi dampak dari sebuah eksploitasi, manajemen dapat mengambil langkah yang lebih tepat untuk menangani risiko tersebut.
- Rekomendasi Perbaikan
  Berdasarkan hasil eksploitasi, seorang pentester dapat memberikan rekomendasi perbaikan yang lebih spesifik dan relevan. Rekomendasi ini biasanya menyasar pada langkah-langkah mitigasi yang tepat untuk memperbaiki kelemahan yang ditemukan, sehingga meningkatkan keamanan sistem secara keseluruhan.
- Evaluasi Efektivitas Pengamanan
  Eksploitasi kerentanan juga digunakan untuk mengevaluasi seberapa efektif mekanisme keamanan yang sudah diterapkan dalam menghadapi serangan nyata. Ini membantu organisasi mengidentifikasi celah dalam sistem pertahanan yang mungkin tidak terlihat dalam pengujian standar.
- Pelatihan dan Edukasi
  Proses eksploitasi kerentanan dapat digunakan sebagai alat untuk melatih tim keamanan dalam merespons insiden keamanan siber secara efektif. Ini mencakup pemahaman tentang bagaimana serangan dilakukan dan bagaimana tindakan pencegahan bisa diterapkan lebih baik di masa mendatang.[10]

Dalam kasus *website e-learning* Politeknik Bhakti Semesta, dilakukan eksploitasi dengan serangan *Cross-Site Scripting (XSS)* menggunakan dua pendekatan, yaitu secara manual dan otomatis. Pendekatan manual dilakukan dengan cara menyuntikkan *script XSS* dan menggunakan *Burp Suite*, sementara pendekatan otomatis dilakukan dengan teknik *reconnaissance (Arjun).*

1. *Manual Injection Script XSS*

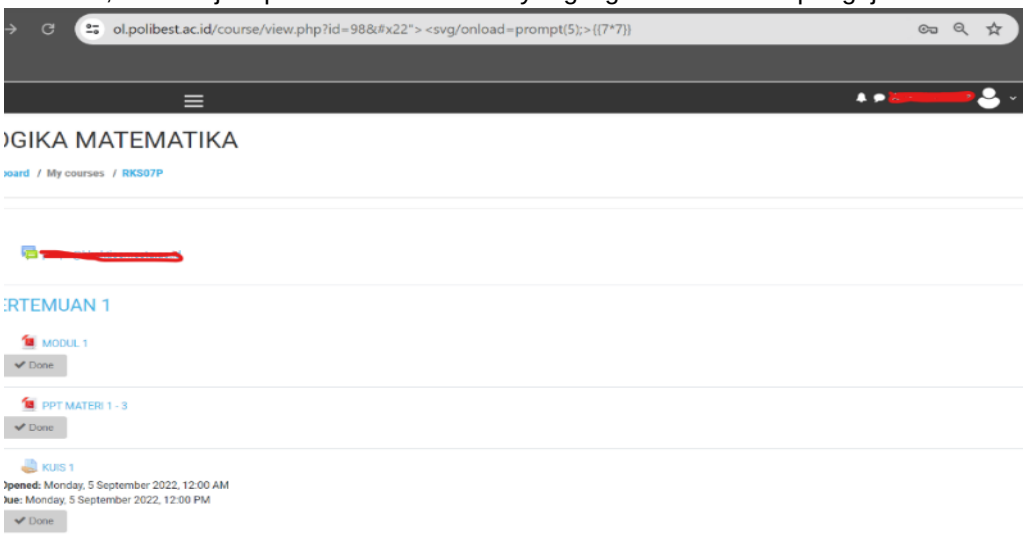
   Pada tahap awal, percobaan dilakukan secara manual dengan menyuntikkan *script XSS* sebagai berikut:



Gambar 7. *Manual Injection XSS*

<img src=x onerror="javascript:alert();">.

Percobaan ini bertujuan untuk membaca jalur rekaman *(record path)* dan *endpoint* saat pengaksesan, serta dilakukan pada akun pengguna mahasiswa. Dari hasil percobaan tersebut, tidak terjadi perubahan atau hasil yang signifikan selama pengujian.



Gambar 8. *Manual Injection XSS*

Langkah berikutnya adalah mencoba memasukkan *script XSS* manual lainnya: *'"><svg/onload=prompt(5);>{{7*7}},* *Script* ini diterapkan pada halaman *course* mahasiswa untuk mendeteksi perubahan atau kerentanan melalui proses *injection script*. Namun, hasil dari percobaan dengan *script* manual ini, sebagaimana ditunjukkan pada gambar di atas, tidak menunjukkan adanya perubahan atau reaksi dari sistem ketika *script* disuntikkan. Hal ini cukup membingungkan karena tidak ditemukan adanya

kerentanan, dan bahkan tidak muncul peringatan dari *Web Application Firewall (WAF)* atau pesan parameter tidak valid dari situs web.

2. *Burpsuite*

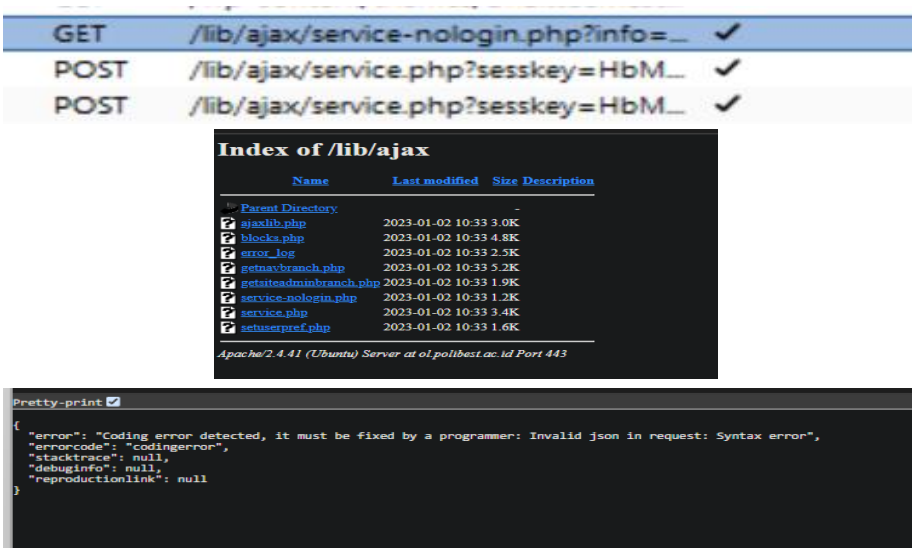
Pada tahap kedua dilakukan *exploitation* menggunakan *tool burpsuite* pengaksesan pada *path /my* dengan parameter yang berisi *username*. Selama proses pengujian, teridentifikasi beberapa *track path* yang muncul selain dari hasil pengaksesan pada *path "/my/"*. Beberapa dari *track path* tersebut dianggap relevan dan menarik untuk ditelusuri lebih lanjut, baik dari sisi *backend* maupun dari sisi komunikasi *client-server*.

https://ol.polibest.ac.id	GET	/my/?username=8315	✓	200	211827	HTML	Dashboard
https://www.bhaktisemesta.a...	GET	/wp-content/themes/bhaktisemest...		301	654	HTML	svg
https://ol.polibest.ac.id	POST	/lib/ajax/service.php?sesskey=HbM...	✓	200	36985	JSON	php
https://ol.polibest.ac.id	POST	/lib/ajax/service.php?sesskey=HbM...	✓	200	146995	JSON	php
https://bhaktisemesta.ac.id	GET	/wp-content/themes/bhaktisemest...		404	104005	HTML	svg
https://ol.polibest.ac.id	GET	/lib/ajax/service-nologin.php?info=...	✓	200	403	JSON	php
https://ol.polibest.ac.id	POST	/lib/ajax/service.php?sesskey=HbM...	✓	200	252165	JSON	php
https://ol.polibest.ac.id	POST	/lib/ajax/service.php?sesskey=HbM...	✓	200	351	JSON	php

Gambar 9. *Exploitation* dengan *Burpsuite*

Selanjutnya, dilakukan upaya untuk mengakses *path* tersebut guna meninjau lebih lanjut isi di balik situs *web*. Namun, akses ke *path* ini memerlukan proses autentikasi menggunakan kredensial mahasiswa, dosen, atau kampus.

Pada *path /lib/ajax,* ditemukan adanya berkas yang mengalami kesalahan *(error)* dan perlu segera diperbaiki oleh tim admin IT kampus. Kesalahan ini berpotensi menjadi salah satu alasan mengapa serangan *XSS* tidak terpicu, kemungkinan akibat adanya *error coding* pada sisi *client-server*.

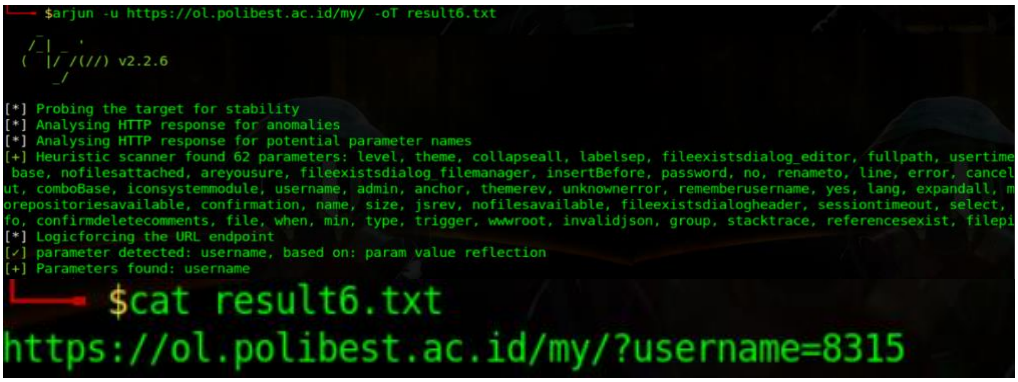


Gambar 9. *Exploitation* pada *path lib/ajax*

3. *Reconnaissance (Arjun)*

Pada tahap ini, proses pengumpulan informasi awal *(reconnaissance)* dilakukan untuk mengidentifikasi potensi kerentanan yang dapat dieksploitasi lebih lanjut. Salah satu alat yang digunakan adalah *Arjun*, yang berfungsi untuk menemukan parameter kueri yang digunakan pada *endpoint URL*. Aplikasi *web* sering kali memanfaatkan

parameter kueri untuk menerima input dari pengguna. Contoh penggunaan parameter kueri adalah sebagai berikut: *http://api.example.com/v1/userinfo?id=751634589*



Gambar 10. *Exploitation* dengan *Arjun*

Dari hasil pencarian *endpoint URL* pada situs *web*, ditemukan puluhan parameter yang digunakan dalam komunikasi aplikasi. Di antara parameter-parameter tersebut, teridentifikasi satu kerentanan yang signifikan, yaitu parameter *"username"*. Hasil keluaran berupa berkas *.txt* yang diperoleh melalui pencarian menggunakan *tool Arjun* mengungkapkan daftar *path* yang rentan terhadap serangan. Berkas tersebut berisi informasi penting terkait *endpoint* dan parameter yang dapat dieksploitasi, termasuk jalur mana saja yang memiliki potensi kerentanan terhadap serangan *Cross-Site Scripting (XSS)* maupun jenis serangan lainnya.

5) Generating Report

Tahapan pembuatan laporan merupakan fase akhir dari proses *vulnerability assessment* dan *penetration testing* yang dilakukan pada sistem *e-learning* Politeknik Bhakti Semesta. Pada fase ini, seluruh hasil identifikasi kerentanan yang ditemukan selama proses pengujian dirangkum dan disusun secara sistematis.

Berikut adalah beberapa poin utama beserta rekomendasi perbaikan yang perlu diperhatikan untuk meningkatkan keamanan sistem di masa mendatang:

1. Kerentanan dalam enkripsi *TLS/SSL* meliputi *Breach (CVE-2013-3587)* dan penggunaan algoritma lama seperti *CBC*. Rekomendasi: Nonaktifkan kompresi *HTTP* dan gunakan algoritma *modern* seperti *GCM*."
2. Kekurangan pada Pengaturan *Header* Keamanan: *Header* keamanan seperti *X-Frame-Options* dan *X-Content-Type-Options* tidak diterapkan dengan benar, sehingga memungkinkan terjadinya serangan *Clickjacking* dan *MIME sniffing*. Rekomendasi: Implementasi *X-Frame-Options: SAMEORIGIN* dan *X-Content-Type-Options: nosniff* untuk mencegah serangan ini.
3. *Token Anti-CSRF* yang tidak ditemukan: Beberapa formulir *HTML* tidak dilengkapi dengan token *Anti-CSRF*, yang membuka peluang untuk serangan *Cross-Site Request Forgery (CSRF)*. Rekomendasi: Pastikan setiap formulir *HTML* menggunakan *token Anti-CSRF* guna mencegah serangan *CSRF*.
4. *Cookie* Tanpa Atribut Keamanan: Terdapat *cookie* yang tidak menggunakan *atribut HttpOnly* dan *SameSite*, sehingga rentan terhadap serangan *Cross-Site Scripting (XSS)* dan pencurian *session*. Rekomendasi: Terapkan atribut *HttpOnly* dan *SameSite* pada semua *cookie* untuk melindungi data pengguna dari akses yang tidak sah.
5. Penggunaan Protokol *HTTPS* yang Tidak Konsisten: Beberapa halaman masih menggunakan konten yang dimuat melalui *HTTP*, sehingga rentan terhadap serangan *Man-in-the-Middle*. Rekomendasi: Terapkan *HTTPS* di seluruh halaman dan tambahkan *header Strict-Transport-Security (HSTS)* untuk memastikan semua permintaan menggunakan protokol yang aman.
6. Eksposur Informasi Versi *Server*: *Server web* mengekspos informasi versi yang digunakan, yang dapat dimanfaatkan oleh penyerang untuk menemukan kerentanan lain. Rekomendasi: Sembunyikan informasi versi *server* dengan mengonfigurasi *ServerTokens* dan *ServerSignature*.

7. Kerentanan terhadap Serangan *Cross-Site Scripting (XSS):* Beberapa parameter seperti *username* masih memiliki potensi kerentanan terhadap serangan *XSS*. Rekomendasi: Implementasikan validasi input yang lebih ketat dan terapkan *filter XSS* di sisi *server* dan klien.
8. Implementasi *Web Application Firewall (WAF):* Tidak ada reaksi yang signifikan dari *Web Application Firewall (WAF)* selama pengujian, menunjukkan bahwa sistem *WAF* tidak diaktifkan atau tidak berfungsi optimal. Rekomendasi: Aktifkan dan konfigurasikan *WAF* untuk memblokir serangan *XSS, SQL Injection,* dan serangan berbasis *website* lainnya.
9. Penggunaan *Platform Moodle: Header* perangkat lunak menunjukkan bahwa *website* menggunakan *platform Moodle* yang rentan jika tidak diperbarui secara berkala. Rekomendasi: Selalu perbarui *Moodle* dan modul terkait untuk memastikan kerentanan versi lama telah diperbaiki.

Kesimpulan

Hasil dari pendekatan *Vulnerability Assessment* dan *Penetration Testing* menunjukkan bahwa ada kelemahan pada *header* keamanan, seperti tidak adanya implementasi yang benar terhadap *X-Frame-Options* dan *X-Content-Type-Options*, yang membuat sistem rentan terhadap serangan *Clickjacking* dan *MIME sniffing*. Selain itu, beberapa *cookie* yang tidak dilengkapi dengan *atribut HttpOnly* dan *SameSite* berpotensi dimanfaatkan oleh penyerang untuk melakukan serangan *session hijacking* dan serangan *XSS*. Pengujian juga menemukan bahwa *website e-learning* masih memiliki konten campuran yang diakses melalui *HTTP*, bukan *HTTPS*, yang membuka peluang untuk serangan *Man-in-the-Middle*. Selain itu, *server web* secara eksplisit mengekspos informasi versi yang digunakan, memberikan celah bagi penyerang untuk mengeksploitasi kerentanan yang terkait dengan perangkat lunak tersebut. *Penetration Testing* dilakukan untuk memvalidasi apakah kerentanan yang ditemukan benar-benar dapat dieksploitasi.

Eksploitasi ini menunjukkan bahwa ada potensi risiko keamanan nyata, meskipun beberapa serangan *XSS* yang dicoba tidak berhasil memicu reaksi yang diharapkan. Hal ini disebabkan kemungkinan adanya *error* pada sisi *client-server* atau kelemahan dalam konfigurasi *Web Application Firewall (WAF)* yang tidak optimal. Dengan menerapkan rekomendasi perbaikan yang telah rumuskan, sistem *e-learning* Politeknik Bhakti Semesta diharapkan menjadi lebih aman terhadap ancaman siber."

Daftar Pustaka

[1] Muhammad Agreindra Helmiawan, “Keamanan E-Learning Menggunakan Metode Square (Studi Kasus Stmik Sumedang),” no. February 2013, 2018, doi: 10.13140/RG.2.2.31309.33761.

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

[3] A. Dharmawan, Y. Prihati, and H. Listijo, “Penetration testing menggunakan OWASP top 10 pada domain xyz.ac.id,” *Jelc*, vol. 8, no. 1, pp. 1–9, 2022.

[4] M. A. Aziz, “Vulnerability Assesment Untuk Mencari Celah Keamanan Web Aplikasi E-Learning Pada Universitas Xyz,” *J. Eng. Comput. Sci. Inf. Technol.*, vol. 2, no. 1, 2023, doi: 10.33365/jecsit.v1i1.13.

[5] M. Fronita, S. Informasi, S. Teknologi, and U. I. N. S. Riau, “21823-68674-2-Pb,” vol. 9, no. 1, pp. 1–7, 2023.

[6] D. Bayu Rendro and W. Nugroho Aji, “ANALISIS MONITORING SISTEM KEAMANAN JARINGAN KOMPUTER MENGGUNAKAN SOFTWARE NMAP (STUDI KASUS DI SMK NEGERI 1 KOTA SERANG),” vol. 7, no. 2, 2020.

[7] M. Dewi, A. Budiono, and U. Y. K. S. Hediyanto, “Vulnerability Assessment pada Website Rekruitasi Asisten (IRIS) Fakultas Rekayasa Industri menggunakan Nikto dan Nessus,” *e-Proceeding Eng.*, vol. 10, no. 2, pp. 1631–1636, 2023.

[8] M. Gibran, A. Danialdo, F. A. Bakhtiar, and M. Data, “Pengujian Efektivitas OWASP ZAP dalam Menemukan Kerentanan dari Metasploitable,” vol. 7, no. 7, pp. 3431–3433, 2023.

[9] J. J. B. H. Yum Thurfah Afifa Rosaliah, “Pengujian Celah Keamanan Website Menggunakan Teknik Penetration Testing dan Metode OWASP TOP 10 pada Website SIM,” *Senamika*, vol. 2, no. September, pp. 752–761, 2021.

[10] M. L. B. Hikam, F. Dewi, and D. Praditya, “Analisis Manajemen Risiko Informasi Menggunakan Iso/Iec 27005:2018 (Studi Kasus: Pt.Xyz),” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 2, pp. 728–734, 2024, doi: 10.29100/jipi.v9i2.4709.