Analisis Celah Keamanan Pada *Website* PDDIKTI Menggunakan Metode *Penetration Testing* Dan *Framework ISSAF*

Nabila Daniasti Darno [1] **, Edwin Lesmana Tjiong** [2]

[1,2] Informatika, Fakultas Ilmu Komputer dan Desain, Universitas Kalbis
Jalan Pulomas Selatan Kav. 22, Jakarta 13210
[1] *Email: daniastinab@gmail.com*
[2] *Email: edwin.tjiong@kalbis.ac.id*

Abstract: *Website security is essential for maintaining the integrity, confidentiality, and availability of data, especially in systems that store sensitive information such as the Higher Education Database (PDDikti). As an official platform for higher education data in Indonesia, PDDikti is vulnerable to cyberattacks like SQL Injection, Cross-Site Scripting (XSS), and Denial of Service (DoS). This study aims to analyze security vulnerabilities on the PDDikti website using Penetration Testing combined with the Information System Security Assessment Framework (ISSAF). The research follows three main phases: Planning, Assessment, and Reporting. Findings from the Assessment phase reveal critical issues in user authentication and data encryption. These are further analyzed to provide security improvement recommendations. This study is expected to enhance the security of the PDDikti system and serve as a reference for similar systems to conduct regular security evaluations.*
Keywords: *website security, pddikti, penetration testing, issaf, vulnerability.*

Abstrak: *Keamanan website merupakan aspek krusial dalam menjaga integritas, kerahasiaan, dan ketersediaan data, terutama pada sistem informasi yang menyimpan data sensitif seperti Pangkalan Data Pendidikan Tinggi (PDDikti). Sebagai platform resmi penyedia data pendidikan tinggi di Indonesia, PDDikti rentan terhadap serangan siber seperti SQL Injection, Cross-Site Scripting (XSS), dan Denial of Service (DoS). Penelitian ini bertujuan menganalisis kerentanan keamanan pada website PDDikti menggunakan metode Penetration Testing yang dipadukan dengan kerangka kerja Information System Security Assessment Framework (ISSAF). Penelitian dilakukan melalui tiga fase utama: Perencanaan, Penilaian, dan Pelaporan. Hasilnya menunjukkan adanya celah kritis pada autentikasi pengguna dan enkripsi data. Temuan ini kemudian dianalisis untuk memberikan rekomendasi perbaikan. Penelitian ini diharapkan dapat meningkatkan keamanan sistem informasi PDDikti serta menjadi acuan bagi pengelola sistem serupa dalam melakukan evaluasi keamanan secara berkala.*
Kata kunci: *keamanan website, pddikti, penetration testing, issaf, kerentanan.*

I. PENDAHULUAN

Perkembangan teknologi informasi yang berkembang pesat telah mendorong transformasi digital dalam berbagai aspek kehidupan manusia. Salah satu teknologi yang menjadi bukti perkembangan teknologi adalah website (situs), yang merupakan kumpulan halaman web yang saling terhubung dan berada dibawah satu domain yang dapat diakses melalui internet menggunakan browser [1]. Website biasanya dijadikan sebagai platform digital bagi individu, organisasi, atau perusahaan untuk menyajikan informasi, menawarkan layanan bahkan melakukan transaksi secara online.Situs web *(website)* merupakan salah satu produk teknologi yang saat ini memiliki peran penting dalam penyampaian informasi, komunikasi dan layanan publik. Penggunaan *website* tidak hanya sebatas pada sektor komersial, tetapi juga merambah ke sektor pemerintahan, pendidikan, dan pelayanan sosial. Seiring dengan semakin luasnya penggunaan *website,* isu mengenai keamanan siber menjadi hal yang tidak dapat diabaikan. Keamanan *website* tidak hanya berkaitan dengan perlindungan data pengguna, namun juga berpengaruh terhadap kredibilitas dan kepercayaan publik terhadap suatu sistem informasi [2]. Salah satu platform penting dalam sektor pendidikan di Indonesia adalah *website* pangkalan data pendidikan tinggi (PDDikti). PDDikti sendiri merupakan sebuah sistem informasi yang dikelola oleh Direktorat Jendral Pendidikan Tinggi, Riset dan Teknologi Republik

Indonesia [3]. PDDikti menyediakan data dan informasi mengenai institusi pendidikan tinggi, mahasiswa, dosen, kurikulum, serta kegiatan akademik lainnya. Informasi yang disediakan oleh pddikti digunakan sebagai dasar dalam pengambilan kebijakan pendidikan tinggi di Indonesia dan menjadi acuan bagi berbagai pemangku kepentingan, mulai dari mahasiswa, dosen, institusi pendidikan, hingga pemerintah.

Namun, seiring dengan meningkatnya penggunaan PDDikti, potensi ancaman terhadap keamanan sistem ini juga turut meningkat. Ancaman yang dapat menggangu integritas dan ketersediaan data dalam sistem PDDikti antara lain:

- SQL Injection
- Cross-site Scripting (XSS)
- Distributed Denial of Service (DDoS)

Berdasarkan data dari badan siber dan sandi negara (BSSN), pada tahun 2022 terjadi lebih dari 370 juta serangan siber di Indonesia, dengan sektor pemerintah yang menjadi salah satu target utama [4]. Oleh karena itu, upaya identifikasi dan mitigasi celah keamanan pada sistem seperti PDDikti sangat penting dilakukan untuk menjaga keberlangsungan dan keamanan layanan yang disediakan.

Salah satu metode yang dapat digunakan untuk mengidentifikasikan celah keamanan pada sistem informasi adalah *penetration testing. Penetration testing* adalah metode yang digunakan untuk mengevaluasi keamanan suatu sistem atau jaringan komputer dengan melakukan simulasi serangan [5]. Tujuan dari uji penetrasi ini adalah untuk menemukan dan mengevaluasi kerentanan yang ada dalam sistem sebelum dimanfaatkan oleh penyerang sesungguhnya. Dalam konteks penelitian ini, *penetration testing* dilakukan dengan mengacu pada kerangka kerja *Information Systems Security Assessment Framework (ISSAF). ISSAF* memberikan pendekatan sistematis dalam proses perencanaan, pelaksanaan, dan pelaporan uji keamanan informasi.

Sebelumnya, sudah ada penelitian lain yang memiliki tujuan yang sama, yaitu menganalisis kerentanan keamanan pada laman PDDikti yang dilakukan oleh Hassanah, Ryansyah, Setiawan, dan Alamsyah pada 2025. Fokus penelitian ini bertujuan mengidentifikasi kerentanan pada halaman PDDikti menggunakan kombinasi OWASP ZAP dan pengujian manual [6].

Peneliti sebelumnya secara konsisten menyoroti pentingnya analisis keamanan pada *website* menggunakan metode *Penetration Testing* dan *Framework ISSAF* [7], sehingga melalui penelitian ini, peneliti akan melakukan analisis terhadap celah keamanan pada *website* PDDIKTI dengan pendekatan *Penetration Testing* berbasis *framework ISSAF*. Penelitian ini tidak hanya berfokus pada pengujian teknis, tetapi juga mempertimbangkan aspek etika dan perizinan dalam pelaksanaan pengujian terhadap sistem milik instansi pemerintah. Dengan demikian, hasil dari penelitian ini diharapkan dapat memberikan kontribusi nyata dalam meningkatkan keamanan sistem informasi di sektor pendidikan tinggi Indonesia.

A. Rumusan Masalah

Berdasarkan latar belakang diatas, maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana metode *penetration testing* dapat diterapkan dalam menganalisis celah keamanan pada *website* PDDIKTI?
2. Apa saja jenis dan tingkat kerentanan keamanan yang ditemukan pada *website* PDDIKTI berdasarkan hasil *penetration testing*?
3. Bagaimana *framework ISSAF* dapat membantu dalam pelaksanaan *penetration testing* secara sistematis dan terstruktur?

B. Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut:

1. Untuk menerapkan metode *penetration testing* dalam mengidentifikasi dan menganalisis celah keamanan pada *website* PDDIKTI.
2. Untuk mengetahui jenis dan tingkat kerentanan yang terdapat pada sistem PDDIKTI.
3. Untuk mengevaluasi efektivitas penerapan *framework* ISSAF dalam pelaksanaan *penetration testing* pada sistem informasi pemerintahan.

Dengan tercapainya tujuan-tujuan tersebut, diharapkan hasil penelitian ini dapat digunakan sebagai referensi dan masukan dalam peningkatan

keamanan sistem PDDIKTI serta menjadi acuan dalam pengembangan kebijakan keamanan informasi pada sistem serupa di lingkungan pendidikan tinggi.

II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan studi kasus dengan menggunakan metode penetration testing berbasis pada kerangka kerja *ISSAF (Information System Security Assesment Framework)*. Pendekatan ini dipilih untuk melakukan pengujian terhadap sistem keamanan pada website PDDikti untuk menemukan celah keamanan yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab.

Penelitian ini menggunakan pendekatan studi kasus dengan menggunakan metode *penetration testing*. Metode ini bertujuan untuk melakukan simulasi serangan siber terhadap sistem target yang dituju secara terkendali guna untuk mengidentifikasi celah keamanan. Metode ini akan diselaraskan dengan kerangka kerja *ISSAF*, yang menyediakan panduan sistematis dengan melakukan pengujian keamanan secara menyeluruh.

A. Kerangka Kerja *ISSAF*

ISSAF merupakan kerangka kerja yang dikembangkan oleh *OISSG (Open Information Systems Security Group),* kerangka kerja ini memiliki beberapa fase penting dalam melakukan pengujian, yaitu;

1. *Planning and Preparation,*
2. *Assesment,*
3. *Reporting.*

Setiap fase memiliki tujuan dan tahapan-tahapan khusus yang akan dijelaskan lebih lanjut di subbab berikut ini.

B. Tahapan Penelitian

Berikut ini adalah tahapan pelaksanaan penelitian berdasarkan metode penetration testing dan sistematis mengikuti kerangka kerja ISSAF.

1. *Planing and Preparation*

Fase *Planing and Preparation* merupakan fase awal, fase ini mencakup langkah-langkah untuk bertukar informasi, merencanakan, dan mempersiapkan pengujian. Tahap ini melibatkan mendapatkan informasi awal, merencanakan proses pengujian, menentukan metodologi, dan mendapatkan persetujuan untuk kasus pengujian spesifik [8]. Persetujuan penilaian akan ditandatangani oleh kedua belah pihak untuk memberikan perlindungan hukum [8]. Dalam penelitian ini, situs yang menjadi objek pengujian adalah laman Pangkalan Data Pendidikan Tinggi (PDDikti). Kegiatan utama yang dilakukan pada fase ini mencakup:

1. Perolehan izin dan persetujuan etis dari pihak-pihak terkait atas pelaksanaan kegiatan pengujian keamanan.
2. Identifikasi target sistem, yang meliputi penentuan struktur sistem dan cakupan teknis dari situs PDDikti yang akan diuji.
3. Perumusan ruang lingkup pengujian, untuk menetapkan batasan-batasan pengujian yang relevan dengan tujuan penelitian.
4. Penentuan perangkat atau *tools* yang akan digunakan dalam proses pengujian, baik *tools* otomatis maupun semi-otomatis.
5. Pengumpulan dokumen pendukung, seperti dokumentasi sistem, struktur halaman, dan informasi teknis lainnya yang dibutuhkan dalam proses pengujian. Namun dikarenakan dokumen pendukung ini bersifat *confidential* maka penulis tidak dapat melampirkan ke dalam penelitian ini.

Untuk memperoleh informasi yang akurat dan relevan dalam mendukung kegiatan di atas, penulis melakukan diskusi secara luring (tatap muka) dengan *Product Owner* dan *Project Manager* (PM) dari laman PDDikti. Diskusi ini bertujuan untuk mendapatkan pemahaman yang komprehensif terkait sistem yang diuji, serta memastikan bahwa seluruh kegiatan pengujian dilakukan sesuai dengan prinsip etika dan kebijakan yang berlaku.

2. *Assessment*

Fase ini adalah fase pelaksanaan uji penetrasi dengan pendekatan bertingkat untuk memberikan akses yang lebih luas ke aset informasi yang diinginkan [8]. Adapun sub tahapan yang akan dilakukan antara lain:

- *Information Gathering* Pada tahap ini, peneliti melakukan proses pengumpulan informasi awal terhadap target sistem, yaitu laman PDDikti yang beralamat di

https://pddikti.kemdiktisaintek.go.id/. Informasi yang dikumpulkan meliputi data teknis seperti *IP address,* informasi pemilik *domain*, konfigurasi *SSL/TLS,* dan teknologi web yang digunakan. Proses ini bertujuan untuk memperoleh pemahaman menyeluruh terhadap sistem sebelum dilakukan *network mapping* dan *vulnerability assesment.*

- *Network Mapping* tahap ini dilakukan untuk mengidentifikasi struktur jaringan dan layanan yang berjalan di *server* target, dalam hal ini *website* PDDikti (https://pddikti.kemdiktisaintek.go.id). Tujuan dari proses ini adalah untuk mengetahui *port-port* mana saja yang terbuka serta jenis layanan (*service*) apa yang berjalan di balik port tersebut. Informasi ini sangat penting untuk menentukan potensi celah keamanan yang bisa dieksplorasi pada tahap pengujian selanjutnya.
- *Vulnerability Assesment* Pada fase ini peneliti akan menggunakan metode pemindaian otomatis untuk mendeteksi komponen atau konfigurasi yang rentan terhadap eksploitasi. Dalam penelitian ini, peneliti menggunakan *tool OWASP ZAP (Zed Attack Proxy),* yang merupakan salah satu tools open-source paling umum digunakan untuk menemukan kerentanan aplikasi web.
- Tahapan *Penetration* merupakan fase terakhir dalam proses *Assessment* pada *framework ISSAF*. Setelah kerentanan berhasil diidentifikasi melalui proses *vulnerability assessment,* tahap ini bertujuan untuk menguji dan memverifikasi apakah celah-celah tersebut dapat benar-benar dieksploitasi dalam situasi nyata. Penyerangan dilakukan dalam lingkungan yang terkendali dan tidak merusak sistem. Dalam penelitian ini, peneliti melakukan tiga jenis pengujian penetrasi yang umum pada aplikasi web, yaitu *XSS injection, SQL injection* dan *Distributed Denial of Service.* Dengan mensimulasikan skenario serangan nyata, tahap *penetration testing* ini memberikan wawasan langsung terhadap risiko yang dihadapi sistem, serta menguji efektivitas mekanisme pertahanan yang telah diterapkan oleh pengelola sistem

3. *Reporting*

Tahapan ini merupakan tahap akhir, tahap ini melibatkan analisis dan pembuatan laporan dari hasil pengujian penetrasi yang telah dilakukan berdasarkan Framework ISSAF [9]. Elemen-elemen yang disusun dalam laporan mencakup:

- Ringkasan temuan,
- Deskripsi masing-masing celah keamanan,
- Tingkat keparahan (*severity*) berdasarkan standar CVSS,
- Bukti eksploitasi berupa tangkapan layar (*screenshot*) atau *log*,
- Rekomendasi mitigasi untuk setiap kerentanan yang ditemukan.

Laporan hasil pengujian kemudian dikirimkan kepada pihak PDDikti. Mengingat isi laporan bersifat rahasia (*confidential*) antara peneliti dan pihak PDDikti, maka peneliti juga meminta umpan balik secara langsung dari pihak PDDikti untuk menilai kejelasan, kelengkapan, serta relevansi laporan yang telah disampaikan

4. Alat Yang Digunakan

Beberapa *tools* yang digunakan peneliti dalam penelitian ini antara lain:

Tabel 1 Alat Yang Digunakan

NO	NAMA TOOLS	FUNGSI
1	Whois	Mendapatkan informasi mendetail mengenai laman PDDikti.
2	Sslscan	Memindai dan mengidentifikasi protocol SSL/TLS yang didukung oleh server laman PDDikti.
3	Dnscan	Mendapatkan informasi mengenai domain apa saja yang dimiliki oleh laman PDDikti.
4	Infoga	Mendapatkan informasi mengenai email dari laman PDDikti.
5	Whatweb	Mendapatkan informasi mengenai cms yang digunakan oleh laman PDDikti
6	Nmap	Mendapatkan informasi mengenai port mana saja yang terbuka dan juga untuk mengetahui layanan yang digunakan.
7	OWASP ZAP	Mencari kerentanan pada website laman PDDikti.

Untuk menjalankan alat-alat tersebut, penulis menggunakan sistem operasi Kali Linux karena Kali Linux telah digunakan secara luas dalam pelatihan keamanan siber, termasuk dalam kursus

pengujian penetrasi profesional. Alat-alatnya memungkinkan pengujian keamanan tingkat lanjut seperti serangan Man-in-the-Middle, analisis jaringan nirkabel, dan injeksi SQL otomatis [10].

5. Teknik Analisis Data

Data yang diperoleh dari hasil pengujian akan dianalisis secara kualitatif dengan mengacu pada standar kemanan aplikasi *web* dan kerangka kerja *ISSAF*. Setiap temuan yang ditemukan akan dikategorikan berdasarkan tingkat resiko, jenis kerentanan, serta potensi hingga dampaknya terhadap sistem.

III. HASIL DAN PEMBAHASAN

A. Information Gathering

Pada bagian ini menampilkan seluruh hasil proses analisis celah keamanan.

1. Mendapatkan IP Address Target

Guna mendapatkan ip address pada laman pada laman https://pddikti.kemdiktisaintek.go.id/ peneliti menggunakan alat (tools) terminal yang sudah disediakan oleh kali linux. Dengan menjalankan perintah ping https://pddikti.kemdiktisaintek.go.id/ maka secara otomatis peneliti mendapatkan informasi mengenai alamat IP yang digunakan pada https://pddikti.kemdiktisaintek.go.id/ yaitu 1*3.5*.19*.**5. Alamat IP ini dapat digunakan untuk mengakses laman dari https://pddikti.kemdiktisaintek.go.id/.

2. Mendapatkan Informasi Target

Guna mendapatkan informasi lebih lanjut mengenai target, peneliti menggunakan *tools whois* Untuk menggunakan *tools* ini peneliti menuliskan perintah *whois* 1*3.5*.19*.**5 pada terminal kali linux. Informasi yang didapatkan dapat dilihat pada tabel dibawah ini.

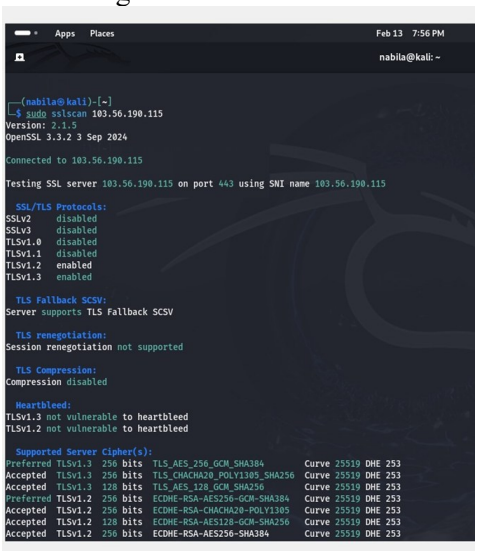
Tabel 2 Informasi Website

DATA	NILAI
NAMA	Deny Kurniawan
NAMA PERUSAHAAN	Kementrian Riset, Teknologi dan Pendidikan Tinggi.
ALAMAT	Jl. Raya Jendral Sudirman Pintu I, Senayan Jakarta, 10270.
EMAIL	dkurniawan@dikti.go.id
NO. HP	082157946074

Hasil dari pengujian menggunakan *tools whois*, peneliti mendapatkan informasi berupa nama perusahaan penyedia layanan *website*, alamat, email serta no hp.

3. *SSL (Secure Socket Layer)*

Guna dapat melakukan pengujian ini, peneliti menggunakan alat *sslscan* sehingga dengan mengetikkan perintah *sslscan*. https://pddikti.kemdiktisaintek.go.id di terminal pada kali linux. Maka peneliti mendapatkan informasi sebagai berikut:



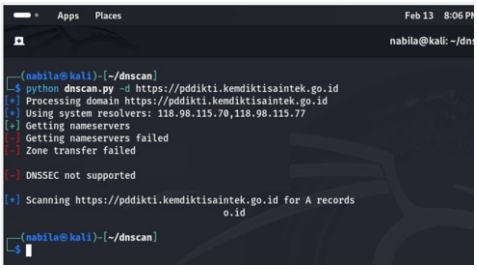
Gambar 1 hasil ssl scan

Dari pengujian ini didapatkan hasil bahwa website tersebut menggunakan TLSv1.2 serta TLSv1.3. TLSv1.2 dan TLSv1.3 merupakan sebuah protokol kriptografi yang digunakan untuk keamanan dalam berkomunikasi antara *client* dengan *web server* dalam jaringan internet. Perbedaan antara kedua protocol tersebut yaitu pada kecepatan dalam merespon permintaan client serta tingkat keamanan yang diberikan. Protocol ini bekerja ketika client melakukan request kepada web server kemudian webserver akan memastikan bahwa kunci yang diberikan oleh client sesuai dengan yang dimiliki oleh server. Selanjutnya server akan mengirim permintaan client tersebut. Serta dari pengujian ter-sebut, peneliti menemkan informasi bahwa

website tersebut telah memiliki sertifikat ssl yang ber-laku sampai dengan 3 Desember 2025.

4. Mendapatkan DNS Target

Guna mendapatkan *domain* apa saja yang dimiliki oleh *website* https://pddikti.kemdiktisaintek.go.id, peneliti menggunakan alat dnscan. Untuk menjalankan alat ini peneliti perlu *mengclone repository* dnscan yang ada pada github, lalu setelah itu user menjalankan perintah python dnscan.py -d https://pddikti.kemdiktisaintek.go.id. Dibawah ini merupakan hasil dari perintah yang di jalankan:

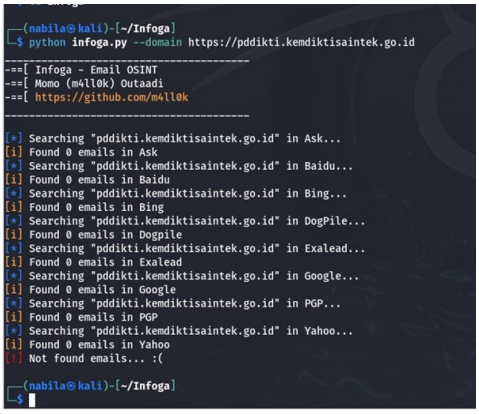


Gambar 2 Hasil DNS Scan

Dari hasil yang didapatkan, peneliti tidak menemukan domain yang berkaitan dengan laman https://pddikti.kemdiktisai.ntek.go.id.

5. Mendapatkan Informasi Mengenai Email

Langkah selanjutnya yang akan peneliti lakukan adalah mendapatkan informasi mengenai *email*, untuk mendapatkan informasi tersebut peneliti menggunakan alat yang bernama infoga, sama seperti dnscan, peneliti perlu *mengclone* infoga melalui github *repository* yang menaungi alat tersebut. Setelah itu peneliti dapat menjalankan perintah python infoga.py –domain https://pddikti.kemdiktisaintek.go.id/ pada terminal di kali linux. Dibawah ini

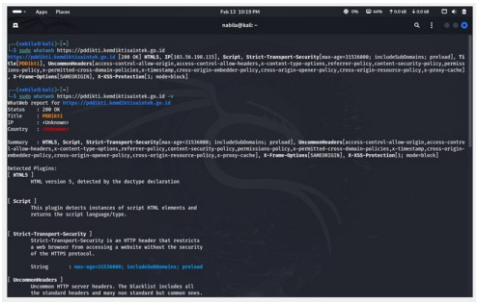


Gambar 3 Hasil Infoga

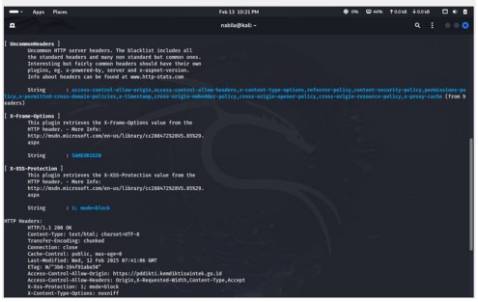
Merupakan gambar dari hasil pengujian yang didapatkan. Dapat dilihat hasil dari pengujian ini tidak terdapat informasi mengenai *email* yang ada pada laman https://pddikti.kemdiktisaintek.go.id/.

6. Identifikasi CMS

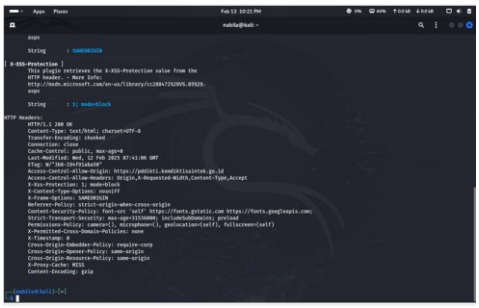
Tahapan ini merupakan tahapan terakhir dalam pengumpulan informasi yang dilakukan oleh peneliti, pada tahap ini peneliti menggunakan alat bernama whatweb. Tujuan dari pengujian ini adalah ingin mengetahui cms yang digunakan oleh laman https://pddikti.kemdiktisaintek.go.id/. Berikut ini adalah hasil dari pengujian yang telah dilakukan:



Gambar 4 Hasil Identifikasi CMS 1



Gambar 5 Hasil Identifikasi CMS 2



Gambar 6 Hasil Identifikasi CMS 3

B. Network Mapping

Pengujian ini dilakukan untuk mengetahui *port* mana saja yang terbuka dan juga untuk mengetahui jenis layanan yang digunakan. Pengujian ini dilakukan dengan menggunakan alat bernama nmap. Dengan mengetikkan perintah nmap -sV 10*.5*.1*0.1** pada terminal di kali linux maka peneliti dapat merangkum informasi yang didapatkan dari pengujian ke dalam table dibawah ini:

Tabel 3 Informasi Port dan Service Yang Terbuka

NO	PORT	SERVICE
1	80/tcp	http
2	443/tcp	Ssl/ http
3	3000/tcp	http
4	8010/tcp	Xmpp

Dalam tabel tersebut terdapat port 80/tcp yang merupakan port yang digunakan untuk layanan HTTPS (*Hypertext Transfer Protocol Secure*), https sendiri merupakan layanan yang digunakan untuk mengamankan data yang dikirim antara pengguna dengan situs web. Selanjutnya, terdapat *port* 3000/tcp yang digunakan untuk layanan PPP, PPP (*point-to-point protocol*) sendiri merupakan layanan yang digunakan untuk mengangkut data *multiprotocol*, biasanya *protokol* ini digunakan untuk membuat koneksi antara dua *router* secara langsung. Dan *port* terakhir yang terbuka adalah *port 8010/tcp port* satu ini digunakan untuk layanan Xmpp (*Extensible Messaging and Presence Protocol*), merupakan layanan atau protokol komunikasi terbuka yang berbasis XML (*Extensible Markup Language*). XMPP digunakan untuk mengirim pesan instan (IM), informasi kehadiran, dan memelihara daftar kontak.

C. Vulnerability Identification

Untuk mencari kerentanan pada website, peneliti menggunakan alat yang bernama OWASP ZAP. Dari identifikasi kerentanan yang dihasilkan dari menggunakan alat OWASP ZAP. Peneliti Berhasil mengidentifikasi 9 kerentanan pada laman https://pddikti.kemdiktisaintek.go.id/. Berikut ini rangkuman dari hasil yang didapatkan berserta dengan tingkatan kerentanan:

Tabel 4 Jenis dan Tingkatan Kerentanan.

JENIS KERENTANAN	LEVEL KERENTANAN
VULNERABLE JS LIBRARY	High
CSP: WILDCARD DIRECTIVE (9)	Medium
CSP: SCRIPT-SCR UNSAFE-INLINE	Medium
CSP: STYLE-SRC UNSAFE-INLINE (9)	Medium
HIDDEN FILE FOUND (4)	Medium
TIMESTAMP DISCLOSURE-UNIX (210)	Low
INFORMATION DISCLOSURE - SUSPICIOUS COMMENTS	Informational
MODERN WEB APPLICATION (9)	Informational
RE-EXAMINE CACHE-CONTROL DIRECTIVES (11)	Informational

Setelah melakukan pemindaian kerentanan pada laman menggunakan OWASP ZAP, peneliti menemukan bahwa setiap kerentanan yang teridentifikasi dapat membuka celah bagi berbagai jenis serangan yang membahayakan keamanan laman PDDIKTI.

Kerentanan dengan tingkat paling tinggi adalah *Vulnerable JS Library*, yaitu penggunaan pustaka JavaScript yang memiliki celah keamanan. Pustaka yang sudah usang atau tidak diperbarui dapat dieksploitasi oleh penyerang untuk menyusup ke dalam aplikasi web. Serangan yang dapat terjadi akibat kerentanan ini meliputi *Cross-Site Scripting (XSS), Remote Code Execution (RCE),* dan *Supply Chain Attack.*

Selanjutnya, terdapat beberapa kerentanan dengan tingkat risiko menengah (*medium*), antara lain:

1. ***Content Security Policy (CSP)* menggunakan *wildcard (**)***
   Penggunaan simbol *wildcard (**) dalam aturan direktif *CSP* memungkinkan pemuatan sumber daya dari domain yang tidak terpercaya. Hal ini meningkatkan risiko serangan seperti *Cross-Site Scripting (XSS)* dan *code injection*.
2. ***CSP* mengizinkan *unsafe-inline* dalam direktif *script-src***
   Dengan mengizinkan *unsafe-inline*, skrip yang ditulis langsung dalam tag <script> dapat dieksekusi. Hal ini membuka peluang bagi serangan *XSS* karena penyerang dapat menyisipkan skrip berbahaya ke dalam halaman web, yang berpotensi menyebabkan *data theft*.

3. ***CSP: style-src unsafe-inline*** Kerentanan ini memungkinkan penggunaan gaya *inline* (misalnya, <style> atau atribut style= pada elemen HTML). Hal ini dapat dieksploitasi dalam serangan *XSS berbasis CSS* serta meningkatkan risiko *clickjacking*.
4. ***Hidden File Found*** Kerentanan ini terjadi ketika terdapat file tersembunyi yang dapat diakses oleh pengguna atau penyerang. File ini bisa berisi informasi sensitif atau konfigurasi yang seharusnya tidak dapat diakses dari luar, sehingga meningkatkan risiko eksploitasi.

Sementara itu, terdapat satu jenis kerentanan dengan tingkat risiko rendah (low), yaitu: ***Timestamp Disclosure – Unix*** yaitu Sistem mengungkapkan informasi tentang *timestamp Unix*, yang dapat memberikan wawasan kepada penyerang mengenai waktu pembuatan atau modifikasi suatu sumber daya. Informasi ini bisa dimanfaatkan untuk serangan *Information Leakage* atau *Predictable Session ID*.

Selain itu, ada beberapa kerentanan yang tergolong dalam kategori informational, di antaranya:

1. ***Information Disclosure - Suspicious Comments*** Kerentanan ini terjadi ketika terdapat komentar mencurigakan dalam kode sumber yang mungkin mengandung informasi sensitif, seperti kredensial, jalur sistem, atau petunjuk untuk eksploitasi lebih lanjut. Hal ini dapat menyebabkan serangan *Information Leakage* dan *Reconnaissance*.
2. ***Modern Web Application*** Meskipun ini bukan merupakan sebuah kerentanan, informasi ini menunjukkan bahwa aplikasi menggunakan teknologi web modern. Namun, hal ini tetap dapat dimanfaatkan oleh penyerang untuk menyesuaikan metode serangan mereka.
3. ***Re-examine Cache-Control Directives*** Kerentanan ini mengindikasikan bahwa konfigurasi *Cache-Control* perlu diperiksa ulang untuk memastikan data sensitif tidak di-*cache* secara tidak aman oleh browser atau *proxy*. Risiko yang dapat timbul akibat kerentanan ini meliputi *Data Exposure* dan *Session Hijacking*.

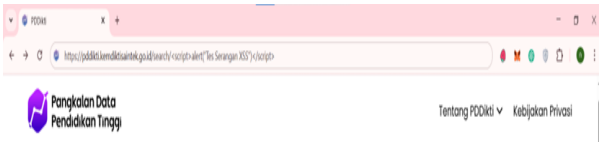
Untuk mengurangi risiko serangan yang disebabkan oleh kerentanan tersebut, perlu dilakukan penerapan langkah-langkah keamanan yang lebih ketat, seperti:

1. Pembaruan pustaka secara berkala.
2. Penguatan kebijakan Content Security Policy (CSP).
3. Pemeriksaan kode dan konfigurasi aplikasi secara berkala untuk memastikan keamanan sistem tetap terjaga.

D. Penetration Testing

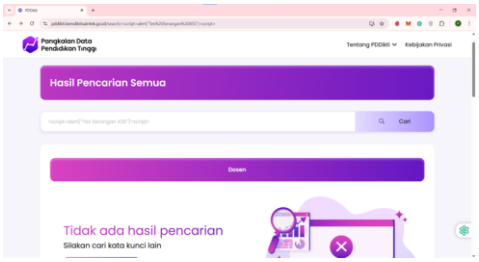
1. XSS Injection

Serangan XSS merupakan jenis serangan yang dilakukan dengan cara menginput script javascript ke dalam website PDDikti untuk melihat apakah script yang diinput dapat dijalankan atau malah membalikan nilai kembali ke sebelumnya. Serangan ini dilakukan berdasarkan hasil dari identifikasi yang sudah dilakukan sebelumnya oleh peneliti menggunakan alat penguji kerentanan otomatis OWASPZAP. Serangan XSS ini bertujuan untuk mendaptkan informasi pengguna, cookies, session, tokens, serta informasi lainnya, namun dikarenakan laman PDDikti merupakan sebuah situs yang berbasis search engine maka peneliti hanya akan memastikan apakah peneliti dapat melakukan defacing atau memanipulasi tampilkan laman PDDikti dengan menginput perintah <script>alert(“Tes Serangan XSS”)</script> pada alamat domain PDDikti, *form search* yang ada pada *landing page*, program studi, perguruan tinggi, publikasi, dan pengumuman.

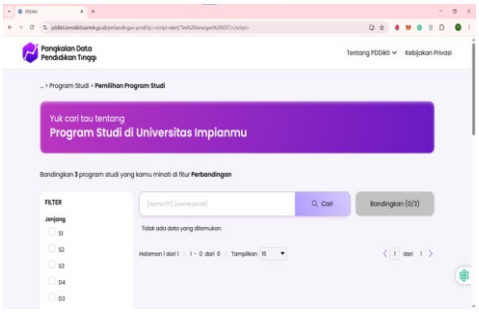


Gambar 7 Pentest XSS Injection 1

Hasil dari serangan XSS terhadap *website* PDDikti ketika peneliti menginput script ke domain PDDikti gagal, dikarenakan setelah menginput script website mengembalikan perintah serangan ke halaman beranda dari website lama PDDikti.



Gambar 8 Pentest XSS Injection 2

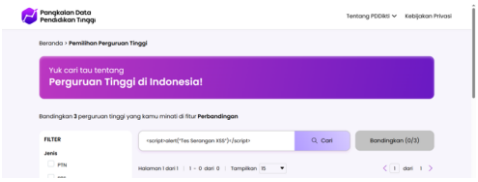


Gambar 9 Pentest XSS Injection 3

Peneliti berhasil mengidentifikasi kerentanan *Cross-Site Scripting* (XSS) pada *form* pencarian yang terdapat di halaman *landing page* PDDikti serta di laman program studi. Kerentanan ini ditemukan ketika peneliti menginput *payload XSS* ke dalam form pencarian. Meskipun sistem menampilkan pesan "tidak ada hasil pencarian" karena tidak ada data yang sesuai dengan input tersebut, *payload XSS* tetap terefleksi dalam halaman hasil pencarian tanpa dilakukan proses penyaringan (sanitasi) terlebih dahulu. Hal ini mengindikasikan adanya kerentanan *Reflected XSS*, yang dapat dieksploitasi oleh penyerang untuk:

- Menyisipkan skrip berbahaya ke dalam URL,
- Mencuri data pengguna seperti cookies atau token sesi,
- Melakukan redirect ke situs phishing,
- Atau menyalahgunakan tampilan halaman untuk tujuan penipuan (*spoofing*).

Walaupun tidak ada data yang ditampilkan dalam pencarian, fakta bahwa skrip dapat terefleksi menunjukkan bahwa halaman tersebut tidak aman dan dapat dimanfaatkan untuk serangan lebih lanjut terhadap pengguna lain.



Gambar 10 Pentest XSS Injection 4

Gambar 11 Pentest XSS Injection 5

Gambar 12 Pentest XSS Injection 6

Dalam Pengujian kali ini, peneliti menemukan potensi kerentanan XSS, pada *form* pencarian yang terdapat pada laman perguruan tinggi, publikasi, dan pengumuman yang ada pada situs PDDikti. Peneliti mencoba memasukkan XSS *payload* ke dalam *form* pencarian. Namun, berdasarkan hasil pengujian:

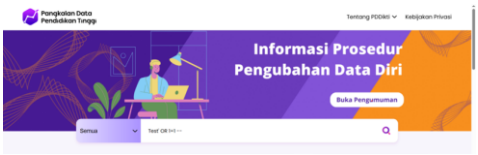
- Tidak ada *payload* yang berhasil terefklesi di laman hasil pencarian.
- *Website* memberikan *response error* dengan *status code* 403 (*forbidden*) pada beberapa permintaan yang dicurigai sebagai inputan berbahaya.

Meskipun tidak ditemukan refleksi langsung dari *payload* yang diuji, kemunculan HTTP 403 dan *error behavior* yang tidak biasa tetap menjadi indikasi bahwa perlu dilakukan pengecekan dan observasi lebih lanjut terhadap sistem, terutama pada mekanisme penyaringan dan penanganan input. Potensi kerentanan ini perlu diperhatikan karena error tersebut dapat menjadi celah untuk eksploitasi di masa mendatang apabila penanganan input tidak dilakukan secara menyeluruh dan konsisten.

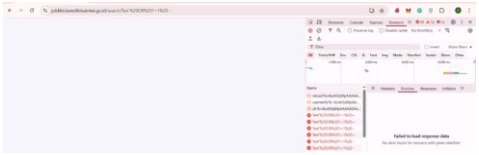
2. SQL Injection

Serangan yang sering terjadi pada website yang berbasis search *engine* seperti laman PDDikti biasanya tidak jauh dari serangan seperti input *validation* dan *injection*, dikarenakan laman PDDikti merupakan website dengan data yang cukup penting dan bersifat cukup sensitif maka peneliti mencoba untuk menguji laman PDDikti

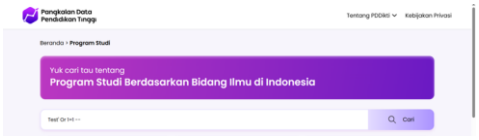
dengan menyuntikan perintah SQL berbahaya kedalam inputan (*form*) *search* yang ada pada laman PDDikti dengan tujuan untuk memanipulasi database. Peneliti akan menyuntikan perintah *basic SQL Injection* terlebih dahulu pada *form search* yang ada pada *landing page*, program studi, perguruan tinggi, publikasi, dan pengumuman untuk menguji apakah input akan langsung diproses ke *query SQL*.



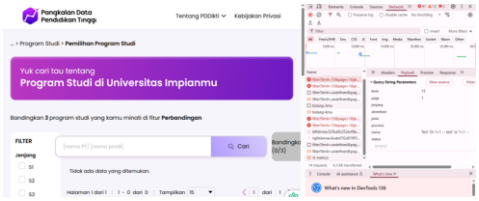
Gambar 13 *Pentest Basic Boolean SQL Injection 1*



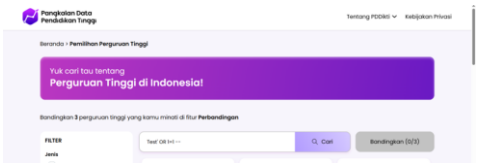
Gambar 14 Hasil Pentest Basic Boolean SQL Injection 1



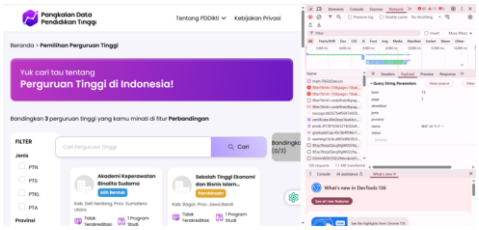
Gambar 15 Pentest *Basic Boolean SQL Injection 2*



Gambar 16 Hasil *Pentest Basic Boolean SQL Injection 2*

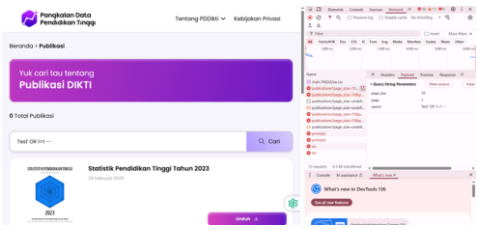


Gambar 17 *Pentest Basic Boolean SQL Injection 3*

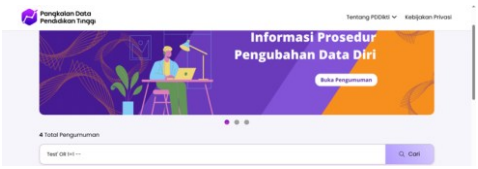


Gambar 18 Hasil *Pentest Basic Boolean SQL Injection 3*

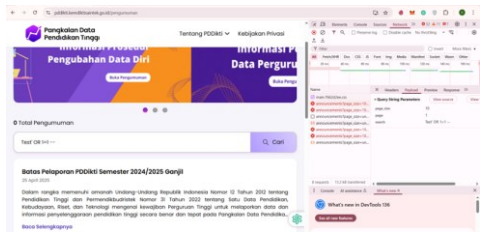
Gambar 19 *Pentest Basic Boolean SQL Injection 4*



Gambar 20 Hasil *Pentest Basic Boolean SQL Injection 4*



Gambar 21 *Pentest Basic Boolean SQL Injection 5*



Gambar 22 Hasil *Pentest Basic Boolean SQL Injection 5*

Setelah dilakukan serangan *basic boolean SQL injection* terhadap laman target yang ada pada laman PDDikti, dapat disimpulkan bahwa serangan yang peneliti lakukan belum berhasil dikarenakan serangan di blokir oleh WAF (*Web Application Firewall*) dengan tipe *signature-based* dimana firewall ini dapat mendeteksi serangan berdasarkan pola kata kunci terentu dengan payload, seperti *OR, UNION, SELECT*, ‘,’ dan sebagainya.

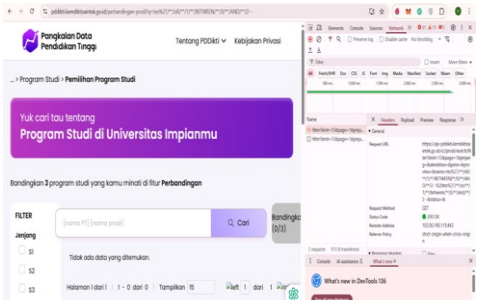
Namun demikian, peneliti berencana untuk melakukan *bypass* terhadap *firewall* ini dikarenakan WAF dengan *signature-based* memiliki kelemahan dimana firewall ini hanya mencocokkan pola teks mentah (*raw text*) tanpa menganalisis cara eksekusi sebenarnya. Dengan menggunakan teknik *obfuscation*, serangan dapat disamarkan agar lolos dari deteksi WAF.

Sebelum melanjutkan serangan *SQL Injection* dengan pendekatan *obfuscation*, terdapat beberapa kejanggalan sistem yang ditemukan selama proses pengujian. Pada laman Perguruan Tinggi, Publikasi, dan Pengumuman, ketika dilakukan pencarian dengan input yang p*ayload basic boolean SQL injection*, sistem tetap menampilkan seluruh data yang ada. Hal ini

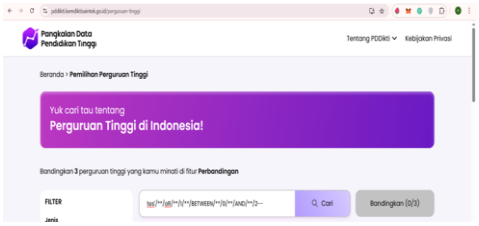
menunjukkan kemungkinan adanya *bug* pada logika pencarian di ketiga laman tersebut, sehingga diperlukan pengecekan dan perbaikan lebih lanjut untuk memastikan fungsionalitas pencarian berjalan sebagaimana mestinya.

Gambar 23 *Pentest SQL Injection Bypass WAF 1*

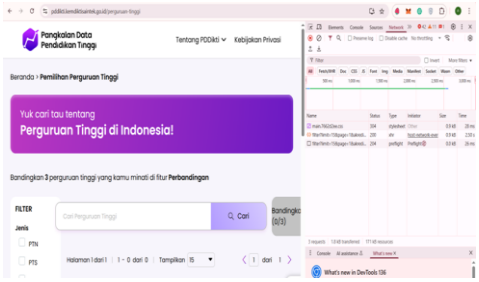
Gambar 24 *Pentest SQL Injection Bypass WAF 2*



Gambar 25 Hasil *Pentest SQL Injection Bypass WAF 2*

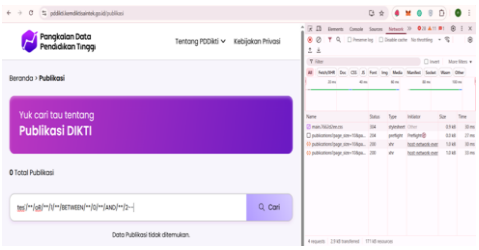


Gambar 26 *Pentest SQL Injection Bypass WAF 3*



Gambar 27 Hasil *Pentest SQL Injection Bypass WAF 3*

Gambar 28 *Pentest SQL Injection Bypass WAF 4*



Gambar 29 *Hasil Pentest SQL Injection Bypass WAF 4*

Setelah melakukan serangan *SQL Injection* dengan menggunakan teknik *obfuscation*, *payload SQL injection* berhasil melewati WAF. Namun hanya laman *landing page* PDDikti saja yang berhasil menggagalkan serangan *SQL* Injection, di mana sistem secara otomatis mengarahkan pengguna kembali ke halaman beranda ketika *payload SQL* disisipkan ke dalam form pencarian.

Namun sebaliknya, pada halaman program studi, perguruan tinggi, publikasi dan pengumuman, saat *SQL payload* disisipkan pada *form search,* sistem pada laman PDDikti tidak mengalami *crash* dan tidak mengembalikan pesan *error*, tapi tidak juga menunjukkan hasil data. Berdasarkan kondisi tersebut, peneliti menyimpulkan bahwa *SQL Injection* berhasil dilakukan, namun termasuk dalam kategori "*Blind SQL Injection" (boolean-based atau time-based)*, di mana *query* termodifikasi, tapi aplikasi tidak memberikan umpan balik langsung terhadap eksekusinya.

Untuk mencegah serangan serupa di masa mendatang, disarankan agar dilakukan langkah-langkah berikut:

1. Meninjau ulang *logika backend* dan *query* yang berinteraksi dengan *database*.
2. Melakukan validasi dan sanitasi input secara ketat.
3. Membatasi hak akses user terhadap database hanya sesuai kebutuhan (*principle of least privilege*).
4. Menerapkan mekanisme *error handling* yang aman dan tidak menampilkan informasi sensitif.
5. Melakukan audit kode secara berkala guna mendeteksi potensi kerentanan lebih dini.

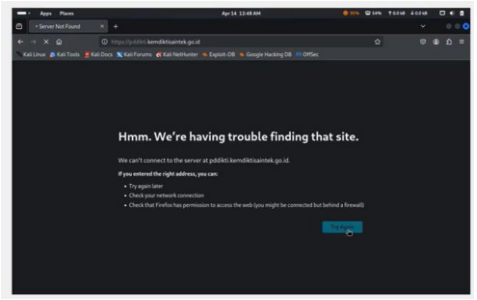
Dengan penerapan langkah-langkah tersebut, risiko terjadinya serangan *Blind SQL Injection*

oleh pihak yang tidak bertanggung jawab dapat diminimalisir secara signifikan.

3. *DDoS Attack*

Serangan DDoS merupakan jenis serangan yang akan memenuhi lalu lintas pada server target. Sehingga seolah-olah *server* terbanjiri oleh akses yang berlebihan dari client, sehingga jika server sudah melebihi batas limit maka laman PDDikti tidak akan dapat diakses. Peneliti menggunakan tools LOIC untuk melakukan pengujian serangan DDoS ini.

Gambar 30 Serangan DDoS Menggunakan Tools LOIC



Gambar 31 Hasil Serangan DDoS Menggunakan *Tools* LOIC

Setelah jendela LOIC (*Low Orbit Ion Cannon*) dibuka, peneliti menginput *URL* laman PDDikti ke dalam *form* target, menggunakan port 80, dengan metode UDP, dan jumlah *thread* sebanyak 1000. Pada tahap pengujian ini, peneliti berhasil melakukan simulasi serangan DDoS (*Distributed Denial of Service*) terhadap *server* PDDikti. Serangan ini membanjiri server dengan permintaan akses secara berlebihan sehingga menyebabkan situs tidak dapat diakses untuk sementara waktu, yakni selama *tools* LOIC dijalankan. Tujuan dari simulasi serangan ini adalah untuk mengidentifikasi potensi kerentanan serta dampak nyata dari serangan DDoS terhadap performa layanan *website*. Hasil pengujian menunjukkan bahwa server PDDikti tidak memiliki proteksi yang cukup untuk menghadapi beban lalu lintas *abnormal* yang tinggi. Untuk mencegah kejadian serupa, disarankan agar pengelola sistem melakukan peningkatan pada sisi keamanan baik dari segi perangkat keras (*hardware*) maupun perangkat lunak (*software*).

4. Umpan Balik

Dalam penelitian ini, dikarenakan objek penelitian merupakan situs web milik suatu instansi, maka diperlukan umpan balik dari pihak klien. Umpan balik ini bertujuan agar peneliti dapat mengetahui kekurangan dari penelitian yang sedang dilakukan. Berikut ini adalah hasil umpan balik yang diberikan oleh klien.

Tabel 5 Hasil Umpan Balik

No	Pertanyaan	Jawaban
1	Apakah ruang lingkup penelitian sudah sesuai dengan kebutuhan Anda?	1 Sangat Sesuai, 2 Sesuai
2	Apakah Anda merasa teknik *penetration testing* yang digunakan sudah tepat?	1 Sangat Tepat, 2 Tepat
3	Apakah laporan hasil uji keamanan mudah dipahami oleh pihak teknis dan non-teknis?	1 Sangat Mudah, 1 Mudah, 1 Cukup Mudah
4	Seberapa lengkap temuan yang disampaikan dalam laporan akhir?	2 Lengkap, 1 Sangat Lengkap
5	Apakah temuan celah keamanan (*XSS, SQL Injection, DDoS*) relevan dengan kondisi aktual sistem?	3 Relevan
6	Apakah rekomendasi keamanan yang diberikan bersifat praktis dan dapat diimplementasikan?	1 Sangat Dapat Diterapkan, 1 Dapat Diterapkan, 1 Cukup Dapat Diterapkan
7	Bagaimana kualitas dokumentasi teknis yang disediakan oleh peneliti?	2 Sangat Baik, 1 Cukup Baik

IV. SIMPULAN

Penelitian ini bertujuan untuk menganalisis celah keamanan pada *website* PDDikti menggunakan metode *penetration testing* yang terstruktur berdasarkan *framework* ISSAF. Berdasarkan hasil pengujian yang telah dilakukan, berikut adalah kesimpulan yang dapat ditarik, sekaligus menjawab rumusan masalah:

1. Penerapan metode *penetration testing* pada *website* PDDikti dilakukan dengan pendekatan sistematis melalui tiga tahapan utama sesuai f*ramework ISSAF*, yaitu *Planning, Assessment,* dan *Reporting*. Pengujian dilakukan secara langsung ke sistem target dengan *tools* seperti Nmap, OWASP ZAP, WhatWeb, dan LOIC.
2. Celah keamanan yang ditemukan pada *website* PDDikti meliputi:
    - *Reflected XSS* pada *form* pencarian halaman *landing page* dan program studi, di mana

payload dapat terefleksi tanpa disaring.

- *Blind SQL Injection* pada beberapa *form* pencarian (halaman perguruan tinggi, publikasi, dan pengumuman), yang menunjukkan bahwa *query* dapat dimodifikasi namun tidak memberikan umpan balik eksplisit.
- Kerentanan terhadap serangan DDoS, di mana simulasi serangan menggunakan LOIC menyebabkan situs tidak dapat diakses sementara waktu.

Secara keseluruhan, penelitian ini menunjukkan bahwa situs PDDikti masih memiliki beberapa celah keamanan yang perlu segera ditangani. Dengan menerapkan langkah-langkah mitigasi yang disarankan, diharapkan sistem ini dapat menjadi lebih aman dan andal untuk digunakan sebagai platform penyedia data pendidikan tinggi di Indonesia.

DAFTAR RUJUKAN

[1] Liputan6, "Apa Itu Situs Web dan Contohnya, Ini Panduan Lengkapnya.," 2024, Liputan 6, Jakarta. [Online]. Available: https://www.liputan6.com/feeds/read/5805830/apa-itu-situs-web-dan contohnya-ini-panduan-lengkapnya

[2] I. Hanif, "Teknologi Website: Definisi, Sejarah, Hingga Tren Terkini," 2024.

[3] O. Ichsandrya, Analisis Kepuasan Pengguna Website Pddikti Dengan Menggunakan Webqual. 2020. [Online]. Available: https://repository.its.ac.id/80599/

[4] BPPTIK, "Jenis-Jenis Serangan Siber di Era Digital," 2023, BPPTIK. [Online]. Available: https://bpptik.kominfo.go.id/Publikasi/detail/jenis jenis-serangan-siber-di-era-digital

[5] D. Satria, A. Alanda, A. Erianda, and D. Prayama, "Network security assessment using internal network penetration testing methodology," Int. J. Informatics Vis., vol. 2, no. 4–2, pp. 360–365, 2018, doi: 10.30630/joiv.2.4 2.190.

[6] F. A. Hassanah, E. Ryansyah, A. Susilo, and Y. Irawan, "Analisis Kerentanan Keamanan Menggunakan OWASP ZAP dan Pengujian Manual pada Tampilan Antarmuka Laman PDDIKTI," no. February, 2025, doi: 10.24843/JLK.2025.v13.i03.p14.

[7] S. Andriyani, M. F. Sidiq, and B. P. Zen, "Analisis Celah Keamanan Pada Website Dengan Menggunakan Metode Penetration Testing Dan Framework Issaf Pada Website SMK Al-Kautsar," J. Inform. Inf. Technol., vol. 8798, pp. 1–13, 2023.

[8] T. Dirgahayu, Y. Prayudi, and A. Fajaryanto, "Penerapan Metode ISSAF dan OWASP versi 4 Untuk Uji Kerentanan Web Server," J. Ilm. NERO, vol. 1, no. 3, pp. 190–197, 2015, [Online]. Available: http://nero.trunojoyo.ac.id/index.php/nero/article/download/29/27

[9] Z. A. Khan, N. Safaat, M. Irsyad, and T. Darmizal, "Penetration Testing Information System Security Assessment Framework (ISSAF)," Kaji. Ilm. Inform. dan Komput., vol. 4, no. 3, pp. 1593–1601, 2023, doi: 10.30865/klik.v4i3.1503.

[10] "Penetration Testing with Kali Linux - A Complete Guide." [Online]. Available: https://www.udemy.com/course/penetration-testing-with-kali linux-a-complete-guide/