Journal Computer and Technology
Vol. 3, No. 1, Juli 2025, hlm. 26-38
ISSN : 3048-1880 (Online)
DOI : https://doi.org/10.69916/comtechno.v3i1.333

ANALISIS KERENTANAN *WEBSITE* PRODI TEKNOLOGI INFORMASI UBB MENGGUNAKAN METODE *APPLICATION SCANNING*

Femiliana Septarita[*1], Wenny Anggraini[2]

[1,2]Prodi Teknologi Informasi, Fakultas Sains dan Teknik, Universitas Bangka Belitung, Indonesia
Email: [1]femilianaseptarita34@gmail.com, [2]Anggrainiwenny79@gmail.com

SEJARAH ARTIKEL
Diterima: 30.05.2025
Direvisi: 08.07.2025
Diterima: 21.07.2025

ABSTRAK

Penelitian ini dilakukan untuk menganalisis keamanan pada *website* program studi teknologi informasi universitas bangka belitung menggunakan metode *application scanning*, yaitu teknik otomatis yang mampu mendeteksi berbagai celah keamanan aplikasi web. Penelitian dilakukan dengan cara pemindaian langsung pada website program studi menggunakan alat bantu pemindai yang mampu mengidentifikasi jenis-jenis kerentanan seperti *SQL Injection*, *Cross-Site Scripting* (XSS), dan kesalahan konfigurasi. Hasil penelitian menunjukkan bahwa *website* memiliki beberapa celah keamanan yang perlu segera diperbaiki untuk menghindari potensi eksploitasi oleh pihak tidak bertanggung jawab. Penelitian ini diharapkan dapat memberikan kontribusi dalam meningkatkan perlindungan informasi digital dan dasar evaluasi teknis dan penguatan sistem keamanan situs web akademik secara berkelanjutan. Metode *application scanning* terbukti efektif, efisien, dan sesuai diterapkan di lingkungan institusi pendidikan tinggi sebagai bagian dari manajemen keamanan informasi.

Kata Kunci: keamanan data, pengujian, pemindaian kerentanan, *website*, analisis sistem.

ABSTRACT

This study aims to analyze the security of the Information Technology Study Program website at Universitas Bangka Belitung by employing the application scanning method, an automated technique capable of detecting various web application vulnerabilities. The research involved directly scanning the website using specialized scanning tools designed to identify vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and configuration errors. The findings reveal several security weaknesses that require immediate remediation to prevent potential exploitation by unauthorized parties. This study is expected to contribute to enhancing digital information protection and serve as a technical reference for the continuous improvement of the academic website's security system. The application scanning method has proven to be effective, efficient, and appropriate for implementation within higher education institutions as part of information security management.

Keywords: *data security, scanning, vulnerability scanning, website, system analysis.*

1. PENDAHULUAN

Di zaman transformasi digital ini telah menjelma menjadi perubahan yang sangat besar tepat nya pada cara manusia berinteraksi, berkomunikasi, serta mengolah data dan menjalankan tugas aktifitas sehari - hari seperti perkembangan teknologi informasi yang berkembang sangat pesat untuk mendukung berbagai institusi pendidikan tinggi yang menyediakan layanan berbasis web guna mendukung aktivitas akademik dan administrasi [1]. Situs web sebagai bagian dari infrastruktur teknologi informasi menjadi sarana penting dalam penyampaian informasi, pendaftaran, dan interaksi antara mahasiswa dan dosen melalui aplikasi. Namun dengan meningkatnya ketergantungan pada layanan yang berbasis web, risiko ancaman terhadap aplikasi web keamanan informasi juga semakin meningkat [2].

Beberapa studi telah mengungkapkan bahwa 78% institusi pendidikan telah mengalami satu insiden keamanan dalam periode satu tahun terakhir, dengan server web menjadi sasaran utama para penyerang [3]. Penemuan ini diperkuat oleh penelitian yang menunjukkan bahwa kelemahan pada situs web dapat menyebabkan berbagai dampak serius, seperti kebocoran data sensitif, gangguan dalam layanan operasional, dan kerugian reputasi yang substansial bagi institusi pendidikan [4]. Banyak situs web institusi pendidikan memiliki kerentanan yang dapat dimanfaatkan

oleh individu yang tidak bertanggung jawab, seperti data akademik,informasi pribadi mahasiswa, dan data keuangan merupakan aset digital paling sering menjadi target ancaman [4]. menemukan berbagai celah kemanan dalam situs web dengan menggunakan metode *application scanning* kerentanan terhadap serangan XSS dan *SQL injection*. *Website* teknologi informasi Universitas bangka belitung sebagai pusat aktivitas digital yang mendukung proses pendidikan menyajikan informasi tentang rencana pembelajaran, pendapat mahasiswa dan profil prodi. Website teknologi informasi Ini dapat diakses secara umum. *Website* ini dikelola oleh 1 admin dan 1 operator yang bertanggung jawab dalam pengelolaan segala informasi keamanan *website* prodi. Namun, maraknya serangan siber dan eksploitasi celah keamanan menunjukkan bahwa aspek keamanan siber sering kali belum menjadi prioritas utama dalam pengelolaan situs akademik [5].

Berdasarkan permasalahan penelitian ini untuk melakukan analisis keamanan pada *website* program studi teknologi informasi ubb menggunakan metode *application scanning* sebagai respon terhadap kebutuhan keamanan teknologi informasi dalam institusi pendidikan, khususnya pada website program studi teknologi informasi ubb merupakan platfrom digital publik yang menyediakan informasi yang tidak hanya bagi aktivitas akademik, tetapi bagi masyarakat umum[6]. Namun, karena tidak adanya dokumentasi resmi tentang pengujian keamanan yang dilakukan secara berulang pada *website* tersebut, muncul kekhawatiran terhadap eksploitasi celah keamanan yang terjadi. Pengamatan observasi awal menunjukkan bahwa tidak ada sistem deteksi terhadap ancaman dan tidak tersedia log audit keamanan yang dapat melacak aktivitas mencurigakan seacra akurat. Disisi lain metode application scanning memungkinkan pendeteksi otomatis terhadap kerentanan memberikan solusi menggunakan pendekatan yang efektif dan efisisen untuk mendeteksi celah kelemahan keamanan pada *website*. Metode ini telah terbukti mengidentifikasi jenis serangan seperti *SQL injection*, XXS, dan kesalahan konfigurasi server, oleh karena itu, penggunaan metode ini pada website prodi teknologi informasi ubb dinilai tepat untuk mendeteksi kelemahan sistem secara menyeluruh, terutama mengingat situs sering diakses oleh banyak pengguna namun belum didukung oleh sistem pengujian kemanan profesional [7].

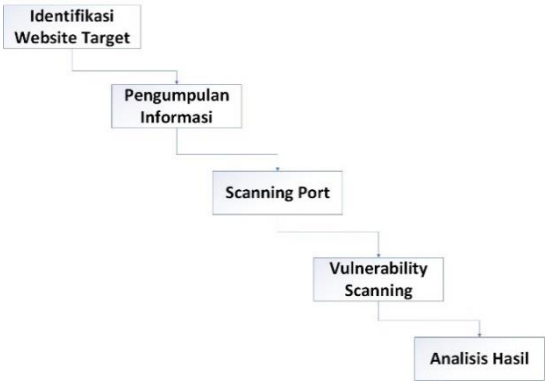
Penelitian ini menggunakan *tools* spesifik seperti OWASP ZAP (*Zed Attack Proxy*) versi 2.16.1 untuk *application scanning*, Who.is dan Nslookup untuk pengumpulan informasi domain dan IP, serta Advanced Port Scanner untuk *port scanning*. Jenis serangan yang diuji meliputi *SQL Injection*, *Cross-Site Scripting (XSS)*, *Cloud Metadata Exposure*, *Absence of Anti-CSRF Tokens*, *Missing Anti-clickjacking Header*, *Cookie Without Secure Flag*, *Cookie without SameSite Attribute*, *Cross-Domain JavaScript Source File Inclusion*, *Strict-Transport-Security Header Not Set*, *X-Content-Type-Options Header Missing*, *Information Disclosure - Suspicious Comments*, *Hidden File Found*, *Modern Web Application Identification*, *Session Management Response Identified*, dan *User Agent Fuzzer*. Konfigurasi awal OWASP ZAP akan melibatkan *Passive Scan* untuk menganalisis lalu lintas yang ada dan *Active Scan* untuk mengidentifikasi kerentanan secara mendalam, dengan *scan depth* yang disesuaikan untuk cakupan yang komprehensif. Hasil dari penelitian ini diharapkan dapat membantu serta memberikan manfaat pada pengembangan analisis keamanan pada *website* Program Studi Teknologi Informasi UBB menggunakan metode *application scanning*, dan juga diharapkan dapat memberikan rekomendasi untuk memenuhi kebutuhan peningkatan perlindungan informasi digital, mendukung keberhasilan upaya transformasi digital yang berkelanjutan di lingkungan pendidikan tinggi.

2. METODE PENELITIAN

Pada penelitian ini, penulis menggunakan dua metode utama: wawancara dan metode *application scanning*.

2.1. Metode *Application Scanning*

Metode *application scanning* adalah teknik pemindaian otomatis pada aplikasi web yang bertujuan mendeteksi berbagai kerentanan keamanan, seperti *SQL Injection*, *Cross-Site Scripting (XSS)*, dan kelemahan konfigurasi keamanan lainnya. Metode ini menggunakan perangkat lunak khusus, salah satunya OWASP ZAP (Zed Attack Proxy) versi 2.16.1, yang mampu melakukan pemindaian secara efisien dan sistematis sehingga membantu pengelola *website* dalam mengidentifikasi dan memperbaiki celah keamanan yang ada [8][9][10]. Berikut adalah langkah – langkah dalam metodologi ini:



Gambar 1. Metode *Application Scanning*

Pada penelitian ini, penulis menggunakan metode *application scanning* untuk menganalisis keamanan *website* Prodi Teknologi Informasi Universitas Bangka Belitung. Metode ini dilakukan dengan memanfaatkan *tools* OWASP ZAP (Zed Attack Proxy) versi 2.16.1 sebagai alat utama untuk melakukan pemindaian otomatis terhadap aplikasi web. *Tools* ini dipilih karena kemampuannya yang handal dalam mendeteksi berbagai jenis kerentanan keamanan, seperti *SQL Injection*, *Cross-Site Scripting (XSS)*, dan kelemahan konfigurasi lainnya secara efisien dan komprehensif. Penggunaan OWASP ZAP memungkinkan proses pemindaian dilakukan secara otomatis dan sistematis, sehingga meminimalisir kesalahan manusia dan mempercepat identifikasi celah keamanan yang mungkin tersembunyi di dalam aplikasi web. Selain itu, OWASP ZAP adalah salah satu *tools open-source* yang biasanya digunakan dalam komunitas keamanan siber, sehingga memiliki dukungan dokumentasi dan pembaruan yang rutin, menjadikannya alat yang sangat relevan dan mutakhir dalam konteks pengujian keamanan aplikasi web [11].

Dalam pengujian ini, OWASP ZAP dikonfigurasi untuk melakukan:

a. Passive Scan: Dilakukan secara otomatis saat *website* dijelajahi, menganalisis lalu lintas HTTP yang ada tanpa mengirim *request* baru. Ini membantu mengidentifikasi kerentanan pasif seperti *missing security headers* atau *information disclosure*.
b. Active Scan: Dilakukan setelah *passive scan*, mengirimkan *request* yang dimodifikasi ke target untuk menemukan kerentanan secara aktif. *Active scan* ini mencakup pengujian untuk *SQL Injection*, *XSS*, dan kerentanan berbasis injeksi lainnya.
c. Scan Depth: Pemindaian dilakukan dengan kedalaman yang komprehensif untuk mencakup seluruh halaman dan *endpoint* yang dapat diakses publik pada *website*.
d. Authentication: Karena *website* Prodi TI UBB dapat diakses secara umum tanpa autentikasi khusus untuk fitur yang diuji, *authenticated scan* tidak diterapkan. Pemindaian berfokus pada kerentanan yang dapat dieksploitasi oleh pengguna anonim.
e. Exclusions: Tidak ada URL atau parameter spesifik yang dikecualikan dari pemindaian untuk memastikan cakupan yang maksimal.

Tahapan penelitian dimulai dengan identifikasi *website* target, yaitu *website* resmi Prodi Teknologi Informasi Universitas Bangka Belitung. Selanjutnya, dilakukan pengumpulan informasi domain dan alamat IP menggunakan *tools* seperti Who.is dan Nslookup. Pengumpulan informasi ini penting untuk memahami struktur jaringan dan domain yang digunakan oleh *website*, sehingga proses pemindaian dapat dilakukan dengan lebih tepat sasaran. Setelah itu, dilakukan *port scanning* menggunakan Advanced Port Scanner untuk mengetahui *port* dan layanan yang terbuka pada server. *Port scanning* merupakan langkah krusial karena *port* terbuka dapat menjadi titik masuk bagi serangan siber jika tidak dikonfigurasi dengan benar. Dengan mengetahui *port* yang aktif, peneliti dapat mengidentifikasi potensi risiko yang mungkin timbul dari layanan yang berjalan pada server tersebut [12].

Setelah tahap pengumpulan informasi dan *port scanning* selesai, langkah berikutnya adalah melakukan pemindaian kerentanan aplikasi web menggunakan OWASP ZAP. Pada tahap ini, OWASP ZAP akan melakukan *scanning* secara aktif dan pasif untuk mendeteksi berbagai kerentanan yang ada pada aplikasi web, mulai dari celah injeksi SQL, XSS, hingga kelemahan konfigurasi keamanan seperti penggunaan *header* HTTP yang tidak aman atau autentikasi yang lemah. Hasil pemindaian ini kemudian dianalisis secara mendalam untuk mengidentifikasi jenis kerentanan dan tingkat risiko yang ditimbulkan. Berdasarkan hasil analisis tersebut, penulis akan memberikan rekomendasi perbaikan yang spesifik dan terukur guna meningkatkan keamanan *website* Prodi Teknologi Informasi Universitas Bangka Belitung secara signifikan [10].

Metode *application scanning* menggunakan OWASP ZAP dipilih karena kemampuannya memberikan gambaran menyeluruh mengenai kondisi keamanan *website* secara otomatis dan sistematis. Hal ini sangat membantu dalam proses identifikasi dan mitigasi risiko keamanan yang mungkin terjadi, sehingga pengelola *website* dapat mengambil langkah cepat dan tepat dalam memperbaiki celah keamanan yang ditemukan.

Adapun penelitian [13] menjelaskan bahwa OWASP ZAP tidak hanya mampu mendeteksi kerentanan umum seperti XSS dan SQL Injection, tetapi juga mampu melakukan *scanning* terhadap kelemahan konfigurasi keamanan yang sering terabaikan, seperti pengaturan *cookie* yang tidak aman, penggunaan protokol komunikasi yang rentan, serta masalah autentikasi dan otorisasi. Hal ini menunjukkan bahwa metode *application scanning* dengan OWASP ZAP tidak hanya fokus pada kerentanan aplikasi, tetapi juga aspek keamanan infrastruktur pendukung aplikasi web. Dengan demikian, metode ini memberikan pendekatan yang komprehensif dan holistik dalam menjaga keamanan website [2]. Secara keseluruhan, penggunaan metode *application scanning* berbasis OWASP ZAP dalam penelitian ini sangat relevan dan efektif untuk meningkatkan keamanan website Prodi Teknologi Informasi Universitas Bangka Belitung.

2.2. Pengumpulan Data

Penulis melakukan pengumpulan data dengan menggunakan 2 metode utama, yaitu wawancara dan pemindaian aplikasi web. Dalam tahap wawancara ini dilakukan secara terstruktur dengan Koordinator Program Studi Teknologi Informasi Universitas Bangka Belitung untuk memperoleh informasi terkait pengelolaan website, teknologi yang digunakan, kebijakan keamanan, serta kendala yang dihadapi dalam menjaga keamanan sistem. Pendekatan wawancara ini penting untuk memahami aspek manajerial dan prosedural yang memengaruhi keamanan website, sebagaimana dijelaskan dalam penelitian yang menekankan pentingnya data kualitatif dalam evaluasi keamanan sistem informasi [14], [15].

Wawancara dilakukan dalam format semi-terstruktur dengan Koordinator Program Studi. Berikut pertanyaan yang diajukan, yakni:

a. Kapan *website* Prodi TI UBB mulai dikembangkan dan menggunakan *framework* apa?
b. Siapa saja yang terlibat dalam pengembangan dan pengelolaan *website*?
c. Mekanisme keamanan dasar apa saja yang sudah diterapkan pada *website*?
d. Bagaimana prosedur pemeliharaan dan pembaruan keamanan *website* dilakukan?
e. Apakah ada rencana untuk peralihan ke versi *website* baru atau pengujian keamanan otomatis?
f. Apakah pernah terjadi insiden keamanan atau serangan siber pada *website*?
g. Kendala utama apa yang dihadapi dalam menjaga keamanan *website*?
h. Apakah ada SOP atau panduan khusus untuk penanganan insiden keamanan?
i. Apa harapan pengelola terhadap pengembangan *website* ke depan terkait keamanan dan fitur?

Kemudian hasil wawancara dianalisis secara tematik untuk mengidentifikasi pola, kendala, dan harapan terkait keamanan *website*. Informasi kualitatif ini digunakan untuk memberikan konteks operasional terhadap temuan teknis dan merumuskan rekomendasi yang lebih relevan. Selain itu, data teknis dikumpulkan melalui pemindaian menggunakan beberapa tools, seperti Who.is untuk memperoleh informasi domain, Nslookup untuk mengetahui alamat IP, Advanced Port Scanner untuk mendeteksi port terbuka, serta OWASP ZAP untuk mengidentifikasi kerentanan pada web. Penggunaan kombinasi *tools* ini sesuai pada metodologi pengumpulan data yang menggabungkan teknik *footprinting* dan *vulnerability scanning* guna mendapatkan gambaran menyeluruh tentang potensi risiko keamanan [16][12][2]. OWASP ZAP sebagai alat utama dalam pemindaian kerentanan yang mendeteksi berbagai celah keamanan seperti *Cross-Site Scripting* (XSS), *SQL Injection*, dan konfigurasi yang tidak aman, yang sering ditemukan pada aplikasi web [14], [2].

Data yang diperoleh dari kedua metode ini kemudian dianalisis secara komprehensif untuk memberikan gambaran menyeluruh mengenai kondisi keamanan website yang diteliti. Pendekatan gabungan antara data kualitatif dari wawancara dan data teknis dari pemindaian memungkinkan peneliti tidak hanya mengidentifikasi kerentanan teknis, tetapi juga memahami faktor-faktor manajerial dan kebijakan yang mempengaruhi keamanan sistem secara keseluruhan. Hal ini penting agar rekomendasi perbaikan yang diberikan dapat bersifat menyeluruh dan aplikatif sesuai kebutuhan pengelola *website* [9], [15].

2.3. Analisis Kerentanan

Analisis kerentanan bertujuan untuk mengevaluasi hasil pemindaian yang diperoleh dari proses *application scanning* menggunakan OWASP ZAP. OWASP ZAP merupakan tools open-source yang biasanya digunakan untuk mengidentifikasi berbagai jenis kerentanan pada aplikasi web secara otomatis, seperti *injeksi SQL*, *Cross-Site Scripting* (XSS), konfigurasi yang tidak aman, dan celah keamanan lainnya [11][17][14]. Dalam penelitian ini, data hasil pemindaian dianalisis untuk mengidentifikasi jenis kerentanan yang ditemukan serta mengelompokkan tingkat risikonya menjadi tinggi, sedang, dan rendah sesuai dengan standar OWASP [17], [16].

Penilaian tingkat risiko kerentanan didasarkan pada kombinasi potensi dampak (kerugian yang mungkin terjadi jika kerentanan dieksploitasi) dan kemungkinan (seberapa mudah kerentanan dapat dieksploitasi). Kategori risiko yang digunakan adalah:

a. Tinggi (*High*): Kerentanan yang sangat mudah dieksploitasi dan dapat menyebabkan dampak serius seperti kebocoran data sensitif, pengambilalihan sistem, atau gangguan layanan yang signifikan [17].
b. Sedang (*Medium*): Kerentanan yang memerlukan upaya lebih untuk dieksploitasi atau memiliki dampak yang moderat, seperti *information disclosure* non-sensitif atau potensi *denial of service* lokal [17].
c. Rendah (*Low*): Kerentanan yang sulit dieksploitasi atau memiliki dampak minimal, seringkali terkait dengan praktik terbaik keamanan yang belum optimal [16].
d. Informasional (*Informational*): Bukan kerentanan langsung, tetapi informasi yang dapat membantu penyerang dalam tahap *reconnaissance* atau menunjukkan area yang perlu diperhatikan lebih lanjut [16].

Metode pemindaian OWASP ZAP mencakup pemindaian pasif dan aktif yang mampu mendeteksi berbagai celah keamanan, termasuk ancaman yang umum terjadi seperti *Cross-Site Scripting* (XSS), *Clickjacking*, dan kesalahan konfigurasi header keamanan [17], [12]. Hasil analisis ini memberikan gambaran menyeluruh mengenai kondisi keamanan *website* yang diteliti, sehingga dapat menentukan prioritas penanganan berdasarkan tingkat risiko yang ditemukan. Penilaian risiko ini penting untuk memfokuskan upaya mitigasi pada kerentanan yang memiliki potensi dampak paling besar terhadap keamanan sistem [16], [10]. Selain itu, analisis kerentanan juga membantu dalam mengidentifikasi kelemahan yang mungkin tidak terlihat secara kasat mata, seperti pengaturan *cookie* yang tidak aman atau kurangnya proteksi terhadap serangan CSRF (*Cross-Site Request Forgery*) [12], [17]. Dengan demikian, hasil analisis ini tidak hanya berfungsi sebagai laporan temuan, tetapi juga sebagai dasar dalam merancang rekomendasi perbaikan yang efektif untuk meningkatkan keamanan aplikasi web secara menyeluruh. Beberapa penelitian sebelumnya menunjukkan bahwa penggunaan OWASP ZAP dalam *vulnerability scanning* dapat memberikan akurasi yang cukup baik dalam menemukan kerentanan pada aplikasi web, meskipun tingkat keberhasilan dapat bervariasi tergantung pada kompleksitas aplikasi dan konfigurasi target [11]. Oleh karena itu, analisis yang cermat dan evaluasi hasil scanning sangat diperlukan agar temuan yang diperoleh dapat dimanfaatkan secara optimal dalam penguatan keamanan *website*.

2.4. Evaluasi Hasil dan Rekomendasi

Evaluasi hasil pengujian menunjukkan bahwa rekomendasi teknis yang disusun perlu mempertimbangkan konteks operasional institusi [18], [19]. Website Prodi TI UBB dikelola oleh tim terbatas tanpa tim khusus keamanan dan belum memiliki SOP penanganan insiden. Oleh karena itu, rekomendasi diarahkan pada langkah-langkah yang realistis dan terukur, seperti penyusunan SOP keamanan singkat, pelatihan internal dasar keamanan bagi pengelola, serta penjadwalan pemindaian keamanan secara berkala minimal dua kali dalam setahun. Prioritas diberikan pada perbaikan kerentanan dengan tingkat risiko tinggi, dengan pendekatan bertahap sesuai kapasitas SDM dan jadwal pemeliharaan yang tersedia. Rekomendasi ini disusun agar dapat diimplementasikan secara efektif dalam lingkungan operasional yang terbatas, namun tetap berdampak signifikan terhadap peningkatan keamanan sistem [20] [21] [22].

3. HASIL DAN PEMBAHASAN

3.1. Hasil

a. Hasil Wawancara

Berdasarkan hasil wawancara semi-terstruktur dengan Koordinator Prodi Teknologi Informasi Universitas Bangka Belitung, diperoleh informasi sebagai berikut:

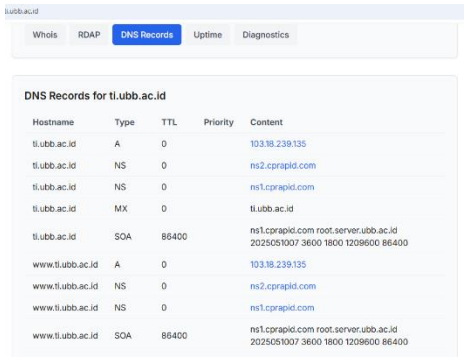
a. *Website* Prodi Teknologi Informasi mulai dikembangkan pada tahun 2021 menggunakan *framework* CodeIgniter (CI). Pengembangan dan pengelolaan *website* dilakukan oleh beberapa individu tanpa pembentukan tim khusus.
b. Dalam fitur keamanan, *website* telah menerapkan beberapa mekanisme dasar, antara lain penggunaan protokol HTTPS untuk komunikasi yang aman, perlindungan terhadap serangan *Cross-Site Request Forgery (CSRF)* yang juga merupakan fitur bawaan *framework*, serta pengaturan integrasi dengan Google Scholar.
c. Untuk keamanan penyimpanan *password*, *website* menggunakan metode *hashing* MD5 yang merupakan standar umum untuk *website* biasa. Prosedur pemeliharaan dan pembaruan keamanan dilakukan secara berkala dengan frekuensi sekitar satu kali dalam setahun, dan pembaruan terakhir dilakukan pada akhir tahun 2023.
d. Saat ini, belum ada rencana peralihan ke versi *website* baru. Pengujian keamanan menggunakan *tools application scanning* otomatis belum dilakukan; pemeriksaan keamanan masih dilakukan secara manual dengan memeriksa satu per satu fitur dan fungsi *website*. Tidak terdapat sistem otomatis untuk mendeteksi anomali atau serangan siber, dan sejauh ini belum pernah terjadi insiden keamanan atau serangan pada *website*.
e. Kendala utama yang dihadapi dalam menjaga keamanan *website* adalah keterbatasan waktu pengelola yang juga memiliki tanggung jawab lain di program studi. Selain itu, tidak terdapat SOP (*Standard Operating*

Procedure) atau panduan khusus untuk penanganan insiden keamanan apabila terjadi serangan atau masalah keamanan lainnya.

f. Harapan pengelola terhadap pengembangan *website* ke depan adalah agar *website* dibuat lebih sederhana dan tidak menyimpan data-data yang bersifat sensitif. Hal ini bertujuan untuk mengurangi risiko peretasan dan memudahkan proses *backup* serta pemulihan data. Selain itu, pengelola juga berkeinginan untuk mengurangi jumlah fitur pada *website*, karena semakin banyak fitur yang disediakan, semakin tinggi pula risiko keamanan yang mungkin muncul.

b. Hasil Cek Domain melalui Who.is

Adapun hasil analisa yang didapatkan oleh peneliti melalui *website* Who.is dengan domain https://ti.ubb.ac.id/ yakni sebagai berikut :

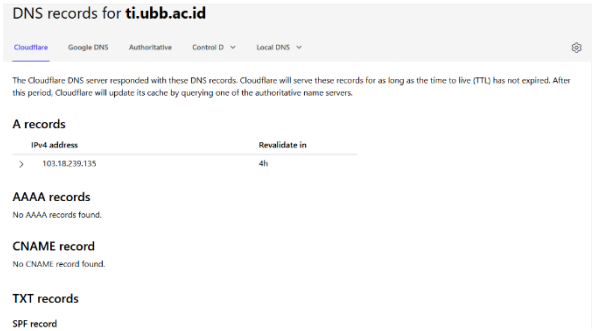


Gambar 2. Hasil Cek website https://ti.ubb.ac.id/

Gambar di atas merupakan pengambilan domain dari *website* Prodi Teknologi Informasi UBB menggunakan who.is, sehingga didapatkan informasi terkait domain, kemudian *content* serta *type* dari *website* tersebut. Informasi kepemilikan domain, kontak administratif, dan data teknis masih dapat diakses secara publik, yang meningkatkan risiko teknik *footprinting*.

c. Hasil Analisis NsLookup

Adapun hasil pengecekan yang dilakukan di NsLookup ini didapatkan informasi berupa IP terkait dari domain *website* Prodi Teknologi Informasi UBB. Berikut IP yang didapatkan :

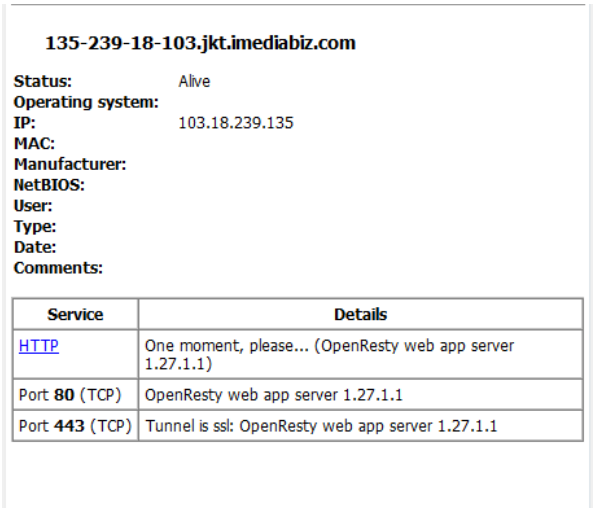


Gambar 3. Hasil Analisa NsLookup *Website* TI UBB

Dari gambar *scanning port* di atas, dapat melihat alamat *website* dan IP *address* pada *website* Prodi Teknologi Informasi Universitas Bangka Belitung. Pada hasil *scanning* di atas dapat diketahui bahwa informasi IP TI UBB adalah sebagai berikut: "103.18.239.135". Ditemukan bahwa beberapa *record* DNS masih menggunakan konfigurasi standar tanpa perlindungan tambahan.

d. Hasil Analisis *Port Scanning*

Adapun hasil yang didapati dari *scanning website* adalah :



135-239-18-103.jkt.imediabiz.com

Status: Alive
Operating system:
IP: 103.18.239.135
MAC:
Manufacturer:
NetBIOS:
User:
Type:
Date:
Comments:

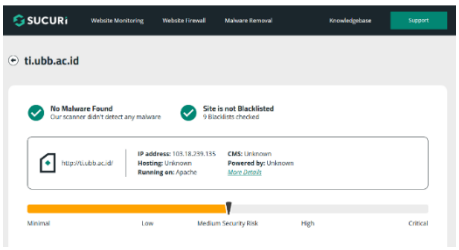
Service	Details
HTTP	One moment, please... (OpenResty web app server 1.27.1.1)
Port **80** (TCP)	OpenResty web app server 1.27.1.1
Port **443** (TCP)	Tunnel is ssl: OpenResty web app server 1.27.1.1

Gambar 4. Hasil *Scanning Advanced Port Scanner*

Hasil gambar diatas menunjukkan beberapa port yang terbuka untuk layanannya sendiri dengan menggunakan HTTP pada rincian One moment, please... (OpenResty web app server 1.27.1.1) serta menggunakan *Apache httpd* 2.4.29. Port yang terbuka yakni Port 80 dan Port 443.

e. Hasil Analisis *Sucure*

Analisis Sucuri digunakan untuk mengetahui kondisi keamanan *website* secara cepat dan menyeluruh.



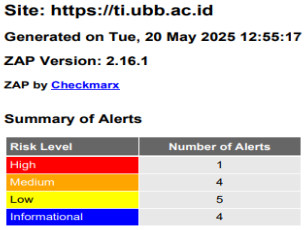
Gambar 5. Hasil cek *Sucuri SiteCheck*

Setelah dilakukan pemindaian keamanan menggunakan *Sucuri SiteCheck*, diperoleh hasil yang menunjukkan bahwa *website* ti.ubb.ac.id dalam kondisi cukup baik dari sisi keamanan dasar. Berdasarkan hasil pemindaian, tidak ditemukan adanya *malware* yang terdeteksi pada website ini. Selain itu, website juga tidak masuk ke dalam daftar hitam (*blacklist*) pada sembilan layanan keamanan yang diperiksa oleh *Sucuri*. Hal ini menandakan bahwa *website* belum terindikasi sebagai sumber ancaman atau distribusi *malware* di *internet*.

Dari hasil analisis, diketahui juga bahwa *website* ini berjalan di atas server *Apache* dengan alamat IP 103.18.239.135. Namun, *Sucuri* tidak dapat mengidentifikasi *Content Management System* (CMS) yang digunakan maupun detail *hosting* dan sumber daya pendukung lainnya. Berdasarkan indikator risiko keamanan yang ditampilkan, *website* ini berada pada kategori medium *security risk*. Artinya, meskipun tidak ada *malware* dan tidak *diblacklist*, masih terdapat beberapa aspek keamanan yang perlu diperhatikan dan ditingkatkan untuk menurunkan tingkat risiko tersebut.

f. Hasil Penilaian Kerentanan Web

Adapun diperoleh hasil pengujian yang dilakukan dengan menggunakan OWASP *Zed Attack Proxy* (ZAP) dari *Open Web Application Security Project* (OWASP), sehingga didapati hasil sebagai berikut:



Site: https://ti.ubb.ac.id
Generated on Tue, 20 May 2025 12:55:17
ZAP Version: 2.16.1
ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	4
Low	5
Informational	4

Gambar 6. Grafik Penilaian Kerentanan

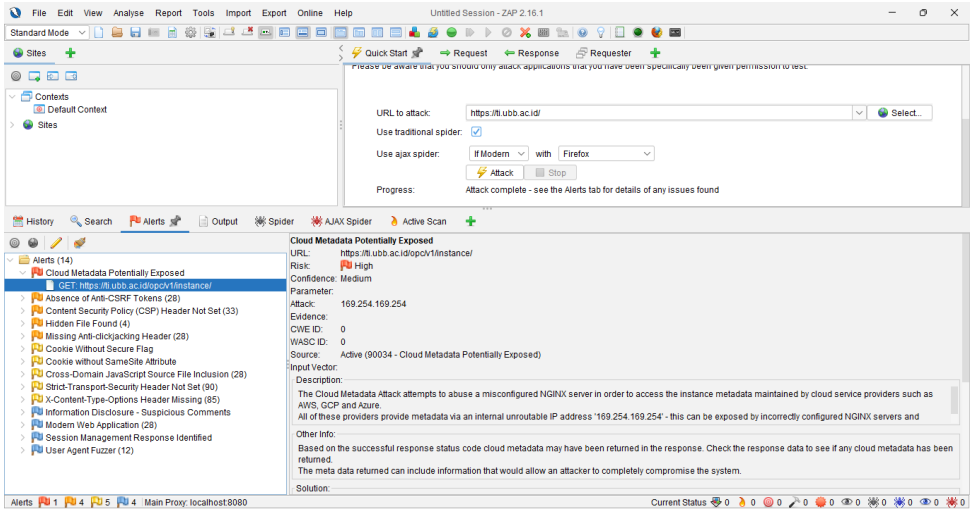
Berdasarkan hasil pemindaian menggunakan OWASP ZAP, terdapat total 14 peringatan keamanan pada *website* ti.ubb.ac.id. Rinciannya terdiri dari 1 risiko tinggi, 4 risiko sedang, 5 risiko rendah, dan 4 bersifat informasional. Hasil ini menunjukkan adanya beberapa potensi kerentanan yang perlu segera dievaluasi dan ditangani, terutama pada kategori risiko tinggi dan sedang, guna menjaga keamanan *website*.

Alerts

Name	Risk Level	Number of Instances
Cloud Metadata Potentially Exposed	High	1
Absence of Anti-CSRF Tokens	Medium	28
Content Security Policy (CSP) Header Not Set	Medium	33
Hidden File Found	Medium	4
Missing Anti-clickjacking Header	Medium	28
Cookie Without Secure Flag	Low	1
Cookie without SameSite Attribute	Low	1
Cross-Domain JavaScript Source File Inclusion	Low	28
Strict-Transport-Security Header Not Set	Low	90
X-Content-Type-Options Header Missing	Low	85
Information Disclosure - Suspicious Comments	Informational	1
Modern Web Application	Informational	28
Session Management Response Identified	Informational	2
User Agent Fuzzer	Informational	12

Gambar 7. *Alert*

Berdasarkan hasil pemindaian OWASP ZAP, terdapat 14 jenis peringatan keamanan pada *website*. Satu peringatan berisiko tinggi terkait potensi terbukanya *metadata cloud*. Selain itu, terdapat empat peringatan berisiko sedang, seperti tidak adanya token *Anti-CSRF, Content Security Policy* (CSP) yang belum diterapkan, *file* tersembunyi yang ditemukan, dan header *anti-clickjacking* yang belum diatur. Lima peringatan berisiko rendah umumnya berkaitan dengan konfigurasi *cookie* dan header keamanan yang belum optimal. Sisanya berupa peringatan informasional, seperti komentar mencurigakan, identifikasi aplikasi web modern, serta respon manajemen sesi. Hasil ini menunjukkan perlunya peningkatan konfigurasi dan pengamanan website untuk meminimalkan risiko serangan siber.



Gambar 8. Hasil Pemindaian OWASP ZAP

Dari hasil pemindaian OWASP ZAP diatas ditemukan celah Cloud Metadata Potentially Exposed pada URL https://ti.ubb.ac.id/opc/v1/instance/. OWASP ZAP mengidentifikasi celah ini sebagai risiko tinggi, karena metadata cloud dapat diakses dari IP internal 169.254.169.254. Respon server menunjukkan data metadata kemungkinan dikembalikan. Celah ini muncul akibat konfigurasi server NGINX yang salah dan dapat dimanfaatkan untuk pengambilalihan sistem.

Berikut adalah tabel pembobotan atau model scoring formal berdasarkan data dari tabel "Alerts" yang Anda berikan.

Tabel 1. Tabel Bobot

Risk Level	Weight (Score)	Description
High	5	Menunjukkan risiko kritis yang memerlukan tindakan segera
Medium	3	Menandakan risiko sedang yang perlu dikelola

Low	1	Menunjukkan risiko rendah yang biasanya tidak memerlukan perhatian langsung
Informational	0	Informasi yang tidak memerlukan tindakan tetapi harus diperhatikan

Tabel bobot diatas berfungsi untuk mengklasifikasikan ancaman atau risiko yang dapat dihadapi suatu sistem atau organisasi berdasarkan tingkat keparahannya.

Tabel 2. Standar *scoring*

Alert Name	*Risk Level*	*Weight*	*Number of Instances*	*Total Score (Weight x Instances)*
Cloud Metadata Potentially Exposed	*High*	5	1	5
Absence of Anti-CSRF Tokens	*Medium*	3	28	84
Content Security Policy (CSP) Header Not Set	*Medium*	3	33	99
Hidden File Found	*Medium*	3	4	12
Missing Anti-clickjacking Header	*Medium*	3	28	84
Cookie Without Secure Flag	*Low*	1	1	1
Cookie without SameSite Attribute	*Low*	1	1	1
Cross-Domain JavaScript Source File Inclusion	*Low*	1	28	28
Strict-Transport-Security Header Not Set	*Low*	1	90	90
X-Content-Type-Options Header Missing	*Low*	1	85	85
Information Disclosure - Suspicious Comments	*Informational*	0	1	0
Modern Web Application	*Informational*	0	28	0
Session Management Response Identified	*Informational*	0	2	0
User Agent Fuzzer	*Informational*	0	12	0

Dari analisis risiko, diperoleh total skor yang menunjukkan berbagai tingkat risiko. Terdapat 5 insiden yang terdeteksi sebagai High Risk, yang diwakili oleh eksposur metadata cloud, memerlukan perhatian segera untuk mencegah potensi ancaman yang lebih serius. Sementara itu, 279 insiden berisiko Medium, termasuk absennya beberapa parameter keamanan seperti anti-CSRF tokens dan Content Security Policy (CSP) header, yang menunjukkan perlunya manajemen dan mitigasi yang baik.

Risiko Low tercatat sebanyak 205 insiden, yang terdiri dari masalah seperti cookie tanpa secure flag dan atribut SameSite, yang meskipun tidak mendesak, tetap perlu dicatat untuk pemantauan lebih lanjut. Terakhir, tidak ada insiden Informational yang memerlukan tindakan segera, namun tetap diharapkan untuk diperhatikan agar tidak menjadi masalah di masa depan. Respons yang tepat terhadap berbagai level risiko ini penting untuk menjaga integritas dan keamanan sistem. Hasil pengujian ini memberikan gambaran mengenai jenis-jenis alert yang ditemukan beserta tingkat risikonya, sehingga dapat menjadi acuan dalam melakukan perbaikan dan peningkatan konfigurasi keamanan *website* secara menyeluruh untuk meminimalkan risiko serangan siber.

Berikut adalah hasil pengujian analisa keamanan *website* Prodi TI UBB yang dirangkum dalam Tabel 1.

Tabel 3. Pengujian Analisa *Website* Prodi TI UBB

No	Name	High	Medium	Low	Informational
1	Cloud Metadata Potentially Exposed	✓			
2	Absence of Anti-CSRF Tokens		✓		
3	Content Security Policy (CSP) Header Not Set		✓		
4	Hidden File Found		✓		
5	Missing Anti-clickjacking Header		✓		
6	Cookie Without Secure Flag			✓	
7	Cookie without SameSite Attribute			✓	
8	Cross-Domain JavaScript Source File Inclusion			✓	
9	Strict-Transport-Security Header Not Set			✓	
10	X-Content-Type-Options Header Missing			✓	
11	Information Disclosure - Suspicious Comments				✓
12	Modern Web Application				✓
13	Session Management Response Identified				✓
14	User Agent Fuzzer				✓

Berdasarkan Tabel diatas diperoleh keterangan sebagai berikut :

a. *Cloud Metadata Potentially Exposed*: Adanya potensi metadata cloud dapat diakses dari luar, sehingga berisiko membocorkan informasi sensitif.
b. *Absence of Anti-CSRF Tokens*: Tidak ada perlindungan terhadap token Anti-CSRF form, sehingga rawan serangan pemalsuan permintaan (CSRF).
c. *Content Security Policy (CSP) Header Not Set*: Header CSP belum diterapkan, sehingga website lebih rentan terhadap serangan injeksi skrip (XSS).
d. *Hidden File Found*: Ditemukan file tersembunyi yang bisa diakses publik, berpotensi mengungkap informasi penting.
e. *Missing Anti-clickjacking Header*: Tidak ada header *anti-clickjacking*, sehingga website bisa dimasukkan ke dalam frame situs lain (*clickjacking*).
f. *Cookie Without Secure Flag*: *Cookie* dikirim tanpa flag "*Secure*", sehingga bisa diakses melalui koneksi yang tidak terenkripsi.
g. *Cookie without SameSite Attribute*: *Cookie* tidak membatasi pengiriman hanya pada domain yang sama, meningkatkan risiko CSRF.
h. *Cross-Domain JavaScript Source File Inclusion*: *Website* memuat *JavaScript* dari domain luar, berpotensi membawa skrip berbahaya.
i. *Strict-Transport-Security Header Not Set*: Header HSTS belum diaktifkan, sehingga koneksi ke *website* tidak selalu aman (HTTPS).
j. *X-Content-Type-Options Header Missing*: Tidak ada header untuk mencegah sniffing tipe konten, sehingga file bisa dijalankan sebagai tipe lain.
k. *Information Disclosure - Suspicious Comments*: Ada komentar mencurigakan di kode sumber yang bisa mengungkap informasi internal.
l. *Modern Web Application*: *Website* terdeteksi sebagai aplikasi web modern, informasi ini relevan untuk pengujian lanjutan.
m. *Session Management Response Identified*: Sistem manajemen sesi terdeteksi, perlu dipastikan sudah aman dari pembajakan sesi.
n. *User Agent Fuzzer*: *Website* merespons berbeda pada user agent tertentu, bisa dimanfaatkan untuk uji keamanan lebih lanjut.

Tabel 4. Rekomendasi Aplikasi Pengujian

No	Name	Risk Level	Number of Instances
1	Cloud Metadata Potentially Exposed	High	1
2	Absence of Anti-CSRF Tokens	Medium	28
3	Content Security Policy (CSP) Header Not Set	Medium	33
4	Hidden File Found	Medium	4
5	Missing Anti-clickjacking Header	Medium	28
6	Cookie Without Secure Flag	Low	1
7	Cookie without SameSite Attribute	Low	1
8	Cross-Domain JavaScript Source File Inclusion	Low	28
9	Strict-Transport-Security Header Not Set	Low	90
10	X-Content-Type-Options Header Missing	Low	85
11	Information Disclosure - Suspicious Comments	Informational	6
12	Modern Web Application	Informational	28
13	Session Management Response Identified	Informational	6
14	User Agent Fuzzer	Informational	12

Berdasarkan tabel diatas maka didapati hasil rekomendasi perbaikan sebagai berikut :

a. Pastikan *endpoint metadata cloud* tidak dapat diakses dari luar jaringan internal. Terapkan *firewall* dan kontrol akses yang ketat agar hanya layanan yang sah yang dapat mengakses *metadata*. Selain itu, lakukan audit berkala terhadap konfigurasi cloud untuk memastikan tidak ada kebocoran informasi sensitif yang dimanfaatkan oleh pihak tidak bertanggung jawab.
b. Gunakan perpustakaan atau *framework* yang menyediakan perlindungan terhadap serangan CSRF, seperti OWASP CSRFGuard atau fitur *built-in* pada *framework modern*. Selalu sertakan token CSRF pada setiap form dan endpoint yang memproses data pengguna. Pastikan aplikasi juga terlindungi dari XSS, karena serangan CSRF dapat dilewati jika ada celah XSS. Selain itu, lakukan validasi pada header HTTP Referer untuk memastikan permintaan berasal dari sumber yang diharapkan, meskipun cara ini memiliki keterbatasan karena beberapa pengguna atau *proxy* dapat menonaktifkan pengiriman referer demi privasi.
c. Aktifkan dan konfigurasikan header *Content-Security-Policy* di seluruh halaman aplikasi. CSP membatasi sumber daya eksternal yang dapat dimuat oleh browser, sehingga dapat mencegah serangan seperti *Cross-Site Scripting* (XSS) dan injeksi konten. Pastikan hanya sumber yang terpercaya yang diizinkan untuk dimuat, dan lakukan pengujian secara berkala untuk memastikan kebijakan CSP sudah berjalan efektif.
d. Lakukan audit terhadap file dan direktori yang ada pada server web. Hapus file tersembunyi yang tidak diperlukan atau atur permission agar tidak dapat diakses secara publik. File tersembunyi sering kali berisi informasi sensitif atau konfigurasi yang dapat dimanfaatkan oleh penyerang jika ditemukan.
e. Pastikan salah satu atau keduanya disetel pada seluruh halaman web. Jika halaman tidak perlu dibingkai oleh situs lain, gunakan *X-Frame-Options* dengan nilai DENY. Jika hanya boleh dibingkai oleh domain sendiri, gunakan SAMEORIGIN. Untuk pengaturan yang lebih fleksibel, gunakan CSP dengan *frame-ancestors* yang spesifik.
f. Pastikan cookie tersebut selalu dikirim melalui saluran terenkripsi (HTTPS) dengan mengaktifkan *flag Secure*. Dengan demikian, cookie tidak akan pernah dikirim melalui koneksi HTTP yang tidak aman, sehingga mengurangi risiko pencurian data sesi.
g. Tambahkan atribut *SameSite* pada setiap *cookie* yang digunakan. *SameSite=Strict* membatasi pengiriman *cookie* hanya pada permintaan dari domain yang sama, sedangkan *SameSite=Lax* memberikan sedikit kelonggaran untuk navigasi tertentu. Penggunaan atribut ini dapat membantu mencegah serangan CSRF secara efektif.
h. Minimalkan pemuatan file *JavaScript* dari domain eksternal. Jika harus menggunakan sumber eksternal, pastikan hanya dari vendor atau sumber yang benar-benar terpercaya. Lakukan *review* berkala terhadap skrip eksternal yang digunakan untuk menghindari risiko penyisipan kode berbahaya.
i. Aktifkan header HTTP *Strict-Transport-Security* (HSTS) pada server. Header ini akan memaksa *browser* untuk selalu menggunakan HTTPS saat mengakses *website*, sehingga mencegah *downgrade attack* dan memastikan komunikasi antara pengguna dan server selalu terenkripsi.
j. Pastikan setiap server web menggunakan header *X-Content-Type-Options* ke nosniff pada seluruh respons. Ini akan mencegah *browser* melakukan sniffing tipe konten yang dapat menyebabkan file dijalankan sebagai

tipe yang tidak semestinya. Selain itu, pastikan juga header *Content-Type* sudah diatur dengan benar untuk setiap file yang dikirimkan.

k. Hapus komentar mencurigakan atau sensitif dari kode sumber sebelum dipublikasikan. Komentar di dalam kode dapat berisi informasi penting tentang struktur aplikasi, konfigurasi, atau bahkan kredensial yang dapat dimanfaatkan oleh penyerang.

l. Pastikan seluruh komponen aplikasi web modern yang digunakan selalu diperbarui ke versi terbaru. Lakukan audit keamanan secara berkala pada *framework, library*, dan *plugin* yang digunakan untuk mengurangi risiko dari kerentanan yang sudah diketahui.

m. Tinjau dan perbaiki mekanisme manajemen sesi agar menggunakan praktik terbaik, seperti rotasi ID sesi setelah *login*, pengaturan waktu kedaluwarsa sesi yang wajar, dan penggunaan *cookie* dengan *flag Secure* dan *HttpOnly*. Hindari penyimpanan informasi sensitif dalam sesi yang mudah diakses.

n. Pantau respons aplikasi terhadap berbagai *user agent* untuk memastikan tidak ada perilaku tak terduga yang bisa dimanfaatkan penyerang. Terapkan validasi dan *filter* pada input *user agent* untuk mencegah eksploitasi celah keamanan melalui manipulasi header *user agent*.

3.2. Pembahasan

Berdasarkan rangkaian pengujian yang telah dilakukan, diperoleh gambaran menyeluruh mengenai kondisi keamanan *website* Prodi Teknologi Informasi Universitas Bangka Belitung. Pada tahap awal, hasil cek domain melalui Who.is menunjukkan bahwa informasi kepemilikan domain, kontak administratif, dan data teknis masih dapat diakses secara publik. Kondisi ini dapat meningkatkan risiko teknik *footprinting*, di mana penyerang dapat mengumpulkan informasi sensitif sebagai langkah awal dalam perencanaan serangan. Oleh karena itu, pembatasan akses terhadap data domain menjadi langkah penting untuk meminimalisir potensi penyalahgunaan. Selanjutnya, hasil analisis NsLookup mengungkapkan konfigurasi DNS yang digunakan oleh *website*, termasuk alamat IP dan *record* penting seperti A, MX, serta CNAME. Ditemukan bahwa beberapa *record* DNS masih menggunakan konfigurasi standar tanpa perlindungan tambahan, sehingga berpotensi dimanfaatkan untuk serangan DNS *spoofing* atau pemalsuan identitas domain. Penguatan konfigurasi DNS, seperti penerapan DNSSEC, sangat disarankan untuk meningkatkan keamanan pada lapisan ini.

Secara keseluruhan, hasil pengujian ini menegaskan bahwa *website* Prodi Teknologi Informasi UBB masih memiliki sejumlah kelemahan yang perlu segera diperbaiki. Setiap hasil analisis dari *tools* yang digunakan saling melengkapi dan memberikan gambaran komprehensif mengenai kondisi keamanan *website*, mulai dari aspek infrastruktur, konfigurasi, hingga aplikasi. Oleh karena itu, diperlukan upaya perbaikan secara menyeluruh, seperti pembatasan akses informasi domain, penguatan konfigurasi DNS dan *firewall*, penutupan port yang tidak diperlukan, pembaruan perangkat lunak, serta penerapan kebijakan keamanan aplikasi web yang sesuai standar. Dengan langkah-langkah tersebut, diharapkan tingkat keamanan website dapat ditingkatkan secara signifikan dan risiko eksploitasi dapat diminimalisasi.

4. KESIMPULAN

Berdasarkan hasil analisis keamanan pada *website* program studi teknologi informasi universitas bangka belitung menggunakan metode *application scanning*, disimpulkan bahwa metode ini efektif dalam mengidentifikasi berbagai celah keamanan secara otomatis dan sistematis. Penulis berhasil menemukan berbagai jenis-jenis kerentanan potensial yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab, seperti kelemahan pada autentifikasi, input data, dan konfigurasi *server*. Penelitian ini juga tidak hanya memberikan gambaran nyata terhadap kondisi keamanan situs akademik berbasis *website*, tetapi juga mendorong pentingnya implementasi manajemen keamanan berbasis teknologi secara berkelanjutan untuk menjamin layanan digital yang andal dan aman.

DAFTAR PUSTAKA

[1] N. Albalawi, N. Alamrani, R. Aloufi, M. Albalawi, A. Aljaedi, and A. R. Alharbi, “The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities,” *Electron.*, vol. 12, no. 12, 2023, doi: 10.3390/electronics12122664.

[2] E. Nurelasari and D. Gumilang Al Farabi, “Analisis Keamanan Sistem Website Menggunakan Metode Open Web Application Security Project (Owasp) Pada Simantep.Id,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3049–3054, 2024, doi: 10.36040/jati.v8i3.9314.

[3] I. Riadi, A. Yudhana, and Y. W, “Analisis Keamanan Website Open Journal System Menggunakan Metode Vulnerability Assessment,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, pp. 853–860, 2020, doi: 10.25126/jtiik.2020701928.

[4] H. Herman, I. Riadi, Y. Kurniawan, and I. A. Rafiq, “Analisis Keamanan Website Menggunakan Information System Security Asessment Framework(ISSAF),” *J. Teknol. Inform. dan Komput.*, vol. 9, no. 1, pp. 126–136, 2023, doi: 10.37012/jtik.v9i1.1439.

[5] A. Y. Eshetu, E. A. Mohammed, and A. O. Salau, “Cybersecurity vulnerabilities and solutions in Ethiopian

university websites," *J. Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00980-z.

[6] R. Mayasari, A. Ali Ridha, D. Juardi, and K. Ahmad Baihaqi, "Analisis Vulnerability pada Website Universitas Singaperbangsa Karawang menggunakan Acunetix Vulnerability," *Systematics*, vol. 2, no. 1, p. 33, 2020, doi: 10.35706/sys.v2i1.3450.

[7] Rizki Alam Ramdhani and Nunu Nurdiana, "Rancang Bangun Aplikasi Analisis Standar Keamanan Website Dengan Metode Scanning Vulnerability Menggunakan Module Requests Python," *Semin. Teknol. Majalengka*, vol. 6, pp. 271–277, 2022, doi: 10.31949/stima.v6i0.699.

[8] M. D. Putra, I. Artikel, and A. Info, "Analisis Website E-learning Bina Darma Menggunakan Metode Web Application Security Project Zap (Owasp Zap)," vol. 4, no. 1, pp. 8–15, 2025.

[9] M. F. A. Ramadhan *et al.*, "ANALISIS ANCAMAN KEAMANAN PADA SISTEM INFORMASI AKADEMIK KAMPUS MENGGUNAKAN METODE OWASP ZAP," vol. 8, no. 4, pp. 7985–7991, 2024.

[10] A. F. Hasibuan and D. Handoko, "Analisis Keretanan Website Dengan Aplikasi Owasp Zap," *J. Ilmu Komput. dan Sist. Inf.*, vol. 2, no. 2, pp. 257–270, 2023, [Online]. Available: https://jurnal.unity-academy.sch.id/index.php/jirsi/article/view/51

[11] 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.

[12] G. Kusuma, "Implementasi Owasp Zap Untuk Pengujian Keamanan Sistem Informasi Akademik," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 16, no. 2, pp. 178–186, 2022, doi: 10.47111/jti.v16i2.3995.

[13] F. Ayu, E. Ryansyah, F. Maulana, R. Alamsyah, and A. Susilo, "Analisis Kerentanan Keamanan Menggunakan OWASP ZAP dan Pengujian Manual pada Tampilan Antarmuka Laman PDDIKTI," vol. 13, no. 3, pp. 671–678, 2025.

[14] M. Amirul, N. Tristanti, G. Pramuja, and I. Fanani, "Analisis dan Pengujian Kerentanan Website Menggunakan OWASP ZAP," vol. 3, no. 1, pp. 36–50, 2024.

[15] K. Nisa, M. A. Putra, R. A. Siregar, and M. Dedi Irawan, "Analisis Website Tapanuli Tengah Menggunakan Metode Open Web Application Security Project Zap (Owasp Zap)," *Bull. Inf. Technol.*, vol. 3, no. 4, pp. 308–216, 2022, doi: 10.47065/bit.v3i4.389.

[16] M. D. Abdillah, J. Gunawan, R. A. Atsil, and A. M. Harahap, "Analisis Kerentanan Website Mtss Al-Washliyah Bah Gunung Menggunakan Metode Open Web Application Security Project ZAP (OWASP ZAP)," *J. Sains dan Teknol.*, vol. 3, no. 1, pp. 61–67, 2023, doi: 10.47233/jsit.v3i1.487.

[17] D. K. Linux, C. Bernandra, P. Pura, T. Y. Maulana, A. Februri, and T. Ariyadi, "Analisis Celah Keamanan Website Menggunakan Tools OWASP ZAP," vol. 4, no. 1, 2025.

[18] B. T. K. & M. A. S. Dewi, "Kajian Literatur: Metode dan Tools Pengujian Celah Keamanan Aplikasi Berbasis Web," *Automata*, vol. 3, no. 1, pp. 1–8, 2022, [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/view/21883/12030

[19] L. F. Burhani and D. Priyawati, "Analisis Pengujian Keamanan Website Pengelolaan Internet Desa Kragan Menggunakan Metode Penetration Testing Execution Standard (Ptes)," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 1, pp. 307–319, 2024, doi: 10.29100/jipi.v9i1.4455.

[20] F. Septian, M. H. Arfian, J. S. Asri, and B. Tjahjono, "Pengujian Keamanan Website dengan Metode Penetration Testing (Studi Kasus : Universitas Esa Unggul)," vol. 4, pp. 3629–3647, 2024.

[21] M. Rifaldi, R. Satra, A. Widya, and M. Gaffar, "Analisis Keamanan Website dengan Metode Penetration Testing pada PT . PLN (Persero)," vol. 1, no. 3, pp. 293–301, 2024.

[22] N. A. S. Akmal, Alif Muhammad Heryana, "Analisis Keamanan Website Universitas Singaperbangsa Karawang Menggunakan Metode Vulnerability Assessment," *Volume 4 Nomor 4 tahun 2022*.