Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Vulnerability Analysis and Mitigation of Village Website Using Vulnerability Scanner Based on PTES Method

Miftahullah[1]*, Ondi Asroni[2], Muhammad Haris Nasri[3]

[1,2,3]Fakultas Teknik, Universitas Bumigora
*miftahullah173@gmail.com[1]**

Abstract

Web application security is a top priority in the digital era, especially for public services such as village websites. This research aims to analyze and mitigate web application vulnerabilities using the Penetration Testing Execution Standard (PTES) method with Acunetix and Nessus tools. The test was conducted on the website of Kediri Selatan Village, West Lombok Regency. To avoid risks to the production system, the test used a replica of the website. The testing process follows the PTES stages: pre-engagement, intelligence gathering, vulnerability analysis, exploitation, and reporting. The scan results showed medium and low-category vulnerabilities from Acunetix and two critical vulnerabilities from Nessus. Although automated tests did not detect SQL injection and XSS vulnerabilities, manual exploitation proved their existence. Mitigation was performed with input validation and script filters, which proved to eliminate the vulnerabilities on retest. This research provides an applicable security implementation model that can be adapted by other web services for villages.

Keywords: *Acunetix, Nessus, Penetration Testing, PTES, Web Security,*

1. Introduction

The rapid development of information technology has prompted governments, including village administrations, to provide web-based digital services to support transparency and effective public services. However, the development of village websites is often carried out without adequately considering web application security aspects, making them potential targets for cyberattacks. According to the National Cyber and Cryptography Agency (BSSN), in 2023 there were more than 403 million anomalous traffic incidents, 189 cases of website defacement, and over 1.6 million data breaches affecting the public administration sector in Indonesia [1].

The two most common vulnerabilities targeting web applications are SQL Injection (SQLi) and Cross-Site Scripting (XSS) [2][3]. SQLi allows attackers to inject illegal SQL commands into user input forms, while XSS enables the execution of malicious scripts on the client side. These vulnerabilities are still commonly found in information systems belonging to schools and government agencies that lack adequate input sanitization [4].

To identify and address these vulnerabilities, penetration testing is an effective and structured approach. One commonly used framework is the Penetration Testing Execution Standard (PTES), which includes five main stages: pre-execution interaction, information gathering, vulnerability analysis, exploitation, and reporting [5][6]. This approach can accommodate the need for comprehensive testing of web-based systems and internal networks[7][8].

In practice, penetration testing is not only done manually but also aided by automated tools such as Acunetix and Nessus. Acunetix is a widely used vulnerability scanner for detecting security flaws in web applications such as SQLi, XSS, and CSRF. Meanwhile, Nessus is used to detect weaknesses on the server and network side, such as incorrect configurations, open ports, and outdated protocols [9][10]. The combined use of these two tools can provide more comprehensive detection results than using either tool alone, especially when combined with manual exploitation.

The Kediri Selatan Village website was selected as the research object because it has characteristics and features commonly found on other village websites in Indonesia. This site provides services such as village profiles, community contacts, and information request forms. However, previous research by Ramadhani and Heriyanto revealed that this system has not implemented input sanitization or protection against malicious scripts, which allows attacks such as SQLi and XSS to occur. Furthermore, Pranata et al. demonstrated that on some government systems, the login form can be bypassed with a simple payload like ‘OR '1’='1'.

Therefore, this study aims to analyze and mitigate the vulnerabilities found on the Kediri Selatan Village website using Acunetix and Nessus tools, with testing stages following the PTES method. With this approach, it is hoped that this study can contribute to strengthening the security system of village websites and serve as a reference for other digital public service managers in supporting national cybersecurity strategies.

2. Research Method

This research uses the penetration testing execution standard method. Penetration testing is a controlled attack simulation that helps identify vulnerabilities in applications, networks, and operating system branches [11].

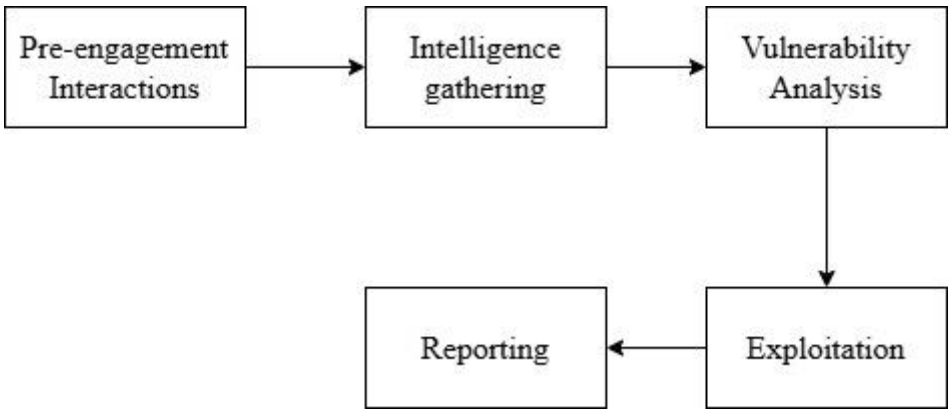


Fig. 1: Stages of the Standard Penetration Test Implementation Method

Figure 1 shows the penetration testing process flow based on the Penetration Testing Execution Standard (PTES) framework. PTES consists of five main interconnected stages that form a comprehensive security testing cycle, namely pre-engagement interactions, intelligence gathering, vulnerability analysis, exploitation, and reporting.

2.1. Pre-engagement Interactions

The pre-engagement interactions phase aims to provide and explain the tools or techniques available to assist in initiating a penetration test. Selecting the right tools for a penetration test will depend on the type and depth of testing.
The testing was conducted using a grey-box approach, in which the author had limited access to the internal system but knew the general structure of the application to be tested. This was done to simulate an attack from an external party with partial information about the system.

2.2. Intelligence Gathering

During the intelligence-gathering stage, various preliminary information about the target website, "kediriselatan.web.id", was collected. The information gathered included open-source technical data such as domain details (via WHOIS) and security system identification. This stage is very important as it provides a basis for understanding the initial condition of the system before proceeding to the vulnerability analysis stage.

```
Domain Name: KEDIRISELATAN.WEB.ID
Registry Domain ID: PANDI-DO14326378
Registrar WHOIS Server:
Registrar URL: www.rna.id
Updated Date: 2025-05-02T07:50:04Z
Creation Date: 2025-05-02T07:48:38Z
Registry Expiry Date: 2026-05-02T23:59:59Z
Registrar: PT Registrasi Neva Angkasa
Registrar IANA ID: 1
Registrar Abuse Contact Email: admin@rna.id
Registrar Abuse Contact Phone:
Domain Status: clientTransferProhibited
Domain Status: serverTransferProhibited
Name Server: NSX1.DOMAINESIA.COM
Name Server: NSX2.DOMAINESIA.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
```

Fig. 2: WhoIs scan results

Figure 2 shows the WHOIS scan results for the "kediriselatan.web.id" domain, it is known that this domain was registered through PT Registrasi Neva Angkasa (RNA). This domain was created on May 2, 2025, and will expire on May 2, 2026. The domain status shows "client transfer prohibited" and "server transfer prohibited," which means that the domain cannot be transferred without approval. Additionally, this domain uses the name servers NSX1.DOMAINESIA.COM and NSX2.DOMAINESIA.COM, indicating that its services are likely hosted by DomaiNesia. It is also important to note that the DNSSEC for this domain is still unsigned, indicating that there is no additional security layer in the DNS system.

2.3. Vulnerability Analysis

Vulnerability analysis is used to identify and evaluate security risks posed by identified vulnerabilities. This analysis work is divided into two areas, namely identification and validation. Vulnerability analysis was conducted using two approaches, namely automatic and manual, in order to obtain comprehensive results. The automatic approach used Acunetix tools to detect vulnerabilities on the web application side, such as SQL Injection, XSS, and weak security configurations, as well as Nessus to identify weaknesses on the server and network side, such as open ports and obsolete protocols. Meanwhile, manual testing was performed by inserting the payload ‘OR '1’='1' into the login form to test for SQL Injection vulnerabilities, and <script>alert(‘XSS Attack’)</script> into the text input field to test for Cross-site Scripting (XSS) vulnerabilities. The combination of these two methods enables validation of findings and provides a comprehensive overview of the security level of the tested system.

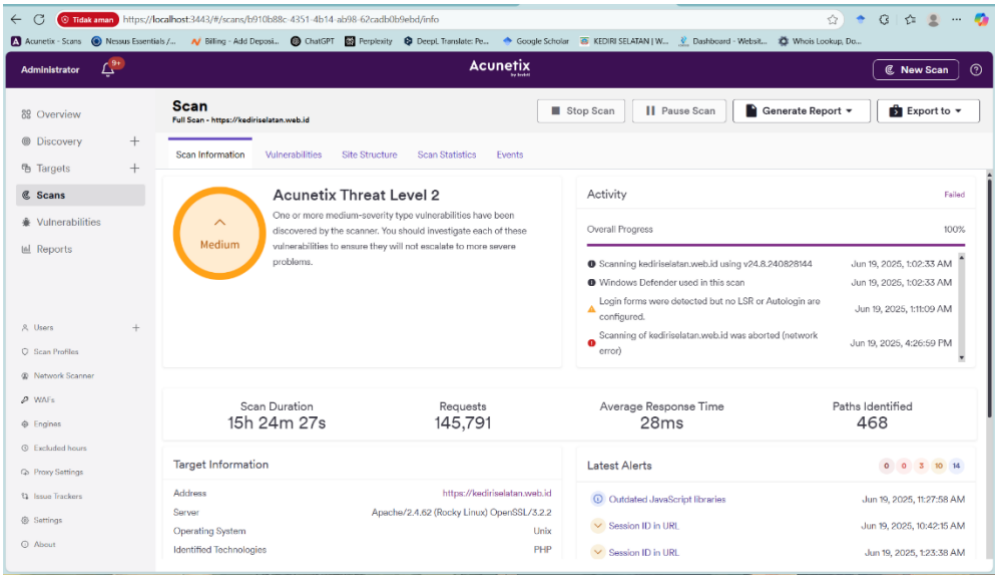
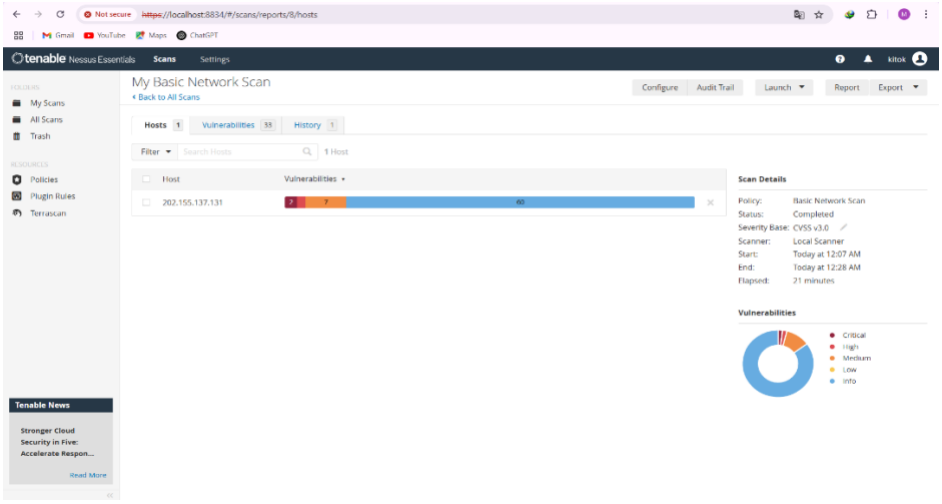


Fig. 3: Display of website security scan results using Acunetix.

Figure 3 above shows the results of a security scan of the website using Acunetix, an automation tool for detecting web application vulnerabilities. Based on the results displayed, the website with the domain https://kediriselatan.web.id/ has a medium threat level (Acunetix Threat Level 2). During the 25-minute scanning process, 1,850 requests were sent, and Acunetix successfully identified 83 vulnerabilities across various paths. The average server response time was 28 milliseconds. Some of the latest warnings include findings such as reflected cross-site scripting (XSS), cookies not marked as secure, generic HTML injection, and permissive cross-domain policy.



These findings

Fig. 4: Details of vulnerabilities found by Nessus

indicate security vulnerabilities that need to be addressed immediately, particularly regarding input sanitization and cookie configuration. Figure 4 shows the results of the scan conducted using Nessus, which found 2 alerts with a critical risk level, 7 alerts with a medium risk level, and 60 alerts of an informational nature, while no vulnerabilities were found at the high or low levels. These results indicate that the target system has serious potential security gaps, particularly in the critical category, which are highly vulnerable to exploitation by attackers and require immediate attention.

2.4. Exploitation

After performing automatic scans using Acunetix and Nessus, no vulnerabilities to SQL injection and XSS were found on the kediriselatan.web.id website. However, to ensure the overall security of the system, manual testing was performed using common payloads often used by attackers.

Manual testing was performed on the login form by inserting the payload ‘OR '1’='1 in the username and password fields. This payload successfully bypassed the authentication process and directed users directly to the dashboard page without valid credentials. The success of this exploit shows that the system does not perform input sanitization and is still vulnerable to SQL query logic manipulation.

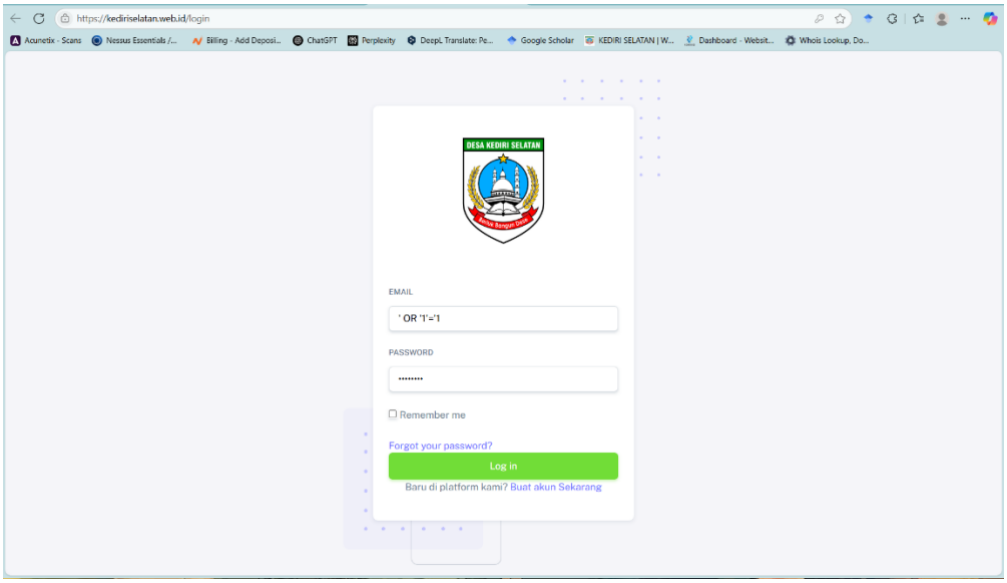


Fig. 5: Manual SQL Injection Exploitation Process

After the payload is entered into the username and password fields as shown in Figure 5, the system directs the user to the dashboard page without requiring valid credentials.

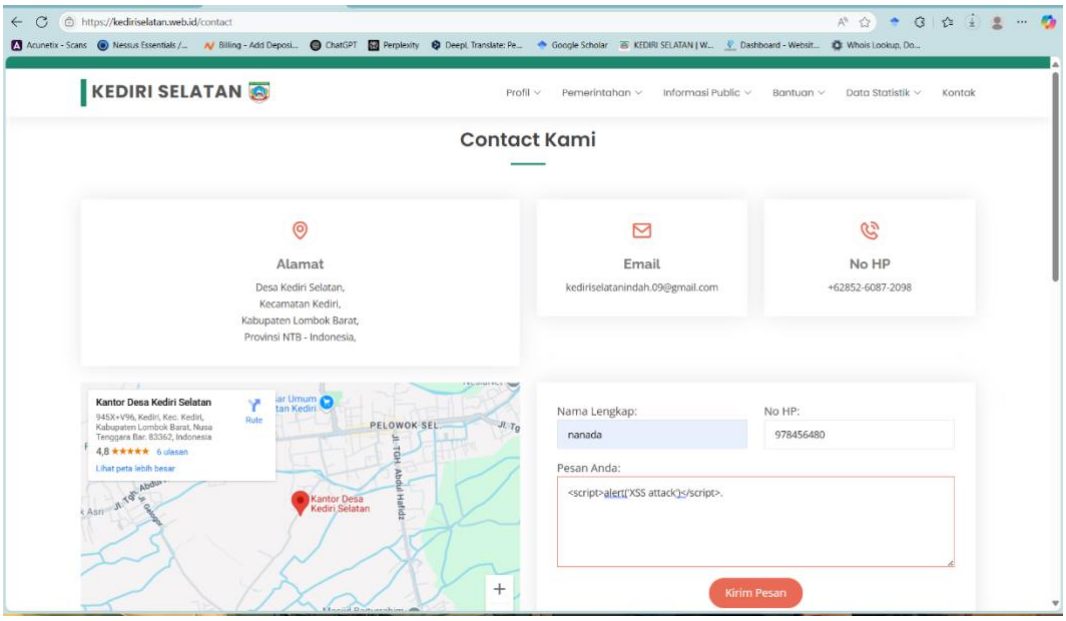


Fig. 6: Manual XSS Exploitation Process

XSS testing was performed on the contact form shown in Figure 6 by inserting the code <script>alert(‘XSS attack’)</script> in the message field. When the message was displayed on the page, the JavaScript code was executed and displayed a pop-up saying “XSS attack,” indicating that the system displayed user input without adequate filtering or validation on the client side.

3. Result and Discussion

3.1. Reporting and Recommendations

The reporting phase is the final stage in the penetration testing process, which aims to systematically document all findings, security risks, and mitigation measures that have been taken against the South Kediri Village website. This documentation is very important so that system administrators can understand the security vulnerabilities that have been found and take appropriate corrective action.

Table 1: Vulnerability mitigation reports and recommendations

No.	Type of Vulnerability	Payload Used	Impact	CVSS Score (Estimated)	Status	Mitigation Recommendations
1	SQL Injection	' OR '1'='1	Bypass admin login	8.6 (High)	Found	Use prepared statements (such as Auth::attempt() or Query Builder in Laravel) and validate and sanitize user input.
2	Cross-Site Scripting (XSS)	<script>alert('XSS Attack')</script>	Executing arbitrary script (reflected XSS)	6.1 (Medium)	Found	Display output using auto escaping ({{ }} in Laravel Blade), avoid {!! !!} for user input, and validate input.
3	Cookie Without HttpOnly Flag	-	Cookies can be stolen by JavaScript	5.3 (Medium)	Found	Add the `HttpOnly`, `Secure`, and `SameSite` attributes to the cookie.
4	Insecure Frame (Missing X Frame Options)	-	Potential clickjacking attack	4.9 (Medium)	Found	Add the header `X-Frame-Options: DENY` or `SAMEORIGIN`.
5	Subresource Integrity Not Implemented	-	External files are vulnerable to being replaced by third parties	3.7 (Low)	Found	Use the `integrity` and `crossorigin` attributes when calling external files.
6	SSL/TLS Weak Cipher Supported	-	Traffic can be intercepted through weak ciphers	9.0 (Critical)	Found	Disable weak ciphers, enable TLS 1.2+, and reconfigure the web server.
7	Missing Security Headers	-	Vulnerable to XSS & content sniffing	6.5 (Medium)	Found	Add `Content-Security-Policy`, `X-XSS-Protection`, and `X-Content-Type-Options`.
8	Directory Listing Enabled	-	Directory information is publicly accessible	5.0 (Medium)	Found	Disable the directory listing option in the server configuration (`Options -Indexes`).
9	Input Without Validation (Contact Form)	<script>alert('XSS Attack')</script>	Potential XSS, spam, and input manipulation	4.5 (Low)	Found	Use server-side validation and whitelist filters for user input.
10	No Session Timeout	-	Session remains active even if the user is idle	3.8 (Low)	Found	Apply session time limits and automatic logout after a certain period of inactivity.

Table 1 shows a summary of the results of testing websites that have a number of vulnerabilities originating from automatic scanning using Acunetix and Nessus tools, as well as manual testing.

3.2. Mitigation of SQL Injection and XSS Vulnerabilities

SQL injection was found in the login form during manual testing by inserting the payload `‘ OR '1’='1` in the email or username field. As a result, the system granted access without valid authentication, indicating that user input was directly processed into the SQL query without validation or the use of prepared statements.

```php
public function authenticate(): void
{
    $this->ensureIsNotRateLimited();

    if (! Auth::attempt($this->only('email', 'password'), $this->boolean('remember'))) {
        RateLimiter::hit($this->throttleKey());

        throw ValidationException::withMessages([
            'email' => trans('auth.failed'),
        ]);
    }

    RateLimiter::clear($this->throttleKey());
}
```

Fig. 7: Code implementation for SQL injection mitigation

After mitigation, the code was fixed using Laravel's `Auth::attempt()` method, as shown in Figure 7, which automatically applies prepared statements and validates passwords through a hashing process. This approach is effective in preventing SQL injection attacks.
An XSS vulnerability was discovered during manual testing of the contact form. An attacker could insert a script such as `<script>alert('XSS Attack')</script>` into the message field. Because the system displays the message using the `{!! !!}` syntax in Blade, the script is executed in the browser.

Fig. 8: Code implementation for XSS Attack mitigation

After the mitigation shown in Figure 8, the system uses the syntax `{{ $g->message }}` to display the input. Blade automatically converts HTML characters to safe entities, preventing scripts from being executed in the user's browser.
These two major vulnerabilities were identified without the help of automated tools, but rather through direct manual exploration and testing. Code-based mitigation measures proved to be effective in securing the server side, particularly in the Laravel framework, which provides built-in security features.

4. Conclusion

Based on the research conducted, it can be concluded that testing was carried out on a dummy website developed independently by mimicking the structure and functions of the South Kediri Village website to avoid disrupting the original production system. The results of automatic scanning using Acunetix and Nessus successfully identified various vulnerabilities, such as insecure cookies, missing subresource integrity, and two critical vulnerabilities from the Nessus results. Meanwhile, manual testing was able to detect SQL injection and XSS vulnerabilities that were not found by automatic tools, thus proving the importance of combining automatic and manual approaches in security testing. Exploitation was successfully carried out on the login form and contact form features, indicating weak input validation mechanisms and the need for input sanitization and parameterized queries. After mitigation and code fixes were implemented, retesting showed that the vulnerabilities could no longer be exploited, indicating that the implemented fixes were effective in enhancing system security.

References

[1] BADAN SIBER DAN SANDI NEGARA RI, "Laporan Keamanan Siber Indonesia (Bssn)," 2023.
[2] A. Gustiyono, E. I. Alwi, and S. M. Abdullah, "Analisa Kerentanan Website Terhadap Serangan Cross-Site Scripting (XSS) Metode Penetration Testing Analyze Website Vulnerability To Cross-Site Scripting (XSS) Attacks Using Penetration Testing," vol. 7, no. 1, pp. 25–33, 2024.
[3] D. P. I. Kusuma, N. H. Maulida, M. Ma'rifat, and D. Hariyadi, "Evaluasi Potensi Celah Keamanan SQL Injection Menggunakan Nearest Neighbor pada Security-Software Development Life Cycle," *J. Repos.*, vol. 2, no. 9, pp. 1273–1280, 2020, doi: 10.22219/repositor.v2i9.999.
[4] A. Zirwan, "Pengujian dan Analisis Kemanan Website Menggunakan Acunetix Vulnerability Scanner," *J. Inf. dan Teknol.*, vol. 4, no. 1, pp. 70–75, 2022, doi: 10.37034/jidt.v4i1.190.
[5] R. M. Fauzi, R. Hermawan, D. R. Adhy, and S. Maesaroh, "Analisis Kerentanan Keamanan Web Menggunakan Metode Owasp Dan Ptes Di Web Pemerintahan Desa Xyz," *Power Elektron. J. Orang Elektro*, vol. 13, no. 2, pp. 225–231, 2024, doi: 10.30591/polektro.v13i2.6711.
[6] Muhammad Risky Ardiansyah *et al.*, "Analisis Kerentanan Keamanan Website Menggunakan Metode PTES (Penetration Testing Execution And Standart)," *Nuansa Inform.*, vol. 18, no. 2, pp. 145–153, 2024, doi: 10.25134/ilkom.v18i2.119.
[7] G. Arna, J. Saskara, U. P. Ganesha, and U. P. Ganesha, "PENGUJIAN KEAMANAN DENGAN METODE PENETRATION TESTING EXECUTION STANDARD (PTES) UNTUK MENEMUKAN KERENTANAN MISCONFIGURATIONS PADA PERANGKAT SECURITY TESTING WITH PENETRATION TESTING EXECUTION STANDARD (PTES) METHODS TO FIND MISCONFIGURATIONS VULNERABIL," *J. Elektro Luceat*, vol. 10, no. 2, 2024.
[8] M Hasym Azwar and Alfina Yuliana, "Analisis Kualitas Layanan Jaringan Internet Wifi Pusdiskom Dengan Metode Peneration Testing," *JICode J. Inform. Dan Komput.*, vol. 1, no. 1, pp. 9–12, 2024.
[9] M. R. Syaifudin, M. A. Murtadho, M. S. Wafa, and M. Masrur, "Analisis Keamanan Website Kampus UNIPDU Melalui Metode Vulnerability Assessment (VA) dengan Menggunakan Tools Acunetix UNIPDU Campus Website Security Analysis Through Vulnerability Assessment (VA) Metho," *KOMPUTA J. Ilm. Komput. dan Inform.*, vol. 14, no. 1, pp. 7–12, 2025, doi: 10.34010/komputa.v14i1.
[10] M. A. Muin, K. Kapti, and T. Yusnanto, "Campus Website Security Vulnerability Analysis Using Nessus," *Int. J. Comput. Inf. Syst.*, vol. 3, no. 2, pp. 79–82, 2022, doi: 10.29040/ijcis.v3i2.72.
[11] S. Utoro, B. A. Nugroho, M. Meinawati, and S. R. Widianto, "Analisis Keamanan Website E-Learning SMKN 1 Cibatu Menggunakan Metode Penetration Testing Execution Standard," *Multinetics*, vol. 6, no. 2, pp. 169–178, 2020, doi: 10.32722/multinetics.v6i2.3432.