Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.6, December 2025, pp. 3689~3696
e-ISSN: 2548-6861

Comparative Analysis of Penetration Testing Frameworks: OWASP, PTES, and NIST SP 800-115 for Detecting Web Application Vulnerabilities

Muhamad Bunan Imtias [1]*, Khothibul Umam[2]*, Hery Mustofa[3]*, Moh Hadi Subowo[4]*
* Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Walisongo, Semarang
bunanimtias@gmail.com [1], khothibul_umam@walisongo.ac.id [2], herymustofa@walisongo.ac.id [3], hadi.subowo@walisongo.ac.id [4]

Article Info

Article history:

Received 2025-06-16
Revised 2025-11-22
Accepted 2025-11-26

Keyword:

Penetration Testing,
OWASP,
PTES,
NIST SP 800-115,
Web Applcation Vulnerabilities,
Comparative Analysis.

ABSTRACT

Web application security faces increasingly complex challenges as digital architectures evolve, necessitating the selection of appropriate and effective penetration testing methods. This study presents a comparative analysis of the OWASP Testing Guide, PTES, and NIST SP 800-115 frameworks in detecting web application vulnerabilities. Through experiments on DVWA and OWASP Juice Shop, the frameworks were evaluated based on detection speed, vulnerability count, and severity. The results highlight a clear trade-off: OWASP proved the most efficient (85 minutes average, 59 total vulnerabilities), making it ideal for rapid assessments. PTES demonstrated the most comprehensive technical depth (63 vulnerabilities, highest severity) but required the most time, while NIST SP 800-115 (49 vulnerabilities) excelled in compliance and risk management integration. The study recommends selecting OWASP for efficiency, PTES for deep technical audits, and NIST for regulatory alignment.

I. PENDAHULUAN

Keamanan aplikasi web merupakan salah satu aspek yang paling krusial dalam dunia teknologi informasi saat ini. Dengan semakin berkembangnya teknologi digital, ancaman terhadap aplikasi web, seperti peretasan, kebocoran data, dan eksploitasi kerentanannya, semakin kompleks dan beragam. Menurut laporan OWASP Top Ten 2021, aplikasi web terus menjadi sasaran utama bagi serangan dunia maya, dengan kerentanannya yang sering kali dapat dimanfaatkan untuk merusak integritas, kerahasiaan, dan ketersediaan data [1]. Oleh karena itu, penting bagi pengembang dan profesional keamanan untuk melakukan pengujian penetrasi (penetration testing) yang efektif guna mengidentifikasi dan mengatasi kerentanannya sebelum dieksploitasi oleh pihak yang tidak bertanggung jawab [2]. Keberagaman framework pentest mengharuskan pemahaman menyeluruh tentang efektivitas masing-masing metode dalam mendeteksi kerentanan aplikasi web secara tepat dan efisien [3].

Pengujian penetrasi adalah metode yang digunakan untuk mengevaluasi keamanan aplikasi web dengan mensimulasikan serangan dari perspektif seorang peretas [4]. Dalam praktiknya, berbagai kerangka kerja dan standar digunakan untuk melakukan pengujian ini, masing-masing dengan pendekatan dan metodologi yang berbeda. Tiga kerangka kerja yang banyak digunakan dalam pengujian penetrasi adalah OWASP Testing Guide, Penetration Testing Execution Standard (PTES), dan NIST SP 800-115 [5]. Masing-masing memiliki kelebihan dan kekurangan, serta area fokus yang berbeda, yang dapat memengaruhi hasil deteksi kerentanannya.

OWASP Testing Guide menyediakan pendekatan terstruktur untuk mendeteksi kerentanan aplikasi web. Panduan ini fokus pada teknik praktis dan penggunaan alat pengujian yang relevan dengan ancaman terbaru [6][7][8]. PTES menawarkan tahapan yang lebih formal dengan penekanan pada perencanaan, pengumpulan informasi, eksploitasi, hingga pelaporan. Standar ini sering digunakan ketika evaluasi yang lebih komprehensif diperlukan [9][10]. NIST SP 800-115, di sisi lain, adalah standar yang diterbitkan oleh National Institute of Standards and Technology (NIST) untuk pengelolaan risiko keamanan informasi secara holistik. Meskipun mencakup panduan pentest, NIST SP 800-115 lebih berorientasi pada integrasi pengujian keamanan dalam

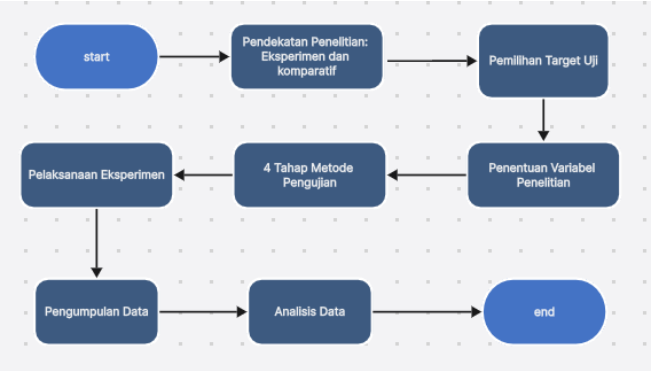
kerangka manajemen risiko dan kepatuhan regulasi, bukan hanya deteksi teknis kerentanan web [11].

Perbedaan pendekatan, cakupan, dan tujuan dari ketiga framework tersebut menjadikan analisis perbandingan menjadi penting. Setiap framework dapat menghasilkan efektivitas yang berbeda dalam konteks pengujian aplikasi web, sehingga pemilihan metode yang tepat perlu mempertimbangkan kebutuhan pengujian yang spesifik [12].

Penelitian ini bertujuan untuk membandingkan OWASP, PTES, dan NIST SP 800-115 dalam mendeteksi kerentanan aplikasi web. Penelitian difokuskan pada analisis perbedaan metodologi serta evaluasi kinerja ketiganya menggunakan parameter objektif seperti jumlah kerentanan yang terdeteksi, waktu deteksi, dan tingkat keparahan berdasarkan CVSS. Penelitian juga menyusun metodologi uji yang terstandarisasi untuk memastikan proses dapat direplikasi. Kontribusi ilmiah penelitian ini terletak pada penyediaan data kuantitatif, analisis efisiensi framework, serta pemetaan kriteria pemilihan framework yang dapat digunakan oleh peneliti dan praktisi keamanan siber.[13].

II. METODE

Metodologi penelitian ini bertujuan untuk melakukan analisis komparatif terhadap tiga framework penetration testing (pentest), yaitu OWASP, PTES, dan NIST SP 800-115, dalam mendeteksi kerentanannya pada aplikasi web [14]. Kerangka penelitian dirancang untuk mengevaluasi efektivitas masing-masing framework dalam mengidentifikasi kerentanan yang ada pada aplikasi web dengan menggunakan simulasi serangan dari perspektif seorang peretas [15]. Prosedur umum penelitian ditampilkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

A. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan eksperimen dan komparatif. Pendekatan eksperimen digunakan untuk menguji ketiga metodologi pentest secara langsung, dengan tujuan untuk mendapatkan data yang jelas dan dapat diukur mengenai efektivitasnya dalam mendeteksi kerentanannya pada aplikasi web [16]. Untuk menjaga konsistensi dan keakuratan hasil, setiap framework diuji pada objek yang sama. Guna memperkuat validitas penelitian dan kemampuan generalisasi, penelitian ini menggunakan dua target uji yang berbeda secara arsitektural:

1. Damn Vulnerable Web Application (DVWA) versi 1.10: Dipilih untuk merepresentasikan arsitektur aplikasi web tradisional yang berbasis PHP/MySQL.
2. OWASP Juice Shop (v19.0.0): Dipilih untuk merepresentasikan aplikasi web modern yang sangat bergantung pada JavaScript (seperti *Single Page Application* / SPA) dan arsitektur berbasis API, yang seringkali memiliki vektor serangan berbeda dari aplikasi tradisional.

DVWA diuji pada level keamanan *medium*, sementara Juice Shop diuji melalui *challenge-based vulnerability verification.* Ruang lingkup pengujian mencakup semua kategori kerentanan yang tersedia di kedua aplikasi target. Setiap kombinasi framework-target dialokasikan durasi pengujian terstruktur untuk mengidentifikasi dan mencoba eksploitasi kerentanan.[17].

Metode pengumpulan data dilakukan melalui penggunaan tools dan teknik pentest yang relevan dengan setiap framework. Data yang dikumpulkan mencakup hasil deteksi kerentanannya, waktu deteksi, serta keparahan dari kerentanan yang ditemukan. Setiap kombinasi framework–target diuji sebanyak tiga kali (3 run) untuk mengurangi bias hasil dan meningkatkan reliabilitas data eksperimen. Dengan menggunakan data eksperimen ini, analisis komparatif akan dilakukan untuk menilai keunggulan dan kelemahan masing-masing framework dalam konteks aplikasi web [18].

B. Variabel Penelitian

Variabel penelitian dalam eksperimen ini terbagi menjadi tiga jenis: variabel bebas, variabel terikat, variabel kontrol.

1) *Variabel Bebas:* Framework pentest yang diuji, yaitu OWASP, PTES, dan NIST SP 800-115.

2) *Variabel Terikat:* Hasil dari eksperimen yang mengukur efektivitas dari ketiga framework pentest, yaitu

- Jumlah kerentanan yang ditemukan: Mengukur seberapa banyak kerentanannya yang berhasil terdeteksi oleh masing-masing framework.
- Tingkat keparahan kerentanan: Menggunakan sistem penilaian CVSS 3.1 untuk menilai keparahan kerentanannya.
- Kecepatan deteksi: Mengukur waktu yang dibutuhkan oleh setiap framework untuk mendeteksi kerentanannya setelah pengujian dimulai.

3) *Variabel kontrol:* Eksperimen dilakukan pada perangkat laptop dengan spesifikasi prosesor AMD Ryzen 5 5500U, RAM 16 GB, dan sistem operasi Kali Linux 2025.3 yang dijalankan pada VirtualBox 7.2.4. Lingkungan pengujian mencakup aplikasi DVWA v1.10 dengan Medium Security dan OWASP Juice Shop v19.0.0 dalam container Docker sebagai target uji.

C. Metode Pengujian

Metode pengujian dilakukan dengan menggunakan tools dan teknik pentest yang ada dalam setiap metodologi pentest yang diuji. Pengujian ini mencakup empat kategori utama dalam pentest, yaitu reconnaissance, vulnerability scanning, exploitation, dan post-exploitation. Setiap kategori memiliki tujuan dan tools yang digunakan untuk mendeteksi dan mengeksploitasi kerentanannya pada aplikasi web [19].

1) *Reconnaissance (Pengumpulan Informasi):* Pada tahap ini, tools yang digunakan Adalah Nmap v7.95 untuk mengumpulkan informasi tentang target (aplikasi web) baik secara pasif maupun aktif. Pengumpulan informasi ini adalah langkah pertama dalam menemukan kerentanannya [20].

2) *Vulnerability Scanning (Pemindaian Kerentanan):* Di sini, tools yang digunakan adalah OWASP ZAP v2.16.1 untuk secara otomatis mendeteksi potensi kerentanannya pada aplikasi web. Pemindaian ini memberikan gambaran awal mengenai kerentanan yang ada, seperti misalnya masalah pada konfigurasi server atau aplikasi [21].

3) *Exploitation (Eksploitasi Kerentanan):* Eksploitasi dilakukan menggunakan fitur manual dan *attack mode* dari OWASP ZAP. Tools diseragamkan agar perbandingan framework tidak terpengaruh faktor alat [20].

4) *Post-Exploitation and Reporting (Pasca-Eksploitasi dan Pelaporan):* Setelah eksploitasi berhasil, tahap ini mengevaluasi lebih lanjut dampak dari eksploitasi untuk menilai potensi kerusakan yang dapat ditimbulkan oleh penyerang [21]. Pada akhir setiap eksperimen, tahap reporting dilakukan untuk mendokumentasikan hasil temuan menggunakan CVSS scoring, serta mencatat log hasil pengujian untuk analisis lebih lanjut [19].

D. Prosedur Eksperimen

Eksperimen dilakukan dengan mengikuti prosedur yang ditetapkan oleh masing-masing framework pentest yang diuji. Masing-masing framework memiliki metodologi dan tahapan yang terstruktur dalam pengujian aplikasi web [22].

1) *OWASP Testing Guide:* memberikan pendekatan berbasis ancaman dan lebih fokus pada pengujian praktis untuk mengidentifikasi kerentanannya melalui berbagai uji, seperti uji authentication, session management, authorization, dan input validation. Pengujian ini bertujuan untuk mengidentifikasi potensi kerentanan yang dapat dimanfaatkan oleh penyerang [23]. Pada framework ini modul yang diuji adalah information gathering, vulnerability scanning, exploitation, dan reporting[24].

2) *PTES:* memberikan pendekatan yang lebih formal dan komprehensif, mencakup tahapan perencanaan yang lebih mendalam. Dimulai dengan information gathering, kemudian diikuti dengan analisis kerentanan, eksploitasi kerentanan, dan diakhiri dengan post-exploitation, dan reporting. PTES memiliki pembagian yang lebih rinci dan menyeluruh, yang memungkinkan evaluasi risiko yang lebih baik [25][26].

3) *NIST SP 800-115:* lebih berfokus pada pengelolaan risiko dan kepatuhan, memberikan pedoman yang jelas untuk mengintegrasikan pentesting ke dalam siklus hidup keamanan informasi. Pengujian yang dilakukan mengikuti prosedur yang mencakup planning, discovery, attack, dan reporting [27][28].

Ketiga framework tersebut diuji pada target berikut.

1) *DVWA:* image vulnerables/web-dvwa:1.10. dengan pengujian pada tingkat keamanan medium. Pengaturan dilakukan melalui file config.inc.php di container DVWA. Jumlah kerentanan DVWA ada 14 yang meliputi Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQLi, Blind SQLi, Weak Session IDs, XSS DOM, XSS Reflected, XSS Stored, CSP Bypass, JavaScript, Open HTTP Redirect [29][30].

2) *Juice Shop:* image bkimminich/juice-shop:19.0.0, konfigurasi default tanpa autentikasi tambahan. Kerentanan yang tersedia terdiri dari 15 kategori utama yaitu, Broken Access Control, Broken Anti Automation, Broken Authentication, Cryptographic Issues, Improper Input Validation, Injection, Insecure Deserialization, Miscellaneous, Security Misconfiguration, Security through Obscurity, Sensitive Data Exposure, Unvalidated Redirects, Vulnerable Components, XSS, dan XXE [31]. Kategori Insecure Deserialization dan XXE dikecualikan dalam pengujian dikarenakan terdapat peringatan membahayakan untuk docker, sehingga website juice shop menghilangkan kategori ini.

Setiap eksperimen dimulai dari snapshot bersih, dan setiap kombinasi framework–target–security level diuji sebanyak 3 kali percobaan.

E. Kriteria Penilaian Metode

Setelah eksperimen dilakukan, hasil dari masing-masing metode pentest akan dinilai berdasarkan kriteria yang relevan untuk mengukur efektivitasnya dalam mendeteksi kerentanannya [4].

1) *Jumlah kerentanan yang ditemukan:* Mengukur seberapa banyak kerentanannya yang berhasil ditemukan oleh masing-masing framework.

2) *Tingkat keparahan kerentanan (CVSS):* Penilaian terhadap seberapa parahnya kerentanan yang ditemukan menggunakan standar CVSS v3.1.

3) *Kecepatan deteksi:* Mengukur waktu yang dibutuhkan oleh setiap framework untuk mendeteksi kerentanan.

4) *Kemudahan implementasi:* Mengukur seberapa mudah metode tersebut dapat diimplementasikan, khususnya untuk pentester yang baru belajar.

F. Teknik Analisis Data

Data yang diperoleh akan dianalisis dengan dua teknik analisis utama:

1) *Deskriptif Kuantitatif:* Data berupa jumlah kerentanan, waktu deteksi, dan tingkat keparahan kerentanan akan disajikan dalam tabel dan grafik untuk memberikan gambaran yang jelas dan mudah dipahami [32].

2) *Analisis Komparatif:* Setelah data disajikan, analisis dilakukan untuk membandingkan ketiga metode pentest berdasarkan kriteria yang telah ditetapkan. Hal ini akan memberikan wawasan yang lebih mendalam tentang kelebihan dan kekurangan masing-masing framework dalam konteks aplikasi web [33].

III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian dan Analisis Kerentanannya

Pengujian dilakukan terhadap dua target aplikasi, yaitu DVWA dan OWASP Juice Shop, dengan menerapkan tiga framework pentest (OWASP Testing Guide, PTES, dan NIST SP 800-115). Hasil lengkap ditampilkan pada Tabel I dan Tabel II.

TABEL I
HASIL PENGUJIAN FRAMEWORK PADA TARGET DVWA

Framework	Vulnerabilities Found	Vulnerabilities Found based on DVWA module	Most Critical Vulnerabilities (CVSS 3.1)	Average Detection Speed	Average Severity Level (CVSS 3.1)
OWASP Testing Guide	26	8 (6 distinct categories)	SQL Injection (CVSS 9.8)	60 minutes	5.87 (Medium)
PTES	24	9 (6 distinct categories)	SQL Injection (CVSS 9.8)	105 minutes	5.76 (Medium)
NIST SP 800-115	20	6 (4 distinct categories)	Path Traversal (CVSS 8.0)	80 minutes	4.83 (Medium)

TABEL II
HASIL PENGUJIAN FRAMEWORK PADA TARGET OWASP JUICE SHOP

Framework	Total Vulnerabilities Found	Challenge Solved	Most Critical Vulnerabilities (CVSS)	Average Detection Speed	Average Severity Level (CVSS)
OWASP Testing Guide	33	16	SQL Injection (CVSS 9.8)	110 menit	5.6 (Medium)
PTES	39	23	SQL Injection (CVSS 9.8)	150 menit	6.41 (High)
NIST SP 800-115	29	9	SQL Injection (CVSS 9.8)	120 menit	6.04 (Medium)

Aplikasi DVWA dan Juice Shop dirancang dengan berbagai kerentanan umum seperti SQL Injection, XSS, CSRF, IDOR, serta kelemahan autentikasi. Hasil penelitian menunjukkan adanya variasi performa di antara ketiga framework dalam hal jumlah kerentanan yang ditemukan, tingkat keparahan, serta kecepatan deteksi.

1) *OWASP Testing Guide:* Pada pengujian DVWA, OWASP Testing Guide berhasil mendeteksi 26 kerentanan (8 valid berdasarkan modul DVWA). Kerentanan paling kritis adalah SQL Injection (CVSS 9.8). Rata-rata waktu deteksi adalah 60 menit, menunjukkan framework ini efektif untuk pengujian dengan waktu terbatas dan memberikan hasil cepat terhadap kerentanan inti seperti SQLi dan XSS. Pada Juice Shop, OWASP mendeteksi 33 kerentanan dan menyelesaikan 16 challenge, dengan kerentanan paling kritis juga berupa SQL Injection (CVSS 9.8). Tingkat keparahan rata-rata berada pada kategori *medium*.

2) *PTES:* Framework PTES menampilkan hasil paling komprehensif. Pada DVWA, PTES menemukan 24 kerentanan dan 9 valid berdasarkan modul DVWA. Pada Juice Shop, PTES menemukan 39 kerentanan, menyelesaikan 23 challenge, dan mencatat tingkat keparahan rata-rata tertinggi yaitu 6.41 (High). PTES membutuhkan waktu deteksi lebih lama (105 menit untuk DVWA, 150 menit untuk Juice Shop) karena mengikuti tahapan eksplorasi yang lebih mendalam (threat modeling, enumerasi manual intensif). Namun demikian, kedalaman analisisnya menghasilkan cakupan deteksi kerentanan paling luas di antara ketiga framework.

3) *NIST SP 800-115:* Framework ini NIST SP 800-115 mendeteksi 20 kerentanan di DVWA dan 29 kerentanan di Juice Shop, dengan tingkat keparahan rata-rata 4.83 (DVWA) dan 6.04 (Juice Shop). Temuan paling kritis adalah Path Traversal (CVSS 8.0) pada DVWA serta SQL Injection (CVSS 9.8) di Juice Shop. Waktu deteksi rata-rata berada di tengah-tengah, yaitu 80 menit (DVWA) dan 120 menit (Juice Shop). Framework ini konsisten dalam proses discovery dan attack, tetapi tidak sekomprehensif PTES pada tahap eksploitasi manual. Beberapa kerentanan tidak terdeteksi oleh NIST.

B. Analisis Berdasarkan Tahapan Pentest

Seluruh framework menunjukkan hasil yang sama pada tahap *Information Gathering* dengan mendeteksi dua service utama, namun PTES membutuhkan waktu lebih lama karena menambahkan identifikasi manual. Pada tahap *Vulnerability Scanning*, PTES menghasilkan temuan paling banyak berkat kombinasi analisis otomatis dan manual, sedangkan NIST

lebih sensitif terhadap alert informational sehingga laporan awalnya tampak lebih detail.

TABEL III
ANALISIS KOMPARATIF BERDASARKAN TAHAPAN PENTEST

Tahapan	OWASP	PTES	NIST SP 800-115
Information Gathering	Menggunakan Nmap, waktu rata-rata 30 detik, 2 service teridentifikasi, baik pada pengujian DVWA maupun OWASP Juice shop	Manual dan Nmap, waktu 2 menit, 2 service teridentifikasi, baik pada pengujian DVWA maupun OWASP Juice shop	Menggunakan Nmap, waktu 1 menit, 2 service teridentifikasi, baik pada pengujian DVWA maupun OWASP Juice shop
Vulnerability Scanning	Sedikit manual testing di awal kemudian menggunakan ZAP dan mendeteksi 33 alert (6 high, 10 medium, 8 low, dan 9 informational) pada pengujian DVWA dan 20 alert (2 high, 6 medium, 6 low, dan 6 informational) pada pengujian Juice shop	Melakukan manual testing secara lengkap pada awal pengujian kemudian menggunakan ZAP dan mendeteksi 34 alert (7 high, 9 medium, 9 low, dan 9 informational) pada pengujian DVWA dan 20 alert (2 high, 6 medium, 5 low, dan 7 informational) pada pengujian Juice shop	Menggunakan ZAP pada setiap endpoint dan mendeteksi 27 alert (2 high, 9 medium, 9 low, dan 7 informational) pada pengujian DVWA dan 21 alert (3 high, 7 medium, 6 low, dan 5 informational) pada pengujian Juice shop)
Exploitation	- Melakukan exploitasi pada setiap modul DVWA dengan mengacu pada hasil scan ZAP (24 valid) - Melakukan percobaan challenge pada juice shop dengan mengacu pada hasil scan ZAP (16 solved)	- Berfokus melakukan exploitasi pada high confidence dan high vulnerable hasil scan ZAP pada DVWA (25 valid) - Melakukan percobaan challenge dengan kesulitan tinggi pada juice shop kemudian dilanjutkan active scan menggunakan ZAP (23 solved)	- Melakukan active scan pada setiap modul DVWA dan endpoint terkait (20 valid) - Melakukan percobaan challenge pada juice shop dengan melakukan active scan menggunakan ZAP di setiap endpoint (9 solved)
Post-Exploitation & Reporting	Format laporan ringkas, fokus ke kerentanan web terdeteksi baik manual maupun otomatis (ZAP). Waktu yang dibutuhkan cukup singkat.	Laporan temuan ZAP ditambah laporan manual dengan ringkasan resiko, narasi serangan, dan kesimpulan. Waktu yang dibutuhkan sedikit lama dibanding owasp.	Memvalidasi ulang setiap temuan kerentanan. Laporan terstruktur dan formal. Waktu yang dibutuhkan cukup lama mengingat struktur laporann yang formal.

Pada tahap *Exploitation*, PTES menyelesaikan eksploitasi terbanyak—lebih dari 24 exploit valid pada DVWA dan 23 challenge pada Juice Shop—karena pendekatannya yang berbasis ancaman dan eksploitasi manual yang agresif. Sebaliknya, NIST menyelesaikan paling sedikit karena fokusnya lebih pada validasi risiko daripada eksploitasi mendalam. Pada tahap *Post-Exploitation & Reporting*, NIST memiliki format pelaporan paling formal dan memerlukan waktu paling lama, OWASP paling ringkas, sementara PTES berada di tengah dengan memberikan konteks risiko yang lebih komprehensif.

C. Jenis Kerentanan

Berdasarkan hasil pemindaian menggunakan tiga framework, yaitu OWASP, PTES, dan NIST SP 800-115, ditemukan variasi jumlah dan jenis kerentanan pada DVWA dan OWASP Juice Shop sebagaimana ditampilkan pada Tabel IV. Secara umum, ketiga framework tersebut mengidentifikasi pola temuan yang serupa, namun dengan tingkat kedalaman dan fokus yang berbeda.

TABEL IV
JENIS KERENTANAN YANG TERDETEKSI

Jenis Kerentanan	OWASP	PTES	NIST
SQL Injection	3 di DVWA, 2 di Juice Shop	4 di DVWA, 4 di Juice Shop	4 di Juice Shop
Broken Access Control	3 di DVWA, 5 di Juice Shop	2 di DVWA, 2 di Juice Shop	2 di DVWA, 5 di Juice Shop
CSRF	1 di DVWA	1 di DVWA	1 di DVWA
Cryptographic Issues	1 di Juice Shop	3 di Juice Shop	-
XSS	2 di DVWA , 2 di Juice Shop	1 di DVWA, 1 di Juice Shop	1 di Juice Shop

Sensitive Data Exposure	4 di DVWA, 5 di Juice Shop	6 di DVWA, 6 di Juice Shop	7 di DVWA, 2 di Juice Shop
Broken Authentication	2 di DVWA, 1 di Juice Shop	2 di DVWA, 4 di Juice Shop	2 di DVWA, 3 di Juice Shop
Broken Anti Automation	1 di Juice Shop	2 di Juice Shop	-
Security Misconfiguration	3 di DVWA, 6 di Juice Shop	2 di DVWA, 6 di Juice Shop	2 di DVWA, 11 di Juice Shop
Unvalidated Redirects	2 di Juice Shop	1 di Juice Shop	1 di Juice Shop
Miscellaneous	2 di Juice Shop	1 di Juice Shop	1 di Juice Shop
Cross-Domain Issues	-	1 di DVWA, 1 di Juice Shop	1 di DVWA
Improper Input Validation	1 di Juice Shop	1 di Juice Shop	-
Security Through Obscurity	-	2 di Juice Shop	-
Observability Failures	5 di Juice Shop	5 di Juice Shop	2 di Juice Shop
Command Injection	1 di DVWA	1 di DVWA	-
XSLT Injection	1 di DVWA	1 di DVWA	1 di DVWA
CSP Bypass	4 di DVWA	4 di DVWA	4 di DVWA

OWASP mendeteksi kerentanan dalam jumlah signifikan pada kategori *Security Misconfiguration*, *Sensitive Data Exposure*, serta *Broken Access Control* pada kedua target aplikasi. PTES menunjukkan cakupan kerentanan yang lebih luas pada kategori seperti *SQL Injection*, *Cross-Site Scripting* (XSS), dan *Broken Anti-Automation*, karena pendekatannya yang lebih berorientasi pada eksploitasi aktif dan perilaku sistem di dunia nyata. Sementara itu, NIST menampilkan jumlah temuan yang relatif stabil dan konsisten, terutama pada kategori seperti *Broken Authentication*, *Security Misconfiguration*, dan *Unvalidated Redirects*, walaupun tidak memberikan kedalaman eksploitasi setinggi PTES karena fokus utamanya adalah kepatuhan prosedural (*compliance-driven*).

D. Kemudahan Implementasi

TABEL V
PERBANDINGAN KEMUDAHAN IMPLEMENTASI

Framework	Kemudahan Implementasi	Keterangan
OWASP Testing Guide	Mudah	Banyak dokumentasi dan tutorial yang tersedia.
PTES	Menengah	Memerlukan pemahaman lebih dalam mengenai tahapan pentest.
NIST SP 800-115	Menengah	Fokus pada manajemen risiko dan kepatuhan, lebih kompleks.

Kemudahan implementasi adalah kriteria penting dalam memilih framework pentest, terutama bagi pentester yang baru memulai atau bagi tim kecil dengan keterbatasan sumber daya. OWASP Testing Guide sangat mudah diimplementasikan karena memiliki dokumentasi yang lengkap dan banyak tutorial yang tersedia. Framework ini memungkinkan pentester untuk dengan cepat memahami metodologi yang digunakan, serta tools yang diterapkan untuk mendeteksi kerentanannya [7]. Dalam eksperimen ini, DVWA dan OWASP Juice Shop menjadi tempat yang baik untuk menguji kerentanannya dengan menggunakan framework ini, karena DVWA sudah dirancang dengan berbagai kerentanannya yang mudah dikenali, sedangkan Juice Shop dirancang dengan tantangan yang masing-masing mencerminkan kerentanan dalam bebrapa struktur web modern. Oleh karena itu, OWASP Testing Guide sangat cocok bagi pentester pemula yang baru memulai dan ingin memahami teknik dasar dalam penetration testing.

Sebaliknya, PTES membutuhkan pemahaman yang lebih mendalam tentang proses pentesting dan lebih kompleks. Framework ini mencakup tahap-tahap seperti information gathering, threat modeling, vulnerability analysis, dan exploitation yang memerlukan pemahaman lebih lanjut tentang risiko dan langkah-langkah pengujian yang terstruktur [9]. Meskipun PTES dapat mendeteksi lebih banyak kerentanannya (termasuk kerentanannya yang lebih tersembunyi), penggunaan framework ini memerlukan lebih banyak waktu dan pengalaman. Oleh karena itu, PTES lebih cocok untuk pentester berpengalaman yang ingin melakukan pengujian penetrasi yang lebih komprehensif dan mendalam, seperti yang ditemukan dalam aplikasi DVWA yang menyimulasikan kerentanannya yang lebih kompleks.

NIST SP 800-115 juga lebih kompleks dibandingkan dengan OWASP, terutama karena berfokus pada pengelolaan risiko dan kepatuhan terhadap standar. Meskipun framework ini memberikan panduan yang lebih terstruktur dan formal, ia lebih cocok untuk organisasi yang memprioritaskan pengujian penetrasi yang terintegrasi dengan kebijakan keamanan [11]. Oleh karena itu, meskipun NIST SP 800-115

berguna bagi organisasi yang perlu memastikan kepatuhan terhadap kebijakan keamanan, framework ini mungkin kurang cocok untuk pentester pemula yang membutuhkan pengujian yang cepat dan lebih langsung pada aplikasi yang lebih dinamis, seperti DVWA yang dirancang untuk eksplorasi kerentanannya.

E. Rekomendasi Framework Pentest

TABEL VI
PERBANDINGAN ASPEK DOKUMENTASI

Aspek Dokumentasi	OWASP	PTES	NIST
Format laporan baku	Tidak ada (manual)	Ada (template PTES)	Ada (SP 800-115 format)
Severity classification	Ada (CVSS otomatis ZAP)	Ada (manual, lengkap)	Ada (berdasarkan standard NIST)
Rekomendasi mitigasi	Ada, tetapi umum	Ada, berbasis hasil eksploitasi	Sangat lengkap dan prosedural

Berdasarkan hasil eksperimen, ketiga framework pentest menunjukkan trade-off antara kecepatan, kedalaman, dan konteks penggunaan. Seperti yang ditunjukkan pada tabel IV, pemilihan framework pentest harus disesuaikan dengan tujuan pengujian dan tingkat keahlian pentester. Berikut adalah rekomendasi penggunaan framework berdasarkan temuan eksperimen:

1) *OWASP Testing Guide:* Cocok untuk pentester pemula yang membutuhkan pengujian cepat dan efisien dengan waktu deteksi yang cepat, seperti yang dilakukan pada aplikasi DVWA. Meskipun jumlah kerentanannya yang ditemukan lebih sedikit dibandingkan dengan PTES, kecepatan dan kemudahan implementasi menjadikannya pilihan utama bagi mereka yang membutuhkan gambaran umum kerentanannya pada aplikasi web. Dengan dokumentasi yang sangat lengkap dan tutorial yang banyak tersedia, OWASP Testing Guide sangat ideal untuk pemula yang ingin memulai dengan cepat.

2) *PTES:* Cocok untuk pentester berpengalaman yang membutuhkan pengujian penetrasi yang lebih komprehensif dan mendalam. PTES berhasil mendeteksi lebih banyak kerentanannya dan menawarkan panduan yang lebih terstruktur untuk mengeksplorasi potensi kerentanannya, termasuk kerentanannya yang lebih tersembunyi seperti Privilege Escalation. Namun, penggunaan PTES memerlukan lebih banyak waktu, dengan waktu deteksi yang lebih lama (rata-rata 25 menit) dan pemahaman yang lebih mendalam mengenai pentesting. Oleh karena itu, PTES lebih cocok untuk pengujian penetrasi yang lebih menyeluruh dan mendalam.

3) *NIST SP 800-115:* Cocok untuk organisasi yang berfokus pada kepatuhan dan pengelolaan risiko. Meskipun sedikit lebih lambat dalam mendeteksi kerentanannya (waktu deteksi rata-rata 20 menit), NIST SP 800-115 menawarkan pendekatan yang lebih terintegrasi dengan kebijakan keamanan dan manajemen risiko. Oleh karena itu, NIST SP 800-115 sangat berguna untuk organisasi yang perlu memastikan kepatuhan terhadap standar keamanan dan regulasi. Framework ini lebih formal dan terstruktur, sehingga cocok untuk penggunaan dalam lingkungan perusahaan atau organisasi yang lebih besar.

IV. KESIMPULAN

Pemilihan framework pentest yang tepat harus disesuaikan dengan tujuan pengujian dan tingkat pengalaman pengguna. Berdasarkan analisis komparatif kuantitatif menggunakan DVWA dan OWASP Juice Shop, penelitian ini menyimpulkan adanya trade-off yang jelas antara kecepatan deteksi, kedalaman analisis, dan kepatuhan standar.

Secara spesifik, penelitian ini mengkuantifikasi perbedaan kinerja. OWASP Testing Guide terbukti sebagai framework yang paling efisien dari segi waktu dengan rata-rata kecepatan deteksi 85 menit (60 menit pada DVWA dan 110 menit pada Juice Shop). Framework ini mendeteksi total 59 kerentanan gabungan, menjadikannya pilihan superior untuk Rapid Assessment dan pentester pemula karena kemudahan implementasi serta dokumentasi yang luas.

PTES (Penetration Testing Execution Standard) terbukti paling komprehensif. Meskipun membutuhkan waktu paling lama (rata-rata 127,5 menit), PTES berhasil mengidentifikasi jumlah kerentanan terbanyak, yaitu total 63 kerentanan (24 pada DVWA dan 39 pada Juice Shop) dengan tingkat keparahan rata-rata tertinggi (High). PTES direkomendasikan untuk Deep Assessment dan tim ahli yang memprioritaskan kedalaman temuan dibanding kecepatan.

NIST SP 800-115 menempati posisi tengah dalam hal kecepatan (rata-rata 100 menit) dan jumlah temuan (total 49 kerentanan). Keunggulan utamanya bukan pada jumlah temuan teknis, melainkan pada struktur pelaporan yang formal. Framework ini paling sesuai bagi organisasi dan lingkungan korporat yang memprioritaskan integrasi manajemen risiko dan kepatuhan regulasi (compliance) di atas sekadar jumlah temuan mentah.

Pada akhirnya, tidak ada framework yang absolut terbaik. OWASP unggul dalam kecepatan, PTES unggul dalam kedalaman teknis, dan NIST SP 800-115 unggul dalam kesesuaian prosedural.

DAFTAR PUSTAKA

[1] O. Bin Tauqeer, S. Jan, A. Omar Khadidos, A. Omar Khadidos, F. Qudus Khan, and S. Khattak, "Analysis of Security Testing Techniques," *Intell. Autom. Soft Comput.*, vol. 29, no. 1, pp. 291–306, 2021, doi: 10.32604/iasc.2021.017260.

[2] M. C. Ghanem and T. M. Chen, "Reinforcement Learning for Efficient Network Penetration Testing," *Information*, vol. 11, no. 1, p. 6, Dec. 2019, doi: 10.3390/info11010006.

[3] R. A. Correa, J. Ram髇 Bermejo Higuera, J. Bermejo Higuera, J. Antonio SiciliaMontalvo, M. S醬chez Rubio, and Alberto Magre 襻n, "Hybrid Security AssessmentMethodology forWeb Applications," *Comput. Model. Eng. Sci.*, vol. 126, no. 1, pp. 89–124, 2021, doi: 10.32604/cmes.2021.010700.

[4] A. G. Bacudio, X. Yuan, B. T. Bill Chu, and M. Jones, "An Overview of Penetration Testing," *Int. J. Netw. Secur. Its Appl.*, vol. 3, no. 6, pp. 19–38, Nov. 2011, doi: 10.5121/ijnsa.2011.3602.

[5] H.-J. Lu and Y. Yu, "Research on WiFi Penetration Testing with Kali Linux," *Complexity*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/5570001.

[6] S. Jain, R. Johari, and A. Kaur, "PJCT: Penetration testing based JAVA code testing tool," in *International Conference on Computing, Communication & Automation*, IEEE, May 2015, pp. 800–805. doi: 10.1109/CCAA.2015.7148483.

[7] M. Peroli, F. De Meo, L. Viganò, and D. Guardini, "MobSTer: A model-based security testing framework for web applications," *Softw. Testing, Verif. Reliab.*, vol. 28, no. 8, Dec. 2018, doi: 10.1002/stvr.1685.

[8] H. Nina, J. A. Pow-Sang, and M. Villavicencio, "Systematic Mapping of the Literature on Secure Software Development," *IEEE Access*, vol. 9, pp. 36852–36867, 2021, doi: 10.1109/ACCESS.2021.3062388.

[9] R. Baloch, *Ethical Hacking and Penetration Testing Guide*. Auerbach Publications, 2017. doi: 10.4324/9781315145891.

[10] B. Arkin, S. Stender, and G. McGraw, "Software penetration testing," *IEEE Secur. Priv. Mag.*, vol. 3, no. 1, pp. 84–87, Jan. 2005, doi: 10.1109/MSP.2005.23.

[11] S. Nagpure and S. Kurkure, "Vulnerability Assessment and Penetration Testing of Web Application," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, IEEE, Aug. 2017, pp. 1–6. doi: 10.1109/ICCUBEA.2017.8463920.

[12] A. Alanda, D. Satria, M. I. Ardhana, A. A. Dahlan, and H. A. Mooduto, "Web Application Penetration Testing Using SQL Injection Attack," *JOIV Int. J. Informatics Vis.*, vol. 5, no. 3, p. 320, Sep. 2021, doi: 10.30630/joiv.5.3.470.

[13] M. Felderer, P. Zech, R. Breu, M. Büchler, and A. Pretschner, "Model-based security testing: a taxonomy and systematic classification," *Softw. Testing, Verif. Reliab.*, vol. 26, no. 2, pp. 119–148, Mar. 2016, doi: 10.1002/stvr.1580.

[14] M. Cova, V. Felmetsger, and G. Vigna, "Vulnerability Analysis of Web-based Applications," in *Test and Analysis of Web Services*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 363–394. doi: 10.1007/978-3-540-72912-9_13.

[15] E. Alata, M. Kaaniche, V. Nicomette, and R. Akrout, "An Automated Approach to Generate Web Applications Attack Scenarios," in *2013 Sixth Latin-American Symposium on Dependable Computing*, IEEE, Apr. 2013, pp. 78–85. doi: 10.1109/LADC.2013.22.

[16] R. Amankwah, J. Chen, P. K. Kudjo, and D. Towey, "An empirical comparison of commercial and open-source web vulnerability scanners," *Softw. Pract. Exp.*, vol. 50, no. 9, pp. 1842–1857, Sep. 2020, doi: 10.1002/spe.2870.

[17] K. Abdulghaffar, N. Elmrabit, and M. Yousefi, "Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners," *Computers*, vol. 12, no. 11, p. 235, Nov. 2023, doi: 10.3390/computers12110235.

[18] R. Akrout, E. Alata, M. Kaaniche, and V. Nicomette, "An automated black box approach for web vulnerability identification and attack scenario generation," *J. Brazilian Comput. Soc.*, vol. 20, no. 1, p. 4, Dec. 2014, doi: 10.1186/1678-4804-20-4.

[19] A. V. -, V. -, R. -, and S. S. -, "Finding Vulnerability in Web Application by using Pentesting," *Int. J. Multidiscip. Res.*, vol. 6, no. 4, Jul. 2024, doi: 10.36948/ijfmr.2024.v06i04.24517.

[20] K. Božić, N. Penevski, and S. Adamović, "Penetration Testing and Vulnerability Assessment: Introduction, Phases, Tools and Methods," in *Proceedings of the International Scientific Conference - Sinteza 2019*, Novi Sad, Serbia: Singidunum University, 2019, pp. 229–234. doi: 10.15308/Sinteza-2019-229-234.

[21] D. K. D. Kongara and S. Krishnama, "A Process of Penetration Testing Using Various Tools," *Mesopotamian J. CyberSecurity*, vol. 2023, pp. 93–103, Apr. 2023, doi: 10.58496/MJCS/2023/014.

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

[23] I. D. G. G. Dharmawangsa, G. M. A. Sasmita, and I. P. A. E. Pratama, "Penetration Testing Berbasis OWASP Testing Guide Versi 4.2 (Studi Kasus: X Website)," *JITTER J. Ilm. Teknol. dan Komput.*, vol. 4, no. 1, p. 1613, Feb. 2023, doi: 10.24843/JTRTI.2023.v04.i01.p06.

[24] E. Saad and R. Mitchell, *OWASP Web Security Testing Guide*.

[25] N. Anand, M. A. Saifulla, R. B. Ponnuru, G. R. Alavalapati, R. Patan, and A. H. Gandomi, "Securing Software Defined Networks: A Comprehensive Analysis of Approaches, Applications, and Future Strategies against DoS Attacks," *IEEE Access*, vol. 13, no. April, pp. 64473–64515, 2024, doi: 10.1109/ACCESS.2024.3520478.

[26] "PTES (Penetration Testing Execution Standard)." Accessed: Sep. 03, 2025. [Online]. Available: http://www.pentest-standard.org/index.php/Main_Page

[27] G. Erdogan, Y. Li, R. K. Runde, F. Seehusen, and K. Stølen, "Approaches for the combined use of risk analysis and testing: a systematic literature review," *Int. J. Softw. Tools Technol. Transf.*, vol. 16, no. 5, pp. 627–642, Oct. 2014, doi: 10.1007/s10009-014-0330-5.

[28] K. A. Scarfone, M. P. Souppaya, A. Cody, and A. D. Orebaugh, "Technical guide to information security testing and assessment.," Gaithersburg, MD, 2008. doi: 10.6028/NIST.SP.800-115.

[29] S. Team, "The Best DVWA (Damn Vulnerable Web Application) 2025 Guide," stationx.net. Accessed: Nov. 01, 2025. [Online]. Available: https://www.stationx.net/dvwa-damn-vulnerable-web-application/

[30] R. Wood, "Damn Vulnerable Web Application (DVWA)," github. Accessed: Oct. 30, 2025. [Online]. Available: https://github.com/digininja/DVWA

[31] B. Kimminich and J. Hollenbach, "OWASP Juice Shop," OWASP. Accessed: Oct. 01, 2025. [Online]. Available: https://owasp.org/www-project-juice-shop/

[32] A. J. Vickers *et al.*, "Guidelines for Reporting of Figures and Tables for Clinical Research in Urology," *Eur. Urol.*, vol. 78, no. 1, pp. 97–109, Jul. 2020, doi: 10.1016/j.eururo.2020.04.048.

[33] Q. Guo *et al.*, "A Survey on Knowledge Graph-Based Recommender Systems," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022, doi: 10.1109/TKDE.2020.3028705.