Jurnal Elektronik Ilmu Komputer Udayana
Volume 13, No 3. February 2025

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Analisis Kerentanan Keamanan Menggunakan OWASP ZAP dan Pengujian Manual pada Tampilan Antarmuka Laman PDDIKTI

Firda Ayu Hassanah[a1], Eddy Ryansyah[a2], Fikri Maulana Setiawan[a3], Ridho Alamsyah[a4], Agung Susilo Yuda Irawan[a5]

[a]Informatika, Universitas Singaperbangsa Karawang
Jl. HS. Ronggowaluyo, Telukjambe Timur, Karawang, Jawa Barat, Indonesia
[1]2110631170017@student.unsika.ac.id (Corresponding author)
[2]eddyryansyah1612@gmail.com
[3]maulamafikri@gmail.com
[4]ridhoalamsyah3@gmail.com
[5]agung@unsika.ac.id

Abstract

Penelitian ini menganalisis kerentanan keamanan pada platform PDDIKTI menggunakan pendekatan kombinasi OWASP ZAP dan pengujian manual dalam kerangka Software Testing Life Cycle (STLC). PDDIKTI yang dikelola oleh Pusdatin Kemristekdikti menyimpan data akademik penting. Metodologi yang diterapkan mencakup enam tahap STLC yaitu analisis kebutuhan, perencanaan sprint, desain pengujian, penyiapan lingkungan, pelaksanaan pengujian, dan penutupan pengujian. Pengujian dilakukan dengan OWASP ZAP untuk mendeteksi kerentanan umum seperti injeksi SQL dan XSS, serta pengujian manual untuk menilai keamanan antarmuka pengguna. Hasil penelitian menunjukkan dari 1.843 skenario pengujian manual terdapat 6 kegagalan pada fitur pengumuman, statistik, dan perbandingan program studi. Sementara itu, pengujian otomatis OWASP ZAP menemukan 5 celah keamanan utama, termasuk konfigurasi CORS yang tidak tepat dan header HTTP yang tidak aman. Hasil ini menunjukkan perlunya perbaikan keamanan guna meningkatkan perlindungan data akademik.

Keywords: *Kerentanan Keamanan, PDDIKTI, OWASP ZAP, Pengujian Manual, Keamanan Data*

1. Pendahuluan

Kemajuan teknologi yang pesat telah meningkatkan ketergantungan pada aplikasi web dalam berbagai sektor termasuk pendidikan tinggi. Namun seiring dengan perkembangan ini, ancaman keamanan siber juga semakin meningkat terutama bagi platform yang mengelola informasi sensitif seperti Pangkalan Data Pendidikan Tinggi (PDDIKTI). PDDIKTI yang dikelola oleh Pusat Data dan Informasi (Pusdatin) di bawah Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbud Ristek) menyimpan data penting tentang mahasiswa, dosen, dan perguruan tinggi di Indonesia. Keamanan sistem ini sangat krusial mengingat meningkatnya jumlah serangan siber terhadap sektor pendidikan. Sebagai contoh dalam beberapa tahun terakhir, terdapat insiden kebocoran data pada institusi akademik yang mengancam privasi serta integritas informasi [1].

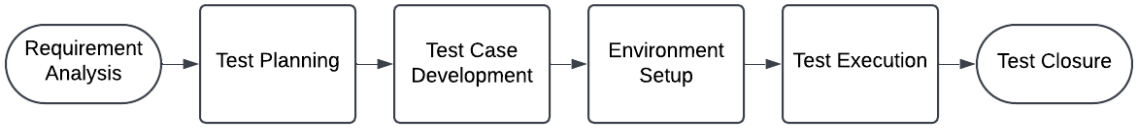
Keamanan perangkat lunak menjadi aspek penting dalam pengelolaan sistem akademik untuk mencegah ancaman seperti injeksi SQL, *cross-site scripting* (XSS), dan konfigurasi keamanan yang tidak memadai [2]. Oleh karena itu, pengendalian kualitas perangkat lunak melalui *Quality Assurance* (QA) menjadi faktor utama dalam menjamin keamanan dan keandalan sistem. QA memungkinkan peningkatan produktivitas, perbaikan proses, serta mitigasi *bug* dan kerentanan dalam perangkat lunak [3]. Penelitian ini bertujuan untuk mengidentifikasi kerentanan pada laman PDDIKTI dengan menggunakan kombinasi OWASP ZAP dan pengujian manual. OWASP ZAP (*Open Web Application Security Project Zed Attack Proxy*) merupakan salah satu alat pengujian keamanan web yang dapat mendeteksi berbagai kerentanan umum dalam aplikasi berbasis web [4]. Penggunaan OWASP ZAP dikombinasikan dengan pengujian manual yang memungkinkan deteksi kelemahan yang mungkin terlewat oleh alat otomatis, sehingga memberikan hasil yang lebih akurat dan komprehensif.

Pengujian manual membutuhkan keterampilan khusus seperti kesabaran, kreativitas, dan inovasi, serta menjadi metode yang lebih efektif dalam mendeteksi skenario eksploitasi yang kompleks [5].

Dengan pendekatan ini, penelitian ini diharapkan dapat memberikan wawasan mendalam mengenai kondisi keamanan PDDIKTI serta rekomendasi perbaikan bagi pengelola sistem. Selain itu, hasil penelitian ini juga dapat menjadi referensi bagi institusi lain dalam meningkatkan keamanan platform akademik terkait, sehingga dapat meminimalisir risiko serangan siber yang dapat mengancam data akademik dan privasi pengguna.

2. Metode Penelitian

Penelitian ini dirancang dengan menggunakan tahapan-tahapan dalam *Software Testing Life Cycle* (STLC) yaitu proses terstruktur yang digunakan dalam pengujian perangkat lunak. STLC sering dianggap sebagai bagian dari siklus hidup pengembangan perangkat lunak di mana setiap tahap memiliki kriteria dan hasil yang telah ditetapkan guna memastikan pengujian dilakukan secara sistematis dan terstruktur [6]. Berikut adalah tahapan yang diterapkan dalam penelitian ini:



Gambar 1. Alur Penelitian

2.1. *Requirement Analysis*

Tahap ini bertujuan untuk menganalisis persyaratan guna mengidentifikasi elemen-elemen yang dapat diuji dari perspektif pengujian. Pihak yang terlibat dalam proses ini meliputi klien, analis bisnis, pemimpin teknis, dan arsitek sistem. Persyaratan yang dikumpulkan terbagi menjadi dua kategori utama, yaitu persyaratan fungsional yang mencakup apa saja yang harus dilakukan oleh perangkat lunak dan persyaratan non-fungsional yang berkaitan dengan aspek kinerja, keamanan, dan keandalan sistem [7].

2.2. *Test Planning*

Pada tahap ini, dilakukan perencanaan pengujian secara mendetail berdasarkan hasil analisis sebelumnya. Penguji mengumpulkan seluruh persyaratan yang diperlukan, melakukan analisis dokumen ringkasan, serta mengidentifikasi kebutuhan sistem dan estimasi biaya jika diperlukan. Hasil dari tahap ini berupa rencana pengujian yang mencakup penyusunan kasus uji serta estimasi waktu yang dibutuhkan untuk pelaksanaan pengujian [7].

2.3. *Test Case Development*

Setelah perencanaan pengujian selesai, tahap berikutnya adalah pengembangan kasus uji. Dalam tahap ini, dilakukan pembuatan, verifikasi, dan pengorganisasian butir uji serta skrip pengujian. Proses ini dimulai dengan identifikasi data uji yang kemudian ditinjau dan disesuaikan dengan persyaratan yang telah ditetapkan [7].

2.4. *Environment Setup*

Tahap ini bertujuan untuk memastikan bahwa lingkungan pengujian, baik perangkat keras maupun perangkat lunak telah dikonfigurasi sesuai dengan rencana yang telah ditetapkan. Proses ini dapat dilakukan bersamaan dengan pengembangan kasus uji karena bersifat independen dari tahap lainnya. Jika lingkungan pengujian telah disiapkan oleh tim pengembang, maka penguji hanya bertugas melakukan *smoke test* yaitu pengujian awal yang memastikan perangkat lunak dalam kondisi stabil sebelum melanjutkan ke pengujian lebih mendalam [7].

2.5. *Test Execution*

Tahap terakhir adalah pelaksanaan pengujian berdasarkan rencana dan kasus uji yang telah disusun sebelumnya. Dalam tahap ini, penguji akan menilai setiap kasus uji dengan status lulus jika berjalan sesuai ekspektasi atau gagal jika terjadi ketidaksesuaian. Jika ditemukan *bug*, maka akan dilaporkan kepada tim pengembang untuk diperbaiki. Setelah perbaikan dilakukan, pengujian ulang akan dilaksanakan guna memastikan bahwa *bug* telah diperbaiki dan sistem berjalan dengan baik [7].

2.6. *Test Closure*

Tahap penutupan dari STLC dikenal sebagai siklus pengujian akhir. Pada tahap ini, dilakukan pelaporan mengenai penyelesaian pengujian, pengumpulan matriks hasil pengujian, serta hasil dari tes yang telah dilaksanakan. Para penguji akan melakukan diskusi dalam tim, melakukan evaluasi, dan menganalisis proses pengujian yang telah dilakukan [7].

3. Hasil dan Pembahasan

Tahap selanjutnya, peneliti akan melaksanakan pengujian kerentanan pada *website* dengan menggunakan OWASP ZAP dan pengujian manual. Pada tahap ini, peneliti akan melakukan pengujian kerentanan secara menyeluruh pada *website* [https://pddikti.kemdikbud.go.id/]. Dalam pengujian manual dan pengujian otomatis terdapat 34 fitur yang diuji, yaitu sebagai berikut:

1. *Landing Page*
2. Profil Lembaga
3. Standar Pelayanan
4. Kebijakan Privasi
5. Hasil Pencarian Semua
6. Hasil Pencarian Perguruan Tinggi
7. Detail Perguruan Tinggi
8. Hasil Pencarian Program Studi
9. Detail Program Studi
10. Hasil Pencarian Dosen
11. Detail Dosen
12. Hasil Pencarian Mahasiswa
13. Detail Mahasiswa
14. Program Studi
15. Bidang Agama
16. Bidang Ekonomi
17. Bidang Humaniora
18. Bidang Kesehatan
19. Bidang MIPA
20. Bidang Pendidikan
21. Bidang Pertanian
22. Bidang Seni
23. Bidang Sosial
24. Bidang Teknik
25. Penyedia Program Studi
26. Perguruan Tinggi
27. Pengumuman
28. Statistik
29. Kategori Perbandingan
30. Pemilihan Perguruan Tinggi
31. Perbandingan Perguruan Tinggi
32. Pemilihan Program Studi
33. Perbandingan Program Studi
34. Kontributor

3.1. *Requirement Analysis*

Pada tahap ini, dilakukan identifikasi fitur yang akan diuji serta potensi kerentanan yang dapat ditemukan. Sebanyak 34 fitur utama laman PDDIKTI dipilih untuk diuji dengan mempertimbangkan aspek keamanan, integritas data, dan kemungkinan eksploitasi. Beberapa parameter keamanan yang diuji meliputi:

a. Validasi input pada formulir *login* dan pencarian data.
b. Pengelolaan sesi pengguna dan autentikasi.
c. Konfigurasi keamanan *header* HTTP.
d. Potensi serangan seperti *SQL Injection* dan *Cross-Site Scripting* (XSS).

3.2. *Test Planning*

Dalam tahap ini, strategi pengujian disusun termasuk pemilihan alat uji dan teknik yang digunakan. Beberapa alat dan metode yang digunakan dalam penelitian ini adalah:

a. OWASP ZAP untuk pengujian otomatis dan pendeteksian celah keamanan.
b. Pengujian manual untuk mengevaluasi keamanan antarmuka pengguna secara langsung.
c. Penyusunan skenario pengujian berdasarkan standar keamanan *web*.

3.3. *Test Case Development*

Pada tahap ini, 1.843 skenario pengujian disusun berdasarkan fitur yang telah diidentifikasi. Setiap skenario mencakup:

a. Pengujian fungsional untuk memastikan fitur bekerja sebagaimana mestinya.
b. Pengujian keamanan untuk mengidentifikasi potensi ancaman dan eksploitasi.
c. Pengujian regresi untuk memastikan bahwa perubahan tidak menyebabkan kerentanan baru.

3.4. *Environment Setup*

Konfigurasi lingkungan pengujian dilakukan untuk memastikan bahwa sistem siap diuji dalam kondisi yang menyerupai lingkungan produksi. Pengaturan mencakup:

a. Pembuatan server uji coba dengan salinan laman PDDIKTI.
b. Instalasi dan konfigurasi OWASP ZAP untuk simulasi serangan.
c. Persiapan skenario pengujian manual dengan berbagai skenario eksploitasi.

3.5. *Test Execution*

Pada tahap ini, pengujian dilakukan menggunakan metode otomatis dan manual. Hasil pengujian adalah sebagai berikut:

a. Pengujian Manual

Pengujian manual dilakukan untuk mengevaluasi fungsionalitas dan keamanan laman PDDIKTI dengan menguji 1.843 skenario yang mencakup berbagai fitur utama, seperti pencarian data akademik, pengelolaan informasi perguruan tinggi, dan kategori bidang studi. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur berjalan sesuai dengan standar yang diharapkan serta mengidentifikasi potensi celah keamanan yang dapat berdampak pada keandalan sistem.

Tabel 1. Hasil Pengujian Manual

No.	*Test Case Feature*	*Total Test Case*	Jumlah *Passed*	Jumlah *Failed*	*Passed* (%)	*Failed* (%)
1	*Landing Page*	56	56	0	100%	0%
2	Profil Lembaga	2	2	0	100%	0%
3	Standar Pelayanan	5	5	0	100%	0%
4	Kebijakan Privasi	6	6	0	100%	0%
5	Hasil Pencarian Semua	60	60	0	100%	0%
6	Hasil Pencarian Perguruan Tinggi	18	18	0	100%	0%
7	Detail Perguruan Tinggi	37	37	0	100%	0%
8	Hasil Pencarian Program Studi	18	18	0	100%	0%
9	Detail Program Studi	11	11	0	100%	0%
10	Hasil Pencarian Dosen	18	18	0	100%	0%
11	Detail Dosen	38	38	0	100%	0%
12	Hasil Pencarian Mahasiswa	18	18	0	100%	0%
13	Detail Mahasiswa	2	2	0	100%	0%
14	Program Studi	14	14	0	100%	0%
15	Bidang Agama	28	28	0	100%	0%
16	Bidang Ekonomi	28	28	0	100%	0%
17	Bidang Humaniora	28	28	0	100%	0%

p-ISSN: 2301-5373
e-ISSN: 2654-5101

No.	*Test Case Feature*	*Total Test Case*	Jumlah *Passed*	Jumlah *Failed*	*Passed* (%)	*Failed* (%)
18	Bidang Kesehatan	46	46	0	100%	0%
19	Bidang MIPA	46	46	0	100%	0%
20	Bidang Pendidikan	46	46	0	100%	0%
21	Bidang Pertanian	46	46	0	100%	0%
22	Bidang Seni	46	46	0	100%	0%
23	Bidang Sosial	46	46	0	100%	0%
24	Bidang Teknik	46	46	0	100%	0%
25	Penyedia Program Studi	174	174	0	100%	0%
26	Perguruan Tinggi	179	179	0	100%	0%
27	Pengumuman	24	22	2	91,67%	8,33%
28	Statistik	266	265	1	99,62%	0,38%
29	Kategori Perbandingan	3	3	0	100%	0%
30	Pemilihan Perguruan Tinggi	179	179	0	100%	0%
31	Perbandingan Perguruan Tinggi	15	15	0	100%	0%
32	Pemilihan Program Studi	173	173	0	100%	0%
33	Perbandingan Program Studi	18	15	3	83,33%	16,67%
34	Kontributor	103	103	0	100%	0%
	Total	1843	1837	6	99,25%	0,75%

Berdasarkan hasil pengujian manual, laman PDDIKTI menunjukkan tingkat keberhasilan 99,25%, dengan 0,75% skenario mengalami kegagalan. Mayoritas fitur utama berfungsi dengan baik, namun beberapa fitur seperti perbandingan program studi dan kontributor mengalami kendala dengan tingkat kegagalan mencapai 33,33%. Hasil ini mengindikasikan bahwa meskipun sistem telah cukup stabil, masih diperlukan optimalisasi pada fitur tertentu untuk meningkatkan keandalan dan keamanan laman PDDIKTI secara menyeluruh.

b. Pengujian Otomatis

Selanjutnya, dalam pengujian otomatis menggunakan OWASP ZAP, pada pengujian keamanan sistem terdapat 5 *alert* atau peringatan di antaranya:

1. *Cross-Domain Misconfiguration*, dengan tingkat risiko sedang yang memiliki jumlah peringatan 3x (60%).
2. *Server Leaks Version Information via "Server" HTTP Response Header Field*, dengan tingkat risiko rendah yang memiliki jumlah peringatan 8x (160%).
3. *Strict-Transport-Security Header Not Set*, dengan tingkat risiko rendah yang memiliki jumlah peringatan 5x (100%).
4. *X-Content-Type-Options Header Missing*, dengan tingkat risiko rendah yang memiliki jumlah peringatan 5x (100%).
5. *Re-examine Cache-control Directives*, dengan tingkat risiko hanya sebagai informatif atau dengan kata lain tidak terlalu rentan yang memiliki jumlah peringatan 5x (100%).

Berikut adalah hasil dari laporan pengecekan keamanan sistem yang telah dilakukan:

Alert type	Risk	Count
Cross-Domain Misconfiguration	Medium	3 (60.0%)
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	8 (160.0%)
Strict-Transport-Security Header Not Set	Low	5 (100.0%)
X-Content-Type-Options Header Missing	Low	5 (100.0%)
Re-examine Cache-control Directives	Informational	5 (100.0%)
Total		5

Gambar 2. Hasil Uji Pengujian Otomatis

Kesimpulannya, masih terdapat celah pada sistem dengan total *alerts* sebanyak 5 dengan tingkat risiko rentan berada di level Sedang, Rendah dan Informatif. Berdasarkan hasil tersebut, pada laman web PDDIKTI masih terdapat celah bagi oknum-oknum tidak bertanggung jawab untuk mengambil data-data pada server.

3.6. *Test Cycle Closure*

Berdasarkan hasil pengujian, dapat disimpulkan bahwa meskipun laman PDDIKTI memiliki tingkat keamanan yang cukup baik, masih terdapat beberapa celah keamanan yang perlu diperbaiki. Langkah-langkah yang disarankan untuk perbaikan adalah:

a. Meningkatkan validasi input untuk mencegah *SQL Injection* dan XSS.
b. Mengaktifkan *header* keamanan seperti *Strict-Transport-Security* dan *X-Content-Type-Options*.
c. Meninjau kembali konfigurasi *cache* untuk menghindari kebocoran data sensitif.
d. Memperbaiki pengelolaan sesi untuk menghindari serangan *session hijacking*.

Dengan menerapkan rekomendasi ini, diharapkan sistem dapat memiliki keamanan yang lebih baik dan lebih tahan terhadap serangan siber.

4. Kesimpulan

Penelitian ini menunjukkan bahwa meskipun laman PDDIKTI memiliki tingkat keamanan yang cukup baik, masih terdapat beberapa celah yang perlu diperbaiki. Pengujian manual menemukan 6 kegagalan pada fitur pengumuman, statistik, dan perbandingan program studi, sementara pengujian otomatis dengan OWASP ZAP mendeteksi 5 celah keamanan, termasuk konfigurasi CORS yang tidak tepat dan *header* HTTP yang tidak aman. Kontribusi utama dari penelitian ini adalah penerapan STLC dalam pengujian keamanan sistem akademik yang menyediakan pendekatan sistematis dalam mengidentifikasi dan mengurangi celah keamanan. Selain itu, penelitian ini memberikan rekomendasi strategis bagi pengelola sistem akademik untuk meningkatkan perlindungan data digital. Implementasi metode STLC dalam pengujian keamanan telah terbukti memberikan hasil yang lebih sistematis dan efektif, sehingga dapat menjadi referensi bagi pengelola sistem akademik lainnya dalam menjaga keandalan data digital.

Referensi

[1] G. Safaat and V. U. Tjhin, “Analysis of Quality Assurance Performance in the Application of Manual Testing and Automation Testing for Software Product Testing”, *iijse*, vol. 7, no. 2, pp. 1987-1996, Mar. 2024.

[2] A. Muni, B. Rianto, and M. Jalil, “Analisis Kerentanan Keamanan Sistem Informasi Akademik Menggunakan OWASP-ZAP di Universitas Islam Indragiri”, *TEKNOFILE: Jurnal Sistem Informasi*, vol. 2, no. 6, pp. 409–420, 2024.
[3] B. Respiar, A. Fernanda, T. Taryadi, F. Maulana, and A. H. Anshor, “Peran Penting Software Quality Assurance Dalam Pengembangan Aplikasi”, *Journal Scientific of Mandalika (JSM)*, vol. 5, no. 12, pp. 535-540, Dec. 2024.
[4] M. G. A. Danialdo, F. A. Bakhtiar, and M. Data, “Pengujian Efektivitas OWASP ZAP dalam Menemukan Kerentanan dari Metasploitable”, *J-PTIIK*, vol. 7, no. 7, pp. 3431–3433, Oct 2023.
[5] F. J. Hulu and R. P. Kristianto, “Pengujian Manual dan Otomatisasi dengan Selenium pada Website SoundCloud”, Jul. 2024.
[6] D. F. N. Utami and R. H. Setyodewi, “Documentation Of Software Testing For Dafbin Aplication With IEEE 829-2008 Standar”, *Restikom*, vol. 5, no. 2, pp. 107-117, Aug. 2023.
[7] A. Arfan and H. Hendrik, “Penerapan STLC dalam Pengujian Black Box dengan Automation Testing Tool (Studi kasus: PT.GIT Solution)”, *AUTOMATA*, vol. 3, no. 2, Aug. 2022.

This page is intentionally left blank.