PERBANDINGAN EFEKTIVITAS OWASP ZAP, ACUNETIX, NIKTO MENGGUNAKAN VULNERABILITY SCANNING UNTUK DETEKSI KERENTANAN APLIKASI WEB

Arnefia Yuzar, Alam Rahmatulloh*
Program Studi Informatika, Fakultas Teknik, Universitas Siliwangi
Jalan Mugarsari, Kel Mugarsari, Tasikmalaya, 46196, Indonesia
alam@unsil.ac.id

ABSTRAK

Keamanan aplikasi web menjadi isu penting seiring meningkatnya penggunaan aplikasi berbasis web, risiko serangan terhadap data sensitif yang dikelola juga meningkat. *Vulnerability scanning* merupakan metode efektif untuk mengidentifikasi dan menilai kerentanan aplikasi web. Penelitian ini bertujuan untuk membandingkan efektivitas tiga alat *vulnerability scanning* diantaranya OWASP ZAP, Acunetix, dan Nikto. Hasilnya dianalisis berdasarkan jumlah, jenis kerentanan, waktu pemindaian, kecepatan dan efisiensi alat. Hasil penelitian menunjukkan Acunetix sebagai alat paling komprehensif, mendeteksi total 20 kerentanan, termasuk seperti *Cross-Site Scripting (XSS)*, dengan risiko tinggi dan menengah mencapai 75%. OWASP ZAP mendeteksi 13 kerentanan seperti *Content Security Policy (CSP)*. Nikto mendeteksi 5 kerentanan seperti *ketidakhadiran header X-XSS-Protection* dan *Expect-CT*. Kombinasi ketiga alat ini memberikan cakupan keamanan yang lebih menyeluruh, OWASP ZAP mendeteksi kelemahan dasar, Acunetix mengidentifikasi kerentanan tingkat lanjut, dan Nikto memverifikasi konfigurasi server. Rekomendasi alat disusun berdasarkan hasil analisis, sehingga dapat menjadi langkah proaktif dalam meningkatkan keamanan aplikasi web terhadap ancaman siber.

Kata kunci : *Acunetix, Kerentanan Aplikasi Web, Nikto, OWASP ZAP, Vulnerability Scanning.*

1. PENDAHULUAN

Keamanan aplikasi web telah menjadi salah satu fokus utama dalam pengembangan teknologi informasi saat ini, mengingat meningkatnya penggunaan aplikasi berbasis web dalam berbagai aspek kehidupan, mulai dari layanan publik, perbankan [1].

Dengan semakin banyaknya data sensitif yang diolah dan disimpan oleh aplikasi-aplikasi ini, potensi risiko terhadap serangan siber juga semakin meningkat [2].

Para penyerang terus mengembangkan teknik-teknik baru untuk mengeksploitasi kerentanan yang ada dalam aplikasi, sehingga penting bagi organisasi untuk secara proaktif mengidentifikasi dan mengatasi potensi celah keamanan sebelum disalahgunakan [3].

Vulnerability scanning menjadi metode penting dalam menilai keamanan aplikasi web [4].

Vulnerability scanning tidak hanya membantu menemukan kerentanan yang ada, tetapi juga memberikan gambaran yang jelas tentang seberapa efektif mekanisme keamanan yang diterapkan [5].

Proses ini melibatkan pemindaian otomatis yang dilakukan oleh alat khusus untuk mengevaluasi keamanan sistem dan mengidentifikasi titik lemah yang rentan dieksploitasi oleh penyerang [6].

Berbagai alat *vulnerability scanning* yang tersedia masing-masing dengan keunggulan dan pendekatan yang berbeda [7].

Dalam penelitian ini, OWASP ZAP, Acunetix, dan Nikto dipilih sebagai objek perbandingan karena ketiganya mewakili pendekatan yang berbeda dalam pemindaian kerentanan aplikasi web [8].

OWASP ZAP dikenal sebagai alat *open-source* yang anyak digunakan dalam pengujian keamanan aplikasi web karena kemampuannya dalam melakukan deteksi kerentanan secara otomatis serta fleksibilitasnya untuk digunakan dalam berbagai skenario pengujian web [9].

Acunetix merupakan alat komersial yang dikenal dengan kemampuannya dalam melakukan pemindaian mendalam dan hasil analisis yang lebih komprehensif, menjadikannya pilihan utama bagi organisasi yang mengutamakan akurasi dan kelengkapan deteksi [10].

Nikto lebih berfokus pada pemeriksaan konfigurasi server serta masalah keamanan yang terkait dengan infrastruktur aplikasi web, menjadikannya alat yang berharga dalam menilai ketahanan server terhadap berbagai ancaman[11].

Masing-masing alat memiliki kekuatan dan kelemahan yang unik, menjadikannya cocok untuk berbagai situasi dan kebutuhan pengujian.

Penelitian ini akan membandingkan efektivitas ketiga alat dalam mendeteksi kerentanan aplikasi web, bertujuan untuk mengidentifikasi alat yang paling efektif dan memberikan rekomendasi praktik terbaik untuk keamanan aplikasi. Diharapkan penelitian ini dapat berkontribusi pada pengembangan keamanan aplikasi web dan meningkatkan kesadaran akan pentingnya perlindungan data di era digital.

2. TINJAUAN PUSTAKA

Tinjauan pustaka ini bertujuan untuk meneliti berbagai penelitian dan sumber yang berhubungan dengan keamanan aplikasi web serta penggunaan metode *vulnerability scanning* dalam mendeteksi

kerentanan sistem. Dengan meningkatnya ancaman siber terhadap aplikasi berbasis web, penting untuk memahami bagaimana berbagai alat pemindai kerentanan dapat membantu dalam mengidentifikasi dan mengatasi celah keamanan.

2.1. Keamanan Aplikasi Web

Keamanan aplikasi web menjadi semakin penting seiring dengan meningkatnya penggunaan aplikasi berbasis web di berbagai sektor. Keamanan aplikasi web memerlukan pendekatan sistematis dalam mengidentifikasi dan mengatasi ancaman yang muncul akibat eksploitasi celah keamanan. Salah satu metode yang digunakan untuk menilai keamanan aplikasi web adalah *vulnerability scanning*, yang memungkinkan deteksi dini terhadap berbagai kerentanan sebelum dimanfaatkan oleh penyerang [12].

2.2. OWASP ZAP

OWASP ZAP merupakan alat *open-source* yang banyak digunakan oleh peneliti keamanan untuk mengidentifikasi kelemahan pada aplikasi web. OWASP ZAP efektif dalam mendeteksi kerentanan seperti *Cross-Site Scripting (XSS)* dan *SQL Injection*, namun kurang optimal dalam mendeteksi kerentanan yang lebih kompleks terkait pustaka eksternal dan konfigurasi server [13].

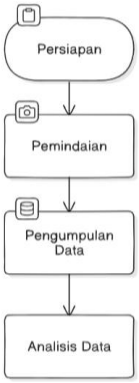
2.3. Acunetix

Acunetix dikenal sebagai alat komersial dengan cakupan pemindaian yang lebih luas dibandingkan OWASP ZAP. Acunetix mampu mendeteksi kerentanan tingkat lanjut, termasuk pustaka eksternal yang rentan serta manipulasi skrip. Keunggulan lain dari Acunetix adalah kemampuannya dalam memberikan rekomendasi mitigasi yang lebih komprehensif [14].

2.4. Nikto

Nikto lebih berfokus pada evaluasi konfigurasi server, termasuk pengaturan *header* keamanan dan sertifikat SSL. Nikto dalam mengidentifikasi kelemahan mendasar pada server, seperti *header* keamanan yang tidak diatur dengan baik atau *file* sensitif yang dapat diakses secara publik [15].

3. METODE PENELITIAN



Gambar 1. Tahapan Penelitian

Tahapan yang dilakukan dalam penelitian ini mengikuti metode *Vulnerability Scanning*, seperti ditunjukkan pada Gambar 1, yang terdiri dari empat langkah utama, yaitu Persiapan, Pemindaian, Pengumpulan Data, dan Analisis Data.

3.1. Persiapan

Pada tahap awal penelitian ini, dilakukan seleksi terhadap aplikasi web yang akan diuji kerentanannya, dengan syarat bahwa aplikasi tersebut tersedia dalam lingkungan yang aman, baik dalam bentuk aplikasi internal maupun yang di-hosting secara online. Aplikasi yang dipilih juga harus menggunakan teknologi-teknologi modern, seperti *framework JavaScript* atau sistem *backend* tertentu, yang berpotensi menyebabkan kerentanan keamanan tertentu.

Penelitian ini memakai alat OWASP ZAP, Acunetix, dan Nikto. Alat-alat ini akan diinstal dan dikonfigurasi dalam lingkungan yang sama untuk memastikan konsistensi dan akurasi hasil pengujian. Semua alat akan dijamin dalam versi terbaru, dan konfigurasi default akan diterapkan untuk menjaga validitas dan reliabilitas hasil pemindaian.

3.2. Pemindaian

Proses pemindaian dimulai dengan vulnerability scanning menggunakan OWASP ZAP digunakan dalam mode Active Scan untuk mendeteksi kerentanan pada aplikasi web, yang mencakup kerentanan seperti *Cross-Site Scripting (XSS)*, *SQL Injection*, *CSRF*, serta kerentanan konfigurasi umum lainnya.

Vulnerability scanning menggunakan Acunetix untuk menjalankan *full web scan* terhadap aplikasi yang diuji, dengan fokus pada identifikasi kerentanan yang lebih kompleks, seperti pustaka rentan, isu konfigurasi *cookie*, serta *header HTTP* yang tidak diatur dengan benar.

Nikto digunakan untuk memeriksa konfigurasi server. Nikto akan melakukan validasi terhadap sertifikat SSL, mengevaluasi *header* keamanan *HTTP*, dan mendeteksi potensi kebocoran *file* sensitif, termasuk *file* dengan ekstensi atau informasi konfigurasi lainnya yang mungkin dapat bersifat sensitif.

3.3. Pengumpulan Data

Pada tahap pengumpulan hasil dari setiap pemindaian kerentanan akan direkam secara sistematis, termasuk jenis kerentanan yang ditemukan oleh masing-masing alat. Hasil temuan akan dikelompokkan berdasarkan kategori, seperti *XSS*, *SQL Injection*, *CSRF*, dan pustaka rentan.

Mencatat waktu pemindaian yang dibutuhkan oleh setiap alat dari awal hingga selesai. Data waktu ini akan memberikan wawasan yang berharga mengenai kecepatan dan efisiensi masing-masing alat dalam melakukan pemindaian terhadap aplikasi yang diuji.

3.4. Analisis Data

Analisis data dilakukan untuk membandingkan hasil yang diperoleh dari pemindaian. Menganalisis jumlah kerentanan yang terdeteksi oleh setiap alat dan membandingkan jenis kerentanan yang teridentifikasi. Kerentanan ini akan dikelompokkan berdasarkan tingkat keparahan, yakni *low*, *medium*, *high*, dan *critical*.

Analisis efektivitas dengan mempertimbangkan beberapa parameter. Pertama, jumlah kerentanan yang berhasil ditemukan oleh masing-masing alat. Kedua, jenis kerentanan yang mampu dideteksi, dengan perhatian khusus pada kerentanan yang lebih kompleks. Terakhir, evaluasi akan mencakup aspek kecepatan pemindaian untuk memahami efisiensi masing-masing alat, khususnya pada aplikasi dengan tingkat kompleksitas yang tinggi.

Berdasarkan analisis yang telah dilakukan kemudian menentukan alat yang paling efektif dalam mendeteksi kerentanan pada aplikasi web secara menyeluruh. Efektivitas alat akan dinilai berdasarkan jumlah kerentanan yang terdeteksi, jenis kerentanan yang dapat diidentifikasi, serta waktu pemindaian yang diperlukan.

4. HASIL DAN PEMBAHASAN

Berdasarkan metode penelitian yang mencakup tahap persiapan, pemindaian, pengumpulan data, dan analisis data, berikut rincian hasil temuan dan pembahasan dari pemindaian kerentanan pada aplikasi web opendata.tasikmalayakab.go.id menggunakan OWASP ZAP, Acunetix, dan Nikto.

4.1. Persiapan

Tahap awal persiapan dilakukan guna menjamin kelancaran dan konsistensi dalam proses pemindaian. Aplikasi web yang dipilih untuk diuji adalah https://opendata.tasikmalayakab.go.id, yang memenuhi syarat sebagai aplikasi yang dapat berjalan dalam lingkungan aman, baik untuk penggunaan internal maupun akses melalui *internet*. Dengan karakteristik tersebut, aplikasi ini dinilai layak untuk diuji menggunakan alat pemindaian yang andal dan kuat.

Pada tahap instalasi alat pemindai, tiga alat utama telah berhasil diinstal dan dikonfigurasi khusus untuk kebutuhan penelitian ini. OWASP ZAP dikonfigurasi dalam mode *Active Scan* untuk mendeteksi berbagai kerentanan yang mungkin terdapat pada aplikasi web yang diuji. Acunetix digunakan untuk melakukan pemindaian penuh terhadap aplikasi web, yang mencakup deteksi kerentanan pada pustaka eksternal, evaluasi konfigurasi *cookie*, dan pemeriksaan kerentanan aplikasi secara umum. Nikto dimanfaatkan untuk mengevaluasi konfigurasi server, dengan fokus pada identifikasi masalah terkait *header HTTP* dan sertifikat *SSL* yang mungkin memengaruhi keamanan aplikasi web.

4.2. Pemindaian

Pada tahap pemindaian, masing-masing alat bekerja secara bertahap untuk mendeteksi kerentanan di aplikasi web. Hasil pemindaian menunjukkan berbagai jenis kerentanan yang ditemukan oleh OWASP ZAP, Acunetix, dan Nikto,

4.3. Rincian Kerentanan Owasp Zap

Owasp Zap ZAP berhasil mengidentifikasi total 13 kerentanan dengan berbagai tingkat risiko dan jenis yang perlu diperhatikan. Rincian lengkapnya disajikan dalam Tabel 1.

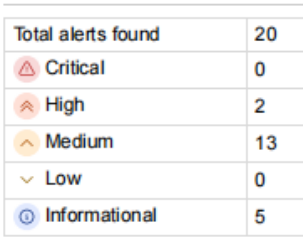
Tabel 1. Rincian Hasil *Scanning* Owasp Zap

Keterangan	Risiko	Temuan
Cloud Metadata Potentially Exposed	*High*	1
Content Security Policy (CSP) Header Not Set	*Medium*	4
Cross-Domain Misconfiguration	*Medium*	1
Missing Anti-clickjacking Header	*Medium*	4
Cross-Domain JavaScript Source File Inclusion	*Low*	4

Pada Tabel 1 dapat dilihat Owasp Zap berhasil mendeteksi total 13 kerentanan pada aplikasi web, dengan fokus utama pada pengaturan keamanan *header HTTP* dan *Content Security Policy (CSP)*. Salah satu temuan penting adalah risiko tinggi yang diidentifikasi pada metadata cloud yang terekspos, yang dapat dimanfaatkan oleh penyerang untuk mengakses informasi sensitif. Selain itu, OWASP ZAP mengungkap beberapa kelemahan dalam penerapan *header* keamanan, seperti *Strict-Transport-Security (HSTS)*, *X-Frame-Options*, dan *X-Content-Type-Options*. Ketidakhadiran atau ketidaksesuaian *header-header* ini membuat aplikasi rentan terhadap beberapa jenis serangan, termasuk *clickjacking* dan *Cross-Domain JavaScript.*

4.4. Rincian Kerentanan Acunetix

Pemindaian Acunetix menemukan 20 kerentanan, termasuk beberapa kerentanan kritis yang berpotensi membahayakan aplikasi web. Terdapat hasil pemindaian keamanan menggunakan Acunetix pada website opendata.kabtasikmalaya menemukan total 20 peringatan dengan rincian 2 peringatan berisiko tinggi, 13 berisiko menengah, dan 5 bersifat informasional, tanpa ada peringatan kritis atau berisiko rendah seperti yang ada pada Gambar 2.



Gambar 2. Hasil *Scanning* Acunetix

Temuan ini menunjukkan adanya potensi celah keamanan yang perlu segera ditangani, terutama pada kategori risiko tinggi dan menengah, untuk memastikan keamanan dan integritas sistem. Detail temuan utama disajikan pada Tabel 2.

Tabel 2. Rincian Hasil *Scanning* Acunetix

Kerentanan	Risiko	Temuan
Cross-site Scripting	*High*	2
HTTP Strict Transport Security Policy Not Enabled	*Medium*	1
jQuery Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability	*Medium*	3
JQuery Prototype Pollution Vulnerability	*Medium*	1
JQuery UI Cross-site Scripting (XSS) Vulnerability	*Medium*	1
JQuery UI Dialog Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability	*Medium*	3
Redis Unauthorized Access Vulnerability	*Medium*	1
Vulnerable JavaScript libraries	*Medium*	3
Access-Control-Allow-Origin header with wildcard (*) value	*Informational*	1
Content Security Policy (CSP) Not Implemented	*Informational*	1
Permissions-Policy header not implemented	*Informational*	1
Subresource Integrity (SRI) Not Implemented	*Informational*	2

Pada Tabel 2 dapat dilihat Acunetix mendeteksi total 20 kerentanan, dengan dua temuan *Cross-site Scripting (XSS)* yang memiliki tingkat risiko tinggi. *XSS* ini memungkinkan penyerang menyisipkan skrip berbahaya ke dalam aplikasi, yang kemudian dapat dieksekusi oleh pengguna akhir, mengakibatkan pencurian data atau manipulasi tampilan halaman web. Selain *XSS*, Acunetix juga menemukan bahwa aplikasi ini menggunakan pustaka *JavaScript* yang rentan, yang meningkatkan risiko serangan tambahan. Pengaturan *header HTTP Strict Transport Security (HSTS)* yang belum diaktifkan juga menjadi perhatian, karena *HSTS* memastikan akses *HTTPS* yang aman. Acunetix juga menyoroti penggunaan *header Access-Control-Allow-Origin* dengan nilai *wildcard*, yang memungkinkan akses dari semua sumber eksternal tanpa pembatasan. Hal ini membuka kemungkinan bagi pihak yang tidak berwenang untuk mengakses dan memanfaatkan data sensitif dari aplikasi.

4.5. Rincian Kerentanan Nikto

Nikto melakukan pemeriksaan konfigurasi server dan menemukan 5 kerentanan yang berhubungan dengan pengaturan *header* keamanan. Hasil rinciannya disajikan pada Tabel 3.

Tabel 3. Rincian Hasil *Scanning* Nikto

Kerentanan	Risiko	Temuan
X-XSS-Protection Header Not Defined	*Medium*	1
Anti-clickjacking X-Frame-Options	*Low*	1
Strict-Transport-Security Header Not Set	*Low*	1
Expect-CT Header Not Present	*Informational*	1
Potentially interesting archive/cert file found	*Informational*	1

Pada Tabel 3 dapat dilihat pemindaian Nikto mengidentifikasi lima kerentanan terkait konfigurasi server, terutama pada *header* keamanan seperti *X-XSS-Protection*, *X-Frame-Options*, dan *Expect-CT*. Tanpa *header* ini, aplikasi berisiko terhadap serangan *XSS*, *clickjacking*, dan masalah validasi sertifikat *SSL*. Nikto menyoroti perlunya konfigurasi server yang lebih ketat untuk melindungi aplikasi dari injeksi dan kebocoran informasi.

4.6. Pengumpulan Data

Pengumpulan data mencakup pencatatan semua kerentanan yang ditemukan oleh masing-masing alat, durasi pemindaian, serta klasifikasi tingkat keparahan dari setiap kerentanan. Setiap alat memiliki fokus yang berbeda, sehingga hasil pengumpulan data ini memberikan pemahaman yang lebih jelas tentang cakupan dan efektivitas dari masing-masing alat. Rincian jenis kerentanan disajikan pada Tabel 4.

Tabel 4. Pengumpulan Data

Kriteria	OWASP ZAP	Acunetix	Nikto
Fokus Pemindaian	Kerentanan aplikasi web	Pemindaian menyeluruh	Konfigurasi server & *header* keamanan
Jumlah Kerentanan	13	20	5
Risiko Tertinggi	*High (Cloud Metadata Exposed)*	*High (Cloud Metadata Exposed)*	*Medium (X-XSS-Protection Not Set)*
Durasi Pemindaian	45 Menit	197 Menit	84 Menit
Keunggulan	*Open-source*, fleksibel, mudah digunakan, bisa kustomisasi	Pemindaian menyeluruh, deteksi paling banyak, rekomendasi mitigasi detail	Fokus pada keamanan server, mendeteksi kelemahan konfigurasi

Kriteria	OWASP ZAP	Acunetix	Nikto
Kelemahan	Tidak optimal dalam mendeteksi pustaka eksternal yang rentan	Pemindaian lama, butuhkan biaya lisensi	Cakupan terbatas dibanding OWASP ZAP dan Acunetix

Pada Tabel 4 dapat dilihat ketiga alat pemindaian menunjukkan peran spesifik dan keunggulan masing-masing alat dalam mendeteksi kerentanan pada aplikasi web https://opendata.tasikmalayakab.go.id. OWASP ZAP unggul dalam mendeteksi kelemahan dasar pada *header HTTP* dan kebijakan keamanan (*CSP*, *Clickjacking*, *Cross-Domain*). Dengan waktu pemindaian 45 menit, kemampuannya dalam mendeteksi kerentanan kompleks seperti pustaka eksternal rentan masih terbatas. Acunetix alat komersial paling komprehensif dengan deteksi 20 kerentanan, termasuk ancaman tingkat tinggi seperti *Cross-Site Scripting (XSS)* dan pustaka *JavaScript* rentan. Keunggulannya terletak pada analisis mendalam dan rekomendasi mitigasi yang detail. Namun, alat ini memiliki waktu pemindaian lebih lama 197 menit dan biaya lisensi yang tinggi. Nikto berfokus pada evaluasi konfigurasi server, mendeteksi kelemahan pada *header* keamanan dan *SSL*. Dengan waktu pemindaian sedang 84 menit, alat ini berguna untuk pemeriksaan awal keamanan server. Namun, cakupan deteksinya terbatas dibandingkan OWASP ZAP dan Acunetix, karena tidak dirancang untuk menemukan kerentanan aplikasi tingkat lanjut.

4.7. Analisis Data

Berdasarkan Berdasarkan hasil analisis data, setiap alat memiliki peran spesifik dan kontribusi unik dalam mengidentifikasi kerentanan pada aplikasi web opendata.tasikmalayakab.go.id. OWASP ZAP mendeteksi 13 kerentanan dengan komposisi risiko sekitar 8% berisiko tinggi, 69% berisiko menengah, dan 23% berisiko rendah. Fokus utama alat ini adalah pada pengaturan *header* keamanan *HTTP*, seperti *Content Security Policy (CSP)*, *X-Frame-Options*, dan *Strict-Transport-Security (HSTS)*, yang kurang optimal. Hal ini menunjukkan bahwa OWASP ZAP sangat berguna dalam mendeteksi kelemahan keamanan dasar pada aplikasi web. Dengan waktu pemindaian 45 menit, alat ini efektif untuk mendeteksi permasalahan pada konfigurasi keamanan dasar namun kurang dalam mengidentifikasi ancaman tingkat lanjut seperti kerentanan pada pustaka eksternal atau skrip injeksi kompleks.

Acunetix mengidentifikasi 20 kerentanan, terdiri dari 10% risiko tinggi, seperti *Cross-Site Scripting (XSS)*, 65% risiko menengah, termasuk pustaka *JavaScript* rentan dan pengaturan *header* keamanan yang lemah, serta 25% tingkat informasi, seperti penggunaan *header Access-Control-Allow-Origin* dengan *wildcard*. Hal ini menunjukkan bahwa Acunetix memiliki cakupan pemindaian yang lebih luas dan lebih efektif dalam mendeteksi kerentanan tingkat lanjut, terutama yang berkaitan dengan skrip, pustaka eksternal, dan kebijakan keamanan *HTTP*. Waktu pemindaian Acunetix 197 menit dan tools ini berbayar.

Nikto mendeteksi 5 kerentanan, dengan 20% risiko menengah dan 80% tingkat informasi. Fokus utama alat ini adalah pada keamanan konfigurasi server, dengan deteksi kelemahan seperti ketidakhadiran *header X-XSS-Protection* dan *Expect-CT*. Hal ini menunjukkan bahwa Nikto berperan penting dalam memastikan konfigurasi server yang lebih aman, meskipun cakupan deteksinya lebih terbatas dibandingkan OWASP ZAP dan Acunetix. Dengan waktu pemindaian 84 menit, Nikto cocok digunakan untuk memverifikasi pengaturan keamanan server, tetapi kurang efektif dalam mengidentifikasi kerentanan tingkat lanjut pada aplikasi web.

Secara keseluruhan, kombinasi ketiga alat ini menghasilkan total 38 kerentanan, dengan 34% dari OWASP ZAP, 53% dari Acunetix, dan 13% dari Nikto. Acunetix terbukti paling efektif dalam mendeteksi kerentanan tingkat lanjut dengan 75% risiko tinggi dan menengah, menjadikannya pilihan utama untuk mendeteksi ancaman kompleks. OWASP ZAP sangat berguna untuk mengidentifikasi kelemahan dasar pada *header* keamanan, menjadikannya alat yang ideal untuk pengujian awal keamanan aplikasi. Nikto unggul dalam mengevaluasi konfigurasi server, membantu memastikan pengaturan yang lebih ketat terhadap potensi ancaman siber

Dalam penelitian sebelumnya juga telah membahas perbandingan alat *vulnerability scanning* seperti OWASP ZAP, Acunetix, dan Nikto, namun menunjukkan adanya perbedaan dalam fokus evaluasi dan hasil temuan. Setiap alat memiliki keunggulan yang unik dalam mendeteksi kerentanan tertentu, seperti kemampuan OWASP ZAP untuk mendeteksi kerentanan dasar, kekuatan Acunetix pada kerentanan tingkat lanjut, dan peran Nikto dalam memeriksa konfigurasi server.

Penelitian sebelumnya oleh Hasibuan et al., mencari kerentanan menggunakan Owasp Zap pada situs lab.scarpe, meliputi 3 risiko menengah dan 5 risiko rendah, dengan penekanan pada masalah seperti *Absence of Anti-CSRF Tokens*, *Missing Anti-clickjacking Header*, dan *Cross-Domain JavaScript Source Inclusion*. Dengan memaparkan langkah teknis mendetail, termasuk penggunaan *Automated Scanner*, eksplorasi manual, dan mitigasi spesifik seperti penerapan *SameSite Attributes*, *HttpOnly Cookies*, dan *X-Content-Type-Options* [16]. Sedangkan pada penelitian ini OWASP ZAP mendeteksi 13 kerentanan, dengan komposisi risiko sekitar 8% berisiko tinggi, 69% berisiko menengah, dan 23% berisiko rendah, dengan fokus pada pengaturan *header* keamanan *HTTP* seperti *Content Security Policy (CSP)*, *X-Frame-Options*, dan *Strict-Transport-Security (HSTS)* yang kurang optimal, menunjukkan

peran pentingnya dalam mendeteksi kelemahan keamanan dasar aplikasi.

Penelitian sebelumnya oleh Zirwan, menggunakan Acunetix untuk mendeteksi 714 kerentanan pada *website ITP*, yang terdiri dari 94 tingkat tinggi (seperti *XSS*), 25 menengah, 46 rendah, dan 549 informasional, serta menunjukkan bahwa mitigasi dapat menurunkan tingkat ancaman ke level rendah, memperkuat peran alat ini dalam identifikasi dan perbaikan kerentanan yang kompleks dan beragam cenderung memberikan data dalam jumlah absolut tanpa penjelasan rinci tentang risiko spesifik yang relevan dengan praktik keamanan web terkini [10].

Sedangkan pada penelitian ini Acunetix berhasil mengidentifikasi 20 kerentanan yang mencakup 10% risiko tinggi, seperti *Cross-Site Scripting (XSS)*, 65% risiko menengah, termasuk pustaka *JavaScript* yang rentan dan pengaturan *header* keamanan yang lemah, serta 25% kerentanan tingkat informasi seperti *header Access-Control-Allow-Origin* dengan konfigurasi *wildcard*. Hasil ini menegaskan keunggulan Acunetix dalam mendeteksi kerentanan tingkat lanjut, khususnya yang melibatkan elemen kompleks seperti skrip dan pustaka eksternal, memberikan wawasan mendalam yang mendukung mitigasi risiko keamanan dengan presisi tinggi. cenderung memberikan data tanpa penjelasan rinci tentang risiko spesifik yang relevan dengan praktik keamanan web terkini.

Pada penelitian sebelumnya oleh Vargehese et al., Nikto difokuskan pada analisis keamanan server, mendeteksi kelemahan seperti *server misconfiguration, default files and programs, insecure files*, serta penggunaan perangkat lunak yang usang, tanpa informasi eksplisit mengenai jumlah kerentanan yang ditemukan atau pembagian risiko dalam kategori tertentu [17].

Sementara itu pada penelitian ini mendapat 5 kerentanan dengan distribusi 20% risiko menengah dan 80% tingkat informasi, yang lebih menyoroti kurangnya *header* keamanan seperti *X-XSS-Protection* dan *Expect-CT*, menunjukkan fokus pada penguatan konfigurasi server yang lebih spesifik. Penelitian ini menekankan hasil kerentanan yang dikelompokkan berdasarkan tingkat risiko, memberikan konteks tambahan tentang signifikansi temuan.

4.8. Evaluasi Performa Alat

Perbandingan efektivitas OWASP ZAP, Acunetix, dan Nikto digunakan untuk mengidentifikasi kelemahan keamanan pada aplikasi web opendata.tasikmalayakab.go.id. Setiap alat memiliki kelebihan dan kekurangan, menghasilkan temuan berbeda sesuai fokus dan kapabilitasnya. Diskusi ini bertujuan untuk mengevaluasi alat-alat tersebut dan menentukan yang paling cocok bagi keamanan aplikasi web ini.

OWASP ZAP mendeteksi 13 kerentanan dengan rincian 8% berisiko tinggi, 69% menengah, dan 23% rendah. Alat ini efektif dalam mendeteksi kelemahan dasar pada *header HTTP*, seperti *Content Security Policy (CSP), X-Frame-Options, dan Strict-Transport-Security (HSTS)*, yang rentan terhadap serangan *clickjacking* dan *XSS*. Namun, OWASP ZAP memiliki keterbatasan dalam mendeteksi kerentanan yang lebih kompleks, seperti pustaka eksternal yang rentan atau skrip injeksi. Secara keseluruhan, OWASP ZAP memberikan wawasan keamanan dasar pada *HTTP*, tetapi kurang efektif untuk mengidentifikasi ancaman yang lebih kompleks, sehingga membutuhkan alat pendukung untuk cakupan yang lebih lengkap.

Acunetix memiliki cakupan deteksi yang paling komprehensif, menemukan 20 kerentanan dengan 10% berisiko tinggi, 65% menengah, dan 25% pada tingkat informasi. Alat ini efektif dalam mendeteksi kerentanan tingkat lanjut seperti *Cross-Site Scripting (XSS)*, pustaka *JavaScript* yang rentan, dan kelemahan pada *header Access-Control-Allow-Origin* dengan *wildcard*, yang rentan terhadap kebocoran data. Keunggulan utama Acunetix adalah kemampuannya mengidentifikasi kerentanan kompleks yang sering tidak terdeteksi oleh OWASP ZAP dan Nikto. Meski pemindaian memakan waktu lebih lama, Acunetix memberikan hasil detail dan rekomendasi berguna, seperti pembaruan pustaka eksternal dan sanitasi input, untuk melindungi aplikasi dari ancaman canggih.

Nikto berfokus pada keamanan konfigurasi server, dengan mendeteksi 5 kerentanan, di mana 20% berisiko menengah dan 80% berupa informasi tambahan yang mencakup pengaturan *header* keamanan server. Nikto mengidentifikasi masalah konfigurasi dasar, seperti ketiadaan *header X-XSS-Protection* dan *Expect-CT*, *serfile* sertifikat yang tidak diamankan dengan baik, yang menunjukkan pentingnya peningkatan pengaturan keamanan server. Meskipun cakupan deteksi Nikto terbatas dibandingkan dengan Acunetix dan OWASP ZAP, alat ini sangat berguna dalam memverifikasi dan meningkatkan konfigurasi server, yang sering kali diabaikan dalam proses *vulnerability scanning*. Alat ini kurang dalam mengidentifikasi kerentanan aplikasi yang lebih canggih tetapi esensial untuk memastikan server memiliki konfigurasi yang lebih aman dan minim risiko

4.9. Penentuan Alat Terbaik

Dari ketiga alat pemindai yang digunakan, Acunetix terbukti paling efektif, alat ini memiliki cakupan deteksi yang lebih luas dan mampu mengidentifikasi kerentanan tingkat lanjut, seperti *XSS* dan pustaka *JavaScript* rentan, yang tidak terdeteksi oleh OWASP ZAP atau Nikto. Selain itu, Acunetix menunjukkan sensitivitas tinggi terhadap kerentanan berisiko tinggi dan memberikan rekomendasi seperti pembaruan pustaka dan sanitasi input untuk meningkatkan keamanan aplikasi.

Meskipun proses pemindaian menggunakan Acunetix memerlukan waktu yang lebih lama, hasil analisis yang komprehensif membuatnya menjadi pilihan terbaik. Kemampuannya untuk mendeteksi

kerentanan kompleks *seperti Cross-Site Scripting (XSS)* dan pustaka eksternal yang rentan menjadi nilai tambah signifikan. OWASP ZAP unggul dalam mengidentifikasi kelemahan pada *HTTP security headers*, yang sering menjadi celah keamanan dasar. Nikto menawarkan keunggulan dalam memeriksa dan memastikan keamanan konfigurasi server. Kombinasi ketiga alat ini memberikan cakupan yang lebih menyeluruh untuk meningkatkan keamanan aplikasi web secara efektif.

4.10. Rekomendasi

Untuk mengoptimalkan keamanan web opendata.tasikmalayakab.go.id, disarankan untuk menggunakan Acunetix sebagai alat utama dalam mendeteksi kerentanan kompleks pada aplikasi. OWASP ZAP sebaiknya digunakan untuk mengevaluasi penerapan kebijakan keamanan *HTTP* secara menyeluruh. Nikto sebaiknya digunakan secara berkala untuk memeriksa dan meningkatkan keamanan konfigurasi server, memastikan tidak ada potensi kebocoran data dari sisi server.

Dengan mengkombinasikan ketiga alat ini, cakupan keamanan yang lebih menyeluruh dapat tercapai, mencakup aspek keamanan dasar hingga kompleks, baik dari sisi aplikasi maupun server. Pendekatan ini akan membantu dalam mengidentifikasi dan menanggulangi potensi ancaman pada berbagai lapisan keamanan aplikasi, memberikan perlindungan proaktif terhadap ancaman siber yang semakin kompleks dan berkembang.

5. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa setiap alat *vulnerability scanning* memiliki keunggulan dan fokus yang berbeda dalam mendeteksi kerentanan pada aplikasi web. Acunetix terbukti sebagai alat paling komprehensif, mendeteksi 20 kerentanan termasuk kerentanan tingkat lanjut, seperti *Cross-Site Scripting (XSS)* dan pustaka *JavaScript* yang rentan, menyediakan rekomendasi yang memperkuat keamanan aplikasi. OWASP ZAP efektif dalam mengidentifikasi kelemahan dasar pada *header* keamanan *HTTP*. Nikto unggul dalam memastikan keamanan konfigurasi server. Kombinasi ketiga alat ini dapat memberikan cakupan keamanan yang menyeluruh, mencakup aspek dasar hingga tingkat lanjut dalam keamanan aplikasi dan server.

Pengujian lebih lanjut terhadap berbagai jenis aplikasi web dengan skala dan kompleksitas yang berbeda untuk mengevaluasi kinerja masing-masing alat dalam lingkungan yang lebih variatif dapat dilakukan untuk penelitian di masa mendatang. Selain itu, menggunakan alat *vulnerability scanning* tambahan yang lebih canggih atau berbasis kecerdasan buatan bisa menjadi langkah untuk mendapatkan hasil yang lebih akurat. Mengeksplorasi metode kombinasi otomatis dari beberapa alat supaya lebih efisien dalam mendeteksi kerentanan dan meminimalkan waktu pemindaian tanpa mengorbankan cakupan deteksi.

DAFTAR PUSTAKA

[1] A. D. Mishra and K. Mustafa, "Formalization of Security Requirements-A Case Study on a Web-Based Application," *Journal of Scientific Research*, vol. 66, no. 02, pp. 108–114, 2022, doi: 10.37398/jsr.2022.660214.

[2] E. Sankar, M. Nikhil, and G. S. Reddy, "Cyber Attacks Prediction using Data Science," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 06, no. 03, Mar. 2022, doi: 10.55041/IJSREM11906.

[3] C. Adam, M. F. Bulut, D. Sow, S. Ocepek, C. Bedell, and L. Ngweta, "Attack Techniques and Threat Identification for Vulnerabilities," *Association for Computing Machinery*, Jun. 2022, [Online]. Available: http://arxiv.org/abs/2206.11171

[4] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, "A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions," *Applied Sciences (Switzerland)*, vol. 12, no. 8, Apr. 2022, doi: 10.3390/app12084077.

[5] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions," Mar. 01, 2023, *MDPI*. doi: 10.3390/electronics12061333.

[6] K. Bennouk, N. Ait Aali, Y. El Bouzekri El Idrissi, B. Sebai, A. Z. Faroukhi, and D. Mahouachi, "A Comprehensive Review and Assessment of Cybersecurity Vulnerability Detection Methodologies," *Journal of Cybersecurity and Privacy*, vol. 4, no. 4, pp. 853–908, Oct. 2024, doi: 10.3390/jcp4040040.

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

[8] 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," *Electronics (Switzerland)*, vol. 12, no. 12, Jun. 2023, doi: 10.3390/electronics12122664.

[9] L. Sakyi Larbi, K. Coll of Educ Komenda, G. Issah Bala Abdulai, R. Amankwah, and A. Amponsah, "Performance Evaluation of Open Source Web Applica-tion Vulnerability Scanners based on OWASP Bench-mark Pious Akwasi Sarpong Daniel Paa Korsah," 2021.

[10] A. Zirwan, "Pengujian dan Analisis Kemanan Website Menggunakan Acunetix Vulnerability Scanner," *Jurnal Informasi dan Teknologi*, pp. 70–75, Mar. 2022, doi: 10.37034/jidt.v4i1.190.

[11] S. Utoro *et al.*, "Analisis Keamanan Website E-Learning SMKN 1 Cibatu Menggunakan Metode

Penetration Testing Execution Standard," *URNAL MULTINETICS*, vol. 6, Nov. 2020, Accessed: Oct. 12, 2024. [Online]. Available: https://doi.org/10.32722/MULTINETICS.V6I2.3432

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

[13] S. Alazmi and D. C. De Leon, "A Systematic Literature Review on the Characteristics and Effectiveness of Web Application Vulnerability Scanners," 2022, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2022.3161522.

[14] S. A. Haryono *et al.*, "Automated Identification of Libraries from Vulnerability Data: Can We Do Better?," in *IEEE International Conference on Program Comprehension*, IEEE Computer Society, 2022, pp. 178–189. doi: 10.1145/3524610.3527893.

[15] A. Carranza, M. Bustamante, H. Carranza, C. DeCusatis, and J. Islam, "Firewall and DNS Filtering Penetration Tests using Parrot OS," in *Proceedings of the LACCEI international Multi-conference for Engineering, Education and Technology*, Latin American and Caribbean Consortium of Engineering Institutions, 2022. doi: 10.18687/LACCEI2022.1.1.675.

[16] A. F. Hasibuan, Tommy, and D. Handoko, "Analisis Kerentanan Website Dengan Aplikasi Owasp Zap," *Jurnal Ilmu Komputer dan Sistem Informasi (JIRSI)*, vol. 2, pp. 257–270, Mar. 2023.

[17] S. Varghese and R. Kurian, "Identifying Vulnerabilities in a Website Using Uniscan and Comparing Uniscan, Grabber, Nikto," *Proceedings of the National Conference on Emerging Computer Applications (NCECA)-2021*, vol. 3, no. 1, 2021, doi: 10.5281/zenodo.5091326