JIPAKIF NUSANTARA
Jurnal Inovasi Pengembangan Aplikasi dan Keamanan Informasi Nusantara
http://jurnal.edunovationresearch.org/

Analisis Celah Keamanan E-Learning Perguruan Tinggi Menggunakan Vulnerability Assessment

Abdul Rohim[1)], Lila Setiyani[2)*]
[1] Program Studi Sistem Informasi, STMIK ROSMA
[2] Program Studi Sistem Informasi, Horizon University Indonesia
Email: lila.setiyani.krw@horizon.ac.id *

ABSTRAK

Perkembangan Teknologi Informasi pada saat ini membawa kemudahan bagi kehidupan manusia. Salah satu hal yang berkembang cukup pesat adalah aplikasi berbasis web. Saat ini konsep *E-Learning* sudah banyak diterima oleh masyarakat dunia, terbukti dengan banyaknya implementasi *E-Learning* di perguruan tinggi dan sekolah. Oleh sebab itu teknologi sistem informasi yang ada pada saat ini dapat memenuhi kebutuhan masyarakat secara langsung. Akan tetapi tidak sedikit pihak yang memanfaatkan perkembangan teknologi secara tidak bertanggung jawab. Mulai dari kasus peretasan, penipuan, bahkan kasus – kasus *cybercrime* yang tidak hanya menyerang individu, namun juga perusahaan/korporasi dan instansi pemerintah. Penelitian ini bertujuan untuk menganalisis dan melakukan pengujian keamanan, kerentanan, pada *website E-Learning* menggunakan metode *Vulnerability assessment* dan *website vulnerability scanning* dengan OWASP tools. Hasil penelitian ini menunjukkan ada 15 kerentanan dengan tiga kategori yaitu 2 kategori medium, 8 kategori low dan 5 kategori informational. Selain itu, penelitian ini juga memberikan alternative solusi untuk menangani kerentanan tersebut sehingga dapat dimanfaatkan oleh perguruan tinggi untuk meningkatkan keamanan informasi pada E-Learning.

Kata kunci: Perguruan Tinggi, Website, *Vulnerability assessment*, OWASP

ABSTRACT

The development of Information Technology at this time brings convenience to human life. One thing that is growing quite rapidly is web-based applications. At present the concept of E-Learning has been widely accepted by the world community, as evidenced by the many implementations of E-Learning in colleges and schools. Therefore, information system technology that exists at this time can meet the needs of the community directly. However, not a few parties take advantage of technological developments irresponsibly. Starting from cases of hacking, fraud, even cybercrime cases that not only attack individuals, but also companies/corporations and government agencies. This study aims to analyze and test security, vulnerabilities, on the STMIK Rosma Karawang E-Learning Website using the Vulnerability Assessment method and the website vulnerability scanning used is OWASP.

Keywords: *System Analysis, Security, Vulnerability assessment, OWASP*

1. PENDAHULUAN

Perkembangan Teknologi Informasi pada saat ini membawa kemudahan untuk semua orang. Aplikasi berbasis web dipilih karena aplikasi tersebut dapat berjalan di berbagai *platform* dan juga termasuk aplikasi yang ringan untuk digunakan. Seiring dengan semakin kompleksnya layanan dan aplikasi web dalam berbagai bidang, maka permintaan layanan web dari pengguna semakin meningkat. Salah satu yang menjadi kebutuhan saat ini adalah aplikasi pembelajaran daring e-learning. Aplikasi tersebut memiliki manfaat dalam proses belajar mengajar yang dilakukan secara *online*, seperti memperjelas informasi pada saat tatap muka, melengkapi dan mempermudah informasi dalam pembelajaran, meningkatkan efektivitas dan efisiensi dalam menyampaikan materi, dan mengatasi keterbatasan ruang dan waktu. Saat ini e-learning, sudah banyak diterima oleh masyarakat dunia, terbukti dengan banyaknya implementasi di perguruan tinggi dan sekolah(Budiman et al., 2021).
E-learning dapat berpotensi memiliki celah keamanan. Hal ini dikarenakan peningkatan penggunaan, dan manfaat yang dirasakan oleh pengguna, mendorong tindakan jahat muncul. Oleh sebab itu perlu adanya analisis terhadap

celah keamanan informasi pada aplikais e-learning tersebut. Menurut (Akmal et al., 2017) bahwa keamanan informasi adalah usaha untuk dapat mencegah penipuan (cheating) atau bisa mendeteksi adanya penipuan pada sistem yang berbasis informasi, dimana informasinya sendiri tidak memiliki artifisik. Menurut John, keamanan komputer adalah tindakan pencegahan dari serangan pengguna komputer atau pengaksesan jaringan yang tidak bertanggung jawab. Sedangkan menurut Gollman, keamanan komputer adalah berhubungan dengan pencegahan diri dan deteksi terhadap tindakan penganggu yang tidak dikenali dalam sistem komputer. Sedangkan menurut Garfinkel dan Spafford sebagai ahli keamanan komputer menyatakan bahwa komputer dikatakan aman jika bisa diandalkan dan perangkat lunaknya bekerja sesuai dengan yang diharapkan (Akmal et al., 2017).

Saat ini, e-learning banyak di sajikan dengan platform website, karena pertimbangan dalam implementasinya. Beberapa peneliti mengungkapkan untuk mengetahui celah keamanan pada sebuah sistem website dapat menggunakan metode *vulnerability assessment.* Hal ini dibuktikan oleh (Akmal et al., 2017)yang telah melakukan pengujian keaman sistem website Universitas Singaperbangsa Karawang. Selain itu penelitian yang dilakukan oleh (Ardita et al., 2022) juga menggunakan metode *Vulnerability assessment* untuk meningkatkan kualitas kemanan aplikasi android. Selain itu penelitian yang dilakukan oleh (Fronita, 2016) juga menggunakan metode *Vulnerability assessment* untuk celah keamanan website sitasi. Penelitian yang dilakukan oleh (Taryana et al., 2023) juga menggunakan metode *vulnerability assessment* untuk keamanan website BPJS kesehatan. Penelitian yang dilakukan oleh (Hendradi, 2022) juga menggunakan OWASP untuk mengetahui celah keamanan *e-learning* menggunakan *open web application.* Penelitian yang dilakukan oleh (Darwis et al., 2022) juga menggunakan OWASP untuk kerentanan website renovaction. Penelitian yang dilakukan oleh (Yudiana et al., 2021) juga menggunakan OWASP TOP 10 untuk kualitas keamanan sistem informasi *e-office*. Hal ini dapat disimpulkan bahwa vurnerability assessment menggunakan OWASP terbukti mampu menganalisis celah keamanan pada aplikasi berbasis website.

Vulnerability assessment adalah proses mendefiniskan, mengidentifikasi, dan mempriotaskan kerentanan dalam sistem komputer, aplikasi, dan infrastruktur jaringan dan memberikan organisasi melakukan penilaian dengan pengetahuan, kesadaran, dan latar belakang risiko yang diperlukan untuk memahami ancaman terhadap lingkungannya dan bereaksi dengan tepat. Proses vulnerability assessment yang dimaksudkan untuk mendefinisikan ancaman dan risiko yang ditimbulkannya biasanya melibatkan penggunaan alat penguji otomatis, seperti pemindaian keamanan jaringan, yang hasilnya terdaftar dalam laporan *vulnerability assessment* (Akmal et al., 2017). Penelitian ini dilakukan untuk mendapatkan informasi celah keamanan dan kerentanan pada *E-learning* yang telah di implementasikan di perguruan tinggi. Keterbaruan dari penelitian ini adalah penggunaan kerangka OWASP Top Ten yang khusus untuk website, sehingga mampu mengidentifikasi celah keamanan E-Learning dengan tepat.

2. KAJIAN PUSTAKA

2.1. E-learning

Proses pembelajaran secara daring dinilai lebih fleksibel, efisien, dan praktis dalam hal menyebarkan konten pembelajaran, serta fleksibel. Untuk dapat melakukan perkuliahan secara daring, dosen dan mahasiswa tidak perlu hadir dalam ruang kelas(Khairani Fadhilah, Ika Wulandari Utami Ningtias, 2016). Pembelajaran adalah kegiatan yang dilakukan pendidik dalam merangsang, membimbing, mengarahkan dan mendorong serta mengorganisir proses belajar peserta didik sehingga memiliki pengetahuan dan dapat mengembangkannya(Khairani Fadhilah, Ika Wulandari Utami Ningtias, 2016). Pembelajaran daring merupakan pembelajaran yang menggunakan jaringan internet dengan aksesibilitas, konektivitas, fleksibilitas, dan kemampuan untuk memunculkan berbagai jenis interaksi pembelajaran (Hadisi & Muna, 2015). Di Indonesia, *E-learning* telah diterapkan dibeberapa institusi akademik. Meningkatnya penggunaan internet sekitar 100% setiap tahun memberikan andil cukup besar dalam kemajuan penggunaan *E-learning*(Hadisi & Muna, 2015). Kelebihan *E-learning* ialah memberikan fleksibilitas, interaktivitas, kecepatan, visualisasi melalui berbagai kelebihan dari masing-masing media (Kuryanti & Sandra, 2016).

2.2. OWASP

The Open Worldwide Appliocation Project atau yang dikenal dengan OWASP merupakan nonprofit foundation yang bekerja untuk meninkatkan keamanan aplikasi. *Foundation* OWASP diluncurkan pada Desember 2021 yang berelasi dengan United State nonprofit charity pada tahun 2004(OWASP, 2021). OWASP dalam artikelnya merilis top 10 *web application security risk*, seperti yang terlihat pada gambar dibawah ini:

Gambar 1. Top 10 *web application security risk*(OWASP, n.d.-h)

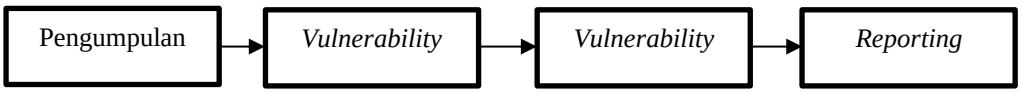
Kategori tersebut diantaranya adalah :
A01:2021 - *Broken Access Control*(OWASP, n.d.-a)
A02:2021 – *Cryptographic Failures*(OWAS, n.d.)
A03:2021 – *Injection*(OWASP, n.d.-b)
A04:2021 – *Insecure Design*(*A04:2021 – Insecure Design*, n.d.)
A05:2021 – *Security Misconfiguration*(*A05:2021 – Security Misconfiguration*, n.d.)
A06:2021 – *Vulnerable and Outdated Components*(OWASP, n.d.-c)
A07:2021 – *Identification and Authentication Failures*(OWASP, n.d.-d)
A08:2021 – *Software and Data Integrity Failures*(OWASP, n.d.-e)
A09:2021 – *Security Logging and Monitoring Failures*(OWASP, n.d.-f)
A10:2021 – *Server-Side Request Forgery (SSRF)*(OWASP, n.d.-g)

2.3. Vulnerability Assessment

Vulnerability Assessment adalah melakukan identifikasi *Vulnerability* dari suatu aplikasi sistem operasi dan infrastruktur jaringan. *Vulnerability Asesment* tidak melakukan celah atau kelemahan dari suatu sistem. Sedangkan *Vulnerability* adalah suatu kelemahan dalam desain sistem implementasi sistem atau operasi dan manajemen yang dapat dimanfaatkan untuk melanggar kebijakan keamanan sistem. *Vulnerability Assessment* lebih fokus untuk menemukan beragam *public Vulnerability* pada seluruh sistem komputer dalam jaringan target(Mulyanto et al., 2021). *Vulnerability Assessment* (VA) merupakan analisa keamanan menyeluruh dan mendalam seperti pada keamanan informasi, hasil scanning jaringan, cara pengelolaan, konfigurasi pada sistem, kesadaran keamanan aktor yang terlibat dan keamanan fisik, untuk mengetahui seluruh potensi kelemahan kritis yang ada(Wibowo et al., 2019).

3. METODE

Pada penelitian ini penulis mencari beberapa celah risiko keamanan website E-Learning, dengan cara mengumpulkan data untuk memperoleh hasil yang dicari, kemudian dilakukan *Scanning* sebagai langkah pengujian pada *E-Learning* yang sudah dipilih guna untuk mengetahui risiko kerentan keamanannya. Metode yang digunakan dalam penelitian ini adalah metode *Vulnerability Assessment* untuk mengetahui celah keamanan dan menentukan kualitas keamanan sistem serta dibantu *Vulnerability Scanning tools* OWASP. Penelitian ini dilakukan dengan beberapa tahapan agar lebih terstruktur dan mudah dipahami. Berikut adalah tahapan dari penelitian ini.



Gambar 2.Tahapan Penelitian

- Pengumpulan Data : dilakukan dengan cara mengamati serta mencatat secara sistematis tentang perangkat dan aplikasi yang digunakan dalam *Scanning* secara langsung untuk memenuhi kebutuhan penilaian kerentanan resiko (*vulnerability assessment*)
- *Vulnerability Assessment* : pada tahapan ini untuk mengidentifikasi adanya resiko celah kerentanan dalam pemindaian aplikasi
- *Vulnerability Scanning* : pada tahapan ini akan dilakukan proses *Scanning* untuk mengetahui resiko celah keamanan menggunakan *tools* OWASP.
- *Reporting* : pada tahapan ini merupakan *fase* terakhir untuk melaporkan dan mendokumentasikan bagian-bagian penting yang terjadi selama uji penetrasi untuk keamanan

4. HASIL DAN PEMBAHASAN

Pada tahapan ini, peneliti melakukan pengujian atau *Scanning* terhadap pontensi serangan keamanan yang terdapat pada *system website* pembalajaran *E-learning*. *Tools* yang akan digunakan adalah *Open Web Application Security Project* (OWASP). Dalam melakukan uji kerentanan peneliti menggunakan hasil *scanning* yaitu OWASP Zap

yang digunakan untuk mengetahui celah keamanan pada website dan menentukan kualitas keamanan sistem informasi *E-learning* Berikut adalah hasil *scaning* menggunakan Owasp Zap.

Tabel 1.*Scannig* Kerentanan

Vulnerability Scanning	***Alert Type***	***Risk***	***Category***
Owasp Zap	*Absence of Anti-CSRF Tokens*	*Medium*	*A01*
	Content Security Policy (CSP) Header Not Set	*Medium*	*A05*
	Big Redirect Detected (Potential Sensitive Information Leak)	*Low*	*A04*
	Cookie No HttpOnly Flag	*Low*	*A05*
	Cookie without SameSite Attribute	*Low*	*A01*
	Cross-Domain JavaScript Source File Inclusion	*Low*	*A08*
	Server Leaks Version Information via "Server" HTTP Response Header Field	*Low*	*A05*
	Strict-Transport-Security Header Not Set	*Low*	*A05*
	Timestamp Disclosure - Unix	*Low*	*A01*
	X-Content-Type-Options Header Missing	*Low*	*A05*
	Information Disclosure - Sensitive Information in URL	*Informational*	*A01*
	Information Disclosure - Suspicious Comments	*Informational*	*A01*
	Modern Web Application	*Informational*	-
	Re-examine Cache-control Directives	*Informational*	-
	User Controllable HTML Element Attribute (Potential XSS)	*Informational*	*A03*

Tabel 1 adalah hasil *scanning* pada *E-learning* dengan menampilkan hasil jumlah sebanyak 15 kerentanan peringatan risiko dan setiap jenis peringatannya memiliki beberapa tingkatan jenis risiko adapun jenis risiko diantaranya *medium, low, and informational.*

Tabel 2 Ringkasan Peringatan

		Confidence				
		User Confirmed	*High*	*Medium*	*Low*	Total
Risk	*High*	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	1 (6.7%)	0 (0.0%)	1 (6.7%)	2 (13.3%)
	Low	0 (0.0%)	2 (13.3%)	5 (33.3%)	1 (6.7%)	8 (53.3%)
	Informational	0 (0.0%)	0 (0.0%)	2 (13.3%)	3 (20.0%)	5 (33.3%)
	Total	0 (0.0%)	3 (20.0%)	7 (46.7%)	5 (33.3%)	15 (100%)

Dari hasil *scanning* pada tabel 2 menampilkan risiko dan kepercayaan diri oleh pengguna untuk *E-learning* dengan beberapa kategori risiko diantaranya *high, medium, low, informational,* dan total berdasarkan setiap kategorinya.

Tabel 3 pemberitahuan atau peringatan *(alerts)*

Vulnerability name	*Vulnerability impact*	*Preventif*
Absence of Anti-CSRF Tokens	*(1) Violation of the principle which allows access to be available to anyone, (2)Bypassing access control check done by modifying URL, internal application state atau HTML page or*	*To prevent it can be done with effectiveness control inside access trusted server-side cide or server-less API, where the attacker cannot*

	by using attck tool modifying API request, (3) Permiting viewing or editing someone else's account, (4) accessing API, (5) Elevation of privilege, (6) Metadata manipulation, (7) CORS misconfiguration allow API access form unauthorized, (8) Force browsing to authenticated page.	*modify the check or metadata control access. implementasi access control mechanisms one and re-use them throughout the application, including minimizing CORS usage.*

Risk=Medium, Confidence=High (1)

Dari tabel 3 kerentanan *Absence of Anti-CSRF Tokens* menampilkan hasil Risiko=Sedang, Keyakinan=Rendah (1) , dengan kata lain risiko yang terjadi menampilkan hasil sedang, sedangkan untuk keyakinan menampilkan hasil rendah.

Tabel 4 pemberitahuan atau peringatan *(alerts)*

Vulnerability name	*Vulnerability impact*	*Preventif*
Content Security Policy (CSP) Header Not Set	*(1) Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services, (2) Unnecessary features are enabled or installed such as ports, services, pages, accounts, or privileges, (3) Default accounts, (4) Error handling displays a stack trace or error message, (5)For upgraded systems, the latest security features are disabled or not configured securely, (6) The security settings in the application servers, application frameworks, (7) The server does not send security headers or directives, (8) The software is out of date or vulnerable.*	*A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.*

Risk=Medium, Confidence=Low (1)

Dari tabel 4 kerentanan *Content Security Policy (CSP) Header Not Set* menampilkan hasil Risiko=Sedang, Keyakinan=Tinggi (1), dengan kata lain risiko yang terjadi menampilkan hasil sedang, sedangkan untuk keyakinan menampilkan hasil tinggi

Tabel 5 pemberitahuan atau peringatan *(alerts)*

Vulnerability name	*Vulnerability impact*	*Preventif*
Server Leaks Version Information via "Server" HTTP Response Header Field	*(1) Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services, (2) Unnecessary features are enabled or installed such as ports, services, pages, accounts, or privileges, (3) Default accounts, (4) Error handling displays a stack trace or error message, (5)For upgraded systems, the latest security features are disabled or not configured securely, (6) The security settings in the application servers, application frameworks, (7) The server does not send security headers or directives, (8) The software is out of date or vulnerable.*	*A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.*
Strict-Transport-Security Header Not Set	*(1) Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services, (2) Unnecessary features are enabled or installed such as ports, services, pages, accounts, or privileges, (3) Default accounts, (4) Error handling displays a stack trace or error message, (5)For upgraded systems, the latest security features are disabled or not configured securely, (6) The*	*A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.*

	security settings in the application servers, application frameworks, (7) The server does not send security headers or directives, (8) The software is out of date or vulnerable.	

Risk=Low, Confidence=High (2)

Dari tabel 5 kerentanan *Server Leaks Version Information via "Server" HTTP Response Header Field*, *Strict-Transport-Security Header Not Set* menampilkan hasil Risiko = Rendah, Keyakinan = Tinggi (2), dengan kata lain risiko yang terjadi menampilkan hasil rendah, sedangkan untuk keyakinan menampilkan hasil tinggi.

Tabel 6 pemberitahuan atau peringatan *(alerts)***.**

Vulnerability name	*Vulnerability impact*	*Preventif*
Big Redirect Detected (Potential Sensitive Information Leak)	*Insecure design is a broad category representing different weaknesses, expressed as "missing or ineffective control design." Insecure design is not the source for all other Top 10 risk categories. Requirements and Resource Management, Compile the technical requirements, including functional and non-functional security requirements. Secure Design, Secure design is a culture and methodology that constantly evaluates threats and ensures that code is robustly designed and tested to prevent known attack methods. Secure Development Lifecycle, Secure software requires a secure development lifecycle.*	*Implement a secure development lifecycle with all AppSec professionals to help evaluate and design controls around security and privacy.*
Cookie No HttpOnly Flag	*(1) Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services, (2) Unnecessary features are enabled or installed such as ports, services, pages, accounts, or privileges, (3) Default accounts, (4) Error handling displays a stack trace or error message, (5)For upgraded systems, the latest security features are disabled or not configured securely, (6) The security settings in the application servers, application frameworks, (7) The server does not send security headers or directives, (8) The software is out of date or vulnerable.*	*A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.*
Cookie without SameSite Attribute	*(1) Violation of the principle which allows access to be available to anyone, (2)Bypassing access control check done by modifying URL, internal application state atau HTML page or by using attck tool modifying API request, (3) Permiting viewing or editing someone else's account, (4) accessing API, (5) Elevation of privilege, (6) Metadata manipulation, (7) CORS misconfiguration allow API access form unauthorized, (8) Force browsing to authenticated page.*	*To prevent it can be done with effectiveness control inside access trusted server-side cide or server-less API, where the attacker cannot modify the check or metadata control access. implementasi access control mechanisms one and re-use them throughout the application, including minimizing CORS usage.*
Cross-Domain JavaScript Source File Inclusion	*Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and content delivery networks (CDNs). An insecure CI/CD pipeline can introduce the potential for unauthorized access, malicious code, or system compromise.*	*(1) Use digital signatures or similar mechanisms to verify the software or data is from the expected source and has not been altered, (2) Ensure libraries and dependencies, such as npm or Maven, are consuming trusted repositories.*

X-Content-Type-Options Header Missing	*(1) Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services, (2) Unnecessary features are enabled or installed such as ports, services, pages, accounts, or privileges, (3) Default accounts, (4) Error handling displays a stack trace or error message, (5)For upgraded systems, the latest security features are disabled or not configured securely, (6) The security settings in the application servers, application frameworks, (7) The server does not send security headers or directives, (8) The software is out of date or vulnerable.*	*A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.*

Dari tabel 6 kerentanan *Big Redirect Detected (Potential Sensitive Information Leak), Cookie No HttpOnly Flag, Cookie without SameSite Attribute, Cookie without SameSite Attribute, Cross-Domain JavaScript Source File Inclusion, X-Content-Type-Options Header Missing,* menampilkan hasil Risiko=Rendah, Keyakinan=Sedang (5), dengan kata lain risiko yang terjadi menampilkan hasil rendah, sedangkan untuk keyakinan menampilkan hasil sedang.

Tabel 7 pemberitahuan atau peringatan *(alerts)***.**

Vulnerability name	*Vulnerability impact*	*Preventif*
Timestamp Disclosure - Unix	*(1) Violation of the principle which allows access to be available to anyone, (2)Bypassing access control check done by modifying URL, internal application state atau HTML page or by using attck tool modifying API request, (3) Permiting viewing or editing someone else's account, (4) accessing API, (5) Elevation of privilege, (6) Metadata manipulation, (7) CORS misconfiguration allow API access form unauthorized, (8) Force browsing to authenticated page.*	*To prevent it can be done with effectiveness control inside access trusted server-side cide or server-less API, where the attacker cannot modify the check or metadata control access. implementasi access control mechanisms one and re-use them throughout the application, including minimizing CORS usage.*

Risk=Low, Confidence=Low (1)

Dari tabel 7 kerentanan *Timestamp Disclosure - Unix* menampilkan Risiko=Rendah, Keyakinan=Rendah (1), dengan kata lain risiko yang terjadi menampilkan hasil rendah, sedangkan untuk keyakinan menampilkan hasil rendah.

Tabel 8 pemberitahuan atau peringatan *(alerts)***.**

Vulnerability name	*Vulnerability impact*	*Preventif*
Information Disclosure - Sensitive Information in URL	*(1) Violation of the principle which allows access to be available to anyone, (2)Bypassing access control check done by modifying URL, internal application state atau HTML page or by using attck tool modifying API request, (3) Permiting viewing or editing someone else's account, (4) accessing API, (5) Elevation of privilege, (6) Metadata manipulation, (7) CORS misconfiguration allow API access form unauthorized, (8) Force browsing to authenticated page.*	*To prevent it can be done with effectiveness control inside access trusted server-side cide or server-less API, where the attacker cannot modify the check or metadata control access. implementasi access control mechanisms one and re-use them throughout the application, including minimizing CORS usage.*
Modern Web Application	*is a modern web application. Being able to explore it automatically with Ajax Spider might be more effective*	*informational alerts so no changes needed.*

Risk=Informational, Confidence=Medium (2)

Dari tabel 8 kerentanan *Information Disclosure - Sensitive Information in URL, Modern Web Application* menampilkan Risiko=Informasi, Keyakinan=Sedang (2), dengan kata lain risiko yang terjadi menampilkan hasil informasi, sedangkan untuk keyakinan menampilkan hasil sedang.

Tabel 9 pemberitahuan atau peringatan *(alerts)***.**

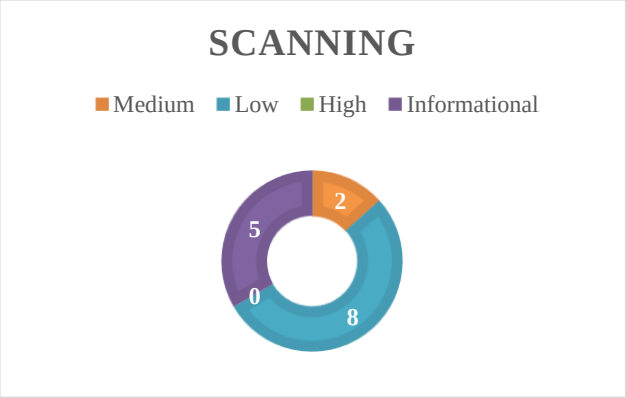
Vulnerability name	*Vulnerability impact*	*preventif*
Information Disclosure - Suspicious Comments	*(1) Violation of the principle which allows access to be available to anyone, (2)Bypassing access control check done by modifying URL, internal application state atau HTML page or by using attck tool modifying API request, (3) Permiting viewing or editing someone else's account, (4) accessing API, (5) Elevation of privilege, (6) Metadata manipulation, (7) CORS misconfiguration allow API access form unauthorized, (8) Force browsing to authenticated page.*	*To prevent it can be done with effectiveness control inside access trusted server-side cide or server-less API, where the attacker cannot modify the check or metadata control access. implementasi access control mechanisms one and re-use them throughout the application, including minimizing CORS usage.*
Re-examine Cache-control Directives	*The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content.*	*make sure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate".*
User Controllable HTML Element Attribute (Potential XSS)	*(1) User-supplied data is not validated, (2) Dynamic queries or non-parameterized, (3) Hostile data is used within object-relational mapping (ORM) search parameters to extract additional, sensitive records., (4) Hostile data is directly used or concatenated, contains malicious structures and data in dynamic queries, commands, or stored procedures*	*The preferred option is to use a safe API, which avoids using the interpreter entirely, provides a parameterized interface, or migrates to Object Relational Mapping Tools (ORMs).*

Risk=Informational, Confidence=Low(3)

Dari tabel 9 kerentanan *Information Disclosure - Suspicious Comments*, *Re-examine Cache-control Directives*, *User Controllable HTML Element Attribute (Potential XSS)* menampilkan Risiko = Informasi, Keyakinan = Rendah (3), dengan kata lain risiko yang terjadi menampilkan hasil informasi, sedangkan untuk keyakinan menampilkan hasil rendah.

Vulnerability Scanning

Pada tahapan terakhir dilakukan proses scanning *website E-learning* untuk mengetahui tingkat kerentanannya, serta dapat melihat hasil proses terakhir dengan tingkatan risiko yang berbeda-beda seperti *medium, low, high*, dan *informational* untuk hasil *scanning* ini bisa dijadikan bahan evaluasi terhadap keamanan *website E-Learning* dapat dilihat pada gambar berikut:



Gambar 3. Chart hasil scanning.

Dari gambar 2 menampilkan hasil akhir dari celah risiko kerentanan pada *E-learning* menggunakan *Vulnerability Scanning tools Open Web Application Security Project* Owasp Zap dengan jumlah risiko sebanyak 15 celah kerentanan dengan berbeda-beda risiko diantaranya:

- ***Medium :***
  Mendapat 2 risiko kerentanan pada E-learning dengan nama kerentanan *Absence of Anti-CSRF Tokens*, dan *Content Security Policy (CSP) Header Not Set.*
- ***Low :***
  Mendapat 8 risiko kerentanan pada E-learning dengan nama kerentanan *Big Redirect Detected (Potential Sensitive Information Leak), Cookie No HttpOnly Flag, Cookie without SameSite Attribute, Cross-*

Domain JavaScript Source File Inclusion, Server Leaks Version Information via "Server" HTTP Response Header Field, Strict-Transport-Security Header Not Set, Timestamp Disclosure – Unix, X-Content-Type-Options Header Missing.

- ***Informational :***
  Mendapat 5 risiko kerentanan pada E-learning dengan nama kerentanan Information *Disclosure - Sensitive Information in URL, Information Disclosure - Suspicious Comments, Modern Web Application, Re-examine Cache-control Directives, User Controllable HTML Element Attribute (Potential XSS).*
- ***High :***
  Sedangkan risiko kerentanan pada *E-learning* untuk hasil *high* tidak ada celah risiko kerentanan dengan mendapat 0 risiko.

5. KESIMPULAN

Berdasarkan hasil penelitian risiko celah keamanan pada *website E-learning* dapat diketahui risiko celah keamanan yang dapat dilihat dari hasil *vulnerability scanning tools* Owasp Zap. Hasil *Scanning* menunjukkan tingkatan yang berbeda-beda seperti medium, low, high, dan informational. Pada hasil *Scanning* yang telah di uji ditemukan adanya celah risiko kerentanan sebanyak 15 kerentanan, dan ditemukan 10 kategori yang berbeda seperti A01-A10 dengan mengambil kategori terbaru pada tahun 2021 top 10 web application. Dari hasil keseluruhan menggunakan owasp pada kerentanan tingkatan *High* mendapat hasil celah risiko kerentanan 0 atau tidak ada, sehingga penelitian ini tidak adanya celah risiko kerentanan yang paling tinggi.

DAFTAR PUSTAKA

A04:2021 – Insecure Design. (n.d.). Retrieved June 3, 2023, from https://owasp.org/Top10/A04_2021-Insecure_Design/

A05:2021 – Security Misconfiguration. (n.d.). Retrieved June 3, 2023, from https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

Akmal, A. M., Heryana, N., & Solehudin, A. (2017). Analisis Keamanan Website Universitas Singaperbangsa Karawang Menggunakan Metode Vulnerability Assessment. *Al-Irsyad*, *105*(2), 79.

Ardita, I. K. A. O., Anom Cahyadi Putra, I. G. N., Kustiadie, M. R., Dika Varuna, G. N. M., & Eka Prananda, M. Y. (2022). Analisis Keamanan Aplikasi Android Dengan Metode Vulnerability Assessment. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, *10*(3), 279. https://doi.org/10.24843/jlk.2022.v10.i03.p04

Budiman, A., Ahdan, S., & Aziz, M. (2021). Analisis Celah Keamanan Aplikasi Web E-Learning Universitas Abc Dengan Vulnerability Assesment. *Jurnal Komputasi*, *9*(2), 1–10.

Darwis, E., Junaedy, & Musdar, I. A. (2022). Analisis Kerentanan Website Renovaction Menggunakan Rangkaian Security Tools Project Berdasarkan Framework Owasp. *KHARISMA Tech*, *17*(1), 1–15. https://doi.org/10.55645/kharismatech.v17i1.170

Fronita, M. (2016). ANALISIS CELAH KEAMANAN WEBSITE SITASI MENGGUNAKAN VULNERABILITY ASSESSMENT. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, *9*(1), 1–23.

Hadisi, L., & Muna, W. (2015). Pengelolaan Teknologi Informasi Dalam Menciptakan Model Inovasi Pembelajaran (E-learning). *Jurnal Al-Ta'dib*, *8*(1), 117–140.

Hendradi, P. (2022). Analisis Keamanan E-learning Menggunakan Open Web Application Security Project (OWASP) studi kasus MOCA UNIMMA. *Jurnal Informatika*, *22*(02), 132–138.

Khairani Fadhilah, Ika Wulandari Utami Ningtias, F. D. (2016). Analisis Kebutuhan Multimedia Interaktif Perkuliahan E-Learning pada Mata Kuliah Landasan Kependidikan. *JURNAL BASICEDU*, *1*(9), 1–23.

Kuryanti, & Sandra, J. K. (2016). Rancang Bangun Sistem E-Learning sebagai Sarana Pemberlajaran Sandra. *Jurnal Khatulistiwa Informatika*, *4*(1), 84–92. https://doi.org/10.1089/pho.2010.2784

Mulyanto, Y., Haryanti, E., & Jumirah, J. (2021). Analisis Keamanan Website Sman 1 Sumbawa Menggunakan Metode Vulnerability Asesement. *Jurnal Informatika Teknologi Dan Sains*, *3*(3), 394–400. https://doi.org/10.51401/jinteks.v3i3.1260

OWAS. (n.d.). *A02:2021 – Cryptographic Failures*. Retrieved June 3, 2023, from https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

OWASP. (n.d.-a). *A01:2021 – Broken Access Control*. Retrieved June 3, 2023, from https://owasp.org/Top10/A01_2021-Broken_Access_Control/

OWASP. (n.d.-b). *A03:2021 – Injection*. Retrieved June 3, 2023, from https://owasp.org/Top10/A03_2021-Injection/

OWASP. (n.d.-c). *A06:2021 – Vulnerable and Outdated Components*. Retrieved June 3, 2023, from https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/

OWASP. (n.d.-d). *A07:2021 – Identification and Authentication Failures*. Retrieved June 4, 2023, from https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

OWASP. (n.d.-e). *A08:2021 – Software and Data Integrity Failures*. Retrieved June 4, 2023, from

https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/
OWASP. (n.d.-f). *A09:2021 – Security Logging and Monitoring Failures*. Retrieved June 4, 2023, from https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/
OWASP. (n.d.-g). *A10:2021 – Server-Side Request Forgery (SSRF)*. Retrieved June 4, 2023, from https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/
OWASP. (n.d.-h). *OWASP Top Ten*. Retrieved June 3, 2023, from https://owasp.org/www-project-top-ten/
OWASP. (2021). *OWASP*. https://owasp.org/about/
Taryana, Y., Heryana, N., Komputer, I., Karawang, U. S., & Sistem, K. (2023). *ANALISIS KEAMANAN WEBSITE BPJS KESEHATAN MENGGUNAKAN*. *8*(1), 31–37.
Wibowo, F., Harjono, H., & Wicaksono, A. P. (2019). Uji Vulnerability pada Website Jurnal Ilmiah Universitas Muhammadiyah Purwokerto Menggunakan OpenVAS dan Acunetix WVS. *Jurnal Informatika*, *6*(2), 212–217. https://doi.org/10.31311/ji.v6i2.5925
Yudiana, Y., Elanda, A., & Buana, R. L. (2021). Analisis Kualitas Keamanan Sistem Informasi E-Office Berbasis Website Pada STMIK Rosma Dengan Menggunakan OWASP Top 10. *CESS (Journal of Computer Engineering, System and Science)*, *6*(2), 185. https://doi.org/10.24114/cess.v6i2.24777