Jurnal Algoritma
P-ISSN: 1412-3622; E-ISSN:2302-7339
Sekolah Tinggi Teknologi Garut

Penetration Testing pada *Website* Universitas ARS Menggunakan *Open Web Application Security Project* (OWASP*)*

Syarif Hidayatulloh[1], Desky Saptadiaji[2]

Jurnal Algoritma
Sekolah Tinggi Teknologi Garut
Jl. Mayor Syamsu No. 1 Jayaraga Garut 44151 Indonesia
Email : jurnal@itg.ac.id

[1]syarif@ars.ac.id
[2]deskyysapta@gmail.com

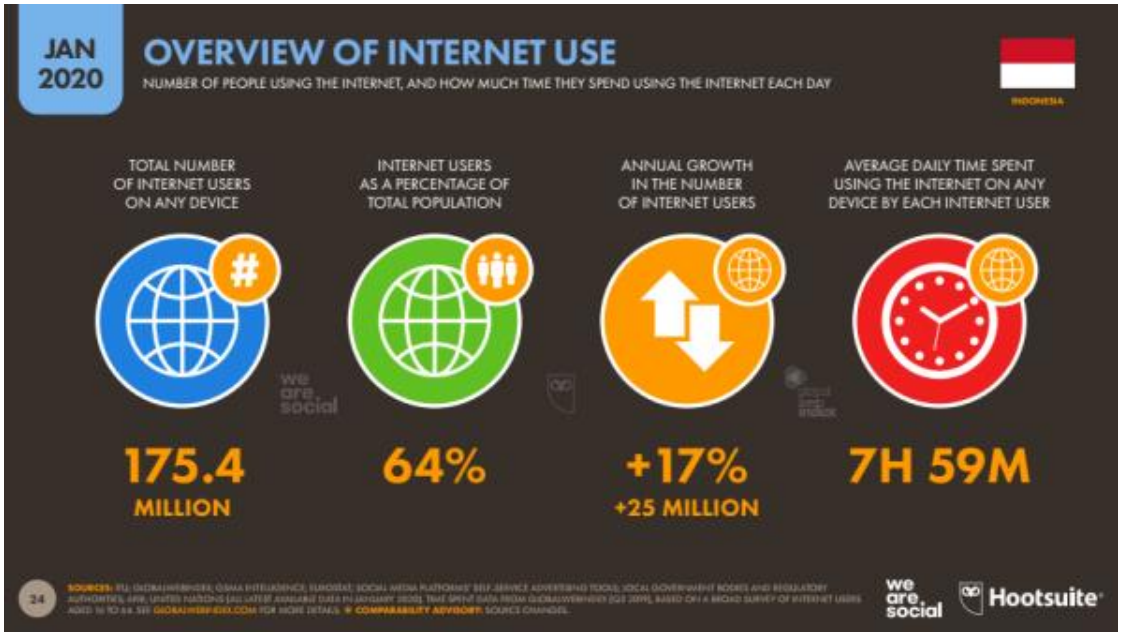
Abstrak **–** Universitas ARS adalah perguruan tinggi yang memanfaatkan website dalam melakukan kegiatan perkuliahannya. Seluruh informasi yang berkaitan dengan perkuliahan dimuat di website Universitas ARS. Banyak resiko yang akan terjadi apabila web server yang digunakan oleh website Universitas ARS tidak memiliki keamanan yang baik, banyak ancaman dari pihak yang tidak bertanggung jawab memanfaatkan celah keamanan untuk merugikan Universitas ARS. Tujuan penelitian ini adalah melakukan identifikasi kerentanan yang terdapat dalam website Universitas ARS dan melakukan pengujian serta analisis untuk mengetahui kondisi kerentanan website Universitas ARS menggunakan Open Web Application Security Project (OWASP). Metode penelitian yang digunakan sebagai parameter keamanan website adalah OWASP Top-10 2017. Jumlah subdomain yang diuji adalah 5 subdomain yang teridentifikasi dengan melakukan scanning menggunakan tool TheHarvester. Hasil dari penelitian ini adalah ditemukannya kerentanan website Universitas ARS yang berhasil dipindai adalah 13 kerentanan. Dari 13 kerentanan tersebut ada 1 kerentanan yang berada pada tingkat ancaman yang sedang dan 12 berada pada tingkat ancaman yang rendah. Berdasarkan seluruh pengujian kerentanan yang dilakukan dapat disimpulkan bahwa website Universitas ARS memiliki keamanan yang sangat baik, memenuhi ketiga aspek keamanan informasi, memiliki web server dan software sistem informasi akademik yang aman.

Kata Kunci **–** Domain; Kerentanan; OWASP; *Penetration Testing*; *Website*.

I. PENDAHULUAN

Di era Internet dan World Wide Web, keamanan sistem telah menjadi isu penting dalam sistem informasi berbasis web global. Hal ini dapat dilihat dari komitmen yang kuat dari para profesional keamanan sistem, komunitas riset, dan vendor perangkat lunak [1]. Menurut laporan digital yang dibuat oleh We Are Social (Hootsuite), pengguna internet di Indonesia pada awal Januari 2020 sudah mencapai 174 juta pengguna. Indonesia mengalami pertumbuhan pengguna yang mengakses internet sebesar 17 persen atau bertambah sebanyak 25 juta pengguna dalam satu tahun terakhir [2].

Article History: Recivied Dec 14, 2021 Revised Aug 15, 2021 Accepted Aug 16, 2021



Gambar 1: Jumlah Pengguna Internet di Indonesia menurut We Are Social dan Hootsuite pada Januari 2020
Sumber: [2]

Dengan semakin bertambahnya pengguna internet di Indonesia menandakan bahwa kini masyarakat Indonesia lebih banyak mendapatkan informasi melalui internet. Menurut [3] menjelaskan bahwa *Website* adalah metode untuk menampilkan informasi di internet, baik itu berupa teks, gambar, video & suara maupun interaktif memiliki keuntungan yang menghubungkan *(link)* dari dokumen dengan dokumen lainnya *(hypertext)* yang dapat diakses melalui *browser*.

Universitas ARS adalah perguruan tinggi yang memanfaatkan *website* dalam melakukan kegiatan perkuliahannya. Seluruh informasi yang berkaitan dengan perkuliahan dimuat di *website* Universitas ARS. Hal ini tentu sangat efektif dalam melakukan proses perkuliahan, karena mahasiswa ataupun dosen akan dimudahkan untuk mengakses informasi. Tetapi, banyak resiko yang akan terjadi apabila *web server* yang digunakan oleh *website* Universitas ARS tidak memiliki keamanan yang baik, banyak ancaman dari pihak yang tidak bertanggung jawab memanfaatkan celah keamanan untuk merugikan Universitas ARS.

G. J. Simons dalam [4] mengemukakan bahwa: Keamanan informasi adalah bagaimana usaha untuk dapat mencegah penipuan (*cheating*) atau bisa mendeteksi adanya penipuan pada sistem yang berbasis informasi, di mana informasinya sendiri tidak memiliki arti fisik. Aspek-aspek yang harus dipenuhi dalam suatu sistem untuk menjamin keamanan informasi adalah informasi yang diberikan akurat dan lengkap (*right information*), informasi dipegang oleh orang yang berwenang (*right people*), dapat diakses dan digunakan sesuai dengan kebutuhan (*right time*), dan memberikan informasi pada format yang tepat (*right form*).

Handisonj [5] menjelaskan bahwa CIA atau yang lebih sering disebut CIA Triad merupakan salah satu aturan dasar dalam menentukan keamanan suatu jaringan atau informasi. Parameter dalam CIA ini digunakan untuk menentukan apakah suatu jaringan atau informasi dikatakan aman atau tidak.

Di antara berbagai aspek keamanan siber, keamanan perangkat lunak memainkan peran penting. [6] Untuk mengamankan suatu *web server* dari serangan pihak yang tidak bertanggung jawab, maka sebaiknya *web server* tersebut harus di uji dengan melakukan *selftest* terhadap sistem *web server* itu sendiri dengan menggunakan metode *penetration testing*.

Mulyadi [7] menjelaskan bahwa *Penetration testing* adalah serangkaian proses berisi prosedur dan teknik mengevaluasi keamanan terhadap sistem komputer atau jaringan dengan melakukan simulasi penyerangan untuk mengetahui letak celah-celah kerawanan pada sistem agar kemudian celah tersebut ditutup atau diperbaiki. *Penetration testing* dilakukan sebagai langkah *preventive* untuk mengatasi terjadinya peretasan pada suatu sistem.

Baloch dan Rafay [8] menjelaskan bahwa *Penetration testing* adalah serangkaian metode dan prosedur yang dilakukan dengan tujuan untuk menguji atau melindungi keamanan suatu organisasi. *Penetration testing* membantu untuk menemukan kerentanan yang ada dalam suatu organisasi dan memeriksa apakah penyerang akan dapat mengeksploitasi hingga mendapatkan akses yang tidak sah.

Meucci dan Matteo [9] menjelaskan bahwa *Penetration testing* adalah teknik umum yang digunakan untuk menguji keamanan jaringan. Pada dasarnya *penetration testing* adalah pengujian aplikasi yang aktif untuk menemukan kerentanan keamanan, tanpa mengetahui cara kerja aplikasi tersebut. Pelaku *penetration testing* bertindak sebagai penyerang dan berupaya untuk menemukan dan mengeksploitasi kerentanan.

Engebretson dan Patrick [10] menjelaskan bahwa *Penetration testing* adalah upaya yang dilakukan secara sah untuk mengeksploitasi sistem komputer dengan tujuan membuat sistem tersebut menjadi aman. *Penetration testing* yang dilakukan dengan baik dapat menghasilkan rekomendasi untuk mengatasi dan memperbaiki masalah yang ditemukan selama pengujian.

Baloch dan Rafay [8] menjelaskan bahwa jenis *penetration testing* terbagi menjadi 2, yaitu:

1. *Black Box Testing* : Pengujian ini dilakukan oleh pihak yang tidak memiliki informasi sama sekali tentang sistem operasi, versi *server* atau jaringan yang terdapat di dalam sistem, sehingga pelaku *penetration testing* harus mencari segala informasi yang dibutuhkan untuk melakukan pengujian terhadap sistem tersebut.
2. *White Box Testing* : Pengujian ini dilakukan oleh pihak yang telah diberikan seluruh informasi tentang sistem operasi, versi *server* atau jaringan yang digunakan dengan tujuan untuk menemukan kerentanan yang ada sehingga dapat langsung diperbaiki oleh pihak pengelola suatu organisasi

Penetration testing terhadap *website* dapat dilakukan dengan berbagai cara, salah satunya menggunakan parameter keamanan yang dibuat oleh organisasi *OWASP* [11]. *OWASP* adalah komunitas terbuka yang memungkinkan organisasi untuk mengembangkan, membeli dan memelihara aplikasi yang dapat dipercaya. *OWASP* tidak terafilisi dengan perusahaan manapun, dan merupakan komunitas non-profit yang memastikan kesuksesan jangka panjang proyek. Hampir semua yang terasosiasi dengan *OWASP* adalah sukarelawan [12].

OWASP Top 10 adalah sebuah daftar yang dibuat oleh komunitas *OWASP* yang berisi 10 daftar teratas kerentanan yang dapat mengancam keamanan suatu *website*. Bertujuan untuk membuat keamanan perangkat lunak terlihat oleh individu dan organisasi untuk membantu mereka mengambil keputusan yang tepat tentang risiko keamanan perangkat lunak mereka [13].

Pada penelitian sebelumnya metode yang digunakan untuk *penetration testing* adalah *framework OWASP* versi 4 dengan kolaborasi beberapa *tools security project* dengan objek yang diteliti adalah *website* www.xyz.com. Pada penelitian kedua metode yang digunakan untuk *penetration testing* adalah *OWASP* 10 tahun 2013. *Website vulnerability scanning tools* yang digunakan adalah *WPScan* dan *OWASPZAP* dengan objek yang diteliti adalah website Universitas Islam Indonesia.

Pada penelitian ketiga metode yang digunakan untuk *penetration testing* adalah *framework Common Vulnerability Scoring System (CVSS). Website vulnerability scanning tools* yang digunakan adalah *NMAP, OWASPZAP,* dan *Accunetix* dengan objek yang diteliti adalah *website* Asosiasi Pekerja Professional Informasi Sekolah Indonesia (APISI).

Pada penelitian keempat metode yang digunakan untuk *penetration testing* adalah *OWASP* versi 4. *Website Vulnerability scanning tools* yang digunakan adalah *Accunetix* dengan objek yang diteliti adalah aplikasi ulangan tengah semester dan ulangan akhir semester yang dinamai aplikasi "Si Ujo".

Berdasarkan penjelasan tersebut penulis akan melakukan penelitian *penetration testing* pada website Universitas ARS menggunakan *OWASP*. Perbedaan penelitian ini dengan penelitian sebelumnya diantaranya adalah objek yang diteliti, versi *OWASP* menggunakan *OWASP* 2017 dan *website vulnerability scanning tools* yang digunakan adalah *OWASPZAP, Uniscan dan Nikto*.

II. URAIAN PENELITIAN

Penelitian ini menggunakan data dan informasi yang diperoleh dari beberapa sumber yang berkaitan dengan *penetration testing* seperti jurnal, karya ilmiah, buku, tugas akhir atau melalui media internet. Parameter yang digunakan dalam pengujian kerentanan *website* Universitas ARS adalah *OWASP Top* 10-2017. Berikut adalah 10 daftar kerentanan yang terdapat di dalam *OWASP Top* 10 2017:

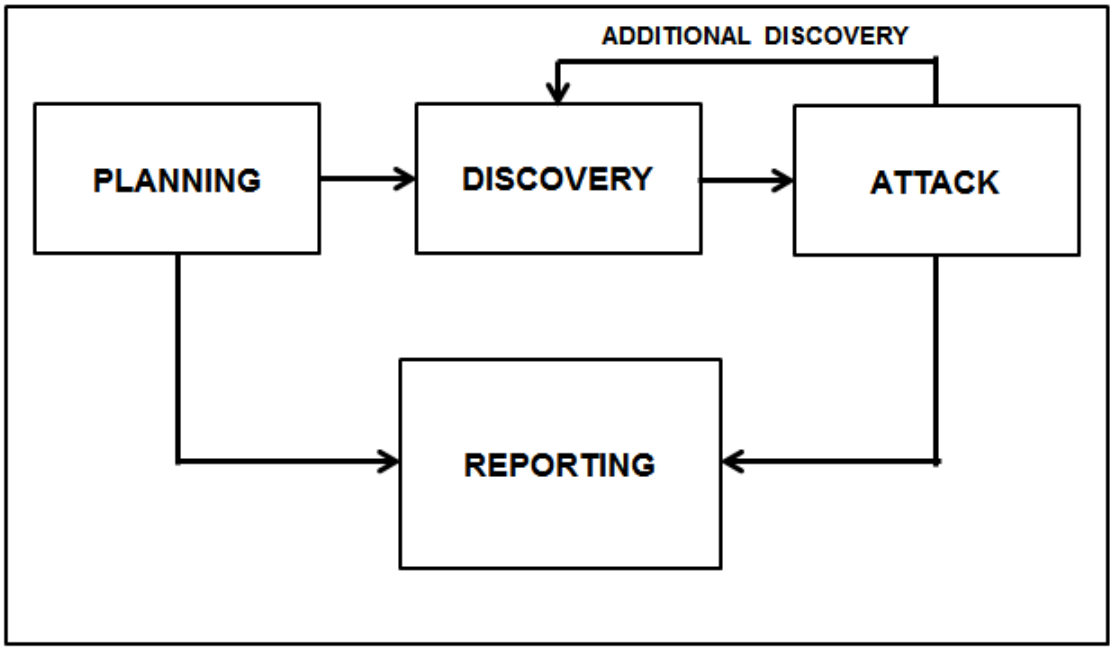
1. *Injection* : *Injection* merupakan serangan yang dilakukan dengan cara menginjeksi *script* ke dalam suatu *website.*
2. *Broken Authentication* : Kelemahan pada sistem login bisa memberikan celah keamanan bagi seorang *hacker* melakukan akses ke dalam akun pengguna.
3. *Sensitive Data Exposure* : Ancaman keamanan ini terjadi ketika *website* tidak melindungi dengan baik mengenai informasi yang sensitif atau sangat penting.
4. *XML External Entities (XEE)* : Serangan XML adalah jenis serangan terhadap *website* dengan mem-parsing input XML.
5. *Broken Access Control* : Kontrol akses ini mengacu kepada sistem kontrol yang mengakses informasi dan fungsionalitasnya.
6. *Security Misconfiguration* : Kesalahan konfigurasi keamanan dapat didefinisikan sebagai kegagalan untuk mengimplementasikan semua kontrol keamanan untuk sebuah *website*.
7. *Cross-Site Scripting (XSS)* : Serangan ini termasuk serangan jenis injeksi, dimana *malicious code* di injeksikan ke dalam suatu *website*.
8. *Insecure Deserialization* : *Insecure Deserialization* merupakan kerentanan terhadap data yang tidak dapat dipercaya atau tidak dikenal yang digunakan untuk menimbulkan serangan *Denial of Service* (DOS), *execute code, bypass authentication.*
9. *Using Components with Known Vulnerabilities* : Beberapa *hacker* biasa mencari kelemahan pada komponen seperti *libraries* dan *framework* yang digunakan pada *website* agar mereka dapat melakukan penyerangan.
10. *Insufficient Logging & Monitoring* : Merupakan kerentanan yang terjadi ketika serangan tidak dicatat dengan benar dan sistem tidak memantau kejadian.

A. Metode Pengumpulan Data

Adapun metode lain yang digunakan pada penelitian ini guna menambah informasi mengenai kerentanan *website* Universitas ARS diantaranya:

1. Observasi : Observasi dilakukan secara langsung pada objek penelitian yaitu *website* Universitas ARS.
2. Studi pustaka : Mempelajari dan mengumpulkan pengetahuan yang berkaitan dengan *penetration testing* yang berupa dokumen skripsi, jurnal dan buku yang ada di internet.

B. Tahapan *Penetration Testing*



Gambar 2: Tahapan *Penetration Testing*
Sumber: [8]

Tahapan *penetration testing* yang pertama adalah perencanaan *(planning).* Tahap ini termasuk juga ke dalam tahap penemuan *(discovery)* yang dibagi menjadi dua bagian, yang pertama adalah bagian pengumpulan informasi, pemindaian jaringan, identifikasi layanan dan mendeteksi sistem operasi dan bagian yang kedua adalah penilaian kerentanan yang terdapat di dalam sistem.

Kemudian tahap selanjutnya adalah tahap serangan *(attack)* yang merupakan inti dari sebuah *penetration testing.* Apabila serangan berhasil dilakukan dan dapat menemukan kerentanan lainnya yang terkoneksi, maka kembali ke tahap penemuan *(discovery)* dan dulakukan sampai tidak ada lagi kerentanan yang tersisa. Tahap terakhir adalah laporan *(reporting)* yang berisi hasil *penetration testing* yang sudah dilakukan [8].

III. PETUNJUK TAMBAHAN

Pada bagian ini akan dibahas mengenai hasil yang diperoleh dari tahapan *penetration testing* yang diantaranya adalah *scanning, discovery and attack*.

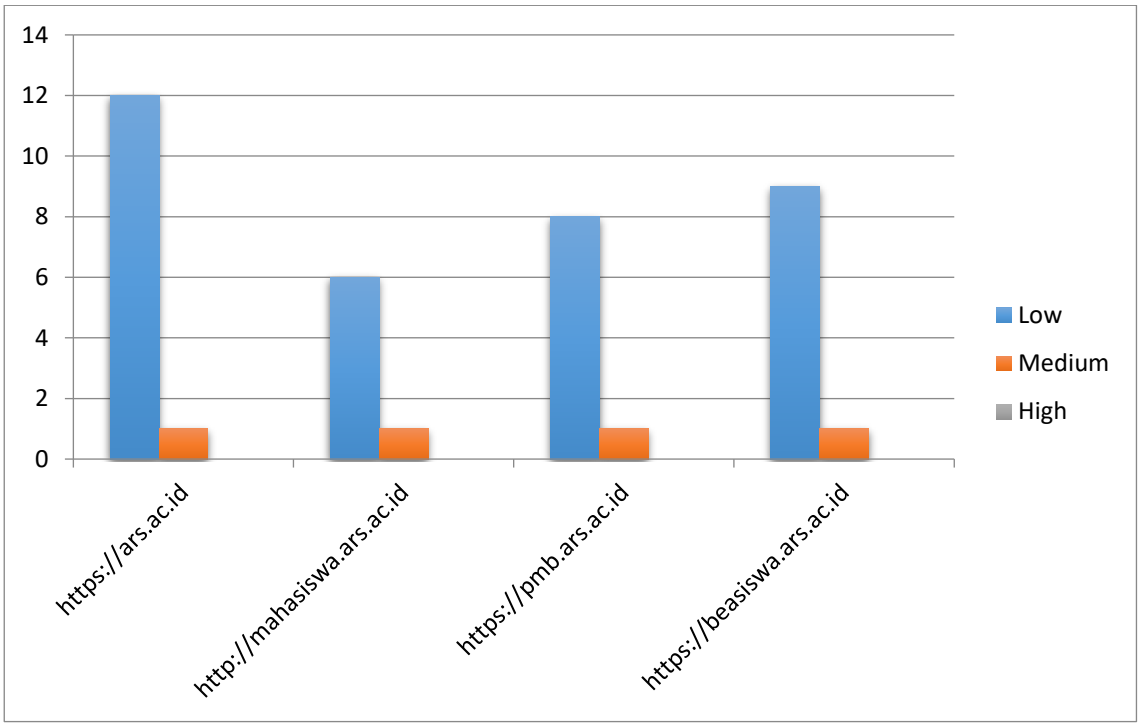
A. *Scanning and Discovery*

Tahapan yang pertama dilakukan adalah proses *planning* dan proses *discovery*. Dari proses yang dilakukan penulis mendapatkan banyak informasi mengenai *website* Universitas ARS dengan menggunakan *tools information gathering* diantaranya [12]:

1. *Whois* : Informasi yang di dapatkan adalah masa berlaku *domain*, lokasi pengelola *domain*, kontak yang tersedia pada *domain*, dan server yang digunakan.
2. *Host* : Informasi yang didapatkan adalah *mail server* yang menangani domain.
3. *theHarvester* : Informasi yang didapatkan adalah *email*, jumlah *subdomain* dan *ip address domain*.
4. *Ping* : Informasi yang didapatkan adalah memastikan *domain* dapat di akses dari luar.
5. *Whatweb* : Informasi yang didapatkan adalah *plugin* yang digunakan untuk membangun website pada seluruh domain.
6. *Nmap* : Informasi yang didapatkan adalah jenis sistem operasi yang digunakan, *port* yang terbuka dan

versi layanan yang digunakan pada *domain*.

7. *OWASPZAP* : Informasi yang didapatkan adalah kerentanan yang terdapat pada *website* Universitas ARS dengan melakukan *scanning* menggunakan *proxy OWASPZAP.*
8. *Uniscan* : Informasi yang didapatkan adalah kerentanan yang terdapat pada *website* Universitas ARS.
9. *Nikto* : Informasi yang didapatkan adalah kerentanan yang terdapat pada *website* Universitas ARS.



Gambar 3: Garfik Jumlah Kerentanan Website

Dari grafik di atas menunjukkan bahwa kelima *subdomain* tidak memiliki ancaman kerentanan yang *high*. Yang paling banyak memiliki kerentanan adalah *domain* https://ars.ac.id dengan jumlah 12 kerentanan pada tingkat ancaman *low* dan 1 kerentanan pada tingkat ancaman *medium,* kemudian *domain* https://beasiswa.ars.ac.id dengan jumlah 9 kerentanan pada tingkat ancaman *low* dan 1 kerentanan pada tingkat ancaman *medium,* lalu *domain* https://pmb.ac.id dengan jumlah 8 kerentanan pada tingkat ancaman *low* dan 1 kerentanan pada tingkat ancaman *medium,* lalu *domain* http://ejurnal.ars.ac.id dengan jumlah 7 kerentanan pada tingkat ancaman *low* dan 1 kerentanan pada tingkat ancaman *medium,* dan terakhir *domain* http://mahasiswa.ars.ac.id dengan jumlah 6 kerentanan pada tingkat ancaman *low* dan 1 kerentanan pada tingkat ancaman *medium.*

B. *Attack*

Setelah selesai melakukan proses perencanaan dan proses penemuan, tahapan selanjutnya adalah tahapan inti dari *penetration testing* yaitu melakukan penyerangan. Penyerangan dilakukan terhadap kerentanan yang telah ditemukan pada tahap sebelumnya. Berikut adalah hasil pengujian yang sudah dilakukan:

Tabel 1 : Tabel Hasil Pengujian

No	Jenis Ancaman	Hasil Pengujian	Rekomendasi
1	*SQL Injection*	Tidak berhasil.	-
2	*Cross-Site Scripting (XSS)*	Tidak berhasil.	-

3	*X-Frame-Options Header Not Set*	Berhasil, menggunakan *iframe* pada halaman *web* lain.	Menggunakan Content-Security-Policy: frame-ancestors 'none'; yang dapat mencegah domain apapun membuat *framing.*
4	*Cookie No HttpOnly Flag*	Berhasil, *Header* terbukti tidak memiliki *cookie HttpOnly.*	Sertakan *HttpOnly* ke dalam HTTP *Header Response,* sehingga cookie tidak dapat diakses melalui skrip klien.
5	*Cookie Without Secure Flag*	Berhasil, *Header* terbukti tidak memiliki *cookie secureflag.*	Sertakan *secure flag* ke dalam HTTP *Header Response*, sehingga *browser* tidak akan mengirim *cookie* yang di kirim melalui permintaan HTTP yang tidak terenkripsi.
6	*X-Content-Type-Options Header Missing*	Berhasil, *header X-Content-Type-Options* tidak ditemukan.	Tambahkan header *X-Content-Type-Options* dengan nilai "nosniff". X-Content-Type-Options: nosniff agar terhindar dari serangan *sniffing.*
7	*Cross-Domain JavaScript Source File Inclusion*	Berhasil, halaman terbukti memuat *script* dari *domain* pihak ketiga.	Pastikan *file* sumber *javascript* hanya berasal dari sumber yang terpercaya dan sumber tidak dapat dikontrol oleh *end-user application.*
8	*Incomplete or No Cachecontrol and Pragma HTTP Header Set*	Berhasil, *Header* terbukti tidak menggunakan *cache-control* dan *pragma-control.*	Gunakan Cache-Control: Max-Age untuk menentukan masa berlaku *cache*. Setelah kedaluwarsa *browser* harus menyegarkan versi *cache.*
9	*Cookie Without SameSite Attribute*	Berhasil, *Header* terbukti tidak menggunakan atribut *SameSite*	Gunakan atribut *SameSite* sehingga *browser* dapat memberitahu kapan dan bagaimana mengaktifkan *cookie* dari pihak kedua atau pihak ketiga.
10	*Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)*	Berhasil, *Header memuat X-Powered-By.*	Pastikan *web server, application server, load balancer* dikonfigurasi untuk disembunyikan dari *header*.
11	*Absence of Anti-CSRF Tokens*	Berhasil, token *Anti-CSRF* tidak digunakan pada saat *login.*	Gunakan Anti-*CSRF* pada *form login*.
12	*The X-XSS-Protection header is not defined.*	Berhasil, *header X-XSS-Protection* tidak ditemukan.	Gunakan X-XSS-Protection: 1; mode=block untuk melindungi *website* dari serangan *XXS.*

13	*The site uses SSL and the Strict-Transport-Security HTTP header is not defined.*	Berhasil, HSTS tidak ditemukan.	Gunakan HSTS untuk melindungi pengunjung web tidak dialihkan dari HTTPS ke HTTP. Contohnya https://ars.ac.id dialihkan ke http://ars.ac.id yang misalnya http://ars.ac.id memuat web yang berbahaya, sehingga terjadi kelemahan *man-in-the-middle attack.*
14	*The site uses SSL and Expect-CT header is not present*	Berhasil, *Header Certificate Transparency* tidak ditemukan.	Gunakan CT *header* untuk mencegah penyalahgunaan *certificates.*
15	*ZAP Spidering*	Berhasil, ditemukan banyak *link* yang berisi untuk mengunduh sertifikat seminar, dan *method GET & POST* pada saat mencoba melakukan *login* terlihat informasinya.	Pastikan *admin web* memperbarui atau menyembunyikan *link* yang sudah tidak digunakan untuk tidak ditampilkan di direktori.

C. *Reporting*

Tahap terakhir dalam melakukan *penetration testing* adalah *reporting.* Dengan menggunakan *OWASP Top* 10-2017 sebagai parameter penelitian *penetration testing* ini maka hasil dari penelitian yang dilakukan adalah seperti pada Tabel 2. di bawah ini:

Tabel 2: Daftar Kerentanan *OWASP Top*-10 2017 dan Hasil Pengujian

No	Nama Kerentanan	Celah Keamanan	Hasil Pengujian
1	A1:2017-Injection	Tidak Ditemukan	Tidak Berhasil
2	A2:2017-Broken Authentication	Tidak Ditemukan	Tidak Berhasil
3	A3:2017-Sensitive Data Exposure	Tidak Ditemukan	Hanya mendapatkan data yang tidak terlalu sensitif.
4	A4:2017-*XML* External Entities (XXE)	Tidak Ditemukan	-
5	A5:2017-Broken Access Control	Tidak Ditemukan	Tidak Berhasil
6	A6:2017-Security Misconfiguration	Tidak Ditemukan	-
7	A7:2017-*Cross-Site Scripting* (XSS)	Tidak Ditemukan	Tidak Berhasil
8	A8:2017-Insecure Deserialization	Tidak Ditemukan	-
9	A9:2017-Using Components with Known Vulnerabilities	Tidak Ditemukan	-
10	A10:2017-Insufficient Logging & Monitoring	Tidak Ditemukan	-

Kemudian berdasarkan aspek keamanan CIA TRIAD yaitu *confidentiality, integration* dan *availability* menunjukkan bahwa *website* Universitas ARS memiliki keamanan yang baik dalam ketiga aspek tersebut,

tetapi hanya pada aspek keamanan *integration* sedikit kurang terpenuhi karena *Header X-Frame-Options* tidak diatur pada masing-masing *domain* sehingga *frame* dapat mudah dimuat di dalam *website* lain yang mengganggu keutuhan informasi.

Keamanan *website* yang dimiliki Universitas ARS sangat baik karena menggunakan *server NiagaHoster*. *NiagaHoster* memiliki banyak kelebihan, terutama adalah menggunakan LiteSpeed Http yang membuat *website* sangat responsif dan cepat diakses. Kemudian memiliki keamanan yang sangat baik yang dapat mencegah serangan *DDOS*, *malware* dan serangan jenis lainnya dengan menggunakan perlindungan *Imunify360*. Selain itu, *NiagaHoster* menggunakan *Green Data Center Tier-4* DCI Indonesia berstandar internasional. *Data center* yang digunakan merupakan bagian dari Equinix, penyedia *data center* berkualitas tinggi terbaik di dunia dengan jaminan uptime hingga 99,98%.

Universitas ARS juga memiliki keamanan sistem informasi akademik yang baik dengan menggunakan SIAKAD *Cloud*. SIAKAD *Cloud* merupakan sebuah layanan *Software As Services (SaaS)*. Sistem Informasi Manajemen Akademik terintegrasi yang diperuntukkan universitas atau perguruan tinggi untuk membantu kegiatan operasional akademik perkuliahan mulai dari penerimaan mahasiswa sampai kelulusan dan semuanya telah terintegrasi dengan PDDIKTI.

IV. KESIMPULAN

Setelah melakukan *penetration testing* pada *website* Universitas ARS dengan menggunakan *OWASP Top*-10 2017 sebagai parameter pengujian kerentanan yang dimiliki *website* Universitas ARS, dapat diambil beberapa kesimpulan diantaranya:

1. Jumlah *subdomain* yang diuji adalah 5 *subdomain* yang teridentifikasi dengan melakukan *scanning* menggunakan *tool TheHarvester* diantaranya adalah https://universitas.ars.ac.id, http://mahasiswa.ars.ac.id, https://pmb.ars.ac.id, https://beasiswa.ars.ac.id dan http://ejurnal.ars.ac.id.Jumlah kerentanan *website* Universitas ARS yang berhasil dipindai adalah 13 kerentanan yang dapat di uji. Dari 13 kerentanan tersebut ada 1 kerentanan yang berada pada tingkat ancaman yang sedang, yaitu *X-Frame-Options Header Not* Set dan 12 kerentanan yang berada pada tingkat ancaman yang rendah yang diantaranya adalah *X-Frame-Options Header Not Set, Cookie No HttpOnly Flag, Cookie Without Secure Flag, X-Content-Type-Options Header Missing, Cross-Domain JavaScript Source File Inclusion, Incomplete or No Cache-control and Pragma HTTP Header Set, Cookie Without SameSite Attribute, Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s), Absence of Anti-CSRF Tokens, Blind SQL Injection, The X-XSS-Protection header is not defined, The site uses SSL and the Strict-Transport-Security HTTP header is not defined* dan *The site uses SSL and Expect-CT header is not present.*
2. Keamanan *website* Universitas ARS sudah baik karena berdasarkan aspek keamanan informasi CIA TRIAD yaitu *confidentiality*, *integrity* dan *availability* terpenuhi, tetapi hanya pada aspek keamanan *integration* sedikit kurang terpenuhi karena *Header X-Frame-Options* tidak diatur pada kelima *subdomain* sehingga *frame* dapat mudah dimuat di dalam *website* lain yang mengganggu keutuhan informasi. *Website server* yang digunakan adalah *NiagaHoster* dan *software* sistem informasi akademik yang digunakan adalah SIAKAD *Cloud* yang membuat *website* Universitas ARS memiliki keamanan yang baik.

DAFTAR PUSTAKA

[1] K. A. Sedek, N. Osman, M. N. Osman, and H. K. Jusoff, “Developing a Secure Web Application Using OWASP Guidelines,” *Comput. Inf. Sci.*, 2009, doi: 10.5539/cis.v2n4p137.

[2] B. Ramadhan, “Data Internet di Indonesia dan Perilakunya Tahun 2020,” *Teknoia*, 2020. https://teknoia.com/data-internet-di-indonesia-dan-perilakunya-880c7bc7cd19 (accessed Feb. 16, 2020).

[3] Yuhefizar, *Mudah Membuat Web Profil Multibahasa*. Bantul: Elex Media Komputindo, 2013.

[4] E. Purwanto, “Keamanan Informasi,” *BPPTIK*, 2014.

https://bpptik.kominfo.go.id/2014/03/24/404/keamanan-informasi/.
[5] Handisonj, "CIA (Confidentiality, Integrity, Availability)," *Handisonj*, 2013. https://handisonj.wordpress.com/2013/09/16/cia-confidentiality-integrity-availability/.
[6] B. Elisa, F. Peitro, and S. Fausto, "Security Analysis of the OWASP Benchmark with Juliale," 2017.
[7] Mulyadi, "Bagaimana Melakukan 'Penetration Test'?," *Kompasiana.com*, 2018. https://www.kompasiana.com/moengil/5a4ae2655e13736b135dd7e3/bagaimana-melakukan-penetration-testing.
[8] R. Baloch, *Ethical Hacking and Penetration Testing Guide*. 2014.
[9] M. Meucci, *OWASP Testing Guide v4*. The OWASP Foundation, 2014.
[10] P. Engebretson, "The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy," *Vasa*, 2011.
[11] M. B, "Evaluation of Web Vulnerability Scanners Based on OWASP Benchmark," 2018.
[12] F. E. Cerullo, "OWASP TOP 10 2009," 2010.
[13] D. Pandya and N. Patel, "Owasp top 10 vulnerability analyses in government websites," *Int. J. Enterp. Comput. Bus. Syst.*, 2016.