ANALISIS KERENTANAN APLIKASI BERBASIS WEB MENGGUNAKAN KOMBINASI *SECURITY TOOLS PROJECT* BERDASARKAN *FRAMEWORK* OWASP VERSI 4

Moh Yunus
Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
sunuy165@staff.gunadarma.ac.id

Abstrak

Keamanan informasi merupakan hal penting yang harus diperhatikan bagi setiap individu maupun instansi supaya terhindar dari tindakan kejahatan. Sistem informasi yang kurang baik dapat mengancam infrastruktur penting suatu organisasi. Masalah kerentanan atau gangguan keamanan sistem banyak bertebaran di internet. Masalah tersebut dapat berupa serangan Malware, Eksploitasi, atau Injeksi database. Solusi pengamanan web dari gangguan atau serangan hacker dapat dilakukan dengan cara self test yaitu pengujian yang dilakukan terhadap web secara legal dengan aktifitas menyerupai hacker. Deteksi sejak dini kelemahan suatu sistem merupakan solusi awal dalam pengamanan suatu sistem. Oleh karena itu dibutuhkan sebuah analisis terhadap kerentanan sebuah sistem yang mengacu kepada standarisasi kemanan Open Web Application Security Project (OWASP) Versi 4 dengan kombinasi beberapa tools security. Analisis kerentanan aplikasi berbasis web dengan teknik OWASP versi 4 dengan beberapa bantuan tools security mampu mengetahui tingkat keamanan suatu aplikasi berdasarkan hasil pengujian yang telah dilakukan dimana hampir setiap kategori pengujian mampu menemukan kerentanan, meskipun ada beberapa kategori yang tidak ada celah kerentanan

Kata Kunci: *OWASP version 4, penetration testing, SQL-Injection, vulnerability assessment, XSS.*

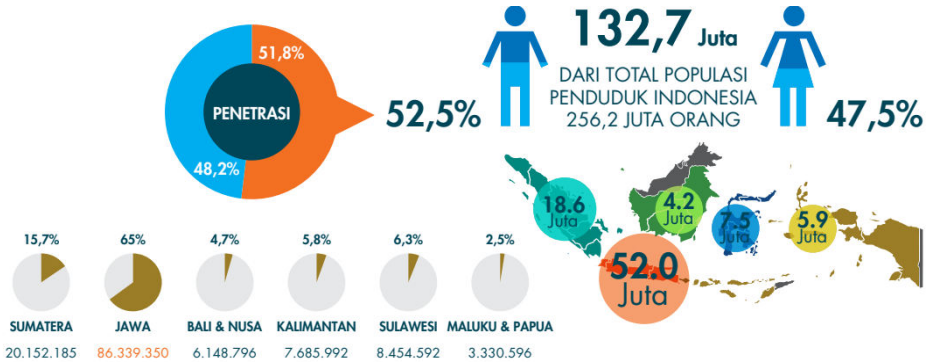
Abstract

Information security is important to be aware of every individual or institution in order to avoid any crime. Poor information systems can threaten an organization's essential infrastructure. Problems with vulnerability or system security are all around the internet. These problems can be Malware attacks, exploits, or database injections. A web security solution from intrusion or hacker attacks can be done by self test that is a test done against the web legally with a hacker-like activity. Early detection of a system weakness is the initial solution in securing a system. Therefore, it takes an analysis of the vulnerability of a system that refers to the standardization of Open Web Application Security Project (OWASP) version 4 with a combination of several Security tools. Analysis of Web-based application vulnerabilities with the technique of OWASP version 4 with some help of security tools is capable of knowing the security level of an application based on the results of tests that have been done where almost every category of testing is able Find vulnerabilitie, although there are some categories that have no gaps in vulnerability.

Keywords*: OWASP Versi 4, penetration testing, SQL-Injection, vulnerability assessment, XSS.*

PENDAHULUAN

Statistik Pengguna Internet di Indonesia Berdasarkan hasil survei yang dilakukan oleh Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) pada Tahun 2016. Jumlah pengguna internet telah mencapai 132,7 juta orang dari total penduduk Indonesia sebanyak 256,2 juta orang. Hal ini menunjukkan bahwa lebih dari 50% penduduk Indonesia kini telah terhubung dengan internet. Penetrasi internet mayoritas masih berada di Pulau Jawa, yaitu sekitar 65% dari total pengguna [1]. Pengguna internet berdasarkan usia, mayoritas didominasi oleh pengguna yang berada pada rentang usia 35-44 tahun, yaitu sebesar 29,2% [2]. Berdasarkan usia justru didominasi oleh pengguna yang berusia antara 25-34 tahun, yaitu sebesar 75,8%. seperti pada Gambar 1.



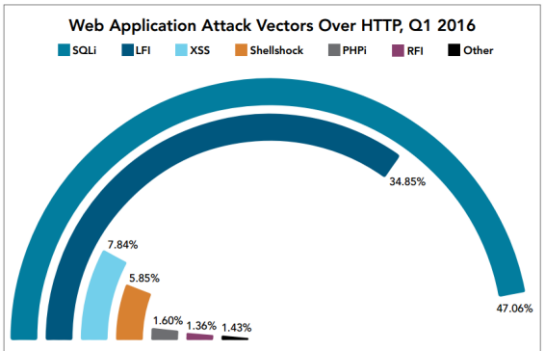
Gambar 1: Penggunaan Internet Pada Tahun 2016

Penggunaan internet yang semakin mudah menimbulkan dampak positif dan negatif tergantung pada pemanfaatannya. Internet ibarat pedang dengan dua sisi mata yang sama tajam jika penggunaanya benar maka akan menghasilkan informasi yang baik akan tetapi, sebaliknya jika penggunaanya salah maka akan mampu melukai diri sendiri. Informasi yang dihasilkan dari internet merupakan informasi yang menyeluruh, sebagai akibat dari meluasnya informasi ini banyak perusahaan maupun instansi berlomba-lomba membangun sistem informasi guna meningkatkan produktifitasnya.

Pembuatan sistem informasi dapat meningkatkan mutu dan kualitas suatu organisasi. Pentingnya nilai informasi menyebabkan informasi yang dihasilkan dari sistem harus dibatasi pengaksesan oleh orang-orang tertentu agar nilai informasi yang disampaikan terjaga integritasnya. Jatuhnya informasi ke pihak lain yang tidak berwenang dapat menimbulkan kerugian bagi organisasi sehingga sistem yang dibuat harus mampu menanggulangi dari tindakan-tindakan yang tidak diinginkan.

Keamanan informasi merupakan hal yang harus diperhatikan bagi setiap instansi agar terhindar dari gangguan atau tindakan

kejahatan. Masalah keamanan atau gangguan banyak bertebaran di internet, gangguan tersebut bisa berupa serangan *Malware*, Eksploitasi, Injeksi *database* dan lain sebagainya. Badan pengawas lalu lintas internet meyimpulkan bahwa pada tahun 2016 sekitar 90% kejahatan internet dilakukan dengan menyerang aplikasi web dengan serangan yang paling populer adalah dengan cara menginjeksi database yang mencapai 47.06 % total serangan yang popular [3]. Pemahaman dan kesadaran yang kurang terhadap isu keamanan sistem selalu mengancam setiap saat khususnya bagi para pengembang. Kebocoran data atau perusakan dapat mengancam setiap saat seiring dengan meningkatnya sumber daya manusia. Solusi pengamanan web dari gangguan atau serangan



Gambar 2: Serangan Populer Pada Aplikasi Web

hacker dapat dilakukan dengan cara *self test* yaitu pengujian yang dilakukan terhadap *web server* secara legal dengan aktifitas menyerupai *hacker*. *Self test* dapat dilakukan dengan beberapa metode *penetration testing* salah satunya adalah *Information Systems Security Assessment Framework* (ISSAF), *Open Web Application Security Project* (OWASP) versi 4 dan *Open Source Security Testing Methodology Manual* (OSSTMM).

OWASP versi 4 merupakan peningkatan pada versi sebelumnya. Adapun 3 kelebihan versi 4 dibanding versi 3 yaitu: versi panduan pengujian terintegrasi dengan produk dokumen, semua bab telah ditingkatkan dan uji kasus diperluas, mendorong penguji keamanan untuk mengintegrasikan penguji perangkat lunak yang lainnya. Metode OWASP bersifat terbuka dan kolaboratif, metode ini didasarkan pada pendekatan *Black Box Testing* dimana penguji sangat minim sekali informasi pada aplikasi yang akan diuji. Sebagai suatu metode pengamanan aplikasi. OWASP Versi 4 menggunakan 11 kategori pendekatan pengujian yang melibatkan analisis aktif dari aplikasi untuk setiap kelemahan. OWASP Versi 4 Menggunakan *tools vulnerability scanner* dengan kolaborasi beberapa *tools security project* dalam mencari celah keamanan, kemudian melakukan pengujian pada beberapa kategori untuk mengetahui keamanan suatu aplikasi.

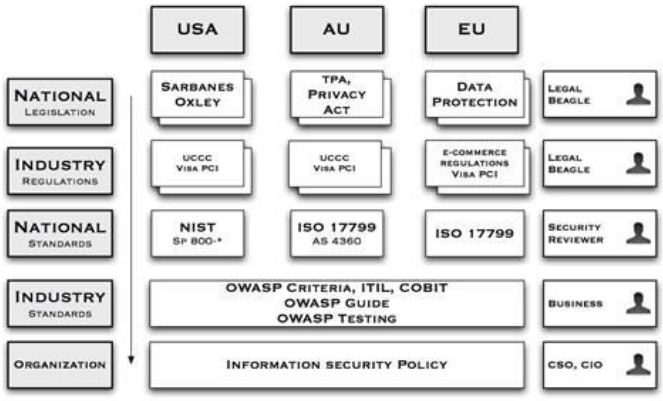
Tujuan penelitian ini adalah menganalisa keamanan aplikasi berbasis web

https://doi.org/10.35760/ik.2019.v24i1.1988

dengan *framework* OWASP versi 4 dengan kolaborasi beberapa *tools security project* untuk mengetahui keamanan suatu aplikasi, sehingga dapat dijadikan sebagai standar penilaian keamanan untuk aplikasi berbasis web. Penerapan metode OWASP Versi 4 pada penelitian ini, dilakukan pada sebuah aplikasi web yang beralamat pada www.xyz.com. Pengujian ini dilakukan sebagai bentuk bantuan teknik pengujian keamanan aplikasi yang nantinya dapat dijadikan sebagai rekomendasi tindak lanjut dalam pengamanan aplikasi.

Open Web Application Security Project (OWASP) Version 4

Open Web Application Security Project (OWASP) adalah komunitas terbuka yang didedikasikan untuk memungkinkan organisasi mengembangkan, membeli, dan memelihara aplikasi yang dapat dipercaya. OWASP adalah jenis organisasi baru. Kebebasan kami dari tekanan komersial memungkinkan kami memberikan informasi terkait keamanan aplikasi yang tidak bias, praktis, efektifbiaya. OWASP tidak terafiliasi dengan perusahaan teknologi manapun, meskipun kami mendukung penggunaan teknologi keamanan komersial. Serupa dengan banyak proyek software open-source, OWASP menghasilkan beragam jenis materi dengan cara kolaborasi dan terbuka. [4]. Pendekatan OWASP Sangat baik untuk dikombinasikan dan dicocokkan pada kontrol COBIT dan ISO 27002 (17799) dan sebagian besar standar keamanan informasi yang lainnya. Diagram berikut ini menunjukkan dimana OWASP cocok dikombinasikan dengan metode lain, dengan cara menyesuaikan dengan kebijakan atau aturan-aturan keamanan informasi yang sudah ditetapkan oleh masing-masing organisasi.



Gambar 3: Diagram Letak Pendekatan OWASP Pada Framework

Teknik Penetration Testing

Tiga strategi pengujian vulnerability assessment berdasarkan lingkup dan jenis audit [5]. Tiga strategi tersebut adalah yaitu: (1) *Black Box Testing*: Pada pendekatan ini penguji tidak memiliki pengetahuan tentang

target yang akan diuji. Mereka hanya mencari tahu semua celah dari sistem berdasarkan pengalaman dan keahlian individu. peguji pada dasarnya bertujuan mengaudit keamanan eksternal target yang diuji dengan mensimulasikan tindakan dan prosedur seperti attacker nyata yang mungkin hadir di beberapa tempat lain di luar batas target uji. (2) *White Box Testing*: Pendekatan ini bertentangan dengan pendekatan Black Box Testing. Pada pendekatan ini penguji disediakan semua informasi lengkap seperti konfigurasi jaringan dan konfigurasi sistem yang diperlukan dan kredensial mengenai target uji, dan penguji mengaudit pengaturan keamanan internal serta mensimulasikan tindakan dan prosedur yang nyata dari ancaman internal seperti bahaya karyawan yang hadir dalam batas-batas target dan kebijakan. Pengujian ini memerlukan pemahaman yang mendalam tentang jaringan pengujian atau sistem untuk memberikan hasil yang lebih baik. (3) *Gray Box Testing*: Pendekatan ini dipahami sebagai kombinasi dari dua pendekatan. Pada pendekatan ini, penguji memiliki pengetahuan parsial jaringan pengujian atau penguji tidak memiliki pengetahuan tentang arsitektur jaringan yang lengkap, akan tetapi dia tahu beberapa informasi dasar dari jaringan pengujian dan konfigurasi sistem. maka tester mengumpulkan lanjut informasi dengan melakukan tes.

Penelitian ini dilakukan tidak terlepas dari hasil penelitian-penelitian terdahulu yang pernah dilakukan sebagai bahan perbandingan dan kajian. Penelitian tersebut diantaranya sebuah kolaborasi teknik pengujian penetrasi dan strategi yang dikembangkan sesuai kebutuhan organisasi dalam hal sifat bisnis dan ukuran dengan menggunakan alat pentest (*Metasploit, Nessus, John The Ripper, accunetix, and Network Fingerprinting*) [6] dan penyerang dapat masuk ke sebuah sistem web dan mengekploitasi lebih lanjut pada sistem [7].

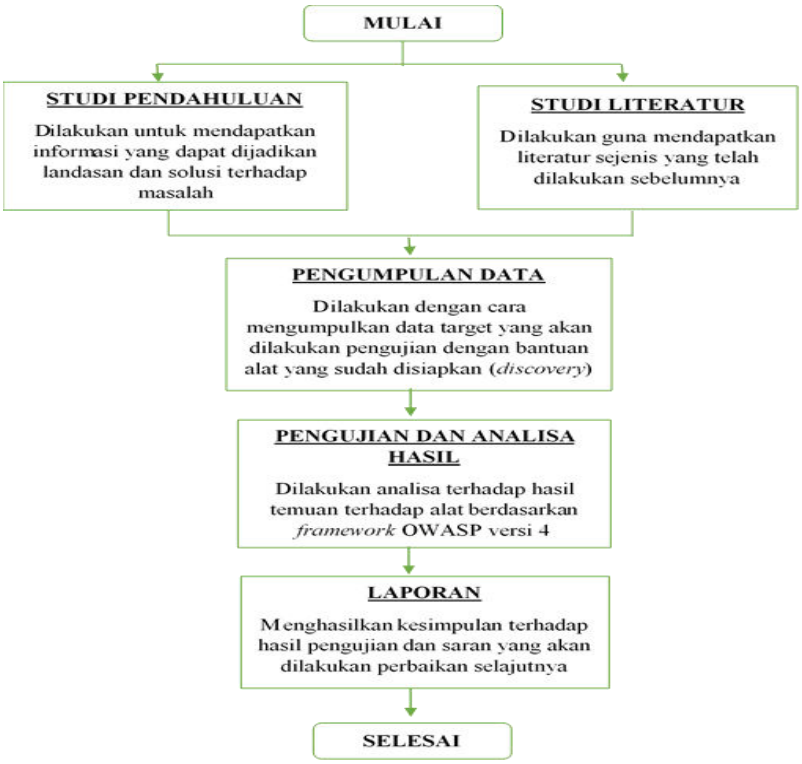
Persamaan penelitian ini dengan hasil-hasil penelitian sebelumnya adalah pada salah satu variabel yang digunakan dalam membahas pokok permasalahan, yaitu variabel keamanan sistem. Secara garis besar persamaan penelitian ini mengacu pada penelitian yang telah dilakukan oleh Chhavi Jain, Isatou Hydara, dkk., Sugandh Shah dan B.M.Mehtre yaitu melakukan pengujian keamanan dengan menggunakan pendekatan OWASP.

Perbedaan antara penelitian ini dengan hasil-hasil penelitian sebelumnya adalah pada pendekatan dan versi OWASP yang digunakan dalam melakukan pengujian serta rekomendasi tindakan lanjut yang diberikan.

METODE PENELITIAN

Secara garis besar skema metode penelitian ini dapat digambarkan sebagaimana pada Gambar 3. Berdasarkan skema pada gambar 3 dapat dijelaskan bahwa metode penelitian ini menggunakan teknik studi literature dan teknik pengumpulan data dalam hal ini

obeservasi non partisipan dimana teknik ini menempatkan diri sebagai pengamat yang tidak secara langsung terlibat dalam sistem. Berdasarkan hasil pengamatan dilakukan pengujian dengan teknik dan bantuan alat yang sudah disiapkan, untuk gambaran umum pengujian dapat dilihat pada Gambar 4. Pada gambar 4 dapat dijelaskan bahwa langkah pengujian dengan metode OWASP versi 4 dimulai dari informasi yang berkaitan dengan target pengujian selanjutnya melakukan scanning target dan berikutnya membuktikan adanya kerentanan dari masing-masing bagian dan yang terakhir adalah memberikan rekomendasi tindak lanjut dan laporan hasil pengujian.



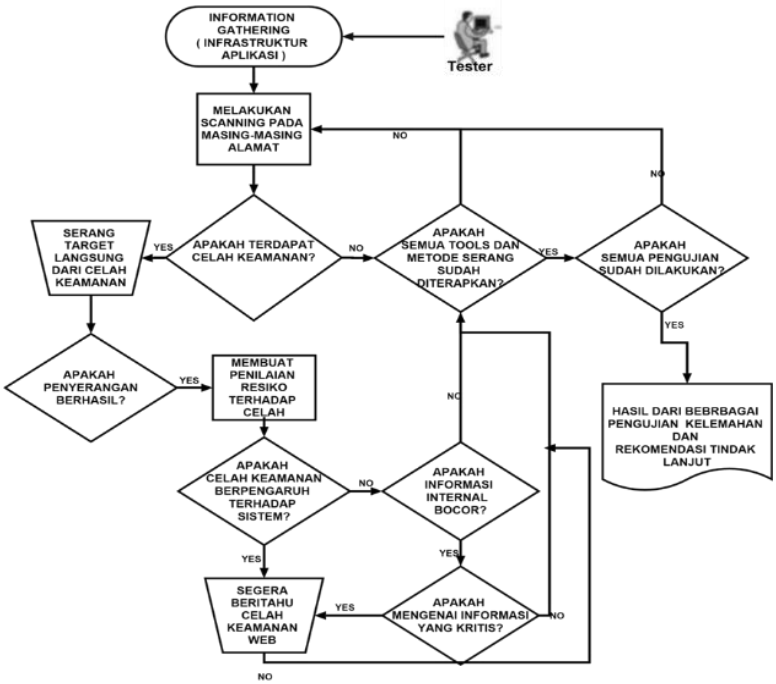
Gambar 3: Skema Penelitian

Gambar 4: Pengujian Keamanan Aplikasi

HASIL DAN PEMBAHASAN

Implementasi OWASP Versi 4

Metode pengujian aplikasi web berdasarkan OWASP versi 4 dilakukan dengan dua pendekatan yaitu black box testing dan gray box testing yang telah dijelaskan sebelumnya. Adapun cara kerja pengujian dapat dilihat pada gambar 5 berikut ini. Secara garis besar metode OWASP terbagi menjadi 11 kategori dengan 91 kontrol dari semua kategori. Pengujian pada penelitian ini difokuskan pada 5 kategori yaitu *Authentication Testing, Authorization Testing Session Management Testing, Input Validation Testing,* dan *Error Handling* dengan masing-masing tahapannya. Adapun tahapan tahapan pengujian OWASP versi 4 sebagaimana Tabel 3 berikut ini:



Gambar 5: *Workflow* Pengujian OWASP

Tabel 2.1: Tahapan Implementasi metode OWASP versi 4

No	Fokus	Tahapan	Aktifitas	Alat
1.	***Authentication Testing***	*Testing For Credentials Transported Over An Encrypted Channel* (OTG-AUTHN-001)	Memastikan bahwa data pengguna kredensial terenkripsi dari web browser ke server dan memastikan halaman login diakses melalui HTTPS	• *Mozilla Firefox*
		Testing For Default Credentials	Memeriksa konfigurasi dan *password default* dari halaman login dengan prediksi	• *Netsparker*

https://doi.org/10.35760/ik.2019.v24i1.1988

		(OTG-AUTHN-002)		
		Testing For Weak Lock Out Mechanism (OTG-AUTHN-003)	Menguji dengan beberapa kali salah login untuk memeriksa apakah terjadi *lock out* atau pemblokiran.	• *ZAP Attack Proxy* • *Mozilla Firefox*
		Testing For Bypassing Authentication Schema (OTG-AUTHN-004)	Menguji skema otentikasi dengan memanfaatkan log dihalaman dengan memotifikasi parameter URL.	• *Mozilla Firefox* • *Netsparker*
		Testing For Vulnerable Remember Password (OTG-AUTHN-005)	Pengujian dengan melihat *log password* yang disimpan dan *autocomplete* yang masih diaktifkan.	• *Mozilla Firefox* • *OWASP ZAP Attack Proxy*
		Testing For Browser Cache Weakness (OTG-AUTHN-006)	Penguji *cache browser* dari tombol "*back*" untuk melihat sumber daya yang ditampilkan sebelumnya.	• *Mozilla Firefox* • *OWASP ZAP Attack Proxy*
		Testing For Weak Password Policy (OTG-AUTHN-007)	Melakukan *brute force* mengunakan kamus password.	• *Mozilla Firefox* • *Netsparker*

Tabel 2.2 : Tahapan Implementasi metode OWASP versi 4 (Lanjutan)

No	Fokus	Tahapan	Aktifitas	Alat
2.	***Authorization Testing***	*Testing Directory Traversal/File Include* (OTG-AUTHZ-001)	Percobaaan akses *web document root* atau *root directory*, dengan menyisipkan string seperti menebak lokasi file seperti \\server_or_ip\path\to\file.abc untuk sistem operasi Windows.	• *OWASP ZAP Attack Proxy* • *Netsparker*
		Testing For Bypassing Authorization Schema (OTG-AUTHZ-002)	Percobaan dilakukan pada halaman *administrator*, Apakah dapat diakses secara langsung tanpa adanya proses autentifikasi pada alamat admin.	• *Netsparker Mozilla firefox*
		Testing For Privilege Escalation (OTG-AUTHZ-003)	Menguji kesalahan pemrograman yang memungkinkan pengguna mendapatkan hak istimewa dengan memeriksa *hidden field HTML.*	• *ZAP Attack Proxy* • *Netsparker* • *Mozilla Firefox*
		Testing For Bypassing Authentication Schema (OTG-AUTHN-004)	Menguji skema otentikasi dengan memanfaatkan log dihalaman dengan memotifikasi parameter URL.	• *Mozilla Firefox* • *Netsparker*

		Testing For Vulnerable Remember Password (OTG-AUTHN-005)	Pengujian dengan melihat *log password* yang disimpan dan *autocomplete* yang masih diaktifkan.	• *Mozilla Firefox* • *OWASP ZAP Attack Proxy*
		Testing For Browser Cache Weakness (OTG-AUTHN-006)	Penguji *cache browser* dari tombol "*back*" untuk melihat sumber daya yang ditampilkan sebelumnya.	• *Mozilla Firefox* • *OWASP ZAP Attack Proxy*
		Testing For Weak Password Policy (OTG-AUTHN-007)	Melakukan *brute force* mengunakan kamus password.	• *Mozilla Firefox* • *Netsparker*

Tabel 2.3: Tahapan Implementasi metode OWASP versi 4 (Lanjutan)

No	Fokus	Tahapan	Aktifitas	Alat
3.	***Session Management Testing***	*Testing For Session Management Schema* (OTG-SESS-001)	Pengujian dilakukan pada *session cookies* yang dikirim oleh server dengan cara melakukan percobaan akses halaman dengan *cookie,* kemudian coba lagi tanpa *cookie* apakah rentan terhadap *hijacking*.	• *OWASP ZAP Attack Proxy* • *Google Chrome (plugin)*
		Testing For Cookies Attributes (OTG-SESS-002)	Pengujian terhadap atribut *cookies* yang digunakan apakah sudah menggunakan *secure attribute*, *httponly attribute, domain attribute, path attribute, expires attribute*	• *OWASP ZAP Attack Proxy Google Chrome (plugin*
		Testing For Session Fixation (OTG-SESS-003)	Melakukan pengujian terhadap *session ID* apakah terjadi pembaharuan *session ID* setelah otentikasi berhasil.	• *OWASP ZAP Attack Proxy* • *Google Chrome (plugin*
		Testing For Exposed Session Variables (OTG-SESS-004)	Pengujian terhadap *cookies, sessionid, hidden field*, apakah apakah setiap kali proses otentikasi berhasil, *user* menerima token sesi yang berbeda melalui channel yang dienkripsi setiap kali melakukan permintaan *HTTP*	• *OWASP ZAP Attack Proxy* • *Google Chrome (plugin*
		Testing For CSRF (OTG-SESS-005)	Memeriksa apakah aplikasi rentan dengan serangan *Cross Site Request Forgery (CSRF)* dimana serangan ini dapat memanipulasi alamat URL yang falid	• *Netsparker* • *Mozilla firefox*
		Testing for logout functionality (OTG-SESS-006)	Pengujian terhadap serangan *cross site scripting* apakah terdapat validasi halaman	• *Mozilla firefox*
		Test Session Timeout (OTG-SESS-007)	Pengujian terhadap batas waktu diam pada aplikasi untuk jumlah waktu tertentu, apakah terdapat *log out* otomatis, dan memastikan tidak dapat menggunakan tombol back pada sesi yang sama.	• *Mozilla firefox*

Tabel 2.4: Tahapan Implementasi metode OWASP versi 4 (Lanjutan)

No	Fokus	Tahapan	Aktifitas	Alat
4.	***Input Validation Testing***	*Testing For Reflected Cross Site Scripting* (OTG-INPVAL-001)	Pengujian dilakukan dengan memeriksa kerentanan terhadap serangan *Hijacking*	• *ZAP Attack Proxy* • *Netsparker* • *Mozilla firefox*
		Testing For Stored Cross Site Scripting (OTG-INPVAL-002)	Pengujian dilakukan dengan memeriksa kerentanan terhadap *Cross Site Scripting* (XSS)	• *ZAP Attack Proxy* • *Netsparker Mozilla firefox*
		Testing For HTTP Verb Tampering (OTG-INPVAL-003)	Memeriksa data yang berpotensi terkena serangan XSS dan juga terhadap *webDAV* yang disebabkan *HEADER HTTP* yang aktif	• *OWASP ZAP Attack Proxy* • *Mozilla firefox*
		Testing For HTTP Parameter Pollution (OTG-INPVAL-004)	Pengujian terhadap parameter *HTTP* apakah dapat dimanipulasi atau digandakan	• *OWASP ZAP Attack Proxy* • *Google Chrome (plugin)*
		Testing For SQL Injection (OTG-INPVAL-005)	Pengujian dilakukan dengan memeriksa kerentanan terhadap *SQL Injection*	• *OWASP ZAP Attack PRoxy* • *HAVIJ 1.15* •
		Testing for logout functionality (OTG-SESS-006)	Pengujian terhadap serangan *cross site scripting* apakah terdapat validasi halaman	• *Mozilla firefox*
		Test Session Timeout (OTG-SESS-007)	Pengujian terhadap batas waktu diam pada aplikasi untuk jumlah waktu tertentu, apakah terdapat *log out* otomatis, dan memastikan tidak dapat menggunakan tombol back pada sesi yang sama.	• *Mozilla firefox*
5.	***Error Handling***	*Testing For Error Code* (OTG-ERR-001)	Pemeriksaan dilakukan apakah terdapat pesan/*error code* pada aplikasi.	• *Netsparker* • *OWASP ZAP Attack PRoxy*
		Testing For Stack Traces (OTG-ERR-002)	Pemeriksaan dilakukan apakah terdapat *link* yang *error* dan *message error* yang belum di *handling*.	• *OWASP ZAP Attack PRoxy* • *Netsparker* • *Mozilla firefox*

Hasil Pengujian

Tahapan ini dilakukan penilaian terhadap pengujian yang telah dilakukan sebelumnya berdasarkan 5 kategori pengujian. Masing-masing dari kategori ini dinilai berdasarkan hasil temuan yang didapatkan. Jika pada setiap tahapan metode OWASP versi 4 mampu dibuktikan dengan hasil temuan makan statusnya adalah "ditemukan" dan jika pada tahapan metode OWASP versi 4 tidak dapat dibuktikan dengan hasil temuan maka statusnya adalah "tidak ditemukan". Adapun hasil pengujian dapat dilihat pada Tabel 4.

Tabel 3.1 Hasil Pengujian Owasp Versi 4

Kategori	Tahapan	Teknik	Status	Hasil Temuan
Authentication Testing	(OTG-AUTHN-001)	Analisa paket *header HTTPS*	Ditemukan	Belum menerapkan *HTTPS*
	(OTG-AUTHN-002)	Konfigurasi *username* dan *password default*	Ditemukan	*Password default* pada *source code*
	(OTG-AUTHN-003)	*Invalid login* beberapa kali	Ditemukan	*No expired* pada user *invalid login*
	(OTG-AUTHN-004)	*Parameter URL, Session ID prediction*	Ditemukan	Dapat memodifikasi *URL*
	(OTG-AUTHN-005)	Analisa terhadap *autocomplete*	Ditemukan	Status *Autocomplete=on*
	(OTG-AUTHN-006)	Analisa *cache browser*	Tidak Ditemukan	Status *no cache*
	(OTG-AUTHN-007)	Analisa *username* dan *password* statis	Ditemukan	Terdapat *username* dan *password* statis
Authorization Testing	(OTG-AUTHZ-001)	Memeriksa jalur *directory traversal*	Ditemukan	*URL* yang mengarah pada *directory traversal*
	(OTG-AUTHZ-002)	Manipulasi HTTP *reguest header*	Ditemukan	*userid* dapat dimanipulsi dengan angka
	(OTG-AUTHZ-003)	Memeriksa *Error Programming*	Ditemukan	Terdapat *error programming*
Session Management Testing	(OTG-SESS-001)	Analisa status *cookies*	Ditemukan	Status *cookies no secure*
	(OTG-SESS-002)	Analisa *sessionid* atau *cookies*	Ditemukan	*Sessionid* dapat diakses melaui *URL*
	(OTG-SESS-003)	Analisa *sessionid* atau *cookies*	Ditemukan	Dapat login dengan *session* yang sama
	(OTG-SESS-004)	Analisa *sessionid* atau *cookies*	Ditemukan	*Sessionid* tidak berubah setelah *log out*
	(OTG-SESS-005)	Analisa *cookies* dan *tag HTML*	Ditemukan	*Tag HTML* dapat dimanipulasi
	(OTG-SESS-006)	*Locked invalid login*	Ditemukan	*No lock mechanism* pada *invalid login*
	(OTG-SESS-007)	*Analisa session time out*	Ditemukan	Tidak ada *session time out* untuk masa ambigu

Tabel 3.2 Hasil Pengujian Metode Owasp Versi 4 (Lanjutan)

Kategori	Tahapan	Teknik	Status	Hasil Temuan
Input Validation Testing	(OTG-INPVAL-001)	Analisa terhadap XSS	Ditemukan	Rentan terhadap *Reflected XSS*
	(OTG-INPVAL-002)	Analisa terhadap XSS	Ditemukan	Rentan terhadap *stored XSS*
	(OTG-INPVAL-003)	Analisa *HTTP Method request*	Tidak Ditemukan	Tidak terdapat metode selain *GET* dan *POST*
	(OTG-INPVAL-004)	Analisa *HTTP Parameter Pollution*	Tidak Ditemukan	Tidak ditemukan kerentanan *parameter pollution*
	(OTG-INPVAL-005)	Analisa *SQL Injection*	Ditemukan	Rentan serangan *SQL Injection*
Error Handling	(OTG-ERR-001)	Analisa *Error Code Customize*	Ditemukan	Ditemukan *Error Code* yang belum di *custome error*
	(OTG-ERR-002)	Analisa *Error Patching*	Ditemukan	Banyak ditemukan *Error Patching*

KESIMPULAN DAN SARAN

Hasil dari penelitian ini dapat disimpulkan bahwa analisis kerentanan aplikasi berbasis web dengan teknik OWASP versi 4 mampu mengetahui keamanan suatu aplikasi. Metode OWASP versi 4 dapat dijadikan sebagai standar penilaian keamanan aplikasi berbasis web berdasarkan hasil pengujian kerentanan pada website yang beralamat di www.xyz.com dari beberapa tahapan kategori yaitu pada tahap *Authentication Testing Authorization*, *Session Management Testing*, *Input Validation Testing,* dan *Error Handling*. Untuk pengembangan lebih lanjut maka penulis memberikan saran diantaranya adalah: (1) Perlunya penambahan status level (*severity*) untuk masing-masing tahapan dari hasil pengujian keamanan yang telah ditemukan berdasarkan panduan OWASP versi 4. (2) Sebaiknya ditambahkan pembahasan mengenai potensi suatu celah keamanan yang dapat berakibat terjadinya cela yang lain. Sehingga satu pencegahan dari suatu cela dapat menutup cela yang lainnya. (3) Perlu dilakukan penelitian lebih lanjut dengan metode ISSAF (*Information System Security Assessment Framework*) agar dapat diketahui kerentanan dari sisi web server.

DAFTAR PUSTAKA

[1] APJII, ”Infografis Penetrasi dan Prilaku Pengguna Internet Indonesia”. Asosiasi Jasa Pengguna Internet Indonesia, 2016 https://apjii.or.id/content/read/39/264/Survei-Internet-APJII-2016.

[2] Kominfo.”*Laporan Tahunan 2016*” Kementrian Komunikasi dan Informatika Republik Indonesia, 2016. https://www.kominfo.go.id/content/detail/10294/laporan-tahunan-kementerian-komunikasi-dan-informatika-tahun-2016/0/laporan_tahunan.

[3] Akamai,” *state of the internet / security report*”, 2016. https://www.akamai.com/us/en/resources/our-thinking/state-of-the-internet-report/archives/state-of-the-internet-security-reports-2016.jsp.

[4] OWASP, ”*The ten Most Critical Web Application Security Risk*”, The Open Web Application Security Project, 2010. http://www.owasp.org.

[5] Jai Narayan Goel.,B.M.Mehtre, “*Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology*”. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), University of Hyderabad. India: Elsevier, PP710 – 715, 2015.

[6] Ms.Sarabjit Kaur.,Mr.Sangram Singh, ”Penetration Testing”. *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, ISSN (*Online*) 2278, ISSN (*Print*) 2319 5940-1021, Vol 5, Issue 3, March 2016.

[7] Savita B. Chavan., Dr.B.B.Meshram," Classification of Web Application Vulnerabilities". *International Journal of Engineering Science and Innovative Technology (IJESIT),* ISSN 2319-5967,Vol 2, Issue 2 March 2013.