

# Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) untuk Penilaian Risk Rating

*Detect Web Application Security Flaws Using the Owasp (Open Web Application Security Project) Method for Risk Assessment*

**Bahrún Ghózali\*<sup>1</sup>, Kusrini<sup>2</sup>, Sudarmawan<sup>3</sup>**

<sup>1,2,3</sup> Magister Teknik Informatika Universitas Amikom Yogyakarta

E-mail: \*[gbahrun@gmail.com](mailto:gbahrun@gmail.com), [kusrini@amikom.ac.id](mailto:kusrini@amikom.ac.id), [sudarmawan@amikom.ac.id](mailto:sudarmawan@amikom.ac.id)

## **Abstrak**

*Mendeteksi kerentanan keamanan aplikasi berbasis website adalah hal yang penting, dan dapat memperkirakan risiko yang ada terhadap keberlangsungan suatu bisnis. Terjadinya transisi bisnis tradisional ke dalam lingkup aplikasi berbasis website dimanfaatkan oleh beberapa pelaku kejahatan dunia maya dengan tujuan mencuri informasi rahasia pengguna demi keuntungan pribadi. Pada Penelitian ini dilakukan mekanisme metode asesmen risiko pada sistem informasi harga komoditas utama yang dibangun oleh PT.Gitsolution. Dimana sistem tersebut merupakan informasi harga pokok untuk kehidupan sehari-hari yang dikelola oleh salah satu instansi pemerintah yang ada di Indonesia. Untuk mengetahui tingkat risiko pada sistem informasi harga komoditas utama menggunakan metode Open Web Application Security Project (OWASP) Risk Rating untuk mendeteksi kerentanan keamanan pada aplikasi berbasis website. Penelitian ini menghasilkan 2 faktor untuk memperkirakan Likelihood dan Impact, dari masing-masing faktor terdapat 3 risiko yang ditemukan yaitu risk severity High, risk severity Medium dan risk severity Low. Hasil penilaian risiko ini dapat membantu para pengelola dan pengembang sistem untuk menyadari risiko yang mungkin terjadi sehingga dapat mengambil tindakan untuk mencegah dan mengatasi risiko tersebut.*

**Kata Kunci** — Vulnerability – OWASP, Risk Rating, Security Assessment.

## **Abstract**

*Detecting website-based security vulnerabilities is important, and can estimate the risks to a business's sustainability. The transition of traditional businesses into the scope of a website-based application is exploited by some cybercriminals with the intention of stealing user confidential information for personal gain. In this research, the mechanism of risk assessment method on the main commodity price information system built by PT. Gitsolution. Where the system is the basic price information for everyday life that is managed by one of the government agencies in Indonesia. To determine the level of risk in the main commodity price information system using the Open Web Application Security Project (OWASP) Risk Rating method to detect security vulnerabilities in website-based applications. This research resulted 2 factors to estimate Likelihood and Impact, from each factor there are 3 risks found that is 1 risk severity High, risk severity Medium and risk severity Low. The results of this risk assessment can help system managers and developers to be aware of possible risks in order to take action to prevent and mitigate those risks.*

**Keywords** — Vulnerability – OWASP, Risk Rating, Security Assessment.

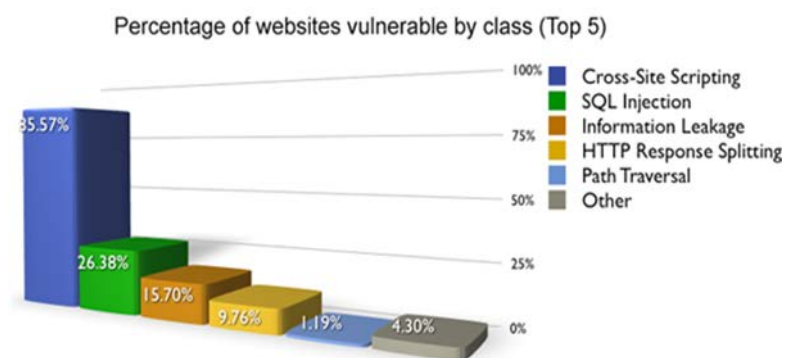
## 1. PENDAHULUAN

Pengujian sistem keamanan aplikasi berbasis website adalah hal yang penting di era perkembangan aplikasi berbasis web yang melaju dengan pesat. Semakin berkembangnya aplikasi berbasis web juga diiringi dengan tingginya serangan keamanan dari berbagai teknik ancaman. Seringkali masalah keamanan berada di urutan kedua, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting [1]. Oleh karena itu organisasi perlu melakukan asesmen pada aplikasi berbasis website agar organisasi mampu mendeteksi kerentanan dan memahami risiko yang dihadapi. Salah satu metode untuk penilaian tingkat risiko kerentanan keamanan aplikasi berbasis website adalah *OWASP Risk Rating Methodology*.

Sebelum melakukan pengujian sistem keamanan aplikasi berbasis website kami melakukan studi literatur dan menemukan beberapa penelitian terkait. Fernando dalam penelitiannya mengusulkan sebuah metode *Open Source Security Testing Methodology Manual (OSSTMM)* untuk melakukan *security testing* pada sistem penerimaan mahasiswa baru Universitas XYZ [2]. Ashraf dalam penelitiannya mengenalkan tentang memberikan gambaran umum resiko keamanan aplikasi berbasis web dengan *Model View Controller (MVC)*, menggunakan ASP.Net atau PHP [3]. Nugroho dalam penelitiannya mempresentasikan untuk Menentukan Dampak Resiko Keamanan Berbasis Pendekatan Owasp pada tiga domain yang terbukti adanya kerentanan *Sql Injenction* dan XSS [4].

Langkah besar dalam mengukur tingkat risiko adalah menentukan dampak buruk yang dihasilkan dari analisa kerentanan [5]. Hasil dari analisa kerentanan dapat membantu pengelola dan pengembang sistem untuk mencegah dan mengatasi dampak risiko yang ditemukan pada sistem. Belum adanya *Security Assessment* pada sistem informasi harga komoditas utama yang dibangun oleh PT. Gitsolution. Saat ini dalam membangun sistem tersebut dengan mengandalkan library untuk mengamankan sistem. Namun dengan menerapkan library belum diiringi dengan pengujian sistem secara langsung dari internal perusahaan, sehingga belum mengetahui secara pasti celah keamanan sistem yang sudah dibangun. Oleh karena itu perlu adanya *Security Assessment (Penilaian keamanan)* pada sistem tersebut.

Ada beberapa faktor yang menyebabkan kurangnya tingkat keamanan pada aplikasi website, diantaranya adalah kesalahan penulisan kode program dan *misconfiguration* [6]. Kesalahan pada penulisan kode program dalam pembuatan aplikasi berbasis website sering dimanfaatkan oleh penyerang, dalam hal ini serangan yang sering dimanfaatkan oleh penyerang diantaranya adalah SQL Injection, Authentication dan XSS [6]. Seperti pada diagram statistic yang dirilis oleh *webappsec.org* (diperbaharui pada januari 2010) pada Gambar 1 menunjukkan bahwa *SQL Injection (26.38%)* dan *XSS (35.57%)* merupakan jenis serangan yang sering digunakan [7].



Gambar 1. Prosentase kerentanan website

Pada penelitian ini, untuk mendeteksi kerentanan keamanan terdapat beberapa metode diantaranya: ISSAF, OSSTMM, OWASP, NIST [8]. Namun diantara ketiga metode tersebut yang tepat untuk *penetration testing* adalah OWASP [8]. OWASP juga dikenal sebagai organisasi non-

profit amal di Amerika Serikat berdiri pada tahun 2004 dan dilengkapi standart Guide untuk mempermudah *penetration testing*.

## 2. METODE PENELITIAN

Metodologi penilaian risiko OWASP adalah pendekatan sederhana untuk menghitung dan menilai risiko yang terkait dengan aplikasi. dimana dengan metode tersebut dapat diputuskan apa saja yang harus dilakukan terhadap resiko-resiko tersebut [7]. Dengan mengetahui resiko yang akan terjadi maka banyak manfaat yang akan diperoleh diantaranya, menghemat waktu dan mengurangi terjadinya resiko yang lebih serius. Perkiraan resiko pada metodologi OWASP dimulai dengan model:

$$\text{Risk} = \text{Likelihood} * \text{Impact}$$

Dimana:

*Likelihood* : Kemungkinan kerentanan untuk menjadi dieksploitasi oleh penyerang.

*Impact* : Dampak dari serangan yang berhasil sukses.

*Risk* : Kemungkinan risiko yang terkait dengan faktor ancaman, kerentanan, dampak teknis dan bisnis.

Sedangkan Untuk persamaan formulasi teorema matematika menggunakan persamaan i:

$$\bar{x} = \frac{\sum x}{n} \quad (1)$$

Dimana:

$\bar{x}$  : nilai Rata-rata hitung (*Risk*)

$\sum x$  : nilai sampel (skor penilaian)

$n$  : Jumlah sampel (Jumlah vulnerability)

Menurut OWASP terdapat beberapa tahap untuk menentukan dan mengkombinasikan besarnya resiko yang ditimbulkan akibat eksploitasi kelemahan yang terdapat pada suatu aplikasi web, Berikut tahapan OWASP Risk Rating Methodology yaitu *Identifying a Risk, Factors for Estimating Impact, Determining the Severity of the Risk, Deciding What to Fix, Customizing the Risk Rating Model*.

### 2.1 Action Research

Metode penelitian yang digunakan dalam penelitian ini menggunakan metode penelitian tindakan atau action research, ada lima tahapan dalam penelitian yang merupakan siklus dari *action research*. Tahap pertama adalah melakukan diagnosa (*Diagnosing*) Pada tahapan ini peneliti akan melakukan identifikasi masalah-masalah yaitu diagnosa sistem keamanan pada aplikasi berbasis web yang dibangun PT. Gitsolution.

Tahap kedua adalah membuat rencana tindakan (*Action Planning*) tahapan ini peneliti melakukan pemahaman pokok masalah yang ada dan menyusun rencana tindakan yang tepat untuk menyelesaikan masalah yang ada. Peneliti akan mulai menyusun rencana pengujian yang akan dilakukan pada aplikasi berbasis web yang dibangun PT. Gitsolution.

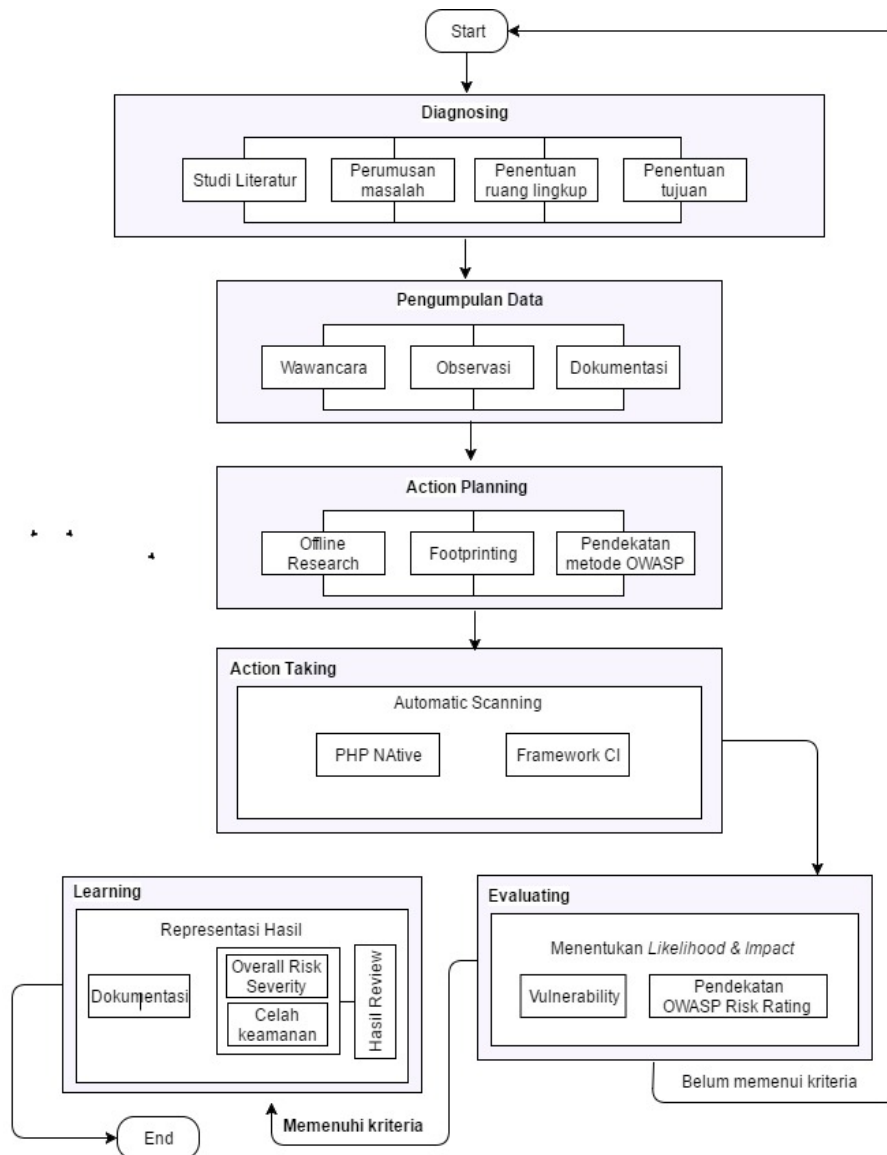
Tahap ketiga adalah melakukan tindakan (*Action Taking*) mengimplementasikan rencana tindakan yang telah disusun. Pada langkah ini peneliti mulai melakukan tahapan-tahapan investigasi guna mendapatkan informasi kelemahan sistem dan mengujinya secara langsung dengan menggunakan tipe-tipe ancaman terhadap aplikasi berbasis web yang dibangun PT. Gitsolution.

Tahap keempat adalah melakukan evaluasi (*Evaluating*) setelah tahapan Action Taking dilaksanakan peneliti mulai melakukan evaluasi pada hasil dari implementasi sebelumnya dan mulai menyimpulkan hasil dari langkah sebelumnya.

Tahap kelima adalah pembelajaran (*Learning*) langkah ini merupakan tahap akhir dari penelitian yaitu melakukan review terhadap hasil dari tahapan-tahapan yang telah dilalui. Hasil dan Pembahasan.

### 2.2 Alur Penelitian

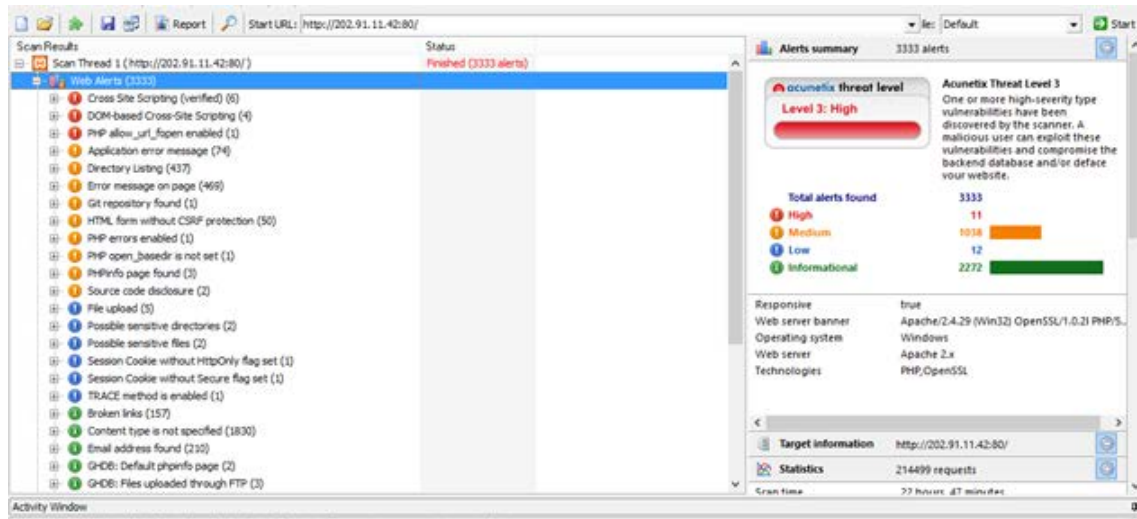
Alur penelitian merupakan langkah-langkah yang harus dilakukan dalam penelitian hingga mencapai suatu kesimpulan. Adapun alur penelitian yang penulis gunakan seperti yang terlihat pada Gambar 2.



Gambar 2. Alur Penelitian

### 3. HASIL DAN PEMBAHASAN

Mendeteksi kerentanan pada penelitian ini menggunakan aplikasi *Acunetix* untuk mengetahui celah keamanan yang ada di aplikasi berbasis website. Dimana pengembangan aplikasi website yang sudah dibangun menggunakan *PHP Native* dan *Framework CI (CodeIgniter)* sebagai *platform web application development*. Gambar 3 adalah tampilan hasil scanning Menggunakan *Tools Acunetix Web Vulnerability Scanner*:



Gambar 3. Tools Acunetix Web Vulnerability Scanner

Setelah dilakukan proses *scanning*. Pada tabel 1 akan menunjukkan celah keamanan yang ada di sistem informasi harga komoditas utama.

Tabel 1. Celah keamanan yang ditemukan

Domain	Kerentanan
http://202.91.11.42/	Cross Site Scripting
	PHP allow_url_fopen enabled
	Error message on page
	Directory Listing
	HTML form without CSRF protection
	Session Cookie without HttpOnly flag set
	Possible sensitive directories
	Possible sensitive files
	Possible sensitive files
http://202.91.11.42/CI	Cross Site Scripting
	PHP allow_url_fopen enabled
	Error message on page
	Directory Listing
	HTML form without CSRF protection
	Session Cookie without HttpOnly flag set
	Slow response time

Berdasarkan metodologi OWASP *Risk Rating* terdapat beberapa tahapan untuk menentukan dan mengkombinasikan besarnya resiko yang ditimbulkan, tahapan tersebut diantaranya Threat Agent Factors, Vulnerability Factors, Technical Impact, Business Impact.

### 3.1 Threat Agent Factors

Kumpulan faktor pertama terkait dengan *Threat agent* yang terlibat. Tujuannya adalah untuk memperkirakan kemungkinan serangan yang berhasil oleh kelompok *threat agent*. Berikut kriteria untuk memperkirakan *Likelihood* kelompok *Threat agent factors* antara lain [10]:

1. Skill level  
Seberapa terampil secara teknis kelompok *threat agent*? Keterampilan penetrasi keamanan (9), keterampilan jaringan dan pemrograman (6), pengguna komputer tingkat lanjut (5), beberapa keterampilan teknis (3), tidak ada keterampilan teknis (1)
2. Motive  
Seberapa kelompok *Threat Agent* termotivasi untuk menemukan dan memanfaatkan kerentanan ini? Tidak ada reward (1), memungkinkan mendapat reward (4), mendapatkan reward yang tinggi (9).
3. Opportunity  
Sumber daya apa yang dibutuhkan kelompok *threat agent* untuk menemukan dan memanfaatkan kerentanan ini? Akses penuh atau membutuhkan sumber daya yang mahal (0), akses khusus atau sumber daya yang dibutuhkan (4), beberapa akses atau sumber daya yang dibutuhkan (7), tidak ada akses atau sumber daya yang diperlukan (9).
4. Size  
Seberapa besar kelompok *threat agent*? Pengembang (2), administrator sistem (2), pengguna intranet (4), mitra (5), pengguna terotentikasi (6), pengguna internet anonim (9).

Rumus untuk mendapatkan hasil *Threat agent* secara keseluruhan mengikuti OWASP *Risk Rating Methodology* menggunakan persamaan 2:

$$\text{Threat Agent} = \frac{\text{Skill level} + \text{Motive} + \text{Oppotunity} + \text{Size}}{4} \quad (2)$$

Berikut adalah hasil pilihan faktor *threat agent* yang sudah disediakan oleh OWASP *risk rating*, seperti pada Tabel 2.

Tabel 2. Skor Threat Agent Factor

http://202.91.11.42/CI				
Jenis Ancaman	Skill level	Motive	Oppurtunity	Size
Cross Site Scripting	9	9	4	9
PHP allow_url_fopen enabled	9	9	4	9
Application Error message	6	4	9	2
Directory Listing	6	3	4	9
HTML form without CSRF protection	9	4	7	9
Session Cookie without HttpOnly flag set	6	4	4	6
Slow response time	6	4	9	9
http://202.91.11.42/				
Cross Site Scripting	9	9	4	9
PHP allow_url_fopen enabled	9	9	4	9
Error message on page	6	4	9	2
Directory Listing	6	3	4	9
HTML form without CSRF protection	9	4	7	9
Session Cookie without HttpOnly flag set	6	4	4	6
Possible sensitive directories	5	4	9	9
Possible sensitive files	5	4	9	9

### 3.2 Vulnerability Factors

Faktor selanjutnya adalah terkait dengan *vulnerability* yang terlibat. Dengan Tujuan untuk memperkirakan kemungkinan *vulnerability* tertentu yang terlibat ditemukan dan dieksploitasi. Asumsikan dengan *threat agent* yang sudah dipilih. Berikut kriteria untuk memperkirakan *Likelihood* kelompok *vulnerability factors* antara lain [10]:

1. Ease of discovery  
Seberapa mudah bagi kelompok *threat agent* untuk menemukan kerentanan ini? cara Praktis tidak mungkin (1), sulit (3), mudah (7), alat otomatis tersedia (9).
2. Ease of exploit  
Seberapa mudah bagi kelompok *threat agent* untuk benar-benar memanfaatkan kerentanan ini? Alat bantu otomatis teoritis (1), sulit (3), mudah (5), tersedia (9).
3. Awareness  
Seberapa terkenal kerentanan ini terhadap kelompok *threat agent*? Tidak diketahui (1), tersembunyi (4), jelas (6), pengetahuan umum (9).
4. Intrusion detection  
Seberapa besar kemungkinan *exploit* untuk dideteksi? Deteksi aktif dalam aplikasi (1), login dan ditinjau (3), login tanpa *review* (8), tidak login (9).

Rumus untuk mendapatkan hasil *Vulnerability Factors* secara keseluruhan mengikuti OWASP *Risk Rating Methodology* menggunakan persamaan 3:

$$Vulnerability = \frac{Ease\ of\ Discovery + Ease\ of\ Exploit + Awareness + Intrusion\ Detection}{4} \quad (3)$$

Berikut adalah hasil pilihan faktor *vulnerability* yang sudah disediakan oleh OWASP *risk rating*, seperti pada Tabel 3.

Tabel 3. Skor Vulnerability Factors

http://202.91.11.42/CI				
Jenis Ancaman	Ease of Discovery	Ease of Exploit	Awareness	Intrusion Detection
Cross Site Scripting	9	5	9	1
PHP allow_url_fopen enabled	9	3	4	1
Application Error message	9	3	4	1
Directory Listing	9	3	4	1
HTML form without CSRF protection	9	3	9	1
Session Cookie without HttpOnly flag set	9	3	9	1
Slow response time	9	5	9	1
http://202.91.11.42/				
Cross Site Scripting	9	5	9	1
PHP allow_url_fopen enabled	9	3	4	1
Application Error message	9	3	4	1
Directory Listing	9	3	4	1
HTML form without CSRF protection	9	3	9	1
Session Cookie without HttpOnly flag set	9	3	9	1
Possible sensitive directories	9	3	4	1
Possible sensitive files	9	3	4	1

### 3.3 Technical Impact

Tujuan utama dari dampak teknis adalah menghitung besarnya dampak jika kerentanan dieksploitasi dari aplikasi. Faktor dampak teknis lebih jauh dibagi menjadi empat kelas yaitu kerahasiaan, integritas, ketersediaan dan akuntabilitas Tujuan informasi Sistem keamanan adalah untuk melindungi kerahasiaan, integritas, tersedianya. Dengan demikian faktor dampak teknis

memainkan peran utama dalam penilaian risiko aplikasi. Dampak teknisnya adalah perkiraan jumlah faktor teknis ini dengan memberikan kecocokan bobot faktor individu. Berikut kriteria untuk memperkirakan *Technical impact* antara lain [10]:

1. Loss of confidentiality  
Berapa banyak data yang bisa diungkapkan dan seberapa sensitif? Data yang diungkapkan minimum dan tidak sensitif (2), minimal data kritis yang diungkapkan (6), data non-sensitif ekstensif yang diungkapkan (6), data kritis dan ekstensif diungkapkan (7), semua data yang diungkapkan (9).
2. Loss of integrity  
Berapa data yang bisa rusak dan seberapa rusaknya? Data korup yang minimal sedikit (1), data korup minimal yang serius (3), data yang agak korup sekali, (7), semua data benar-benar korup (9).
3. Loss of availability  
Berapa banyak layanan yang bisa hilang dan seberapa vitalnya? Layanan sekunder minimal terputus (1), layanan primer minimal terputus (5), layanan sekunder yang luas terganggu (5), layanan utama yang luas terganggu (7), semua layanan benar-benar hilang (9).
4. Loss of accountability  
Apakah tindakan agen ancaman bisa dilacak pada individu? Sepenuhnya dapat dilacak (1), mungkin dapat dilacak (7), benar-benar anonim (9).

Rumus untuk mendapatkan hasil *Technical Impact* secara keseluruhan mengikuti OWASP Risk Rating Methodology menggunakan persamaan 4:

$$\text{Technical impact} = \frac{\text{Confidentiality} + \text{Integrity} + \text{Availability} + \text{Accountability}}{4} \quad (4)$$

Berikut hasil dari skor penilaian *Technical impact*, seperti pada table 4.

Tabel 4. Skor Technical Impact

http://202.91.11.42/CI				
Jenis Ancaman	Loss of Confidentiality	Loss of Integrity	Loss of Availability	Loss of Accountability
Cross Site Scripting	2	1	1	9
PHP allow_url_fopen enabled	6	3	1	7
Application Error message	2	3	1	7
Directory Listing	2	1	1	1
HTML form without CSRF protection	6	3	1	9
Session Cookie without HttpOnly flag set	2	3	1	7
Slow response time	2	1	1	7
http://202.91.11.42/				
Cross Site Scripting	2	1	1	9
PHP allow_url_fopen enabled	6	3	1	7
Application Error message	2	3	1	7
Directory Listing	2	1	1	1
HTML form without CSRF protection	6	3	1	9
Session Cookie without HttpOnly flag set	2	3	1	7
Possible sensitive directories	2	1	1	7
Possible sensitive files	2	1	1	7

### 3.4 Business Impact

Tujuan akhir dari penilaian risiko adalah untuk mengukur dampak bisnis, jika kerentanan sedang dieksploitasi oleh penyerang. Pada umumnya, perusahaan harus mengarahkan resiko perusahaan dengan dampak bisnis, terutama jika pengguna adalah tingkat eksekutif. Resiko bisnis inilah yang memberikan alasan investasi dalam memperbaiki masalah keamanan.



Faktor-faktor di bawah ini adalah area umum bagi banyak bisnis, namun kawasan ini bahkan lebih unik bagi perusahaan daripada faktor yang terkait dengan ancaman, kerentanan, dan dampak teknis. Berikut kriteria untuk memperkirakan *Business impact* antara lain [10]:

1. Financial damage  
Berapa banyak kerusakan finansial yang diakibatkan oleh eksploitasi? Kurang dari biaya untuk memperbaiki kerentanan (1), pengaruh kecil terhadap laba tahunan (3), berpengaruh signifikan terhadap laba tahunan (7), kebangkrutan (9).
2. Reputation damage  
Apakah hasil eksploitasi akan merusak reputasi yang akan merugikan bisnis? Kerusakan minimal (1), Kehilangan akun utama (4), kehilangan niat baik (5), kerusakan merek (9).
3. Non-compliance  
Berapa banyak keterpaparan yang tidak dikenali perkenalan? Pelanggaran ringan (2), pelanggaran yang jelas (5), pelanggaran profil tinggi (7).
4. Privacy violation  
Berapa banyak informasi identitas pribadi yang bisa diungkapkan? Satu orang (3), ratusan orang (5), ribuan orang (7), jutaan orang (9).

Rumus untuk mendapatkan hasil *Business Impact* secara keseluruhan mengikuti OWASP *Risk Rating Methodology* menggunakan persamaan 5:

$$\text{Business impact} = \frac{\text{Financial Damage} + \text{Reputation Damage} + \text{Non-Compliance} + \text{Privacy Violation}}{4} \quad (5)$$

Berikut hasil dari skor penilaian *Business impact*, seperti pada Tabel 5.

Tabel 5. Skor Business Impact

http://202.91.11.42/CI				
Jenis Ancaman	Financial Damage	Reputation Damage	Non-Compliance	Privacy Violation
Cross Site Scripting	1	1	2	3
PHP allow_url_fopen enabled	1	1	2	3
Application Error message	3	4	2	3
Directory Listing	1	1	2	3
HTML form without CSRF protection	1	1	2	3
Session Cookie without HttpOnly flag set	3	1	2	3
Slow response time	1	1	2	3
http://202.91.11.42/				
Cross Site Scripting	1	1	2	3
PHP allow_url_fopen enabled	1	1	2	3
Application Error message	3	4	2	3
Directory Listing	1	1	2	3
HTML form without CSRF protection	1	1	2	3
Session Cookie without HttpOnly flag set	3	1	2	3
Possible sensitive directories	1	1	2	3
Possible sensitive files	1	1	2	3

Domain-domain yang telah di-scanning secara automatic terdapat beberapa kerentanan yang telah ditemukan. Dari kerentanan yang ditemukan dihasilkan tingkat resiko keamanan yang akan dihitung dan diakumulasikan sebagai hasil akhir penilaian tingkat resiko keamanan, seperti di Tabel 7 sampai Tabel 10 menggunakan persamaan 6 dan 7:

$$\text{Likelihood} = \frac{\text{Threat Agent Factor} + \text{Vulnerability } F}{2} \quad (6)$$

$$\text{Impact} = \frac{\text{Technical Impact} + \text{Business Impact}}{2} \quad (7)$$

Skor secara keseluruhan *Likelihood* dan *Impact* dari sistem informasi harga komoditas utama adalah 5.8 dan 2.622 pada domain <http://202.91.11.42/C1>, sedangkan nilai skor domain <http://202.91.11.42/> adalah 5.638 dan 2.577 berikut skor *Likelihood* dan *Impact*, seperti pada tabel 6.

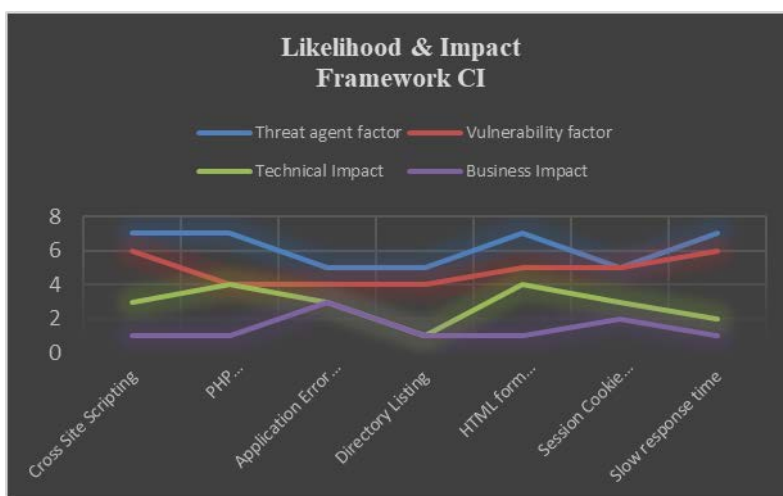
Tabel 6. Hasil Threat Agent Factors

http://202.91.11.42/CI					
Skill Level	Motive	Opportunity	Size	Total	Risk
7.28	5.28	5.85	7.57	25.98	6.495
http://202.91.11.42/					
6.87	5.12	6.25	7.75	25.99	6.4975

#### 4. KESIMPULAN

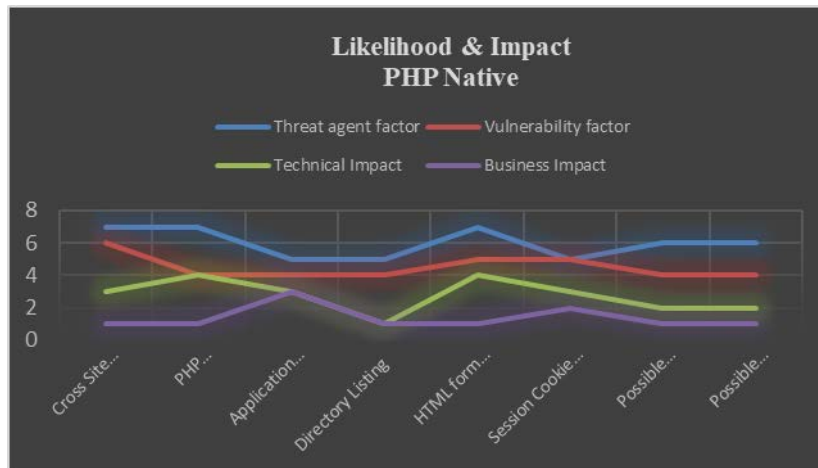
Berdasarkan hasil security assessment dengan menggunakan OWASP Risk Rating Methodology terhadap 2 sampel aplikasi berbasis website yang memiliki karakter aplikasi web yang berbeda maka dapat disimpulkan:

1. Perlu adanya penilaian risiko kerentanan keamanan terhadap aplikasi berbasis website agar bisa terlihat potensi risiko keamanan untuk mencegah dan mengatasi risiko keamanan sebelum aplikasi berbasis website di upload ke server produksi.
2. Terdapat 7 risiko dengan 3 risiko memiliki risk severity high, 2 risiko memiliki risk severity medium, 2 risiko memiliki risk severity low pada domain <http://202.91.11.42/CI>. Pada Gambar 4 menunjukkan grafik Likelihood dan Impact Framework CI sebagai platform web application development



Gambar 4. Faktor Risiko domain <http://202.91.11.42/CI>

3. Terdapat 8 risiko dengan 3 risiko memiliki risk severity high, 2 risiko memiliki risk severity medium, 3 risiko memiliki risk severity low pada domain <http://202.91.11.42/>. Pada Gambar 5 menunjukkan grafik Likelihood dan Impact PHP Native sebagai platform web application development.



Gambar 5. Faktor Risiko domain <http://202.91.11.42>

4. Hasil akhir dari resiko keseluruhan menunjukkan bahwa platform web application development menggunakan framework Codeigniter dan PHP Native sama-sama memiliki tingkat keparahan pada Likelihood dilevel Medium, sedangkan tingkat keparahan pada Impact sama-sama berada dilevel Low. Dengan melihat hasil akhir secara keseluruhan, artinya tidak ada jaminan platform web application development menggunakan framework atau PHP Native terhindar dari celah keamanan.

## 5. SARAN

Berdasarkan kesimpulan ada beberapa saran untuk dilakukan penelitian diantaranya:

1. Menggunakan metode yang lain untuk melengkapi OWASP atau menggabungkan dua metode antara OWASP dan Cobit.
2. Untuk melakukan *Security Assessment* sebaiknya juga dilakukan proses uji penetrasi sistem secara manual. Penetrasi manual membutuhkan proses dan waktu yang lama karena harus melakukan uji coba untuk menemukan dan membuktikan celah keamanan yang ada pada sistem.
3. Aplikasi berbasis web yang dibangun dan akan dilakukan *Security Assessment* ada baiknya membandingkan antara *Framework Codeigniter* dan *Laravel*.

## DAFTAR PUSTAKA

- [1] Aryasa, K., Paulus, Y. T., 2017, Implementasi Secure Hash Algorithm-1 Untuk Pengamanan Data Dalam Library Pada Pemrograman Java, *Citec Journal*, Vol. 1, No. 1, Hal 57 – 66.
- [2] Fernando, Y. I., Abdillah, R., 2016, Security Testing Sistem Penerimaan Mahasiswa Baru Universitas XYZ Menggunakan Open Source Security Testing Methodology Manual (OSSTMM), *Jurnal CoreIT*, Vol. 2, No.1, Hal 33 – 40.
- [3] Rafiq, A., Touseef, P., Ashraf, M. A., Analysis of Risks against Web Applications in MVC. *NFC IEFER Journal of Engineering and Scientific Research*, Vol. 5, No. 1, hal. 1-6
- [4] Hutagalung, R. H., Nugroho, L. E., Hidayat, R., 2017, Menentukan Dampak Resiko Keamanan Berbasis Pendekatan Owasp, *Prosiding SNATI F Ke-4 Tahun 2017*, Kudus, Indonesia.

- 
- [5] Juliharta, I. G. P. K., 2012, Business Impact Analysis Sistem dan Jaringan Komputer Menggunakan Metode Network Security Assessment, *EKSPLORA INFORMATIKA*, Vol. 2, No. 1, Hal 89 – 100.
- [6] Kesuma, M. C., Shiddiqi, A. M., Pratomo, B. A., 2013, Pencari Celah Keamanan pada Aplikasi Web, *Tugas Akhir*, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.
- [7] Web Application Security Consortium, <http://www.webappsec.org/>, yang diakses tanggal 23 Oktober 2017
- [8] Shanley, A., Johnstone, M. N., 2015, Selection of penetration testing methodologies: A comparison and evaluation, *Australian Information Security Management Conference*, Western Australia. 30 November – 2 Desember 2015.
- [9] Rao, R. M., Durgesh, P., 2010, Security risk assessment of Geospatial Weather Information System (GWIS): An OWASP based approach, (*IJCSIS*) *International Journal of Computer Science and Information Security*, Vol. 8, No. 5, Hal 24 – 32.
- [10] OWASP Risk Rating Methodology, [https://www.owasp.org/index.php/OWASP\\_Risk\\_Rating\\_Methodology](https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology), diakses tanggal 21 Desember 2017