MENENTUKAN DAMPAK RESIKO KEAMANAN BERBASIS PENDEKATAN OWASP

Robertus Halomoan Hutagalung[1*], **Lukito Edi Nugroho**[1], **Risanuri Hidayat**[2]
[1] Program Studi Teknik Elektro dan Teknologi Informasi Fakultas Teknik
Universitas Gadjah Mada
[*]Email: robertus.cio15@mail.ugm.ac.id

Abstrak

Menemukan celah keamanan adalah hal yang penting, tetapi dapat untuk memperkirakan resiko yang ada terhadap bisnis juga sama pentingnya. Pada awal siklus kebanyakan orang mengidentifikasi masalah keamanan pada arsitektur atau desain menggunakan model ancaman. Setelah itu, beberapa orang mencari masalah keamanan dengan menggunakan code review atau uji penetrasi. Atau masalah tidak akan ditemukan hingga aplikasi sudah masuk tahap produksi atau hingga aplikasi sudah diretas. Pada penelitian ini diimplementasi mekanisme metode asesmen resiko aplikasi menggunakan metode Open Web Application Security Project (OWASP) Risk Rating untuk asesmen keamanan pada aplikasi berbasis website. Penelitian ini menghasilkan tingkat resiko pada aplikasi berbasis website yang ada di dalam server Departemen Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada (DTETI UGM) menggunakan pendekatan OWASP.

***Kata kunci**: Asesmen Keamanan, Penilaian Resiko, Dampak Resiko*

1. PENDAHULUAN

1.1 Latar Belakang

Asesmen keamanan aplikasi berbasis *website* adalah hal yang penting pada siklus pengembangan aplikasi. Meskipun organisasi mempunyai banyak tindakan pencegahan tradisional pada jaringan mereka seperti *firewalls,* itu tidaklah cukup untuk melindungi aplikasi melalui *internet*. *Firewall* sendiri tidak bisa melindungi aplikasi dari ancaman luar, tetapi *firewall* tetap dibutuhkan sebagai bagaian dari keamanan jaringan. Untuk menahan aplikasi berbasis *website* dari teknik ancaman yang terus berkembang, organisasi harus melakukan asesmen pada aplikasi berbasis *website* organisasi tersebut agar organisasi tersebut memahami resiko yang organisasi tersebut hadapi. Sebagian besar kasus ini dapat ditemukan dengan melakukan *scanning* pada situs dengan menggunakan *tools* yang dapat mendeteksi jumlah celah keamanan yang ada di situs tersebut. Tindakan ini akan memberi kesempatan untuk memperbaiki teknik *coding* untuk menghilangkan celah keamanan.

Mengetahui celah keamanan sendiri tidak akan membantu manajemen untuk meningkatkan keamanan pada aplikasi. Melakukan penilaian pada resiko aplikasi dengan mempertimbangkan perbedaan faktor-faktor yang terkait dengan aplikasi akan memberikan penjelasan yang lebih dan menodong untuk mengamankan aplikasi lebih baik lagi. Dengan mengikuti pendekatan ini, organisasi dapat memperkirakan tingkat keparahan aplikasi dan dapat membuat keputusan mengenai resiko tersebut. Juga faktor resiko akan memprioritaskan masalah pada aplikasi dengan cara yang lebih baik daripada pendekatan yang dilakukan secara acak. Bagian yang memiliki lebih banyak resiko dapat secara cepat di tindak lanjutin dan selanjutnya ke prioritas berikutnya. Pada penelitian ini, percobaan sederhana telah dilakukan menggunakan metodologi *Information Systems Security Framework* (ISSAF) pada layanan berbasis *website* di Departemen Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada (DTETI UGM) dan menentukan dampak dan faktor resiko menggunakan metodologi *Open Web Application Security Project* (OWASP) *risk rating.*

1.2. KERANGKA PENGUJIAN APLIKASI

Pengujian aplikasi telah melalui proses perkembangan. *Market* pengujian aplikasi bernilai $13 *billion*[2]. Secara umum pengembangan aplikasi sampai dengan 40% dari modal merilis produk, tetapi pengujian aplikasi sama juga sampai 40% dari modal pengembangan. Oleh karena itu tahap pengujian

sangat penting untuk beberapa aplikasi untuk menghasilkan aplikasi dengan kualitas lebih baik. Pengujian aplikasi adalah proses yang digunakan untuk mengidentifikasi cara yang benar, kelengkapan, keamanan, dan kualitas aplikasi yang dikembangankan. Hal ini adalah proses dari investigasi dari segi teknik untuk menyatakan kualitas yang terkait dengan infromasi tentang produk atau aplikasi. Pengujian memberikan kritik pada aplikasi untuk tingkat yang lebih lanjut pada aplikasi pada beberapa konteks. Ada tiga pendekatan dasar untuk pengujian otomatis pada aplikasi berbasis *website*[3]. *Black box, white box* dan *gray box* memberikan perbedaan pendekatan untuk melakukan asesmen keamanan pada aplikasi berbasis *website*. *White box* dan *black box* adalah istilah yang digunakan untuk mendeskripsikan sudut pandang usaha pengujian yang diambil ketika mendesain kasus pengujian. *Black box* mengambil sudut pandang *external* dari aplikasi dan *white box* mengambil sudut pandang *internal*. Pengujian *gray box* adalah kombinasi dari pengujian *white box* dan *black box.* Pengujian ini membuat analisis kemanan dapat melakukan uji penetrasi secara secara otomatis dan manual pada aplikasi. tabel 1 menunjukan perbandingan pada teknologi pengujian.

Tabel 4 Perbandingan Teknologi Pengujian

Pros	Cons
Manual: Penetrasi dilakukan oleh satu atau sekumpulan kecil orang yang mengetahui *tools* dan *scripts*	
Pengujian baik dilakukan pada suatu target untuk fungsi aplikasi yang spesifik	1. Pengujian terbatas untuk ahli yang mungkin saja terjadi *bottlenecks.* 2. Dapat membuat *high error* dengan pengeluarn yang berulang kali. 3. Terbatasnya jangkuan aplikasi yang diuji dikarenakan waktu yang tidak banyak.
Otomatis: Pengujian spesifik untuk fungsi individu, dibuat oleh *code developer.* Tim asesmen kualitas melakukan pengujian dari sudut pandan *end user*.	
Biaya yang dikeluarkan diimbangin dengan peningkatan dalam kualitas, mengurangi usaha untuk penerimaan dan proses pengembangan yang bisa dilakukan berulang.	Memerlukan biaya yang lebih banyak untuk membuat dan melakukan perawatan daripada pengujian secara manual.
Blaxb box: mencari hanya pada *ouput* dan *input* sistem, merubah *input* yang dilakukan *user* secara normal untuk membuat aplikasi menjadi menampilkan atau melakukan beberapa hal yang tidak semestinya.	
Menggunakan *tool* pengujian yang *established* yang hanya memerlukan pengetahuan untuk menggunakan aplikasi tersebut.	Kemungkinan hanya ketika komponen aplikasi siap untuk diuji
White box : asesmen komponen individu untuk kesalahan fungsional yang spesifik, sering terjadi pada kombinasi *code tools scanning* dan *peer reviews*	
Penggunaan *tools* yang mempunyai integrasi dengan *developer* IDEs, membuat penemuan yang baik pada celah pada fungsi yang diuji	Tidak menemukan kebutuhan dan celah desain. Mungkin juga tidak menemukan celah keamanan untuk menyerang melibatkan lebih dari satu komponen atau waktu yang spesifik yang tidak diperlihatkan pada *unit testing*.
Gray box (menggunakan *framework* yang digunakan aplikasi) : kombinasi pengujian *black* dan *white box* untuk membuat pengujian yang tidak tersedia pada *commercial tools.*	
Memberikan metode yang lebih luas dengan mengkombinasikan sistem dan *unit level testing.*	Memerlukan *framework* yang yang secara spesifik digunakan pada tahap lainnya dan pada aktifitas desain. memerlukan usaha yang lebih banyak dibanding membuat pengujian pada *framework* untuk membangun aplikasi.

Pemilihan metodologi pengujian tertentu tergantung pada jumlah faktor, seperti waktu yang dialokasikan untuk melakukan asesmen, melakukan akses kedalam sumber daya aplikasi dan tujuan dari pengujian[3].

Kemanan aplikasi digunakan pada *software, hardware* dan metode prosedural untuk melindungi aplikasi dari ancaman luar.[4]

2. METODOLOGI PENELITIAN

Metode penelitian yang digunakan berdasarkan metodologi OWASP *risk rating* terdiri dari tujuh tahapan pengerjaan :

1. Menilai faktor *Threat Agent*.
2. Menilai faktor *vulnerability*,
3. Menilai faktor *technical impact,*
4. Menilai faktor *business impact,*
5. Manilai dampak resiko.

3. HASIL DAN PEMBAHASAN

Tahap uji penetrasi telah dilakukan sebelumnya pada *domain-domain* layanan berbasis *website* pada DTETI UGM yang berada di dalam *server* yang memiliki *hostname* server.te.ugm.ac.id. Pada tabel 2 akan menujukan *domain-domain* apa saja dibawah layanan di *server* DTETI UGM yang terbukti memiliki celah keamanan.

Tabel 5 *domain-domain* yang terbukti memiliki celah keamanan

Domain	Kemungkinan serangan
http://cna.te.ugm.ac.id	*Cross-site scripting*
http://me.te.ugm.ac.id	*Local file inclusion* dan *sql injection*
http://pasca.te.ugm.ac.id	*Cross-site scripting*

Berdasarkan metodologi OWASP *risk rating* maka tahap-tahap untuk menentukan dampak tersusun sebgai berikut :

Menilai Faktor *Threat Agent*

Kumpulan faktor pertama adalah yang berhubungan dengan *threat agent* yang terlibat. Tujuannya disini adalah untuk memperkirakan kemungkinan keberhasilan serangan oleh *threat agent*[4].

Berdasarkan pilihan faktor *threat agent* yang sudah disediakan oleh *OWASP risk rating sebelumnya* maka didapatkan hasil seperti pada tabel 3.

Tabel 6 Hasil *Threat Agent Factors*

	Faktor *Threat Agent*			
Jenis Ancaman	*Skill Level*	*Motive*	*Opportunity*	*Size*
Local File Inclusion	9	4	7	9
SQL Injection	9	4	7	9
Xss Reflected	6	4	7	9

Menilai Faktor *Vulnerability*

Faktor ini berhubungan dengan *vulnerability* yang terlibat. Tujuannya disini adalah untuk memperkirakan kemungkinan *vulnerability* tertentu yang terlibat ditemukan dan dieksploitasi. Asumsikan dengan *threat agent* yang sudah dipilih[4].

Berdasarkan pilihan faktor *Vulnerability* yang sudah disediakan oleh *OWASP risk rating sebelumnya* maka didapatkan hasil seperti pada tabel 4.

Tabel 7 Hasil *Vulnerability Factor*

	Faktor *Vulnerability*			
Jenis Ancaman	*Ease of Discovery*	*Ease of Exploit*	*Awareness*	*Intrusion Detection*
Local File Inclusion	9	3	9	1
SQL Injection	9	3	9	1
Xss Reflected	9	5	9	1

Menilai Faktor *Technical Impact*

Dampak teknis bisa dipecah kedalam faktor yang selaras dengan area keamanan tradisional yang menjadi perhatian : *confidential, integrity, availability,* dan *accountability*. Tujuannya adalah untuk memperkirakan besarnya dampak ke sistem jika celah keamanan itu dieksploitasi[4].

Berdasarkan pilihan faktor *technical impact* yang sudah disediakan oleh *OWASP risk rating sebelumnya* maka didapatkan hasil seperti pada tabel 5.

Tabel 8 Hasil Faktor *Technical Impact*

Jenis Ancaman	Faktor *Technical Impact*			
	Loss of Confidentiality	*Loss of Integrity*	*Loss of Availability*	*Loss of Accountability*
Local File Inclusion	9	9	9	9
SQL Injection	4	1	1	9
Xss Reflected	2	1	1	9

Menilai Faktor *Business Impact*

Dampak bisnis berasal dari dampak teknis, tapi memerlukan pehamanan mendalam tentang apa yang penting dari perusahaan yang menjalankan aplikasi tersebut. Pada umumnya, anda harus bertujuan untuk mengarahkan resiko anda dengan dampak bisnis, terutama jika pengguna anda adalah tingkat eksekutif. Resiko bisnis inilah yang memberikan alasan investasi dalam memperbaiki masalah keamanan.

Banyak perusahaan mempunyai panduan klasifikasi aset atau referensi dampak bisnis untuk membantu merumuskan apa yang penting pada bisnis mereka. Standar ini dapat membantu anda fokus pada hal yang benar-benar penting untuk keamanan. Jika ini tidak tersedia, maka perlu untuk berbicara dengan orang yang paham bisnis agar mereka yang mengambil keputusan tentang apa saja yang penting.

Faktor dibawah ini adalah area umum untuk banyak bisnis. Tapi area ini bahkan lebih khusus lagi untuk perusahaan dibandingkan faktor yang berhubungan dengan *threat agent*, *vulnerability,* dan *technial impact*[4].

Tabel 9 Hasil Faktor *Business Impact*

Jenis Ancaman	Faktor *Business Impact*			
	Financial Damage	*Reputation Damage*	*Non-Compliance*	*Privacy Violation*
Local File Inclusion	3	4	5	7
SQL Injection	1	1	2	3
Xss Reflected	1	1	2	3

Menentukan Dampak Resiko

Pada bagian ini hasil penilaian kemungkinan dan perkiraan dampak yang dikumpulkan sebelumnya, digunakan untuk menghitung tingkat keparahan dampak dan resiko secara keseluruhan. Maka kita masukan data-data yang sebelumnya kita dapatkan ke dalam rumus-rumus mengikuti metodologi *OWASP risk rating* seperti berikut :

a) *Threat*

$$TAR = \frac{SL + M + O + S}{4}$$

TAR = *Threat Agent Rating*

SL = *Skill Level*

M = *Motive*

S = *Size*

O = *Opportunity*

Hasil :

- *Local File Inclusion* $\frac{9+4+7+9}{4} = 7.25$
- *Sql Injection* $\frac{9+4+7+9}{4} = 7.25$
- *Xss Reflected* $\frac{6+4+7+9}{4} = 6.5$

b) *Vulnerability*

$$VR = \frac{EoD + EoE + A + ID}{4}$$

VR = *Vulnerability Rating*

EoD = *Ease of Discovery*

EoE = *Ease of Exploit*

A = *Awareness*

ID = *Intrusion Detection*

Hasil :

- *Local File Inclusion* $\frac{9+3+9+1}{4} = 5.5$
- *Sql Injection* $\frac{9+3+9+1}{4} = 5.5$
- *Xss Reflected* $\frac{9+5+9+1}{4} = 6$

c) *Likelihood*

$$LR = \frac{TAR + VR}{2}$$

LR = *Likelihood Rating*

Hasil :

- *Local File Inclusion* $\frac{7.25+5.5}{2} = 6.375$
- *Sql Injection* $\frac{7.25+5.5}{2} = 6.375$
- *Xss Reflected* $\frac{6.5+6}{2} = 6.25$

d) *Impact*

Dampak bisnis dapat dihitung menggunakan rumus :

$$BI = \frac{FD + RD + Nc + PV}{4}$$

BI = *Business Impact*

FD = *Financial Damage*

RD = *Reputation Damage*

Nc = *Non-compliance*

PV = *Privacy Violation*

Hasil :

- *Local fi*le *inclusion* $\frac{3+4+5+7}{4} = 4.75$
- *Sql Injection* $\frac{1+1+2+3}{4} = 1.75$
- *Xss Reflected* $\frac{1+1+2+3}{4} = 1.75$

Dampak teknis dapat dihitung menggunakan rumus :

$$TI = \frac{LoC + LoA + LoI + LoAc}{4}$$

TI = *Technical Impact*

LoC = *Loss of Confidentially*

LoA = *Loss of Availability*

LoI = *Loss of Integrity*

LoAc = *Loss of Accountability*

Hasil :

- *Local File Inclusion* $\frac{9+9+9+9}{4} = 9$
- *Sql Injection* $\frac{4+1+1+9}{4} = 3.75$
- *Xss Injection* $\frac{2+1+1+9}{4} = 3.25$

Penilaian dampak akhir dihitung sebagai rata-rata dari dua nilai diatas:

$$FIR = \frac{BI + TI}{2}$$

FIR = *Final Impact Rating*

Hasil :

- *Local File Inclusion* $\frac{4.75+9}{2} = 6.875$
- *Sql Injection* $\frac{1.75+3.75}{2} = 2.75$
- *Xss Injection* $\frac{3.25+1.75}{2} = 2.5$

Maka berdasarkan penilaian dari rumus yang merujuk ke metodologi *OWASP risk rating* di atas dapat dilihat hasilnya jika mengikuti tabel 7 adalah seperti berikut :

Tabel 10 Tabel Tingkat Kemungkinan Resiko dan Dampak

Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Berdasarkan penilaian dan rumus yang dibahas sebelumnya, maka dapat dibuat *script HTML* sederhana guna untuk menghitung dan membuat tabel secara otomatis yang hasilnya dapat dilihat *url* https://obet.us/owasprr.html. Merujuk ke tabel 7 maka celah *local file inclusion* terhadap server.te.ugm.ac.id dapat di *generate* dan menghasilkan penilaian kemungkinan dengan angka 6.375 dengan tingkat kemungkinan secara keseluruhan adalah *High* dan dampak teknis secara keseluruhan menghasilkan nilai 9 yang berarti tingkat dampak pada sisi teknis adalah *High,* sedangkan dampak bisnis secara keseluruhan menghasilkan nilai 4.75 yang mana berarti tingkat dampak pada sisi teknis adalah *Medium* sepert pada gambar 1.

Overall Likelihood		Overall Technical Impact		Overall Business Impact	
6.375	HIGH	9	HIGH	4.75	MEDIUM

Gambar 7 Hasil kemungkinan dan dampak keseluruhan pada celah *lfi*

Merujuk ke tabel 7 maka celah *Sql Injection* terhadap server.te.ugm.ac.id dapat di *generate* dan menghasilkan penilaian kemungkinan dengan angka 6.375 dengan tingkat kemungkinan secara keseluruhan adalah *High* dan dampak teknis secara keseluruhan menghasilkan nilai 3.75 yang berarti tingkat dampak pada sisi teknis adalah *Medium,* sedangkan dampak bisnis secara keseluruhan menghasilkan nilai 1.75 yang mana berarti tingkat dampak pada sisi bisnis adalah *Low* seperti pada gambar 2.

Overall Likelihood		Overall Technical Impact		Overall Business Impact	
6.375	HIGH	3.75	MEDIUM	1.75	LOW

Gambar 8 Hasil kemungkinan dan dampak keseluruhan pada celah *Sqli*

Merujuk ke tabel 7 maka celah *Xss Injection* terhadap server.te.ugm.ac.id dapat di *generate* dan menghasilkan penilaian kemungkinan dengan angka 6.25 dengan tingkat kemungkinan secara keseluruhan adalah *High* dan dampak teknis secara keseluruhan menghasilkan nilai 3.25 yang berarti tingkat dampak pada sisi teknis adalah *Medium,* sedangkan dampak bisnis secara keseluruhan menghasilkan nilai 1.75 yang mana berarti tingkat dampak pada sisi teknis adalah *Low* seperti pada gambar 3.

Overall Likelihood		Overall Technical Impact		Overall Business Impact	
6.25	HIGH	3.25	MEDIUM	1.75	LOW

Gambar 9 Hasil Keseluruhan Dampak Bisnis dan Teknis

4. KESIMPULAN

Melihat tahap sebelumnya maka penelitian ini memberikan asesmen pada aplikasi berbasis *website* yang ada pada DTETI UGM, yang mana adalah domain-domain layanan yang berada pada *server* yang menggunakan *hostname* server.te.ugm.ac.id, yang di dalamnya terdapat tiga domain layanan yang terbukti mempunyai celah keamanan yaitu cna.te.ugm.ac.id, pasca.jteti.ugm.ac.id dan

me.te.ugm.ac.id, pada prosesnya penelitian ini menampilkan metodologi resiko ancamanan keamanan, standar manajemen celah keamanan dengan melakukan penilaian pada resiko yan terkait. Penelitian ini menghasilkan kuantifikasi tingkat resiko yang berbeda pada layanan berbasis *website* di DTETI UGM. Dengan hasil ini pihak manajemen pada DTETI UGM dapat memprioritaskan celah mana yang seharusnya lebih dulu dilakukan perbaikan dengan melihat hasil dari dampak teknis, bisnis dan dampak secara keseluruhan agar layanan menjadi lebih baik.

DAFTAR PUSTAKA

[1] OWASP, "OWASP Risk Rating Methodology - OWASP." [Online]. Available: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology. [Accessed: 19-Jun-2017].

[2] L. Corporation, "Software testing: The continuous evolution of software testing." Logigear Whitepaper, 2008.

[3] Dan Cornell, "Web application testing: The difference between black, gray and white box testing." [Online]. Available: http://searchsoftwarequality.techtarget.com/tip/Web-application-testing-The-difference-between-black-gray-and-white-box-testing. [Accessed: 19-Jun-2017].

[4] C. Kane, "Vulnerability Assessment Process," *UC*, 2015.