Penilaian Risiko *Secure Software Development Life Cycle* pada Sistem Informasi Senat Mahasiswa Perguruan Tinggi XYZ Menggunakan Metode OWASP

Hermawan Setiawan[1)], Muhammad Novrizal Ghiffari[2)]
(1) Rekayasa Kriptografi, Politeknik Siber dan Sandi Negara, hermawan.setiawan@poltekssn.ac.id
(2) Badan Siber dan Sandi Negara, novrizal.ghiffari@bssn.go.id

Abstrak

Penelitian ini bertujuan untuk menilai risiko keamanan aplikasi web Sistem Informasi Senat Mahasiswa di Perguruan Tinggi XYZ dengan Mengunakan metode OWASP (Open Web Application Security Project) dalam kerangka Secure Software Development Life Cycle (SDLC). Metode OWASP dikenal sebagai standar internasional untuk mengidentifikasi dan menangani berbagai kerentanan dalam pengembangan aplikasi web. Penilaian ini dilakukan melalui beberapa tahap yang meliputi perencanaan, analisis risiko, desain, implementasi, pengujian, dan pemeliharaan aplikasi. Hasil perhitungan resiko untuk web senatmahasiswa_XYZ adalah 6.05 pada likelilhood dan 3.49 pada impact. Hasil akhir dari penelitian ini menunjukkan 7 risiko dengan tingkatan kerentanan yang berbeda-beda mulai dari high hingga low. Penerapan OWASP dalam siklus pengembangan perangkat lunak yang aman secara signifikan dapat mengurangi risiko keamanan, seperti injeksi SQL, cross-site scripting (XSS), dan kelemahan autentikasi. Selain itu, temuan ini memberikan rekomendasi praktis untuk meningkatkan keamanan aplikasi di lingkungan pendidikan tinggi. Dengan demikian, penelitian ini memberikan kontribusi penting bagi pengembangan sistem informasi yang lebih aman dan terpercaya, khususnya dalam konteks manajemen organisasi mahasiswa.

Kata kunci: penilaian risiko, secure software development life cycle, owasp, keamanan aplikasi web, sistem informasi

Abstract

This study aims to assess the security risks of the Student Senate Information System web application at XYZ College using the OWASP (Open Web Application Security Project) method within the Secure Software Development Life Cycle (SDLC) framework. The OWASP method is known as an international standard for identifying and handling various vulnerabilities in web application development. This assessment is carried out through several stages including planning, risk analysis, design, implementation, testing, and application maintenance. The results of the risk calculation for the XYZ student senate web are 6.05 in likelihood and 3.49 in impact. The final results of this study show 7 risks with varying levels of vulnerability ranging from high to low. The application of OWASP in a secure software development cycle can significantly reduce security risks, such as SQL injection, cross-site scripting (XSS), and authentication weaknesses. In addition, these findings provide practical recommendations for improving application security in higher education environments. Thus, this study makes an important contribution to the development of more secure and reliable information systems, especially in the context of student organization management.

Keywords: risk assessment, secure software development life cycle, owasp, web application security, information systems, colleges.

1. PENDAHULUAN

Web memiliki dampak besar dan permanen pada hidup, dan pekerjaan itu telah mengalami perubahan yang tak dapat diubah. Web telah menjadi platform utama untuk menyebarkan aplikasi bisnis dan sosial serta mengatur sistem informasi. Banyak organisasi telah memperluas cakupan sistem berbasis web yang bisa diakses oleh teknologi selular. Oleh karena itu, aplikasi berbasis web sekarang menyediakan berbagai konten dan fungsi untuk sejumlah besar pengguna, dan memiliki banyak tujuan yang berbeda. Karena keberadaannya di mana-mana, harapan dan persyaratan untuk aplikasi web telah meningkat pesat selama bertahun-tahun [1]. Dengan meningkatnya implementasinya, juga berbanding lurus dengan ancaman yang ada dari berbagai pihak dengan berbagai macam teknik ancaman. Seringkali masalah kerawanan atau keamanan di anggap tidak penting [2]. Padahal hal tersebut berdampak besar di berbagai aspek.

Pada Senat Mahasiswa Perguruan tinggi XYZ , Mengunakan Sistem Informasi berbasis web untuk mempermudah jalannya organisasi dan birokrasi. Namun seiring berjalannya perkembangan, kerawanan dan kerentanan bisa terjadi pada platform tersebut. Untuk menentukan tindakan apa yang dilakukan untuk menanganinya perlu adanya penilaian terhadap risiko yang diukur melalui kemungkinan dampak terburuk.

Ada beberapa faktor yang membuat program pada website memiliki celah, yaitu karena kode yang salah atau konfigurasi yang salah [3]. Pada webappsec.org merilis bahwa *SQL Injection* (26.38%) dan XSS (35.57%) merupakan jenis serangan yang sering digunakanakan. *Open Web Application*

Security Project (OWASP) melaporkan bahwa *SQL injection* tetap menjadi salah satu dari 10 ancaman keamanan aplikasi web terbesar, dengan sekitar 9% dari semua pelanggaran aplikasi web disebabkan oleh *SQL injection* [4], [5]. Menurut laporan Akamai, XSS menyumbang 30% dari semua serangan aplikasi web yang dilaporkan. Serangan *SQL Injection* dapat menyebabkan kerusakan data yang parah, kehilangan data penting, atau pengambilalihan penuh sistem. Hal tersebut tidak hanya dapat mengganggu operasional sistem, menimbulkan downtime, dan merusak reputasi Senat Mahasiswa bahkan perguruan tingginya [6].

Beberapa pendekatan dapat digunakanakan untuk mendeteksi kerentanan seperti, ISSAF, OSSTMM, OWASP dan NIST. Namun di antara metode tersebut yang paling populer dan baik untuk platform web adalah OWASP [1]. Metode OWASP dipilih karena ia menawarkan pendekatan yang komprehensif, terpercaya, dan relevan untuk mengidentifikasi dan mengurangi risiko keamanan yang dihadapi oleh Sistem Informasi Senat Mahasiswa. Fokus metoda OWASP dibandingkan metoda lain yakni pada ancaman paling umum dan kritis. Kelebihan lainnya adalah panduan yang mudah dipahami dan langkah-langkah mitigasi yang jelas, sehingga menjadikannya pilihan yang ideal untuk organisasi yang ingin memastikan bahwa sistem informasi mereka aman dan terlindungi dari serangan. Karena alasan tersebut penelitian ini Mengunakan metode OWASP untuk menilai tingkat risiko Sistem Informasi Senat Mahasiswa Perguruan tinggi XYZ.

2. LANDASAN TEORI

2.1 *Secure Software Development Life Cycle*

Secure Software Development Life Cycle (SSDLC) merupakan proses teknik mengembangkan sebuah perangkat atau aplikasi yang aman secara sistematis baik dari segi pengembangan, pemeliharaan, dan proses deliver sebuah solusi kemanan yang baik [7]. Mengembangkan dengan SSDLC mengikuti proses praktik yang baik pada *desain*, *testing*, *reviews*, *risk management*, serta *people management* [7].

2.2 *Risk Assessment*

Risk Assessment adalah kegiatan penilaian tingkat risiko di setiap bidang dengan kemungkinan yang bervariasi. Secara umum meliputi analisis risiko, evaluasi risiko serta identifikasi [8]. *Risk assessment* menghasilkan nilai melalui hasil komparasi dari berbagai faktor atau kriteria yang ada yang sudah ditentukan nilainya melalui standard tertentu [8], seperti OWASP, NIST, ISO.

2.3 Metode OWASP

OWASP merupakan sebuah metode atau pendekatan untuk melakukan penilaian risiko yang berfokus pada web app. OWASP juga dikenal sebagai organisasi yang berdiri pada tahun 2004 yang memiliki kelengkapan petunjuk untuk melakukan uji penetrasi [1]. OWASP merupakan pendekatan yang sederhana digunakan untuk menghitung dan menilai risiko pada suatu aplikasi, yang bertujuan untuk mengambil keputusan terkait langkah apa yang dilakukan terhadap risiko tersebut [1].

2.4. OWASP ZAP

OWASP Zed Attack Proxy atau yang dikenal dengan sebutan OWASP ZAP merupakan platform yang biasanya digunakan untuk melakukan *penetration testing* dengan tujuan mencari celah keamanan pada aplikasi berbasis web [9]. OWASP ZAP bekerja dengan melakukan *scanning* secara terotomatisasi [10]. OWASP diakui secara global sebagai sumber otoritatif dalam bidang keamanan aplikasi web. *OWASP Top Ten* merupakan standar *de facto* yang digunakan oleh pengembang, auditor keamanan, dan perusahaan di seluruh dunia untuk mengidentifikasi dan mengatasi ancaman keamanan aplikasi [11]. OWASP tidak hanya berfokus pada identifikasi ancaman, tetapi juga memberikan panduan detail tentang cara mencegah dan mengurangi risiko tersebut. Ini termasuk praktik pengkodean aman, pengujian penetrasi, dan strategi mitigasi lainnya [12]. OWASP memiliki komunitas yang luas dan aktif, serta menyedi berbagai sumber daya tambahan, seperti proyek pengujian, alat keamanan, dan dokumentasi yang dapat membantu organisasi mengimplementasikan keamanan yang lebih baik [13].

2.5 *Acunetix Web Vulnerability*

Acunetix Web Vulnerability merupakan sebuah tools yang memberikan layanan untuk pengujian keamanan aplikasi berbasis web secara otomatis, cara kerja aplikasi ini yaitu mengaudit aplikasi dengan memeriksa kerentanan seperti SQL Injection, XSS, serta kerentanan lainnya [11]. *Acunetix* merupakan *tools* yang sudah terotomatisasi yang kebanyakan digunakan oleh perusahaan untuk membantu melakukan *scanning* pada aplikasi web mereka dengan tujuan mengidentifikasi dan menemukan kerentanan [14].

3. METODE PENELITIAN

Metodologi penelitian yang digunakan pada penelitian ini adalah *Secure Software Development Life Cycle* (SDLC) dengan metode OWASP berdasarkan pada penelitian [1]. Pada penelitian tersebut dijelaskan langkah yang dilakukan dalam penelitian ini meliputi *Diagnosing, Action Planning, Action Taking, Evaluating, Learning* [15]. Pada penelitian ini tahap *Action Taking* dilakukan *testing* dengan aplikasi OWASP ZAP dan *acunetix*, dan hasil dari *scanning* kedua aplikasi tersebut digabungkan menjadi satu.

3.1 *Diagnosing*

Pada tahap ini penelitian dilakukan untuk identifikasi masalah dengan melakukan diagnosa pada sistem web Sistem Informasi Senat Mahasiswa Perguruan tinggi XYZ.

3.2 *Action Planning*

Pada tahap ini dilakukan pemahaman terhadap permasalahan yang terkait, kemudian merencanakan tindakan yang tepat untuk menyelesaikan masalah terkait. Penelitian ini dimulai dengan menyusun rencana pengujian pada sistem web Sistem Informasi Senat Mahasiswa Perguruan tinggi XYZ .

3.3 *Action Taking*

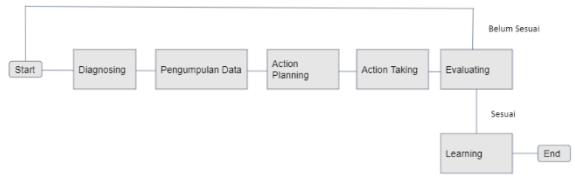
Pada tahap ini dilakukan tindakan implementasi dari rencana yang telah disusun. Dengan pengujian secara langsung dilakukan investigasi untuk mendapatkan informasi kelemahan sistem beserta tipe ancaman yang ada pada sistem web Sistem Informasi Senat Mahasiswa Perguruan tinggi XYZ.

3.4 *Evaluating*

Pada tahap ini dilakukan evaluasi terhadap hasil dari implementasi sebelumnya dan dapat diambil kesimpulan dari langkah sebelumnya.

3.5 *Learning*

Pada tahap terakhir ini dilakukan *review* terhadap hasil dari langkah-langkah yang telah dilakukan. Alur penelitian merupakan serangkaian langkah yang harus dilakukan secara runtun pada penelitian ini sehingga mencapai sebuah kesimpulan. Alur penelitian pada penelitian ini sebagai berikut:



Gambar 1. *Alur penelitian*

Gambar 1 dimulai dari langkah *start, diagnosing, pengumpulan data, action planning, action tacking* lalu *evaluating*. Pada langkah *evaluating* sekiranya tidak sesuai kembali ke langkah *start* dan jika sesuai maka lanjut pada langkah *learning* dan terakhir *end*.

Pada *Diagnosing* melakukan audit keamanan menyeluruh pada kode sumber aplikasi untuk mendeteksi kerentanan. *Gun* alat pemindai untuk mengidentifikasi potensi ancaman seperti *SQL Injection*, XSS, dan CSRF. Langkah *Action Planning* adalah mengurutkan kerentanan berdasarkan tingkat keparahan dan potensi dampaknya. Kerentanan kritis, seperti *SQL Injection* dan otentikasi yang tidak aman, harus diatasi terlebih dahulu.

Pada *Action Taking* merupakanakah langkah memperbaiki kode yang rentan sesuai dengan rencana perbaikan. Setelah perbaikan diterapkan, lakukan pengujian untuk memastikan bahwa kerentanan telah diatasi tanpa memperkenalkan masalah baru. Di akhir yaitu *Evaluating* adalah melakukan audit keamanan kedua setelah tindakan perbaikan diterapkan untuk memastikan bahwa semua kerentanan telah diatasi dan tidak ada masalah baru yang muncul.

4. HASIL DAN PEMBAHASAN

Sistem Informasi Senat Mahasiswa Perguruan tinggi XYZ merupakan sebuah platform sistem informasi yang dibangun dengan *framework codeigniter* dan di-*hosting* mengunakan VPS sehingga dapat diakses secara umum. Sistem informasi ini dapat membantu birokrasi kemahasiswaan yang berjalan pada Senat Mahasiswa, seperti distribusi Tanda Tangan Elektronik (TTE), pendataan puasa, izin bermalam, daftar kegiatan, pelanggaran, dan skor CTF untuk setiap mahasiswa serta memuat data mahasiswa dan tanda tangan elektronik beserta *password*-nya yang menjadi nilai penting yang harus diamankan. Pada penelitian ini dijelaskan juga runtutan alur bisnis dan penggunaan program sehingga dapat disusun skenario dan diagram terkait serangan yang mungkin saja dapat terjadi sebagai berikut:

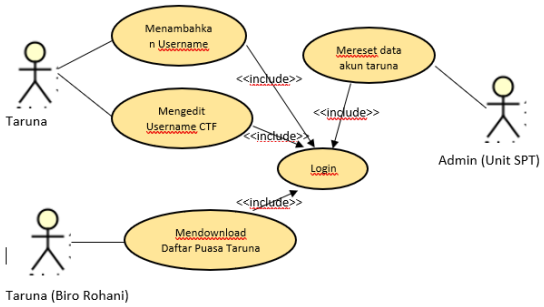
1. *Use Case* Skenario
   a. Menambahkan *Username* CTF Mahasiswa
   Tabel 1 menunjukkan upaya antisipasi adanya serangan pada *use case login*.

Tabel 1. *Use Case* Skenario Menambahkan Username CTF

Tujuan	Mahasiswa dapat menambahkan username CTF untuk kegiatan CTF Day
Aktor	Mahasiswa
Kondisi Awal	Login Valid
Skenario Utama	1. Mahasiswa memilih menu dashboard 2. Mahasiswa mengklik tombol “input username ctf” 3. Mahasiswa menginputkan username pada field yang sudah ada
Skenario Alternatif	1. Jika mahasiswa salah dalam input username maka tidak menampilkan *score* CTF 2. Jika mahasiswa mengisi kosong maka diperingati oleh sistem
Kondisi Akhir	Sistem merekam *username* CTF mahasiswa ke database

   b. Mengedit *Username* CTF Mahasiswa
   Upaya antisipasi adanya serangan *use case edit username* ditunjukkan pada Tabel 2.
   c. Men-*download* Daftar Puasa Mahasiswa
   Tabel 3 menunjukkan upaya antisipasi adanya serangan pada *use case* daftar puasa.
   d. Mereset Data *Password* Mahasiswa
   Upaya antisipasi adanya serangan pada *use case reset password* ditunjukkan pada Tabel 4.
2. *Use Case Diagram*

Diagram alur mengenai keterkaitan antara seluruh user dengan berbagai skenario yang ada ditunjukkan pada Gambar 2.



Gambar 2. *Use Case* Diagram Menambahkan *Username* CTF

Tabel 2. *Use Case* Skenario Mengedit *Username* CTF

Tujuan	Mahasiswa dapat mengedit *username* CTF untuk kegiatan CTF *Day*
Aktor	Mahasiswa
Kondisi Awal	*Login Valid* dan *Username* CTF sudah ada
Skenario Utama	1. Mahasiswa memilih menu dashboard 2. Mahasiswa mengklik tombol "ganti username ctf" 3. Mahasiswa menginputkan *username* pada *field* yang sudah ada
Skenario Alternatif	1. Jika mahasiswa salah dalam *input username* maka tidak menampilkan *score* CTF 2. Jika mahasiswa mengisi kosong maka diperingati oleh sistem
Kondisi Akhir	Sistem merekam username CTF Mahasiswa ke *database*

Tabel 3. *Use Case* Skenario Men-*download* Daftar Puasa Mahasiswa

Tujuan	Untuk pendataan daftar puasa sunnah mahasiswa
Aktor	Untuk pendataan daftar puasa sunnah mahasiswa
Kondisi Awal	*Login Valid* dan Memiliki Jabatan Biro Rohani
Skenario Utama	1. Biro rohani memilih menu puasa 2. Biro rohani mengklik tombol "download data" 3. Biro rohani mendapatkan file berupa .csv atau .pdf
Skenario Alternatif	1. Jika daftar tidak tersedia maka muncul peringatan bahwa data kosong 2. Jika daftar puasa mahasiswa masih menyala maka file daftar tidak bisa di download
Kondisi Akhir	Biro rohani mendownload file daftar puasa dari sistem database

Tabel 4. *Use Case* mereset *password* akun mahasiswa

Tujuan	Mahasiswa yang bermasalah dengan kredensial akun dapat mengakses kembali akun
Aktor	Admin (Unit SPT)
Kondisi Awal	Login Valid dan memilik jabatan Unit SPT
Skenario Utama	1. Admin memilih menu reset akun 2. Admin mencari nama mahasiswa terkait 3. Admin mengklik tombol reset password pada mahasiswa terkait
Skenario Alternatif	1. Jika admin salah dalam menekan, dapat melakukan undo 2. Jika admin tidak valid maka tidak dapat mengakses reset akun
Kondisi Akhir	Sistem mereset *database password* mahasiswa yang bermasalah dengan kredensial akun

3. *Abuse Case* Skenario

Pada bagian *abuse case* skenario Tabel 5 hingga Tabel 8 berturut-turut menunjukkan kemungkinan adanya eksploitasi terhadap suatu komponen, jangkauan eksploitasi berikut dengan hasilnya. Tabel 5 menunjukkan upaya serangan pada *abuse case* injeksi *script*. Tabel 6 menunjukkan upaya serangan pada *abuse case* injeksi *file*. Tabel 7 menunjukkan upaya serangan pada *abuse case privilages escalation*. Tabel 8 menunjukkan upaya serangan pada *abuse case* eksekusi fungsi tanpa izin admin.

a. Menginjeksi *script* berbahaya

Tabel 5. *Abuse Case* Skenario Menginjeksi *script* berbahaya

Tujuan	Membuat tampilan web berubah
Aktor	Penyerang
Komponen yang dieksploitasi	*Input Field Username* CTF
Jangkauan Eksploitasi	1. *Front end fitur score board* 2. Database
Hasil	Merubah tampilan pada *list score board* CTF

b. Menginjeksi *file* berbahaya

Tabel 6. *Abuse Case* Skenario Menginjeksi *file* berbahaya

Tujuan	Menanamkan *backdoor*
Aktor	Penyerang
Komponen yang dieksploitasi	*Input Field Username* CTF
Jangkauan Eksploitasi	1. Direktori sistem publik dari web tersebut 2. Database
Hasil	Penyerang dapat memanggil *backdoor* berupa *shell* kemudian bebas melakukan apapun layaknya *root/admin*

c. *Priveleges Escalation*

Tabel 7. *Abuse Case* Skenario previlages escalation

Tujuan	Mengunakan *privileges* layaknya yang memiliki kepentingan
Aktor	Penyerang
Komponen yang dieksploitasi	*Download* daftar puasa dan buka tutup fitur puasa
Jangkauan Eksploitasi	1. Fitur Puasa 2. File database daftar puasa
Hasil	Penyerang dapat membuka fitur daftar puasa tidak pada waktunya dan dapat mendownload daftar list puasa tanpa harus memiliki kredensial biro rohani

d. Mengeksekusi fungsi tanpa seizin admin

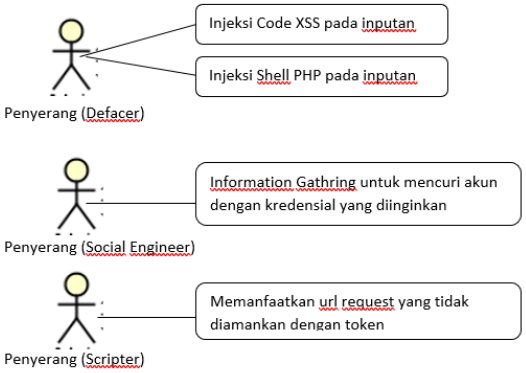
Tabel 8. *Abuse Case* Skenario eksekusi fungsi tanpa izin admin

Tujuan	Mereset akun mahasiswa tanpa login dan tanpa sebagai admin
Aktor	Penyerang
Komponen yang dieksploitasi	Url pada *request method web*
Jangkauan Eksploitasi	1. Database 2. Akun mahasiswa
Hasil	Penyerang dengan melakukan injeksi url *request* tanpa izin dapat melakukan reset akun mahasiswa kepada siapapun

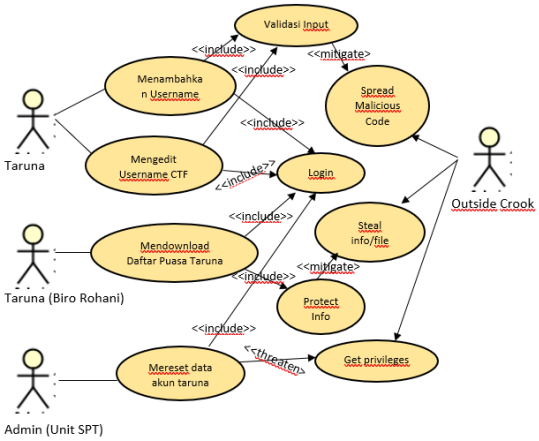
4. *Abuse Case* Diagram

Gambar 3 menunjukan diagram serangan baik

berupa *Defacer, Social Engineering*, maupun *Scripter* yang dapat dilakukan pada komponen yang menjadi target eksploitasi. Gambar 4. menunjukkan *outside crook* dapat melakukan melakukan tiga jenis serangan pada sistem [4].



Gambar 3. Diagram Serangan



Gambar 4. *Abuse Case* diagram

5. HASIL PENGUJIAN KEAMANAN

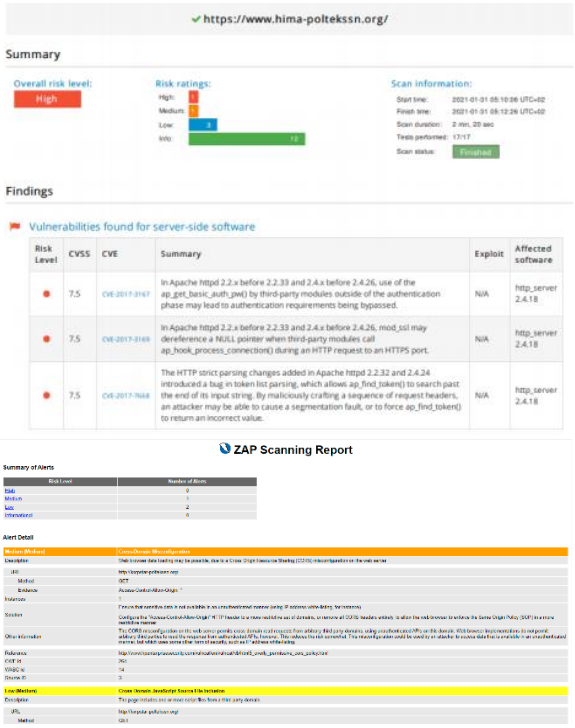
5.1 *Software Security Testing*

Pada tahap *Diagnosing* ini dilakukan *pentesting* dengan *tools* OWASP ZAP dan *Acunetix* untuk mengetahui titik kerentanan dan tingkat kerentanannya. Gambar 5 menunjukkan hasil dari *tools Acuntix* dan OWASP ZAP dengan hasil level risiko medium sebanyak 1 peringatan sedangkan risiko rendah sebanyak 2 peringatan.

Tabel 9 memperlihatkan kerentanan dalam berbagai tingkatan. Pada domain senatmahasiswa_XYZ sendiri masih banyak ditemukan kerentanan baik itu dari sisi *user-side* maupun *server-side*.

5.2 *Manual Testing*

Pada tahap *Diagnosing* ini dilakukan juga *testing* manual yang bertujuan melakukan *scanning* kerentanan untuk menemukan kerentanan yang tidak dapat terdeteksi oleh *software scanning* kerentanan. Proses ini dilakukan dengan cara *manual input* dan *manual request* pada web senatmahasiswa_XYZ.



Gambar 5. Hasil dari *tools Acunetix* dan OWASP ZAP

Tabel 9. Kerentanan dari Web Senatmahasiswa_XYZ

Domain	Kerentanan
https://senatmahasiswa_XYZ.org/	*Cross-Domain Misconfiguration*
	Error Massage on Page
	Cross Site Scripting
	Session Cookie without HttpOnly flag set
	Information Disclosure - Suspicious Comments
	HTML form without CSRF protection

Berikut kerentanan yang ditemukan dari hasil manual testing:

1. *Input Filtering*

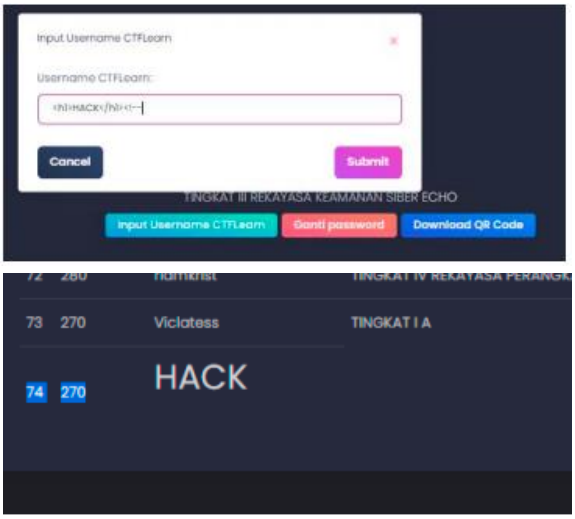
Dilakukan percobaan inputan karakter yang mengandung sintaks *code* pada *field username* CTF dan berhasil di eksekusi pada sisi server, Gambar 6 merupakan hasil dari *manual testing* yang dilakukan pada tahap ini.

2. *Session Hijaking*

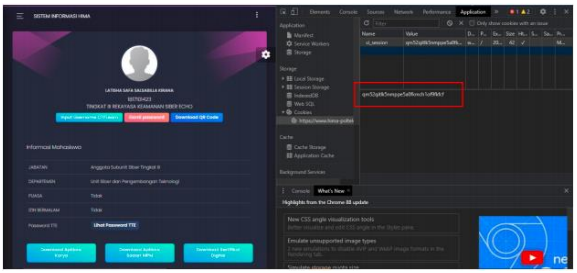
Dilakukan percobaan untuk mencuri sesi akun yang sudah login dan mengunakan akannya pada sesi yang belum login. Mendapatkan hasil pada sesi yang belum login dapat login dengan kredensial sesi yang dicuri sebagaimana ditampilkan pada Gambar 7.

3. *Cross Site Request Forgery*

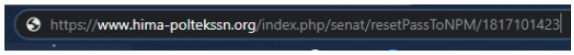
Pada tahap ini dilakukan percobaan untuk melakukan *request post* url tanpa login dengan parameter NPM mahasiswa dan mendapatkan hasil password akun dengan NPM tersebut berhasil di-*reset* tanpa izin admin. Gambar 8 menampilkan hasil dari manual *testing* yang dilakukan.



Gambar 6. Hasil dari *manual testing* pada input *filtering*



Gambar 7. Hasil dari *manual testing* pada *cookie*



Gambar 8. Hasil dari *manual testing* pada CSRF

5.3 Penilaian Risiko

Pada tahap *Action Planning* dengan berdasarkan OWASP *risk rating* terdapat tahapan untuk menentukan besarnya risiko yang ada, di antaranya sebagai berikut [16]:

1. *Threat Agent Factors*

Faktor ini bertujuan untuk mengetahui perkiraan besar kemungkinan serangan yang dapat dilakukan oleh *threat agent*. Berikut kriteria *threat agent factor* dengan hasil penilaian masing-masing keterampilan ada di dalam () [17]:

a. *Skill level*
Keterampilan penetrasi keamanan (9), keterampilan jaringan dan pemrograman (6), pengguna komputer tingkat lanjut (5), beberapa keterampilan teknis (3), tidak ada keterampilan teknis (1).

b. *Motive*
Tidak ada reward (1), memungkinkan mendapat reward (4), mendapatkan reward yang tinggi (9).

c. *Opportunity*
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).

d. *Size*
Pengembang (2), administrator sistem (2), pengguna intranet (4), mitra (5), pengguna terotentikasi (6), pengguna internet anonim (9).

Adapun untuk mendapatkan hasil dari penilaian berdasarkan *threat agent*, digunakan rumus sebagai berikut ini:

$$threat\ agent = \frac{skill\ level + motive + oport + size}{4}$$

Tabel 10 menunjukkan hasil penilaian dari threat agent pada senatmahasiswa_XYZ. Tabel 10. menyajikan tujuh jenis ancaman dengan atribut adalah masing-masing keterampilan pada kriteria *threat agent*.

Tabel 10. Hasil penilaian dari faktor *Threat Agent*

https://senatmahasiswa_XYZ.org/

Jenis Ancaman	*Skill level*	*Motive*	*Opportunity*	*Size*
Input Filtering	9	9	4	9
CSRF	9	4	4	9
XSS	9	9	4	9
Information Disclosure	6	9	9	9
Error Massage on Page	6	4	9	6
Session Hijacking	9	4	7	9
Cross-Domain Misconfiguration	6	3	4	4

2. *Vulnerability Factors*

Faktor ini digunakanakan untuk memperkirakan kemungkinan *vulnerability* tertentu yang ditemui, kriteria penilaian sebagai berikut dengan hasil penilaian masing-masing keterampilan ada di dalam () [18]:

a. *Ease of Discovery* (EoX)
Cara Praktis tidak mungkin (1), sulit (3), mudah (7), alat otomatis tersedia (9).

b. *Ease of Exploit* (EoE)
Alat bantu otomatis teoritis (1), sulit (3), mudah (5), tersedia (9).

c. *Awareness*
Tidak diketahui (1), tersembunyi (4), jelas (6), pengetahuan umum (9).

d. *Intrusion Detection*(ID)
Deteksi aktif dalam aplikasi (1), *login* dan ditinjau (3), *login* tanpa review (8), tidak *login* (9).

Adapun untuk mendapatkan hasil dari penilaian berdasarkan faktor *vulnerability*, digunakan rumus sebagai berikut ini:

$$Vulnerability\ = \frac{EoD + EoX + Awarness + ID}{4}$$

Tabel 11 menunjukkan hasil penilaian dari faktor *Vulnerability*, yang menyajikan tujuh jenis ancaman dengan atribut adalah masing-masing keterampilan pada kriteria *Vulnerability Factors*.

3. *Technical Impact*

Faktor ini bertujuan untuk melakukan perhitungan pada dampak yang timbul jika suatu aplikasi dilakukan eksploitasi.

Tabel 11. Hasil penilaian dari faktor *Vulnerability*

https://senatmahasiswa_XYZ.org/				
Jenis Ancaman	EoD	EoX	Awarness	ID
Input Filtering	9	5	6	3
CSRF	9	3	6	1
XSS	9	5	6	3
Information Disclosure	9	9	9	1
Error Massage on Page	9	9	3	1
Session Hijacking	9	7	6	1
Cross-Domain Misconfiguration	3	3	3	1

Kriteria penilaian sebagai berikut dengan hasil penilaian masing-masing keterampilan ada di dalam () [19]:

a. *Loss of confidentiality* (LoC)
   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).
b. *Loss of integrity* (LoI)
   Data korup yang minimal sedikit (1), data korup minimal yang serius (3), data yang agak korup sekali, (7), semua data benar-benar korup (9).
c. *Loss of availability* (LoAv)
   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).
d. *Loss of accountability* (LoAc)
   Sepenuhnya dapat dilacak (1), mungkin dapat dilacak (7), benar-benar anonim (9).

Adapun untuk mendapatkan hasil dari penilaian berdasarkan faktor *tecnical impact* digunakan rumus sebagai berikut:

$$tecnical\ impact\ = \frac{LoC + LoI + LoAv + LoAc}{4}$$

Tabel 12 menunjukkan hasil penilaian dari dua faktor *Tecnical Impact*, yang menyajikan tujuh jenis ancaman dengan atribut adalah masing-masing keterampilan pada kriteria *Tecnical Impact.*

Tabel 12. Hasil penilaian dari faktor *Tecnical Impact*

https://senatmahasiswa_XYZ.org/				
Jenis Ancaman	LoC	LoI	LoAv	LoAc
Input Filtering	2	1	1	9
CSRF	6	3	1	9
XSS	2	1	1	9
Information Disclosure	9	1	1	9
Error Massage on Page	7	3	1	1
Session Hijacking	7	1	1	7
Cross-Domain Misconfiguration	2	1	1	1

4. *Business Impact*

Faktor ini mempunyai tujuan untuk melakukan pengukuran terhadap dampak bisnis yang terjadi. Penilaian sebagai berikut dengan hasil penilaian masing-masing keterampilan ada di dalam () [20]:

a. *Financial damage* (FD)
   Kurang dari biaya untuk memperbaiki kerentanan (1), pengaruh kecil terhadap laba tahunan (3), berpengaruh signifikan terhadap laba tahunan (7), kebangkrutan (9).
b. *Reputation damage* (RD)
   Kerusakan minimal (1), Kehilangan akun utama (4), kehilangan niat baik (5), kerusakan merek (9).
c. *Non-compliance* (NC)
   Pelanggaran ringan (2), pelanggaran yang jelas (5), pelanggaran profil tinggi (7).
d. *Privacy violation* (PV)
   Satu orang (3), ratusan orang (5), ribuan orang (7), jutaan orang (9).

Adapun untuk mendapatkan hasil dari penilaian berdasarkan faktor *business impact*, menggunakan rumus sebagai berikut ini:

$$bussiness\ impact\ = \frac{FD + RD + NC + PV}{4}$$

Tabel 13 menunjukkan hasil penilaian dari faktor *business impact* pada senatmahasiswa_XYZ.

Tabel 13. Hasil penilaian dari faktor *Business Impact*

https://senatmahasiswa_XYZ.org/				
Jenis Ancaman	FD	RD	NC	PV
Input Filtering	1	5	2	3
CSRF	1	4	2	3
XSS	3	5	2	3
Information Disclosure	7	9	5	9
Error Massage on Page	3	1	2	3
Session Hijacking	7	4	2	3
Cross-Domain Misconfiguration	3	1	2	3

Melalui keempat faktor tersebut selanjutnya dihitung nilai tingkat risiko yang dihasilkan kemudian diakumulasikan sebagai hasil akhir dari penilaian tingkat risiko, dengan rumus sebagai berikut:

$$Likelihood = \frac{Theat\ agent\ F + Vulnerability\ F}{2}$$

$$Impact = \frac{Tecnical\ Impact + Business\ Impact}{2}$$

Dilakukan perhitungan untuk *likelihood* dan *impact* sehingga didapatkan hasil seperti ditampilkan pada Tabel 14.

Setelah semuanya dihitung dilakukan perhitungan dengan rumus *likelihood* dan *impact* sehingga mendapatkan hasil untuk web senatmahasiswa_XYZ 6.05 pada *likelihood* dan 3.49 pada *impact*.

Berdasarkan tabel kategori tingkat kerawanan pada OWASP *risk rating* [21], secara keseluruhan dari hasil tersebut *likelihood* mendapatkan tingkat kerawanan tinggi dan *impact* mendapatkan tingkat kerawanan sedang yang dapat dilihat pada Tabel 15.

Selain itu pada *Overall Risk Severity* [22] dapat ditarik secara keseluruhan hasil dari tingkat kerawanan pada web senatmahasiswa_XYZ dapat

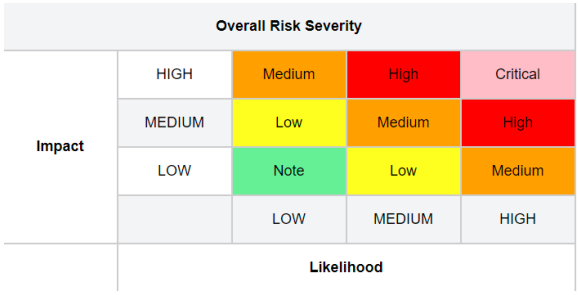
dikategorikan pada tingkatan kerawanan tingkat tinggi. Keseluruhan hasil dapat dilihat berdasarkan Gambar 9 [23].

Tabel 14. Hasil Perhitungan Tingkat Resiko

Threat agent factor					
Skill level	Motive	Opportunity	Size	Total	Risk
7.71	6	5.8	7.8	27.21	6.82
Vulnerability factor					
EoD	EoX	Awarness	ID	Total	Risk
8.14	5.85	5.57	1.57	21.13	5.28
Tecnical impact					
LoC	LoI	LoAv	LoAc	Total	Risk
5	1.57	1	6.42	13.99	3.49
Bussiness impact					
FD	RD	NC	PV	Total	Risk
3.57	4.14	2.42	3.85	13.98	3.49

Tabel 15. Level *likelihood* dan *impact*

Likelihood and impact levels	
0 to < 3	*LOW*
3 to < 6	*MEDIUM*
6 to 9	*HIGH*



Gambar 9. Keseluruhan hasil dari tingkat kerawanan

5.4 Kebijakan

Selanjutnya *Action Taking* dilakukan dengan pendekatan penerpan kebijakan pada wibsite senatmahasiswa_XYZ untuk menangani kerentanan yang belum dapat diatasi oleh *tools*. Kebijakan yang diusulkan pada penelitian ini ada dua yaitu kebijakan teknis dan non-teknis, yang diantarnya sebagai berikut:

1. Kebijakan Teknis
   a. Pada kerentanan input filtering dilakukan filter karakter yang tidak diperlukan pada input box yang ada di web senatmahasiswa_XYZ dengan mengunakan PHP filter_input yang merupakan sebuah fungsi bawaan yang sudah disedia oleh bahasa pemrograman tersebut.
   b. Pada kerentanan dengan CSRF dilakukan perbaikan dengan menerapkan token pada setiap sesi pengaksesan page sehingga tidak dapat dilakukan serangan CSRF. Pembangkitan token mengunakan akan random generator number dan string yang ada pada bahasa pemrogramman PHP.
   c. Memberikan penanganan pesan error atau error handling sehingga *end-user* ketika melihat *error* tidak menampilkan informasi detail *error* yang bisa dimanfaatkan untuk melakukan serangan.
   d. Melakukan enkripsi pada *session* untuk menghindari session *hijacking*. Dengan bahasa pemrogramman PHP dilakukan enkripsi dengan *library* yang ada pada *framework code igniter*.
2. Kebijakan Non-Teknis
   a. Membuat password minimal 8 karakter dan terdapat huruf kapital dan angka. Hal ini dikaren masih banyak user yang mengunakan password yang lemah, sehingga mudah ditebak oleh penyerang.
   b. Membatasi jumlah login akun pada setiap perangkat, Kebijakan ini dapat membantu ketika banyaknya *user* yang melakukan *multi login* sehingga menyebabkan sesi belum dikeluarkan dari salah satu perangkat.
   c. Melarang file format *executable* untuk di-*upload*, Kebijakan ini dapat menjaga server dari kemungkinan serangan virus dan *malware*.
   d. Membatasi jumlah request baik itu *download* maupun *upload file*, Kebijakan ini dapat membantu terpeliharanya kelancaran sistem.

6. KESIMPULAN

Setelah dilakukan penilaian terhadap risiko melalui kerentanan yang ada pada sampel web senatmahasiswa_XYZ, maka dapat disimpulkan Metode OWASP dapat diimplementasikan pada penilaian risiko terhadap kerentanan web senatmahasiswa_XYZ yang menggunakan *framework Codeigniter*. Ditemukan 7 risiko dengan tingkatan kerentanan yang berbeda-beda mulai dari status *high* hingga *low risk*. Pada studi kasus ini juga memberikan kandidat teknis dan non-teknis sebagai solusi untuk permasalahan kerentanan yang belum bisa diatasi oleh *tools* atau aplikasi.

REFERENSI

[1] B. Ghozali, "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," *Dikirim: 09 Februari*, 2018.

[2] K. Aryasa and Y. T. Paulus, "Implementasi Secure Hash Algorithm-1 Untuk Pengamanan Data Dalam Library Pada Pemrograman Java," *Citec Journal*, vol. 1, no. 1, 2013.

[3] P. Apol, S. T. Subriadi, M. T. Bekti, C. Hidayanto, S. Si, and M. Kom, "*Final Project-KS141501 Vulnerability Risk Evaluation Using Open Web Application Security Project*

(Owasp) Methodology For Student Information System Web Application (Case Study : Perguruan Tinggi XYZ)."

[4] Martin Otieno, David Odera, and Jairus Ekume Ounza, "Theory and practice in secure software development lifecycle: A comprehensive survey," *World Journal of Advanced Research and Reviews*, vol. 18, no. 3, pp. 053–078, Jun. 2023, doi: 10.30574/wjarr.2023.18.3.0944.

[5] A. Alanda, D. Satria, M. Isthofa Ardhana, A. A. Dahlan, and A. Mooduto, "Web Application Penetration Testing Using SQL Injection Attack," *International Journal on Informatics Visualization*, 2021, [Online]. Available: www.joiv.org/index.php/joiv

[6] F. Q. Kareem *et al.*, "SQL Injection Attacks Prevention System Technology: Review," *Asian Journal of Research in Computer Science*, pp. 13–32, Jul. 2021, doi: 10.9734/ajrcos/2021/v10i330242.

[7] N. Davis, "Secure Software Development Life Cycle Processes: A Technology Scouting Report," 2005. [Online]. Available: http://www.sei.cmu.edu/publications/pubweb.html

[8] S. Bitaraf and M. Shahriari, "Risk Assessment and Decision Support."

[9] L. Costaner and dan Musfawati, "Analisis Keamanan Web Server Open Journal System (OJS) Menggunakan Metode ISSAF dan OWASP (Studi Kasus OJS Universitas Lancang Kuning)."

[10] A. Saputra, M. Armys Roma Sitorus, and P. Negeri Batam Program Studi Teknik Multimedia dan Jaringan Jalan Ahmad Yani, "Penilaian Ancaman pada Website Transkrip Aktivitas Kemahasiswaan Politeknik Negeri Batam Menggunakan Metode DREAD," 2017. [Online]. Available: http://www.tak.polibatam.ac.id

[11] F. Al Fajar, "Analisis Keamanan Aplikasi Web Prodi Teknik Informatika UIKA Menggunakan Acunetix Web Vulnerability," *Jurnal INOVA-TIF*, vol. x, No.x, no. 2.

[12] V. Casola, A. De Benedictis, C. Mazzocca, and V. Orbinato, "Secure software development and testing: A model-based methodology," *Comput Secur*, vol. 137, Feb. 2024, doi: 10.1016/j.cose.2023.103639.

[13] F. M. Tudela, J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo, and M. I. Argyros, "On combining static, dynamic and interactive analysis security testing tools to improve owasp top ten security vulnerability detection in web applications," *Applied Sciences (Switzerland)*, vol. 10, no. 24, pp. 1–26, Dec. 2020, doi: 10.3390/app10249119.

[14] K. I. Satoto, "Keamanan Sistem Informasi Akademik Berbasis Web, Analisis," 2009.

[15] K. El and H. Ismail, "Action Research: from Theory to Practice." [Online]. Available: https://ijascfrtjournal.isrra.org/index.php/Applied_Sciences_Journal

[16] H. Setiawan, L. E. Erlangga, S. Siddiq, and A. Gunawan, "Analisis Kerawanan Pada Aplikasi Website Menggunakan Standar OWASP Top 10 Untuk Penilaian Risk Rating," 2023.

[17] B. Sugiantoro, M. Anshari, and D. Sudrajat, "Developing Framework for Web Based e-Commerce: Secure-SDLC," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2020. doi: 10.1088/1742-6596/1566/1/012020.

[18] D. Kaur and P. Kaur, "Ranking and Impact of Web Applications" Vulnerabilities," 2014. [Online]. Available: www.ijser.in

[19] S. P. V. Vijayaraghavan and N. Rajarathnam, "iMeasure Security (iMS): A Framework for Quantitative Assessment of Security Measures and its Impacts," *Information Security Journal: A Global Perspective*, vol. 19, no. 4, pp. 213–225, 2010, doi: 10.1080/19393551003762223.

[20] T. Hardiani, D. Wijayanto, and N. Latifah, "Data Security Analysis with OWASP Framework on Website XYZ," *CYBERNETICS*, vol. 6, no. 01, pp. 10–20, 2022.

[21] E. Snorrason, "Relative Impact: A model for Quantitative Risk Assessment," University of Oslo, Oslo, 2022.

[22] S. Reddy *et al.*, "A Review of a Customized OWASP Risk Calculator for Security Risk Analysis," *Ijert*, 2023, [Online]. Available: http://www.ijert.org

[23] F. Putra Utama, R. Muhamad, and H. Nurhadi, "Uncovering the Risk of Academic Information System Vulnerability through PTES and OWASP Method," *CommIT Journal*, vol. 18, no. 1, 2024.