BAB II

LANDASAN TEORI

2.1 Website

Website atau situs merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, suara, animasi, gambar gerak atau diem, dan atau gabungan dari semuanya yang membentuk suatu rangkaian bangunan saling terkait satu sama lain baik yang bersifat statis maupun dinamis, yang masing-masing terhubung dengan jaringan-jaringan halaman (Rahmat Hidayat, 2010).

2.2 Web Application

Merupakan *website* jenis aplikasi yang berjalan melalui jaringan internet dan diakses menggunakan web browser. *Web application* bersifat dinamis yang menyediakan element interaktif sehingga memungkinkan pengunjung *website* untuk berinteraksi didalamnya (Pratama et al., 2019).

2.3 Keamanan Informasi

Menurut G. J. Simons dalam penelitian Ramdhani, keamanan informasi adalah bagaimana kita dapat mencegah penipuan (*cheating*) atau, paling tidak, mendeteksi adanya penipuan di sebuah sistem yang berbasis informasi, dimana informasinya sendiri tidak memiliki arti fisik. Keamanan informasi adalah upaya untuk mengamankan aset informasi terhadap ancaman yang mungkin timbul.

Secara tidak langsung keamanan informasi menjamin kontinuitas bisnis, mengurangi risiko-risiko yang terjadi, mengoptimalkan pengembalian investasi (Ramdhani, 2018).

Sutabri, T (2012) pada bukunya tentang Konsep Sistem Informasi menjelaskan konsep keamanan informasi meliputi:

1. Tujuan keamanan sistem informasi

   Tujuan dari pengamanan sistem informasi ini adalah untuk meyakinkan integritas, kelanjutan, dan kerahasian dari pengolaan data.

2. Kebijakan keamanan sistem informasi

   Kebijakan keamanan sistem informasi merupakan urusan dan tanggung jawab semua karyawan. Langkah keamanan harus sesuai dengan peraturan dan undang-undang.

3. Strategi keamanan sistem informasi

   Meliputi *integrity, confidentiality,* dan *availability.*

2.4 Keamanan Web

Keamanan *website* adalah suatu cabang teknologi yang dikenal dengan nama keamanan informasi untuk melindungi informasih/data dari berbagai serangan yang diterapkan pada *website* (Elu, 2013). Permasalahan yang umum terjadi terhadap keamanan web adalah kesalahan proses pada *layerlogic* yang disebabkan oleh gangguan terhadap aplikasi-aplikasi melalui HTTP. Ada dua cara untuk melakukannya:

a. Memanipulasi aplikasi dengan *Graplucal Web Interface* (GUI) secara langsung

Graphic User Interface (GUI) pada web seperti *form,* memudahkan dalam penggunaan website untuk memberikan input. Penyerang dapat memanfaatkan GUI yang terdapat pada web untuk melakukan serangan seperti *SQL injection*. Beberapa tahun terakhir, banyak peretas telah mencoba menggunakan berbagai *tools* untuk memecahkan logika dasar aplikasi web. Faktanya, beberapa serangan web tingkat lanjut hanya menggunakan *browser*

b. Gangguan pada *Uniform Resource Indentifier* (URI)

Uniform Resource Identifier (URI) adalah sebuah *string* pada *bilah alamat browser* yang digunakan untuk mengakses sebuah halaman web. *Uniform Resource Identifer* (URI) menggambarkan sebuah protokol (skema) untuk mengakses sebuah *resourc*e (*path*) atau aplikasi (*query*) pada sebuah *server* (*authority*). Aplikasi web protokol yang paling sering digunakan adalah HTTP, namun ada versi yang aman dari HTTP yaitu HTTPS dimana *session* data diproteksi lebih baik oleh SLL atau TLS (Surya T Fajri, 2013).

2.5 Penetration Testing

Penetration Testing adalah metode yang digunakan untuk menguji kelemahan sistem komputer, jaringan atau aplikasi web (Yunus, 2019). Berdasarkan definisi dalam modul CEH, *Penetration Testing* merupakan metode evaluasi keamanan sistem komputer atau jaringan dengan mensimulasikan serangan dari sumber yang berbahaya dan merupakan bagian dari *security audit* (Sahren et al.,

2019). *Vulnerability assessment* merupakan bagian dari *risk assessment* yang terdiri dari *risk analysis*, *policy development*, *training and implementation*, dan *vulnerability assessment and penetration testing* (Wibowo F et al., 2019).

2.6 Acunetix

Acunetix Vulnerability Scanner merupakan salah satu perangkat lunak yang dapat melakukan pengujian dan evaluasi keamanan *website* yang dirancang khusus untuk mengetahui kerentanan terhadap suatu sistem atau situs web (Mayasari R et al., 2020). *Acunetix* adalah sebuah *software* yang didesain untuk melakukan pengecekan terhadap celah keamanan pada aplikasi yang berbasis web yang dapat disusupi oleh penyerang yang kemungkinan akan memasuki sistem dan menyalahgunakan data (Saleh S, 2019).

2.7 Open Web Application Security Project Zed Attack Proxy (OWASP ZAP)

Open Web Application Security Project Zed Attack Proxy (OWASP ZAP) adalah aplikasi atau perangkat lunak untuk melakukan *penetration test* terhadap suatu *web application* dengan cara mudah yang bertujuan untuk menemukan *vulnerabilities*, menggunakan OWASP ZAP *scanner automatis* akan menemukan *vulnerabilities* sebaik bila kita mengguna *tool* secara manual (Kusuma G, 2022). OWASP ZAP merupakan sebuah *tools vulnerability scanner* yang disediakan oleh organisasi OWASP, salah satu proyek paling aktif dari organisasi OWASP yaitu OWASP ZAP karena terus dikembangkan dan bersifat *opensource* sehingga siapa saja juga dapat mengembangkan *tools* ini (Yudiana et al., 2021). *Open Web*

Application Security Project (OWASP) sendiri merupakan sebuah organisasi non-profit yang menyediakan banyak alat, panduan, dan metodologi pengujian untuk keamanan cyber yang berfokus pada keamanan web (Pratama Eka I Putu Agus & Bagus, 2019).

Dokumen OWASP yang sering disebut sebagai panduan untuk keamanan *website* salah satunya adalah OWASP Top 10, merupakan *checklist* standar keamanan mengenai kelemahan-kelemahan pada keamanan web yang rentan untuk diserang dan harus segera ditangani (Sunardi et al., 2019). Berikut *checklist* OWASP Top 10:

1. *Injection*

Merupakan kelemahan pada *website* yang dapat di eksploitasi lewat *input text*. Terdapat 2 tipe serangan *injection*; *server-side injection* dan *SQL injection*.

2. *Broken Authentication*

Jenis kelemahan yang memungkinkan penyerang untuk melewati proses otentikasi pada akses login.

3. *Sensitive Data Exposure*

Data atau informasi penting yang tidak terlindungi dengan baik ataupun terekspos seperti *password*, informasi privasi, *credit card*, rekam kesehatan dan lain sebagainya.

4. *XML External Entities* (XXE)

Serangan pada web yang menganalisa input XML, input ini bisa mereferensikan *entity external* untuk mengetahui kelemahan yang ada pada *input* XML nya.

5. *Broken Access Control*

Rusaknya sistem kontrol yang mengakses informasi dan fungsionalitasnya memungkinkan penyerang untuk melewati proess autorisasi dan dapat melakukan hal-hal yang biasanya hanya dilakukan oleh admin.

6. *Security Misconfiguration*

Kesalahan konfigurasi keamanan yang biasanya terjadi jika hanya menggunakan *default* konfigurasi tanpa melihat kebutuhan *website.*

7. *Cross-Site Scripting XSS*

Cross-Site Scripting memanfaatkan kerentanan pada web berupa *input* dan *output* yang tidak di validasi, Jenis serangan berupa code injetion yang menanamkan script berbahaya pada website.

8. *Insecure Deserialization*

Kelemahan pada website yang memungkinkan terjadinya serangan DDOS.

9. *Using Components with Known Vulnerabilities*

Website yang menggunakan komponen *libraries* ataupun *framework* yang tidak dapat dipercaya atau tidak diperbarui memungkinkan penyerang untuk melakukan eksploitasi melalui kelemahan atau *bug* pada *framework* dan *libraries* yang sudah diketahui.

10. *Insufficient Logging & Monitoring*

Pencatatan dan Pemantauan yang tidak memadai, ditambah dengan tidak efektifnya dengan respons insiden, memungkinkan penyerang untuk melakukan serangan dimasa mendatang.

2.8 Laravel

Laravel adalah sebuah *web development framework* MVC yang didesain agar dapat mengurangi biaya pengembangan dan perbaikan untuk meningkatkan kualitas perangkat lunak serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk implementasi (Purnomo, 2016). *Laravel* merupakan framework PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya (Naista & Lokomedia, 2016).

2.9 CodeIgniter

CodeIgniter adalah sebuah *framework* php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal (Destiningrum & Adrian, 2017). Keuntungan dalam penggunaan *codeigniter* diantaranya yaitu penghematan waktu, *script website* mudah dibaca dan diperbarui, terstruktur, dan kode *script* menjadi lebih mudah dikelola sehingga mempermudah pekerjaan (Wardana, 2010).

2.10 Perbandingan Laravel dan CodeIgniter

Yasin K dengan artikelnya yang berjudul "*Laravel vs CodeIgniter:* Manakah yang lebih baik" dalam web Niagahoster.co.id menyebutkan bahwa terdapat 6 dasar perbandingan antara *framework Laravel* dan *CodeIgniter* (Yasin K,

2019). Berikut perbandingan *Laravel* dan *Codeigniter* yang akan ditampilkan pada tabel 2.1.

Tabel 2.1 Dasar Perbandingan *Laravel* dan *CodeIgniter*

Dasar Perbandingan	*Laravel*	*CodeIgniter*
Definisi	*Laravel* merupakan *open source PHP Framework*, menggunakan MVC, yang mana *framework* ini termasuk *powerfull* dan mudah untuk dipelajari. Cocok untuk pengembang yang ingin membuat aplikasi web yang canggih dengan fitur yang elegan dan modern.	*CodeIgniter* merupakan *framework PHP* yang juga *open source. Framework* ini dibuat untuk pengembang yang ingin membuat aplikasi web sederhana namun mempunyai fitur yang sangat lengkap dan elegan.
Database Model	*Object Oriented*	*Relational-Object Oriented*
Programming Paradigm	*Object Oriented Event Driven Functional*	*Component Oriented*
Routing	*Explicit Routing*	*Explicit Routing dan Implicit Routing*
Built-in Modules	Mengizinkan programmer/pengembang untuk membantu project ke dalam modul-modul kecil melalui bundle, dan dapat menggunakan kembali modul di dalam berbagai macam *project* yang berbeda.	Tidak mendukung fitur built-in modules, ini membutuhkan pengembang untuk membuat sendiri dan mengelolanya menggunakan tambahan *Modular Extension.*
HTTPS Support	Mengizinkan pengembang untuk menentukan kustom *HTTPS Routes*. Pengembang juga mempunyai wewenang untuk membuat URL yang spesifik untuk masing-masing *routes. Laravel* lebih jauh menjamin keamanan data yang ditransmisikan dengan menambahkan https:// sebelum URL secara otomatis.	Tidak mendukung penuh penggunaan *HTTPS route.* Pengembang harus mengelola URL *Helper* untuk membuat transmisi data aman dengan pengembangan pats.

2.11 Penelitian Terkait

Penelitian sebelumnya dengan judul "Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project)" dalam penelitiannya mendeteksi kerentanan keamanan aplikasi berbasis *website* terhadap *website* PT.Gitsolution dengan metode *Open Web Application*

Security Project (OWASP) *Risk Rating* ditargetkan terhadap 2 aplikasi berbasis *website* yang memiliki karakter berbeda sebagai sampelnya, yaitu *framework codeigniter* dan *PHP Nativ.* Hasil akhir dari resiko keseluruhan menunjukan tingkat keparahan keamanan serupa, sehingga tidak ada jaminan menghindari kerentanan keamanan dengan menggunakan *framework codeigniter* atau *PHP Nativ.* Berdasarkan penelitian-penelitian terkait disarankan melakukan pengujian kerentanan keamanan aplikasi berbasis *website* antara *framework laravel* dan *codeigniter.*

Penelitian-penelitian sebelumnya yang berkaitan dengan penelitian ini di tampilkan pada tabel 2.1.

Tabel 2.2 Penelitian Terkait

No	Peneliti	Judul	Metode	Hasil
1	Guntoro, Loneli Costaner, & Musfawati (2020)	Analisis Keamanan Web Server Open Journal System (OJS) Menggunakan Metode ISSAF dan OWASP (Studi Kasus OJS Universitas Lancang Kuning)	Information Systems Security Assessment Framework (ISSAF), OWASP	• Journal.unilak.ac.id rentan terhadap serangan DoS • Tidak memiliki kerentanan terhadap SQL Injection • Kelemahan sistem yaitu tidak bisa memblokir ketika user melakukan berkali-kali kesalahan dalam proses login. • Perlunya sistem monitoring untuk melindungi server, misalnya menerapkan Firewall maupun Intrusion Detection System (IDS)
2	Raden Teduh Dirgahayu, Yudi Prayudi, & Adi Fajaryanto (2016)	Penerapan Metode ISSAF dan OWASP versi 4 Untuk Uji Kerentanan Web Server	ISSAF (Information Systems Security Assessment Framework) dan OWASP versi 4	• Metode ISSAF menunjukkan web server IKIP PGRI Madiun masih dapat ditembus dan mengambil alih hak akses administrator • Metode OWASP versi 4 menunjukkan bahwa manajemen otentifikasi, otorisasi dan manajemen sesi belum diimplementasikan dengan baik.
3	Moh Yunus (2019)	Analisi Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi Security Tools Project Berdasarkan Framework OWASP versi 4	Open Web Application Security Project (OWASP) Versi 4	• OWASP versi 4 mampu mengetahui keamanan suatu aplikasi • OWASP versi 4 dapat dijadikan sebagai standar penilaian keamanan aplikasi berbasis web
4	Bahrun Ghozali , Kusrini, & Sudarmawan (2019)	Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) untuk Penilaian Risk Rating	Open Web Application Security Project (OWASP) Risk Rating	• Penelitian pada web PT.Gitsolution yang di bangun dengan framework Codeigniter dan PHP Native • Terdapat 3 risiko yang ditemukan yaitu risk severity High, risk severity Medium dan risk severity Low • Framework Codeigniter dan PHP Native sama-sama memiliki tingkat keparahan pada Likelihood dilevel Medium • Tingkat keparahan pada Impact sama-sama berada dilevel Low.
5	Mohmmad Muhsin & Adi Fajaryanto (2016)	Penerapan Pengujian Keamanan Web Server Menggunakan Metode OWASP versi 4 (Studi Kasus Web Server Ujian Online)	OWASP (Open Web Application Security Project) versi 4	• Pengujian keamanan dari aplikasi Si Ujo (Sistem Ujian Online) berbasis web • Manajemen otentifikasi, otorisasi dan manajemen sesi belum diimplementasikan dengan baik • Perbaikan lebih lanjut diperlukan oleh pihak stake holder Fakultas Teknik Universitas Muhammadiyah Ponorogo

6	I Putu Agus Eka Pratama & Anak Agung Bagus Arya Wiradarma (2019)	Open Source Intelligence Testing Using the OWASP Version 4 Framework at the Information Gathering Stage (Case Study: X Company)	Maltego and OWASP (Open Web Application Security Project) version 4	• Pengujian keamanan pengumpulan informasi faktor penting dari situs web Perusahaan X • Komponen pembangunan website masih dapat ditembus dan dianalisis oleh penyerang
7	Imam Riadi, Rusydi Umar, & Tri Lestari (2020)	Analisis Kerentanan Serangan Cross Site Scripting (XSS) pada Aplikasi Smart Payment Menggunakan Framework OWASP	OWASP (Open Web Application Security Project)	• mengetahui kerentanan aplikasi Smart Payment dengan cara self test menggunakan tool ZAP • Kerentanan yang ditemukan berupa Information Disclosure-Suspicious Comments, X-Frame-Options Header not Set, XContent-Type-Options Header Missing, Timestamp Disclosure-Unix, Web Browser XSS Protection Not Enabled, dan Directory Browsing.
8	Mia Zattu Maharani, Henry Rossi Andrian, & Setia Juli Irzal Ismail (2017)	Analysis Website Security Using Scanning Method And Calculation Of Security Metrics	Security Metrics	• pengujian celah keamanan dengan metode scanning pada web igracias.telkomuniversity.ac.id, ppdu.telkomuniversity.ac.id. • Security metrics dapat menampilkan hasil berlabel High, Medium atau Low yang ditentukan menggunakan rumus BaseScore yang menghasilkan jumlah hasil 1-10. • Hasil vulnerability assessments pada igracias.telkomuniversity.ac.id adalah bernilai high.
9	Anggariyona Saputra, Nelmiawati, & Maya Armys Roma Sitorus (2017)	Penilaian Ancaman pada Website Transkrip Aktivitas Kemahasiswaan Politeknik Negeri Batam Menggunakan Metode Dread	Dread	• Analisis celah keamanan dan memberikan penilaian ancaman terhadap website TAK Politeknik Negeri Batam • Terdapat 4 kategori ancaman dengan jenis ancaman yaitu authentication, kriptografi, dan session management • Risiko yang paling tinggi terdapat pada pengujian otentikasi.
10	Bhaskara Vito Tarigan, Ari Kusyanti, & Widhi Yahya (2017)	Analisis Perbandingan Penetration Testing Tool Untuk Aplikasi Web	Penetration Test	• Mengnanlisa perbandingan ketiga tool uji penetrasi yaitu w3af, wapiti, dan arachni. • Setelah 5 kali percobaan menggunakan tool w3af, wapiti, dan arachni hasilnya tool arachni mendapatkan kerentanan lebih banyak dari tool w3af dan wapiti

Tabel 2.3 Matriks Penelitian

No	Peneliti	JUDUL	Ruang Lingkup															Keterangan
			Metode					Framework		Pengujian		Tools						
			OWASP	*ISSAF*	*OSSIG*	*Security Metrics*	*Dread*	Laravel	Codeigniter	Vulnerability scanner	Penetration Test	Acunetix	Arachni	OWASP ZAP	Skipfish	Vega	W3af	Wapiti	
1	Guntoro, Loneli Costaner, & Musfawati	Analisis Keamanan Web Server Open Journal System (OJS) Menggunakan Metode ISSAF dan OWASP (Studi Kasus OJS Universitas Lancang Kuning)	✔	✔						✔	✔	✔							Perlunya sistem monitoring untuk melindungi server, misalnya menerapkan Firewall

2	Raden Teduh Dirgahayu, Yudi Prayudi, & Adi Fajaryanto	Penerapan Metode ISSAF dan OWASP versi 4 Untuk Uji Kerentanan Web Server	✓	✓							✓	✓							Manajemen otentifikasi, otorisasi dan manajemen sesi belum diimplementasikan dengan baik.
3	Moh Yunus	Analisi Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi Security Tools Project Berdasarkan Framework OWASP versi 4	✓								✓								OWASP versi 4 dapat dijadikan sebagai standar penilaian keamanan aplikasi berbasis web
4	Bahrun Ghozali, Kusrini, & Sudarmawan	Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security	✓						✓	✓		✓							OWASP versi 4 dapat dijadikan sebagai standar penilaian keamanan aplikasi berbasis web

		Project) untuk Penilaian Risk Ratin																	
5	Mohmmad Muhsin & Adi Fajaryanto	Penerapan Pengujian Keamanan Web Server Menggunakan Metode OWASP versi 4 (Studi Kasus Web Server Ujian Online)	✓								✓	✓							Manajemen otentifikasi, otorisasi dan manajemen sesi belum diimplementasikan dengan baik
6	I Putu Agus Eka Pratama & Anak Agung Bagus Arya Wiradarma	Open Source Intelligence Testing Using the OWASP Version 4 Framework at the Information Gathering Stage (Case Study: X Company)	✓								✓								Komponen pembangunan website masih dapat ditembus dan dianalisis oleh penyerang
7	Imam Riadi, Rusydi Umar, & Tri Lestari	Analisis Kerentanan Serangan Cross Site Scripting (XSS) pada	✓							✓				✓					Kerentanan yang ditemukan berupa Information Disclosure-Suspicious

		Aplikasi Smart Payment Menggunakan Framework OWASP																	Comments, X-Frame-Options Header not Set, XContent-Type-Options Header Missing, Timestamp Disclosure-Unix, Web Browser XSS Protection Not Enabled, dan Directory Browsing
8	Mia Zattu Maharani, Henry Rossi Andrian, & Setia Juli Irzal Ismail	Analysis Website Security Using Scanning Method And Calculation Of Security Metrics				✔				✔		✔							Hasil vulnerability assessments pada igracias.telkomuni-versity.ac.id adalah bernilai high.
9	Anggariyona Saputra, Nelmiawati, & Maya Armys Roma Sitorus	Penilaian Ancaman pada Website Transkrip Aktivitas Kemahasiswaan Politeknik					✔			✔				✔					Terdapat 4 kategori ancaman dengan jenis ancaman yaitu authentication, kriptografi, dan

		Negeri Batam Menggunakan Metode Dread																	session management
10	Bhaskara Vito Tarigan, Ari Kusyanti, & Widhi Yahya	Analisis Perbandingan Penetration Testing Tool Untuk Aplikasi Web			✓			✓		✓			✓				✓	✓	Tool w3af, wapiti, dan arachni hasilnya tool arachni mendapatkan kerentanan lebih banyak dari tool w3af dan wapiti
11	Simon Fetrus Sijabat	Analisis Perbandingan Keamanan Web Laravel dan Codeigniter Dengan OWASP	✓					✓	✓	✓	✓	✓							Mengetahui tingkat perbandingan kerentanan pada web laravel dan codeigniter