BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Keamanan yang merupakan salah satu faktor penting dalam merancang sebuah *website*, di mana pada jurnal ini yang di tulis oleh Dewanto, (2018)berjudul "Penetration Testing pada Domain UII.AC.ID Menggunakan OWASP 10". Penelitian ini untuk mengetahui seberapa tingkat kerentanan *website* terhadap serangan dari luar, karena itu peneliti menyimulasikan dirinya sebagai pihak luar yang berusaha untuk masuk dalam *website* sehingga peneliti dapat memberikan saran dari hasil penelitian.

Penelitian yang dilakukan oleh Yanti & Cut, (2019) berjudul "Analisa Keamanan Web Server dari Serangan *Remote Os Command Injection* pada Instansi Pemerintahan Kota Banda Aceh". Di mana penulis jurnal tersebut untuk menguji dan analisis keamanan *web server* instansi yang menggunakan teknik *remote os command injection*. Hasil analisis akan diberikan pada pihak pemilik *website* yang dapat dimanfaatkan dan digunakan untuk meningkatkan keamanan pada *website* tersebut.

Pada jurnal yang di tulis oleh Saputra, Nelmiawati, & Sitorus, (2017) berjudul "Penilaian Ancaman pada *Website* Transkrip Aktivitas Kemahasiswaan Politeknik Negeri Batam Menggunakan Metode DREAD" yang memiliki data dari hasil analisis yang membuktikan pengaruh dari kurangnya keamanan *website*. Bahwa aplikasi *web* yang merupakan salah satu target yang sering di serang, terbukti serangan mencapai 72% dari laporan aplikasi *vulnerability*. Dari hasil penelitian ini *website* tersebut rentan terkena salah satu serangan yaitu *SQL Injection* yang mampu mempelajari struktur dari *SQL Query* dan meng *inject* suatu data untuk mengubah sintak dari *query* tersebut.

Berdasarkan penelitian yang berjudul "Mendeteksi Kerentanan Keamanan Aplikasi *Website* Menggunakan Metode OWASP (*Open Web Application Security Project*) untuk Penilaian *Risk Rating*" yang dilakukan oleh (Ghozali, Kusrini, & Sudarmawan, 2019) dengan menggunakan metode OWASP untuk memperkirakan kemungkinan dampak dari masing-masing faktor yang di bagi dalam tiga bagian

risiko yaitu *risk severity high*, *risk severity medium*, dan *risk severity low*. Hasil dari penilaian risiko ini agar dapat membantu para perancang dan pengembang sistem aplikasi untuk menyadari risiko kemungkinan terjadi serangan dan dapat mengambil tindakan untuk mencegah dan mengatasi risiko tersebut.

Dalam penelitian yang berjudul "*Security Assessment of Libyan Government Websites*" yang di lakukan oleh (Ali & Zamri Murah, 2019). Dalam menganalisis tingkat kerentanan pada *website* pemerintah dengan menggunakan aplikasi testing yaitu *Acunetix* dan *Netsparker* untuk mengetahui celah-celah pada *website* tersebut. Dari hasil testing dapat menyimpulkan bahwa tingkat keamanan dalam *website* pemerintah dari 16 *website* terdapat 7 *website* yang memiliki keamanan yang tinggi, selain itu sisa dari *website* tersebut memiliki kerentanan tinggi dapat ditemukan 5 *website* tersebut yang belum menerapkan enkripsi SSL yang mengakibatkan hilangnya data, gangguan layanan, kehilangan privasi.

Berdasarkan penelitian di atas maka dapat disimpulkan dan ditampilkan dalam Tabel 2.1 sebagai berikut:

Tabel 2.1 Tabel Jurnal Penelitian

Peneliti	Tahun	Kesimpulan Penelitian
Bahrun Ghozali, Kusrini & Sudarmawan	2019	Melakukan *Penetration testing* pada sebuah *website* mengetahui tingkat kerentanan *website* yang dibagi menjadi 3 yaitu *high*, *medium* dan *low* sehingga pengembang bisa dapat memprioritaskan dari *risk severity high* ke *risk severity low*.
Lisa Handasari Yanti & Iqbal Banta Cut	2019	Berdasarkan hasil analisis tingkat kerentanan keamanan *website* dapat memberikan manfaat untuk perancang atau pengembang *website* untuk meningkatkan keamanan *website* tersebut
Abudllah Ahmed Ali & Mohd Zamri Murah	2019	Dengan menggunakan alat testing yaitu *Acunetix* dan *Netsparker* dapat mengetahui celah dan tingkat kerentanan pada *website* yang dapat mengakibatkan hilangnya data, gangguan layanan, dan kehilangan privasi.
Adetya Putra Dewanto	2018	Keamanan dalam sebuah sistem sangat penting dan dengan melakukan penetration testing yang menyimulasikan peneliti sebagai pihak luar yang berusaha untuk masuk ke dalam website.
Anggariyona Saputra, Nelmiawati & Maya Armys Roma Sitorus	2017	Berdasarkan tingkat kerentanan pada sebuah website yang memiliki tingkat serangan yang tinggi yaitu SQL Injection yang dapat mengubah sintak pada sebuah Query.

Berdasarkan hasil penelitian, penulis akan melakukan analisis tingkat kerentanan keamanan pada *website* FKUB dengan menggunakan metode OWASP maka penulis dapat memberikan hasil penelitian dalam tabel OWASP *Top* 10 2017 kepada pihak pengembang *wesbite* serta memberikan solusi untuk meminimalkan tingkat risiko masuknya serangan dari pihak luar.

2.2 Landasan Teori

2.2.1 Keamanan Informasi

Keamanan informasi yang merupakan sebuah praktik untuk melindungi informasi dari akses yang tidak valid dalam penggunaan, pengungkapan, modifikasi, inspeksi, dan menghapus data (Fauzan & Rijayanti, 2018). Keamanan informasi memiliki tiga prinsip dasar yaitu: *Confidentiality*, *Integrity* dan *Availability*.

1. *Confidentiality*

Merupakan sebuah *property*, di mana informasi yang tidak diungkapkan kepada *user* yang tidak sah (Fauzan & Rijayanti, 2018).

2. *Integrity*

Adalah untuk menjaga, menjaminkan kelengkapan data tersebut tidak dapat dimodifikasikan secara tidak sah (Fauzan & Rijayanti, 2018).

3. *Availability*

Merupakan ketersediaan informasi yang dapat melayani tujuan dan tersedia jika dibutuhkan. Di mana sistem komputasi yang digunakan untuk menyimpan data dan memproses informasi (Fauzan & Rijayanti, 2018).

2.2.2 *E-Government*

Adalah bagian tak terpisahkan dari sebuah kota yang memanfaatkan teknologi informasi dan komunikasi dalam mengubah hubungan antara badan pemerintah, warga negara, dan bisnis. *E-Government* memegang masa depan dari kota-kota karena semua otoritas publik lokal membentuk pemerintah yang terhubung ke *E-Government* di seluruh dunia untuk mempersiapkan dan memungkinkan lebih dalam teknologi informasi (Yang, Elisa, & Eliot, 2019).

2.2.3 *Consumer To Government* (C2G)

C2G digunakan untuk membantu konsumen dalam meminta informasi atau mengirim berbagai *feedback* mengenai sector publik langsung ke administrasi pemerintah. Contohnya untuk melakukan pajak atau asuransi kesehatan dalam *website* pemerintah adalah jenis model bisnis dari C2G (Lipu, 2018)

2.2.4 *Website*

Adalah halaman *web* yang menyediakan informasi yang tersedia di jalur internet sehingga dapat diakses dari seluruh dunia selama terhubung dengan jaringan internet berserta file-file pendukung seperti file video, gambar dan file digital lainnya yang dapat disimpan dalam sebuah *web server*. Atau bisa dikatakan dengan sekumpulan folder atau file yang mengandung fungsi-fungsi dan perintah tertentu untuk menangani penyimpanan data dan tampilan (Fazriani & Sanusi, 2019).

2.2.5 *Web Server*

Merupakan sebuah *host website* yang dapat diakses secara publik dan digunakan untuk menyajikan informasi seperti halaman *web*, formulir, iklan, barang dagangan, dan isi keranjang belanja ke konsumen *web browser*. Pengunjung situs *web* dapat melihat, mengambil, mengirim data ke atau dari *database* melalui internet dengan menggunakan *web browser* pilihan mereka. *Web server front-end* menerima sambungan *Hypertext Transfer Protocol* (HTTP) dan *Hypertext Transfer Protocol Secure* (HTTPS) dari *web browser* dan memproses permintaan HTTP atau HTTPS yang dicapai melalui request dari skrip kode program (Lin, 2017).

2.2.6 *Flowchart*

Merupakan bagan (*chart*) yang mengarahkan alir (*flow*) di dalam prosedur atau program sistem secara logika. *Flowchart* yang berfungsi untuk menjelaskan tahap-tahap proses pemecahan masalah atau proses melakukan penelitian sistem secara logika (Syamsiah, 2019).

2.2.7 *Penetration Testing*

Merupakan teknik untuk menemukan kerentanan atau celah keamanan yang ada di halaman *website* untuk dapat membantu mengesampingkan akses ilegal ke dalam halaman *website* dan *database*. Dalam pengujian *penetration testing* pada halaman *web* terus dapat menjadi masalah signifikan, karena semakin banyak fitur

pada aplikasi *web* maka semakin lama dalam pengujian tersebut (Bin Ibrahim & Kant, 2018). Adapun 3 tipe *pentest* yaitu:

1. *Black box* adalah pengujian yang berfokus pada spesifikasi fungsional dari sistem. Tester pada hanya diberikan informasi berupa domain perusahaan, dalam posisi tersebut harus mengumpulkan informasi sendiri untuk melakukan pengetesan pada spesifikasi fungsional sistem dan keamanan sistem (Hidayat & Muttaqin, 2018).

2. *White Box* merupakan pengujian sistem yang didasarkan pada pengecekan detail pada perancangan, dan struktur desain program secara prosedural yang dapat membagi pengujian dalam beberapa kasus (Hidayat & Muttaqin, 2018). Pengujian *white box* diberikan pertunjuk atau informasi sepenuhnya untuk melakukan pengujian program yang benar.

3. *Gray Box* adalah sebuah metode pengujian fungsional yang dilakukan untuk menguji interaksi seorang pengguna terhadap sistem. *Gray box* juga dapat disebut kombinasi dari pengujian *white box* dan *black box*. Pengujian *Gray box* diberikan informasi yang terbatas dalam melakukan *pentest* (Amalia, 2019).

2.2.8 *Open Web Application Security Project* (OWASP)

Yaitu sebuah komunitas terbuka yang bertujuan dalam peningkatan keamanan aplikasi perangkat lunak. OWASP menyediakan beberapa alat, panduan dan metode testing untuk *cyber security* dalam sumber yang terbuka. Panduan OWASP dibagi menjadi tiga macam yaitu kerangka pengujian OWASP dalam pengembangan aplikasi *web*, metode pengujian aplikasi *web*, dan laporan evaluasi sistem (Agus & Pratama, 2019). Yayasan OWASP merilis kan daftar 10 kerentanan paling berbahaya. Daftar itu disebut OWASP *TOP* 10 dari tahun 2003 sampai 2017. Berikut 10 daftar risiko OWASP pada tahun 2017:

1. *SQL Injection* adalah serangan di mana mengandung perintah *SQL* ke dalam kode *SQL* yang memungkinkan merusak *database*, identitas palsu, dan menjadi administrator dari *database* tersebut (Sagar, 2018).

2. *Broken Authentication* adalah mengirim informasi sesi dan data manajemen akun (pembuatan aku, ubah kata sandi, pulihkan kata sandi) yang melalui *website* yang terkena kode sesi dan kredensial yang tidak dilindungi (Dehalwar, Kalam, Kolhe, & Zayegh, 2018).

3. *Sensitive Data Exposure* adalah jenis kerentanan keamanan di mana aplikasi *web* gagal dalam melindungi data rahasia dari suatu organisasi. Data sensitif termasuk dalam informasi pribadi, informasi kesehatan, informasi keuangan yang dapat digunakan dengan baik dalam serangan *phishing*, penipuan kartu, dan penipuan email (Sagar, 2018).

4. *XML External Entities* (XEE) yang bersifat kerentanan esoterik dibandingkan dengan serangan aplikasi *web* lainnya. Penyerangan ini mengirim input data XML. Parser XML yang dikonfigurasi dengan buruk yang berisi referensi ke entitas eksternal. Dalam beberapa kasus, penyerang menjalankan eksekusi kode jarak jauh untuk mengekstrak informasi rahasia, detail pemindaian *port* (Dehalwar et al., 2018).

5. *Broken Access Control* yaitu serangan yang terjadi ketika pembatasan pada hak akses tidak di lindungi dengan benar yang memberikan penyerangan mendapatkan kesempatan untuk mengeksploitasi kelemahan ini dan karenanya dapat mencapai akses ke fungsi dari akun orang lain atau informasi pribadi dari suatu organisasi (Sagar, 2018).

6. *Security Misconfiguration* yaitu serangan yang bersifat yang terjadi di karena kan kesalahan dalam konfigurasi keamanan aplikasi *website* atau *server*. Kesalahan konfigurasi kecil dapat membuat data tersebut dipertaruhkan. Contohnya perangkat lunak yang ketinggalan versi dan mengaktifkan atau menonaktifkan fitur yang tidak diinginkan tanpa mengetahui fungsinya (Sagar, 2018).

7. *Cross-Site Scripting* (XSS) adalah serangan injeksi kode di sisi klien di mana penyerang mengeksekusi skrip / sebuah struktur kode khusus yang dikirimkan ke situs *web* yang sah jika sisi *server* memiliki validasi tersebut maka struktur kode tersebut tidak dijalankan (Sagar, 2018).

8. *Insecure Deserialzation* adalah proses pengambilan data mentah dari file atau soket jaringan untuk merekonstruksi model objek. Dari data yang tidak dipercayai tidak dapat deserialisasi tanpa cukup memverifikasi bahwa data tersebut dihasilkan valid / asli. Data yang di serial dapat digunakan dengan mudah, tetapi data yang tidak disterilisasi dapat mudah dimodifikasi oleh penyerang jika tidak dilindungi oleh fungsi enkripsi (Dehalwar et al., 2018).

9. *Using Components With Know Vulnerabilities* yaitu serangan yang menggunakan sumber terbuka atau *open-source libraries* yang memasang skrip kode serupa dipasang pada server dapat membuat masalah. Dengan menetapkan sebuah kebijakan keamanan yang mengambil atau mengakses semua informasi kesalahan dari *website* dapat meminimalkan risiko menggunakan kerentanan yang diketahui dari serangan di alternatif *website* (Dehalwar et al., 2018).

10. *Insufficient Logging and Monitoring* yang tidak digunakan jika *software* tidak dikonfigurasi dengan benar pada jaringan, maka aktivitas dari jaringan tidak dapat di monitor untuk mengekstrak data tersebut. Selain itu, memerlukan waktu lama untuk diperlukan, terkadang dibutuhkan 200 hari untuk mendeteksi serangan. Dan melaporkan masalah pada waktu yang tepat maka dapat menyediakan solusi sehingga dapat mencegah yang akan datang serangan (Dehalwar et al., 2018).

2.2.9 *Vulnerabilty Assessment*

Merupakan analisis keamanan yang menyeluruh dalam sebuah sistem aplikasi *website* untuk mengetahui seluruh potensi kelemahan kritis yang ada (Subhiyakto & Utomo, 2016), dan memberikan penilaian dari tiap kelemahan yang ditemukan dari *High, Medium,* dan *low severity risk.*

Dari penilaian kerentanan ditemukan dapat dijelaskan sebagai berikut (Dwivedi, 2016):

1. *High severity risk* merupakan kemungkinan bagi seorang peretas dapat mengakses *root, super user* dan berdampak langsung pada operasi sistem. Jenis yang termasuk pada *high severity risk* yaitu *SQL injection* dan *cross-site scripting*.

2. *Medium severity risk* yang dapat menimbulkan gangguan keamanan dan memperoleh akses terbatas pada pengguna. Jenis yang termasuk dalam *medium severity risk* yaitu a*pplication error message* dan *slow http denial of service attack.*

3. *Low severity risk* yaitu sangat sedikit kemungkinan atau peluang bagi seorang peretas dapat mendapatkan akses data. Jenis yang termasuk pada *low severity risk* yaitu *autocomplete enabled* dan *version disclosure*.

2.3.10 *Secure Socket Layer* (SSL)

Merupakan protokol yang digunakan untuk membrowsing *web* secara aman yang dikembangkan pada tahun 1994. SSL yang bertindak sebagai protokol

yang mengamankan komunikasi antara *client* dan *server* untuk memberikan fasilitas enkripsi pada data yang dikirimkan dan membantu menjaminkan integritas informasi dari *web server* dan *web browser* (Aditya, Farida, & Ramadhani, 2018).

2.3 Tools yang digunakan

2.3.1 Geekflare

Adalah sebuah situs *web* yang berfokus pada keamanan *web, cloud computing, hosting, wordpress,* dan *web infrastructure* yang digunakan untuk membaca informasi *header* dari situs target yang menampilkan informasi dari bahasa pemrograman dan server yang digunakan (Zavadskas, Govindan, Antucheviciene, & Turskis, 2016).

2.3.2 Wappalyzer

Yaitu sebuah aplikasi terbuka yang digunakan untuk mengidentifikasi teknologi yang diterapkan pada pembuatan sebuah *web*. Ini umumnya tersedia pada ekstensi browser. Namun dari sumber yang tersedia dapat digunakan sebagai skrip yang diterapkan pada halaman *web* target yang akan di *crawling* (Duarte et al., 2016).

2.3.3 Acunetix

Yaitu sebuah aplikasi pemindai kerentanan web otomatis pada *website* yang dikembangkan untuk melakukan *crawling* dalam menemukan kelemahan dan mengelola tiap lacak dari setiap kerentanan tersebut. Seperti *SQL injection*, *cross-site scripting* dan lainnya. Serta hasil pemindaian memberikan rincian kerentanan, tetap juga memberikan informasi tentang cara memperbaiki kerentanan. (Maharani, Andrian, & Ismail, 2017).

2.3.4 Netsparker

Sebuah aplikasi otomatis yang tidak memerlukan pengetahuan yang lebih untuk menggunakan aplikasi ini. Aplikasi ini dirancang untuk mendeteksi kerentanan keamanan pada *website* dan melakukan audit dari hasil pemindaian yang disusun dalam laporan serta menghemat kan waktu untuk penguji tidak perlu melalukan audit manual dari hasil pemindaian kerentanan yang telah dikonfirmasikan oleh *netsparker* (Joshi & Kumar, 2016).