BAB 1 PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi memberikan kemudahan dalam berbagai aspek kehidupan. Salah satunya dalam pengembangan aplikasi berbasis web. Aplikasi web semakin banyak digunakan untuk berbagai kebutuhan, mulai dari layanan publik hingga transaksi digital. Seiring dengan meningkatnya penggunaan aplikasi *web*, keamanan sistem informasi merupakan elemen penting dalam aplikasi *web*. Sistem yang tidak memiliki keamanan yang memadai beresiko mengalami serangan siber yang dapat menyebabkan pencurian data, pencurian informasi penting, serta gangguan terhadap layanan yang tersedia. Oleh karena itu, penilaian kerentanan *(Vulnerability Assessment*) merupakan langkah penting dalam mengidentifikasi dan mengatasi celah keamanan yang ada pada suatu sistem.

Penelitian yang dilakukan oleh Eshetu et al. [1] menunjukan rentannya situs *web* institusi publik terhadap ancaman keamanan. Mengungkapkan bahwa situs *web* institusi pendidikan di Ethiopia menghadapi ancaman serius akibat lemahnya keamanan siber yang diterapkan. Hal ini menggambarkan bahwa banyak situs *web* institusi publik masih memiliki celah keamanan yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab. Temuan ini menggarisbawahi pentingnya penerapan uji keamanan yang sistematis agar kelemahan tersebut dapat diidentifikasi dan diperbaiki.

Observasi awal pada *website* XYZ ditemukan bahwa *website* tersebut belum menerapkan konfigurasi keamanan secara optimal. Masih ditemukan *header* keamanan yang hilang, dan terdapat kelemahan pada pengaturan *cookie*. Temuan ini menegaskan bahwa meskipun sistem berjalan dengan normal, namun terdapat celah yang berpotensi terjadinya serangan siber jika tidak ditangani.

Vulnerability assessment merupakan metode yang digunakan untuk mengidentifikasi, menganalisis, dan mengklasifikasikan kerentanan keamananan yang terdapat pada sebuah sistem khususnya aplikasi berbasis *web*. Menurut Erick Irawadi Alwi et al. [2]. *Vulnerability assessment* bertujuan untuk mencari celah kerentan yang berpotensi dimanfaatkan oleh pihak yang tidak bertanggung jawab.

Pemilihan kombinasi dua *tools* pemindai keamanan, *OWASP ZAP* dan *Nikto*. Pemilihan *tools* untuk pelaksanaan vulnerability assessment ini didasarkan pada kekuatan spesifik masing-masing *tools*, fokus area pemindaian yang berbeda tetapi komplementer, serta rekomendasi praktik untuk meningkatkan efektivitas dan cakupan deteksi menyeluruh.

Dalam pengujian terhadap *NodeGoat OWASP ZAP* memiliki skor 61% (yang berarti 39% *false positive*) dan *recall* 82%. Sementara itu *Arachni* memiliki skor 100% tetapi recall 18%, menunjukkan bahwa *Arachni* memiliki banyak *true positive*. *Nikto*, dalam sebuah studi, menunjukkan presisi 80% dan *recall* yang sangat rendah (6,2%) di *WebGoat* [3][4].

Dengan penerapan *vulnerability assessment*, diharapakan dapat menjadi persiapan untuk menghindari risiko yang dapat berakibat fatal, serta dapat memberikan panduan praktis bagi pengelola situs *web* XYZ dalam melindungi data dan mengurangi risiko serangan siber.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang diuraikan, maka beberapa permasalahan dirumuskan sebagai berikut:

1. Bagaimana metodologi penilaian kerentanandapat mengidentifikasi dan mengklasifikasikan kerentanan keamanan pada situs *web* XYZ?
2. Seberapa praktis pendekatan kombinasi antara *OWASP ZAP* dan *Nikto* dalam mendeteksi spektrum kerentanan umum pada situs *web* XYZ?
3. Bagaimana perbandingan dan pengukuran akurasi deteksi kerentanan dari *OWASP ZAP* dan *Nikto* berdasarkan metrik evaluasi standar *seperti True Positive, False Positive, Precision,Recall dan F1*?

1.3. Tujuan dan Manfaat

Penelitian ini bertujuan untuk mengidentifikasi dan mengevaluasi kerentanan keamanan pada aplikasi *web* XYZ dengan menggunakan metode *Vulnerability Assessment*, yang memanfaatkan *tools OWASP ZAP* dan *Nikto*. Penelitian ini berusaha untuk memberikan rekomendasi mitigasi berdasarkan tingkat risiko dari

kerentanan yang teridentifikasi, sehingga membantu pengelola situs web dalam meningkatkan keamanan aplikasi. Evaluasi keberhasilan pengujian ini mengacu pada pengelompokkan risiko berdasarkan *CVSS* (Common Vulnerability Scoring System), mengklasfikasikan jenis kerentanan, serta memvalidasi temuan penting untuk mengurangi risiko *false positive*.

Tabel 1. 1. Tabel keterkaitan antara tujuan, pengujian dan kesimpulan.

No.	Tujuan	Pengujian	Kesimpulan
1	Mengidentifikasi kerentanan pada *website* XYZ.	Pemindaian otomatis menggunakan *OWASP ZAP* dan *Nikto*	Berdasarkan hasil pemindaian ditemukan berbagai jenis kerentanan dengan berbagai tingkat risiko dengan rata-rata sedang.
2	Mengevaluasi efektivitas *OWASP ZAP* dan *Nikto* dalam mendeteksi kerentanan.	Membandingkan hasil temuan kedua *tools* terhadap berbagai jenis kerentanan	*OWASP ZAP* unggul pada jenis kerentanan *XSS*, dan *SQLi*. Sedangkan Nikto lebih unggul dalam mendeteksi konfigurasi *server*.
3	Menyusun rekomendasi mitigasi berdasarkan hasil temuan kerentanan	Analisis kerentanan berdasarkan tingkat risiko dan penyusunan rekomendasi mitigasi mengacu pada *CVSS* dan *OWASP Risk Rating*	Kerentanan dan rekomendasi disusun terstruktur sesuai prioritas risiko dari tinggi ke rendah.
4	Mengevaluasi akurasi masing masing *tools* dalam mendeteksi kerentanan valif	Mengacu pada metrik *true positive, false positive, precision, recall, dan .....*	Diperoleh nilai *true positive, false positve, precision*. Terdapat perbedaan akurasi pada kedua *tools*.

1.4. Batasan Masalah

Batasan masalah dalam penelitian ini ditetapkan untuk mengarahkan fokus penelitian dan memastikan penelitian tetap berada dalam ruang lingkup yang sesuai dengan tujuan utama. Batasan dalam penelitian ini meliputi:

- Penilitian ini akan berfokus pada satu *domain*, ini dilakukan untuk menjaga fokus penelitian, mengingat batas waktu pengerjaan.

- Penggunaan alat dalam penelitian ini dibatasi pada *OWASP ZAP* dan *Nikto*. Meskipun ada banyak alat lain seperti *Acunetix*, pilihan ini didasarkan pada pertimbangan bahwa *OWASP ZAP* dan *Nikto* merupakan perangkat lunak *open-source* yang tidak memerlukan biaya, serta secara fungsional saling melengkapi dalam mendeteksi kerentanan pada aplikasi dan *server*.
- Validasi hasil pemindaian dilakukan secara manual melalui observasi dan dokumentasi sistem, tanpa melakukan pengujian penetrasi lanjutan. Hal ini dikarenakan keterbatasan izin keamanan dan untuk menjaga kestabilan layanan pada sistem yang sedang berjalan.
- Hasil penelitian ini direkomendasikan untuk digunakan oleh tim keamanan dan pengelola *website* XYZ. Penelitian ini tidak mencakup penggunaan atau penerapan rekomendasi tersebut oleh pihak eksternal atau pengguna di luar institusi terkait

1.5. Metode Penelitian

Metode yang digunakan dalam peneltian ini adalah metode vulnerability assessment. *Vulnerability assessment* memiliki beberapa tahapan, berikut ini penjelasan lebih rinci:

1.5.1. Perencanaan dan Pengumpulan Informasi

Pengumpulan data tentang *website* XYZ, dengan mengidentifikasi domain utama dan subdomain, mengumpulkan data alamat *IP*, melakukan pemindaian *port* untuk mengidentifikasi layanan yang berjalan pada *server*, yang dapat memberikan indikasi potensi titik masuk bagi penyerang.

1.5.2. Pemindaian

Proses pemindaian kerentanan *website* XYZ ini menggunakan dua *tools* yaitu *OWAZP ZAP* dan Nikto. Pada OWASP ZAP dilakukan automated scanning pemindaian pasif, pemindaian aktif, dan *spidering*. Pemindaian pasif dilakukan untuk menganalisis lalu lintas *HTTP* tanpa mengganggu *server* target. Tujuannya untuk mendeteksi potensi kebocoran informasi [4].

Pemindaian aktif dengan mengirimkan *payload* ke berbagai elemen dalam *website* XYZ untuk menguji kerentanan. *OWASP ZAP* menggunakan

pendekatan berbasis *signature* untuk mendeteksi *server* atau kode aplikasi [10]. Penggunaan fitur *spider OWASP ZAP* untuk memetakan semua halaman yang dapat diakses pada *website* XYZ. Hal ini dapat membantu mengidentifikasi elemen atau *endpoint* tersembunyi yang mungkin terdapat celah keamanan [9].

1.5.3. Analisis Hasil Pemindaian

Setelah pemindaian pada kedua *tools* selesai, hasil dari kedua tools tersebut digabungkan untuk memberikan analisis yang lebih komprehensif. Proses ini memastikan bahwa semua potensi yang ditemukan dari berbagai metode pemindaian dapat teridentifikasi lebih baik [5].

Kerentanan yang ditemukan kemudian diklasifikasikan berdasarkan tingkat risiko dari yang tinggi hingga rendah, ini dilakukan dengan memperhatikan parameter tingkat eksploitasi, dampak pada sistem, dan kemungkinan terjadi eksploitasi [9].

Proses analisis juga melibatkan validasi manual untuk memisahkan hasil *false positive* dari *true positive*. Hal ini bertujuan supaya hanya kerentanan yang valid dilaporkan, sehingga memberikan gambaran yang lebih akurat kepada tim pengelola sistem [5].

1.5.4. Penyusunan Laporan

Berdasarkan tingkatan risiko, rekomendasi mitigasi disusun untuk dapat menangani kerentanan dengan tingkat risiko yang tinggi terlebih dahulu. Langkah ini memungkinan implementasi solusi yang lebih tepat dalam meningkatkan sistem secara efisien [14].

Menyusun laporan yang mencakup temuan kerentanan, analisis dan rekomendasi mitigasi. Laporan ini diharapkan dapat membantu tim pengembang dan pengelola *website* XYZ dalam memperbaiki kerentanan yang ada, memperkuat sistem layanannya, dan juga sebagai panduan.

1.6. Jadwal Pelaksanaan

Tabel 1. 2. Jadwal Pelaksanaan Tugas Akhir

No.	Deskripsi Tahapan	Bulan 1	Bulan 2	Bulan 3	Bulan 4	Bulan 5	Bulan 6
1	Studi Literatur						
2	Pengumpulan Data						
3	Pengumpulan Informasi *Website* XYZ						
4	Proses Pemindaian						
5	Analisis Hasil Pemindaian						
6	Penyusunan Laporan/Buku TA						