BAB III
METODE PENELITIAN

3.1 Metode Penelitian

Dalam penelitian ini terdapat tahapan – tahapan pengumpulan data atau informasi yang diperoleh dari berbagai sumber seperti *literature*, jurnal, buku, karya ilmiah, tugas akhir, dan juga internet. Untuk informasi yang lebih spesifik terkait kerentanan sistem perlu adanya metode untuk menganalisis sistem agar mendapat hasil yang lebih lengkap, yaitu dengan melakukan *scanning* informasi kerentanan sistem menggunakan OWASPZap yang mengacu pada OWASP 10 tahun 2017.

3.2 Alat Kebutuhan Penelitian

Untuk melakukan penelitian ini perlu menggunakan alat yang terdiri dari perangkat lunak dan perangkat keras. Untuk perangkat lunak yang digunakan adalah windows 10, linux Mint, dan OWASPZap *tools*, lalu untuk perangkat keras memerlukan laptop.

Untuk perangkat lunak window 10 adalah salah satu sistem operasi komputer pribadi yang dikembangkan oleh *Microsoft* sebagai bagian dari keluarga sistem operasi Windows NT. Sedangkan linux Mint adalah salah satu sistem operasi dari linux yang berbasis menggunakan debian dengan banyak fitur dan *tools* yang berguna untuk *penetration testing.*

Perangkat keras leptop adalah komputer pribadi yang bentuknya relative lebih kecil dan ringan. Contoh spesifikasi laptop yang dipakai, Tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras

No	***Hardware***	**Spesifikasi**
1	*Processor*	Intel core i5
2	*Memory(RAM)*	8 GB
3	*Hardisk ssd*	128 GB

Lalu untuk perangkat lunak yang akan digunakan untuk pengujian *penetration testing* disajikan dalam, Tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Lunak

No	*Software*	Keterangan
1	Sistem operasi	Windows 10 dan Linux Mint
2	*tools*	OWASPZap 2.10.0

3.3 Tahapan *Penetration Testing*

Didalam pelaksanaan *penetration testing* pada domain usahidsolo.ac.id dengan menggunakan metode OWASP Top 10 mempunyai tahapan yang dapat dilihat pada Gambar 3.1.

Gambar 3.1 Alur *Penetration Testing* Pada Domain Usahidsolo.ac.id

Penjelasan langkah-langkah penetration testing :

a. Mencari informasi mengenai *web* target

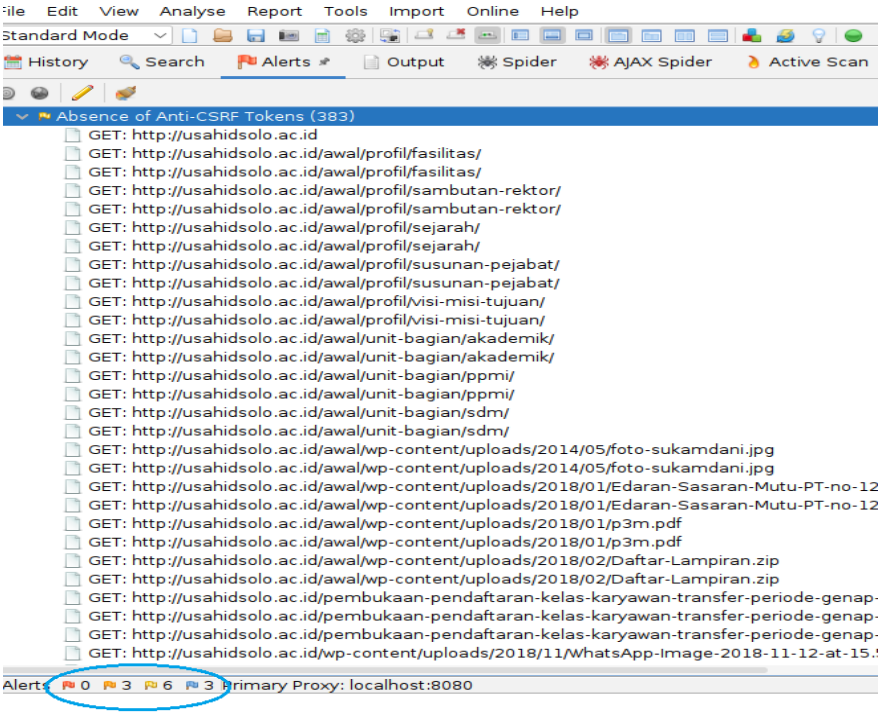
Mulai dari menggali informasi yang sudah ada sebelumnya yang berkaitan dengan kerentanan keamanan website usahidsolo.ac.id ditambah dengan mencari informasi atau data mengenai website usahidsolo.ac.id agar lebih lengkap seperti informasi pada *Google Search Engine* dan WHOIS *website* usahidsolo.ac.id, server hosting pada website usahidsolo.ac.id.

b. Scanning

Selanjutnya adalah proses *scanning* yaitu dimana penyerang atau peneliti mengumpulkan celah keamanan lain yang berhubungan dengan *website* target secara lebih spesifik dan detail. Pada tahap ini tools yang digunakan adalah OWASPZap versi 2.10.0 yang sudah di instal pada sistem operasi Linux Mint.

Tools ini adalah salah satu *tools* otomatis yang akan mencari semua data kemungkinan kerentanan keamanan pada *web* target secara menyeluruh yang akan disajikan dengan kategori tingkat kerentananya.

Tingkat kerentanan pada OWASPZap dibagi menjadi 4 tingkatan berdasar warna, yaitu adalah yang pertama tingkat kerentanan *Hard* dengan simbol bendera berwarna merah, *Medium* dengan simbol bendera berwarna orange, *Low* dengan simbol bendera berwarna kuning, dan terakhir *Informational* dengan simbol bendera berwarna biru. Pada aplikasi OWASPZap dapat dilihat pada bagian *alerts*, Gambar 3.2.



Gambar 3.2 Simbol Bendera Tingkat Kerentanan

c. Pengujian celah keamanan dengan OWASP Top 10 2017

Tahap selanjutnya adalah pengujian celah keamanan dengan berdasarkan OWASP Top10 tahun 2017, data yang didapat dari hasil *scanning* menggunakan OWASPZap sebelumnya dikumpulkan dan diklasifikasikan apakah terdapat kerentanan keamanan yang masuk dalam daftar kerentanan web menurut OWASP top 10 tahun 2017. Sepuluh daftar resiko paling berbahaya menurut OWASP Top10 tahun 2017 adalah sebagai berikut :

1) Serangan injeksi (*SQL Injection)*

Serangan ini terjadi pada hal-hal database baik SQL atau noSQL, sistem operasi, atau server (melewati protokol seperti LDAP). Hal ini terjadi ketika data yang tidak terpercaya dikirim ke *interpreter* sebagai bagian dari kueri atau perintah. Data ini mengelabui *interpreter* agar dapat menjalankan perintah yang seharusnya terlarang bagi orang luar. Contoh : ketika menjalankan e-comerce, dan cara mengakses item tertentu adalah dengan memasukan hal berikut ini ke dalam browser :

- http://www.rownstore.com/catalog/item.asp?itemid=888

Di mana "888" menghasilkan kueri SQL untuk menampilkan item yang sesuai "888". Seorang penyerang dapat memanipulasi hal ini dengan memasukan :

- http://www.rownstore.com/items/item.asp?itemid=888;droptable

Kueri sekarang akan menjadi :

SELECT ItemName, ItemDescription

FROM Item

WHERE ItemNumber = 888; DROP TABLE USERS

Dan hasilnya adalah penghapusan tabel pada database tersebut.

2) Kerusakan Autentikasi *(Broken Authentication)*

Autentikasi yang rusak seperti fungsi autentikasi dan sesi manajemen yang diterapkan tidak benar. Kredensial seperti kata sandi dapat di asumsikan identitas pengguna lain. Serangan memungkinkan akses ke akun admin yang berbahaya bagi sistem. Contoh serangan autentikasi yang rusak adalah *Credential Stuffing,* hal ini terjadi ketika penyerang menggunakan daftar kata sandi yang diketahui (diperoleh dari pelanggaran data) untuk mencoba dan mendapatkan akses dengan aplikasi bertindak sebagai mekanisme validasi setiap upaya sandi. Salah satu solusinya adalah menerapkan pembatasan jumlah upaya login gagal dan menggunakan autentikasi multi faktor.

3) Terpaparnya Data yang Bersifat Sensitif *(Sensitive Data Exposure)*

Eksposur data sensitif terjadi ketika aplikasi web dan API gagal melindungi data sensitif seperti informasi keuangan atau perawatan

kesehatan. Data yang terlindungi secara lemah ini dapat dengan mudah dicuri oleh penyerang untuk melakukan penipuan, pencurian identitas, dan kejahatan lainnya. Contohnya adalah situs web yang tidak menggunakan sertifikat SSL/TLS berkualitas untuk semua halaman, maka penyerang dapat memantau lalu lintas, mengubah koneksi dari HTTPS ke HTTP lalu mencuri sesi cookie untuk mendapatkan akses. Contoh lainnya adalah hash yang kurang. Jika hash sederhana digunakan untuk menyimpan kata sandi dan penyerang memperoleh akses ke database, hash dapat dengan mudah dirusak.

4) *XML External Entities* (XEE)

Serangan ini menargetkan aplikasi web yang mengurai input XML. Prosesor XML yang lebih lama atau tidak dikonfigurasi dengan benar dapat mengevaluasi referensi entitas eksternal (seperti *hard drive*) dalam dokumen XML. Hal ini dapat mengelabui pengurai XML agar mengirimkan data ke entitas eksternal yang tidak sah, yang kemudian dapat mengirim data sensitif langsung ke peretas. Dan penyerang juga bisa mendapatkan informasi tentang jaringan pribadi dengan mengubah baris ENTITY.

Beberapa solusinya dengan menonaktifkan penggunaan entitas eksternal dalam XML serta memvalidasi XML.

5) Kontrol Akses Rusak *(Broken Access Control)*

Terjadi ketika pembatasan tentang apa yang diizinkan dan tidak diizinkan oleh pengguna yang diberlakukan autentikasi secara tidak benar. Lalu serangan dapat memanfaatkan untuk mendapat fungsionalitas yang tidak sah. Contoh mengakses dan mengubah akun pengguna,file sensitif, hak akses, dan data pengguna.

6) Kesalahan Konfigurasi Keamanan *(Security Misconfiguration)*

Kerentanan paling umum pada Daftar OWASP Top Ten dan biasanya disebabkan oleh penggunaan konfigurasi dan kredensial default atau menampilkan pesan kesalahan yang panjang dan tidak perlu. Pesan-pesan ini berpotensi mengungkapkan kerentanan dalam aplikasi. Contohnya web akan menampilkan pesan atau notifikasi error yang menampilkan detail kode aplikasi, yang dapat dimanfaatkan oleh pihak penyerang.

7) *Cross-Site Scripting (XSS)*

Jenis kerentanan ini merupakan hasil dari sesi manajemen yang lemah dan terjadi saat aplikasi web memungkinkan pengguna menambahkan kode khusus ke URL atau situs yang akan ditampilkan kepada orang lain. Contoh seorang peretas berpura-pura dari perusahaan terpercaya dapat mengirim email kepada seseorang, termasuk tautan ke situs perusahaan di dalam email. Tautan tersebut memiliki kode berbahaya yang ditambah pada akhir URL. Jika situs perusahaan tidak diamankan dengan baik maka kode berbahaya tadi akan dijalankan pada browser korban setelah mengkliknya.

8) Deserialisasi yang tidak aman *(Insecure Deserialzation)*

Deserialisasi adalah kebalikan dari serialisasi. Serialisasi adalah ketika objek dari kode aplikasi diubah menjadi format yang dapat digunakan untuk tujuan lain, seperti *streaming.* Dan deserialisasi memungkinkan peretas mengeksekusi kode berbahaya pada server. Contoh: Ketika anda mengirim file dengan data yang lengkap dan penting dalam satu paket file, ketika file yang sudah sampe tujuan deserialisasi tidak aman maka terjadi jika penggerak menambahkan, melepas, dan mengatur ulang file sebelum file dibongkar.

9) Penggunakan Komponen dengan Kerentanan yang Tidak Diketahui *(Using Components With Know Vulnerabilities)*

Web developer sering menggunakan komponen yang ada dalam aplikasi untuk menghindari pekerjaan yang berlebihan sambil menyediakan fungsionalitas yang dibutuhkan. Serangan terjadi ketika penyerang mencari kerentanan di dalam komponen ini yang dapat mereka manfaatkan untuk melakukan serangan pada aplikasi tersebut. Contoh : kasus Pelanggaran Equifax pada tahun 2017 adalah contoh sempurna dari jenis kerentanan ini. Disebabkan oleh penggunaan versi Apache Struts yang memiliki kerentanan yang ditemukan enam bulan sebelum serangan. Andai saja mereka membaca OWASP Top 10 List terlebih dahulu, maka 700 juta USD mereka bisa diselamatkan.

10) Pencatatan dan Pemantauan yang Tidak Memadai *(Insufficient Logging and Monitoring)*

Pencatatan (*logging*) dan pemantauan yang tidak rutin dapat meningkatkan resiko serangan dan menghambat waktu respon suatu situs, hal ini memberi waktu penyerang untuk merusak, mengekstrak, atau menghancurkan data dan mengacak-acaknya. Contoh : kasus Seperti aksi credential stuffing, di mana penyerang berulang kali mencoba menggunakan nama pengguna dan pasangan kata sandi yang bocor. Katakanlah setelah 100 percobaan, mereka akhirnya mencapai kombinasi yang tepat untuk akun tertentu. Karena tidak ada logging atau pemantauan di tempat, tidak ada yang pernah diberitahu tentang tingginya jumlah upaya login pada akun tersebut. Jika tidak, aktivitas tersebut akan dianggap mencurigakan dan pelanggaran dapat dengan mudah dicegah.

d. Pembuatan Laporan Hasil Pengujian

Pembuatan laporan hasil pengujian sebagai tahap akhir laporan ini adalah dengan mencari tingkat kerentanan paling tinggi dengan mengelompokannya pada tabel menurut metode OWASP Top 10 tahun

2017. Dalam tabel 10 daftar standar keamanan OWASP Top tahun 2017 nantinya dapat diketahui seberapa banyak kerentanan website usahidsolo.ac.id dan seberapa tinggi tingkat kerentanannya mulai dari *hard, medium, low* atau *informational.*