SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

Analisis Kerentanan Keamanan Aplikasi Manajemen Aset Berbasis *Web* Menggunakan Metode *OWASP* *(Open Web Application Security Project)* Studi Kasus PT. XYZ

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Strata Satu

Shidqi Anshori Robbani
011021036

PROGRAM STUDI TEKNIK INFORMATIKA
Jakarta
Februari 2021

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : **Shidqi Anshori Robbani**

NIM : **011021036**

Tanda Tangan : **.....**

Tanggal : **23 Februari 2021**

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :
Nama : Shidqi Anshori Robbani
NIM : 0110217036
Program Studi : Teknik Informatika
Judul Skripsi : Analisa Kerentanan Keamanan Aplikasi Manajemen Aset Berbasis *Web* Menggunakan Metode *OWASP (Open Web Application Security Project)* Studi Kasus PT.XYZ

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing

Sirojul Munir, S.Si.,
M.Kom.

Penguji I	Penguji II
Tubagus Rizky Darmawan, S.T.,M.Sc.	Henry Saptono, S.Si., M.Kom.

Ditetapkan di : .....
Tanggal : .....

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Lukman Rosyidi, ST. MM. MT. selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Bapak Ahmad Rio Adriansyah, S. Si M.Si., selaku Wakil Ketua 1 Akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Ibu Tifanny Nabarian, S.Kom. M.T.i., selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Sirojul Munir, S.Si., M.Kom selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
7. Bapak Sirojul Munir, S.Si., M.Kom selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
8. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Jakarta, 23 Februari 2021

Shidqi Anshori Robbani

Jurnal Teknologi Terpadu
Journal of Integrated Technology

https://journal.nurulfikri.ac.id/index.php/JTT
ISSN : 2477-0043 ISSN ONLINE : 2460-7908

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Shidqi Anshori Robbani
NIM : 0110217036
Program Studi : Teknik Informatika
Jenis karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :
ANALISIS KERENTANAN KEAMANAN APLIKASI MANAJEMEN ASET BERBASIS WEB MENGGUNAKAN METODE *OWASP* (OPEN WEB APPLICATION SECURITY PROJECT) STUDI KASUS PT. XYZ

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.
Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
Pada tanggal : 23 Februari 2021
Yang menyatakan

(Shidqi Anshori Robbani)

Diterima (tanggal), Direvisi (tanggal), Diterima untuk publikasi (tanggal

ABSTRAK

Nama : Shidqi Anshori Robbani
NIM : 0110217036
Program Studi : Teknik Informatika
Judul : Analisis Kerentanan Keamanan Aplikasi Manajemen Aset Berbasis Web Menggunakan Metode *OWASP (Open Web Application Security Project)* Studi Kasus PT.XYZ

Tugas Akhir/Skripsi ini membahas tentang (penjelasan ringkas mengenai penelitian) Keamanan dan kerentanan *Website* merupakan tantangan bagi perusahaan serta developer dalam penanganan dan juga perawatan. Pada topik kali ini berfokuskan pada kerentanan apa saja yang ditemukan pada *Website* manajemen aset PT.XYZ untuk dijadikan pertimbangan setelah rilisnya atau perbaikan dikemudian hari. Dengan menggunakan standar yang telah disediakan oleh *OWASP* dan pengujian secara manual pada *Website* manajemen aset, *Website* manajemen aset yang menggunakan laravel versi 8 dan php versi 8 akan diujikan sebelum perilisan yang masih menggunakan server local atau localhost dari sistem operasi windows 10. Penelitian ini diharapkan dalam memberikan informasi serta gambaran dari *Scanning* vulnerability dan *manual testing* untuk menemukan kerentanan.

Kata kunci : *OWASP*, Keamanan *Website*, Kerentanan *Website*, *Penetration Testing*

ABSTRACT

Name : Shidqi Anshori Robbani
NIM : 0110217036
Study Program : Informatics Engineering
Title : Vulnerability Analysis of Web-Based Asset Management Application Using *OWASP* Methodology (Open Web Application Security Project) Case Study of PT.XYZ

This final thesis discusses the topic of security and vulnerability of *Website*s, which pose a challenge for companies and developers in handling and maintenance. This topic focuses on identifying vulnerabilities found in the asset management *Website* of PT.XYZ as consideration for future improvement after its release. The *Website*, which uses Laravel version 8 and PHP version 8, will be tested using the standards provided by *OWASP* and *manual testing* before its release, using a local server on a Windows 10 operating system. This research aims to provide information and an overview of vulnerability *Scanning* and *manual testing* to identify vulnerabilities.

Key words : *OWASP*, *Website* Security, *Website* Vulnerability, Penetration Testing

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABLE

BAB I
PENDAHULUAN

1.1 Latar belakang

Mendeteksi kerentanan sebuah aplikasi berbasis *Webiste* merupakan sebuah kebutuhan dalam menanggapi kemajuan teknologi saat ini, sebab dari berbagai macam kejahatan dalam dunia internet *Website* mudah dijadikan target dari orang-orang yang tidak bertanggung jawab[1]. Termasuk soal pengambilan sebuah data adalah salah satu dari kejahatan yang cukup fatal untuk bisa mengakses sebuah *Website*.

Saat ini aplikasi *Website* menjadi sebuah alternatif termudah untuk diakses oleh siapa pun, kapan pun, dan dimana saja. Pada tahun 2015 saja bahkan Indonesia sudah diperkirakan mengalami lonjakan penggunaan internet yang dimana peningkatan penggunaan internet sudah dimulai sejak tahun 2009 dan setiap tahunnya akan terus meningkat[2].

Kemajuan teknologi setiap tahunnya tidak menutup kecil kemungkinan dari seseorang untuk dapat membuat sebuah kejahatan dalam media informasi yang saat ini perkembangannya akan lebih meningkat lagi. Berbagai kasus kejahatan melalui internet saat ini sudah tidak asing lagi bagi banyak masyarakat dalam kasus yang dapat kita sebut juga sebagai *cyber crime*[1].

Sebuah situs *Website* dapat dengan mudah diretas atau disusupi oleh banyak pihak yang disebut dengan *hacker*, indonesia disebutkan bahwa menjadi negara hacker terbesar ketiga di dunia yang termasuk kota semarang dan kota yogyakarta menjadi peringkat teratas. Karena sebuah kejahatan dalam media internet sangatlah merugikan masyarakat dan negara.

Keamanan pada kerentanan sebuah *Website* menjadi salah satu keharusan yang dilakukan untuk membuat sebuah *Website*[3]. Dengan melakukan analisa terhadap sebuah *Website* yang dirancang ataupun akan dirilis menjadikan kita dapat dengan mudah melakukan sebuah evaluasi untuk mencegah dari sebuah kejahatan terjadi.

Salah satu metode untuk analisa *Website* adalah *Open Web Application Security Project (OWASP). OWASP* dapat menjadi sebuah alternatif dalam analisa kerentanan keamanan dari *Website* guna untuk mengetahui celah dari *Website* kita yang dengan mudah disusupi oleh *hacker*. *OWASP* menyediakan *tools* dan juga beberapa kerentanan yang dapat diindikasikan akan terjadinya sebuah serangan pada sebuah *Website*[4].

Pada penelitian ini akan digunakan metode *OWASP* pada aplikasi manajemen aset berbasis *Website* pada PT.XYZ.

1.2 Perumusan Masalah

Adapun perumusan masalah pada Tugas Akhir ini adalah :

1. Bagaimana tahap dalam menemukan kerentanan terhadap aplikasi manajemen aset dengan menggunakan metode *OWASP*?
2. Melakukan uji coba pada salah satu celah keamanan berdasarkan *OWASP* yang terdapat pada aplikasi manajemen aset PT.XYZ?

1.3 Tujuan Penelitian

Tujuan dari Tugas Akhir ini yang dihasilkan adalah :

1. Melakukan evaluasi pada *Website* manajemen aset PT.XYZ dengan menggunakan metode *OWASP*
2. Mendapatkan hasil uji coba kerentanan keamanan pada aplikasi *Website* manajemen aset PT.XYZ untuk rekomendasi perbaikan

1.4 Manfaat Penelitian

Manfaat yang didapatkan dari dibuatnya Tugas Akhir ini adalah :

1. Mengetahui potensi kerentanan pada salah satu serangan pada *Website* manajemen aset PT.XYZ berdasarkan *OWASP*
2. Mengetahui jenis ancaman dari salah satu kerentanan *OWASP* yang dapat menyerang *Website* manajemen aset PT.XYZ

1.5 Batasan Masalah

Tugas Akhir ini dibatasi dengan beberapa struktural pembahasan yaitu :

1. Penelitian ini hanya dilakukan pada aplikasi *Website* manajemen aset PT.XYZ
2. Pada testing *Website* manajemen aset akan dilakukan menggunakan *testing manual* pada *OWASP Top 10* dan dilakukan minimal 2 kerentanan yang diujikan pada *OWASP Top 10*.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir ini adalah sebagai berikut :

1. **BAB I PENDAHULUAN**

   Merupakan bab pembuka yang memberikan gambaran umum mengenai proses pelaksanaan Tugas Akhir. Bab ini terdiri dari latar belakang, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

2. **BAB II KAJIAN LITERATUR**

   Bab ini akan mengkaji lebih dalam mengenai teori dan literatur yang akan mendukung proses pengkajian dan dijadikan penulis sebagai bahan penelitian seperti pemahaman tentang keamanan sebuah *Website*, pemahaman terkait *OWASP* dan *tools OWASP ZAP*, dan yang lainnya.

3. **BAB III METODOLOGI PENELITIAN**

   Pada bab ini menjelaskan tentang metode yang akan dilakukan oleh penulis serta tahapan-tahapan untuk mengimplementasikan penelitian yang akan dibuat. Pada bab ini juga tahapan-tahapan itu lah yang akan digunakan oleh penulis dalam menunjang penelitian ini.

4. **BAB IV ANALISA DAN RANCANGAN**

   Pada bab ini menjelaskan tentang metode yang akan dilakukan oleh penulis serta tahapan-tahapan untuk melakukan implementasikan penelitian yang akan dibuat. Pada bab ini juga tahapan-tahapan itu lah yang akan digunakan oleh penulis dalam menunjang penelitian ini.

5. **BAB V IMPLEMENTASI**

   Pada bab ini merupakan implementasai dari penelitian yang dilakukan setelah melakukan rancangan serta analisa dan rancangan yang telah disusun. Kemudian hasil implementasi yang juga akan menjadi laporan dari penelitian ini

6. **BAB VI KESIMPULAN DAN SARAN**

   Kesimpulan dari hasil penelitian yang telah dilakukan, dan juga akan dijadikan laporan atau laporan untuk menjadi pertimbangan pada aplikasi manajemen aset untuk dilakukannya perbaikan. Dan terakhir saran dari penulisan akan semua penelitian dan proses yang telah dilakukan

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada bab ini akan membahas terkait referensi atau sumber yang mendukung untuk dibuatnya sebuah tugas akhie, untuk memudahkan menulis dalam melakukan studi literatur. Didalam landasan teori mencakup dalam hal pemahaman keamanan sistem, *Open Web Application Security project, tools* dari *penetration testing,* dan penelitian terkait untuk memudahkan dalam penulisan tugas akhir.

2.1.1 Keamanan Sistem

Keamanan sistem pada sebuah *Website* merupakan sebuah keharusan yang dilakukan oleh seseorang yang akan atau sedang dalam tahap dalam pembangunan aplikasi berbasis *Website*, dikarenakan akan dengan mudah seseorang dalam melakukan penyusupan ataupun tindak kejahatan dalam dunia internet yang sudah semakin maju untuk saat ini.

Dengan adanya suatu sistem keamanan, maka dengan mudah dalam menjaga informasi yang dimiliki oleh seorang pengembang, baik itu dalam melakukan evaluasi ataupun perbaikan untuk menunjang keamanan sebuah sistem. Disebutkan bahwa dalam lebih dari 70% upaya peretasan dilakukan melalui aplikasi *Website* [5]

Dalam jurnal yang disebutkan oleh Guntoro pada tahun 2020, keamanan informasi disimpulkan pada 2 hal, yaitu:

1. *Threats*

   Ancaman terhadap hilangnya dari sebuah informasi yang termasuk kedalam bahaya alam (banjir, gempa bumi, dan tsunami), manusia (perusak, peretas, pengurangan tenaga listrik).

2. *Vulnerability*

   Paramater untuk menganalisa celah sebuah keamanan adalah *Confidentiality* (kerahasiaan), *Integrity* (intergritas) dan *Aviability* (ketersediaan) menjadi sebuah acuan standar.

2.1.1.1 Open Web Application Security project

Open Web Application Security Project (OWASP) adalah sebuah organisasi yang berfokus pada sebuah keamanan *Website* yang dimana mereka memberikan sebuah informasi baik dalam dokumentasi, *tools*, dan forum. Mereka menargetkan pengembang aplikasi berbasis *Website* untuk menjadikan *Website* yang sedang mereka kerjakan atau bahkan siap untuk dirilis agar sistem tetap aman.

Gambar 1 OWASP

OWASP menerapkan *open source* bagi penggunanya, yang dimana bagi penggunanya bebas jika tertarik untuk memperbaiki keamanan aplikasi mereka, dengan mudahnya dalam penggunaan dan download *tools* maupun dokumentasi yang disediakan menjadikan setiap pengembang aplikasi berbasis *Website* untuk membuat *Website* mereka lebih aman dan terjaga dari ancaman *hacker*.

Dalam perilisannya *OWASP* menyediakan 10 resiko yang sering terjadi dalam sebuah aplikasi *Website*. Pada 10 jenis resiko dari serangan ini yang dapat diketahui serta mendapatkan hasil dari sebuah serangan tersebut. Berikut merupakan yang disebut dengan *Top 10 Web Application Security Risk* yang disediakan oleh sebuah perusahaan bernama Sucuri dengan judul *OWASP Top 10 Vulnerabilities,* dalam bukunya (Sucuri Corporation, 2019) :

Gambar 2 Top 10 OWASP 2021

1. ***Broken Access Control***

   Kerentanan ini terjadi ketika kontrol akses tidak diterapkan dengan benar pada aplikasi atau sistem, sehingga pengguna dapat mengakses sumber daya yang tidak diizinkan. Penyerang dapat memanfaatkan kerentanan ini untuk melakukan tindakan yang tidak diinginkan, seperti mencuri data sensitif atau merusak sistem.

2. ***Cryptographic Failures***

   Cryptographic Failures terjadi ketika teknik kriptografi yang tidak aman digunakan untuk menyimpan atau mengirim data sensitif, memungkinkan penyerang untuk mencuri data sensitif dan merusak integritas data.

3. ***Injection***

Pengambilan atau pencurian terhadap data yang dimiliki sebuah *Website* adalah salah satu hal yang perlu diperhatikan setiap *developer* sebuah *Website*, ketika input dari pengguna tidak divalidasi dengan benar. Hal ini dapat memungkinkan serangan *SQL injection* atau *NoSQL injection*, di mana penyerang dapat mengakses atau mengubah data di dalam database dan merusak integritas data.

4. ***Insecure Design***

Ketika kelemahan keamanan terdapat pada perancangan atau arsitektur aplikasi. Penyerang dapat menyerang sistem dengan memanfaatkan kelemahan ini dan mencuri data sensitif atau merusak sistem.

5. ***Security Misconfiguration***

Membatasi pengguna dengan admin adalah celah yang mudah untuk digunakan oleh peretas, ketika konfigurasi keamanan tidak diatur dengan benar pada sistem atau aplikasi. Penyerang dapat memanfaatkan konfigurasi yang salah dan mengakses sumber daya yang tidak diizinkan atau melakukan tindakan yang tidak diinginkan.

6. ***Vulnerable and Outdated Components***

Peretas selalu mencari cara salah satunya terjadi ketika kelemahan keamanan terdapat pada perancangan atau arsitektur aplikasi, memungkinkan penyerang untuk menyerang sistem dengan memanfaatkan kelemahan tersebut. ketika komponen perangkat lunak yang digunakan oleh aplikasi tidak diperbarui atau memiliki kerentanan. Penyerang dapat memanfaatkan kerentanan pada komponen ini dan merusak sistem atau mencuri data sensitif.

7. **Identification and Authentication Failures**

Penulisan *code* yang membuat seorang peretas memasukan konten lain dari *Website* kita sehingga memaksa pengguna untuk mendatangi *code* yang

tidak valid itu sehingga meminta untuk memasukkan informasi terkait pada *Website*. Hasilnya identitas atau autentikasi pengguna tidak dikelola dengan benar. Penyerang dapat memanfaatkan kerentanan ini untuk mendapatkan akses tidak sah ke sistem atau mencuri data sensitif.

8. Software and Data Integrity Failures

Pemanfaatan *input* data serta *output* dari data yang dihasilkan sering kali dimanfaatkan oleh peretas untuk para pengguna menyadari bahwa hasil unduhan yang didapatkan berasal dari *Website*. Kerentanan ini terjadi ketika keutuhan perangkat lunak atau data terancam. Penyerang dapat memanfaatkan kerentanan ini untuk merusak sistem atau mengubah data dengan cara yang tidak diinginkan.

9. Security Logging and Monitoring Failures

Penggunaan *framework* atau *code* program yang sudah usang dapat memicu kerentanan pada keamanan, perawatan dan pembaruan yang tidak rutin dilakukan akan dimanfaatkan oleh peretas untuk mendapatkan informasi dari *Website* yang kita miliki. Kerentanan ini terjadi ketika *log* keamanan atau sistem pemantauan tidak berfungsi dengan benar. Hal ini dapat menyebabkan penyerang dapat melakukan tindakan yang tidak diinginkan tanpa terdeteksi.

10. Server-Side Request Forgery

Pemindaian *log* yang lemah atau tidak melakukan perawatan pada *session* jarak jauh menyebabkan sebuah *Website* rentan untuk disusupi oleh peretas, sehingga mudah untuk dilakukan peretasan berkala pada sebuah *Website*, aplikasi tidak memvalidasi input yang diterima dari pengguna dan memungkinkan penyerang untuk mengirimkan permintaan ke *server internal*. Penyerang dapat memanfaatkan kerentanan ini untuk mendapatkan akses ke sistem atau mencuri data sensitif.

2.1.2 *Tools Pengembangan*

2.1.2.1 Framework Laravel

Framework laravel merupakan sebuah kerangka atau bagan dalam pembuatan sebuah aplikasi berbasis *Website* yang menyediakan segala aspek dalam kebutuhan seorang *developer*. *Laravel* juga merupakan *framework bundle*, migrasi dan *artisan CLI (Command Line Interface)* yang menawarkan seperangkat alat dan arsitektur aplikasi yang menggabungkan banyak fitur terbaik dari kerangka kerja seperti *Codeigniter, Yii, ASP.NET MVC, Ruby on Rails, Sinatra* dan lain-lain [6].

Pada alur kerja sebuah *framework laravel* juga mengambil skema *MVC* atau disebut juga dengan *Model, Views*, dan *Controller*. Skema ini merupakan alur yang memisahkan antara logika dalam menampilkan komponen aplikasi seperti manipulasi data, *user interface*, dan yang lainnya [6]. Dengan menerapkan skema atau alur dari *MVC* ini lah maka akan digunakan dengan mudah oleh seorang *developer* dalam mengembangkan aplikasi *Website* yang akan dia buat. *Website* manajemen aset pun terhubung dengan *database* yang menggunakan *MySQL*, untuk pemrosesan datanya.

2.1.3 *Pengujian Sistem*

2.1.3.1 Penetration Testing

Penetration testing adalah sebuah pengujian pada keamanan *Website* yang bertujuan untuk mendapatkan hasil dari sebuah kerentanan pada aplikasi *Website* [7]. Hasil dari pengujian itu kemudian akan dilakukan sebuah pengujian yang berguna untuk menambah keamanan yang sebelumnya sangat rentan terhadap sebuah serangan. Evaluasi pun dilakukan dengan melakukan sebuah simulasi serangan pada *Website* kita dan mencari celah keamanan. Hasil ini pun akan dibuat dalam sebuah laporan yang menunjukan celah keamanan yang berhasil lolos dari keamanan, maka akan dilakukan tindak pencegahan untuk memperbaikinya.

Penetration testing merupakan sebuah langkah penting dalam pengujian dalam pengembangan sistem pertahanan [8].

2.1.3.3 BlackBox Testing

Pengujian Black box testing merupakan salah satu metode pengujian perangkat lunak yang dilakukan dengan melihat perangkat lunak sebagai sebuah kotak hitam tanpa mengetahui detail dan cara kerja di dalamnya. Dalam metode ini, pengujian dilakukan berdasarkan input yang diberikan dan output yang dihasilkan oleh perangkat lunak tersebut. Tujuan dari black box testing adalah untuk mengetahui apakah perangkat lunak sudah berjalan sesuai dengan spesifikasi yang diinginkan dan mengidentifikasi potensi kesalahan atau bug yang mungkin terjadi.

Metode pengujian black box testing dapat dilakukan secara manual atau otomatis. Pada pengujian manual, tester akan melakukan simulasi penggunaan perangkat lunak seperti pengguna biasa. Sedangkan pada pengujian otomatis, pengujian dilakukan dengan menggunakan program komputer yang dapat menghasilkan input dan memeriksa output yang dihasilkan oleh perangkat lunak secara otomatis. Pengujian otomatis lebih efisien dan efektif dalam mengidentifikasi bug, namun pengujian manual tetap diperlukan untuk menguji aspek-aspek yang sulit diidentifikasi secara otomatis seperti usability dan user experience. Dalam pengujian perangkat lunak, black box testing sangat penting untuk memastikan kualitas dan keamanan perangkat lunak sebelum diluncurkan ke publik.

2.1.3.3 Tahapan OWASP

OWASP (Open Web Application Security Project) memiliki 4 fase utama dalam mengatasi masalah keamanan aplikasi Web:

1. Mengidentifikasi Ancaman (Threat Identification): tahap ini adalah langkah pertama dalam mengatasi masalah keamanan aplikasi Web. Pada tahap ini, para pengembang dan profesional keamanan harus mengidentifikasi ancaman-ancaman yang mungkin terjadi pada aplikasi Web yang sedang dikembangkan.
2. Memperkirakan Risiko (Risk Estimation): setelah mengidentifikasi ancaman-ancaman, dilakukan penilaian terhadap risiko yang muncul dari

ancaman tersebut. Pada tahap ini, para pengembang dan profesional keamanan harus menentukan tingkat risiko yang ditimbulkan dari ancaman tersebut.

3. Mengimplementasikan Langkah-langkah Keamanan (Security Implementation): setelah menentukan tingkat risiko, pada tahap ini dilakukan rancangan dan pengimplementasian langkah-langkah keamanan untuk mengurangi risiko yang telah diidentifikasi pada tahap sebelumnya.

4. Memantau dan Meningkatkan Keamanan (Security Monitoring and Improvement): setelah aplikasi Web diluncurkan, aplikasi tersebut harus terus dipantau untuk memastikan bahwa aplikasi tersebut tetap aman dan terhindar dari ancaman keamanan yang muncul. Pada tahap ini juga dilakukan peningkatan terhadap langkah-langkah keamanan yang telah diimplementasikan sebelumnya, berdasarkan hasil dari monitoring keamanan.

Setiap tahap dalam siklus *OWASP* harus dilakukan dengan teliti dan benar, untuk memastikan bahwa aplikasi Web yang dikembangkan memiliki tingkat keamanan yang optimal.

2.2 Penelitian Terkait

2.2.1 Tabel Penelitian Terkait

Adapun penelitian yang terkait dengan penelitian ini adalah :

No	Judul Penelitian	Tahun	Kesimpulan
1	*Penetration Testing* Pada *Domain* uii.ac.id Menggunakan *OWASP* 10	2018	Pada penelitian ini difokuskan kepada *penetration testing* menggunakan berbagai macam *tools* salah satunya adalah *OWASP ZAP* dengan menggunakan metode *OWASP* dan memfokuskan terhadap *Website* Universitas Islam Indonesia pada setiap aspeknya.

2	ANALISIS KERENTANAN APLIKASI BERBASIS *WEB* MENGGUNAKAN KOMBINASI *SECURITY TOOLS PROJECT* BERDASARKAN *FRAMEWORK OWASP* VERSI 4	2015	Pada penelitian ini menggunakan tools *vulnerability scanner* dan beberapa *tools* security project dalam pengujiannya dengan pemodelan atau metode terhadap *OWASP* versi 4, penelitian ini akan menetapkan struktur pada fitur apa saja kerentanan akan diujikan.
3	Mendeteksi Kerentanan Keamanan Aplikasi *Website* Menggunakan Metode *OWASP (Open Web Application Security Project)* Untuk Penilaian *Risk Rating*	2019	Penelitian ini terfokus pada hasil nilai *risk rating* dari hasil pengujian dengan menggunakan *OWASP* dan melakukan perbandingan kerentanan keamanan *Website laravel* dengan *code igniter*, pada setiap pengujiannya maka akan dihitung dalam perhitungan *risk rating*. Dengan menggunakan aplikasi *acunetix*
4	Analisis Kerentanan Keamanan Aplikasi Manajemen Aset Berbasis *Website*	2021	Pada penelitian ini berfokus pada *Website* manajemen aset untuk melakukan evaluasi dan perbaikan dalam hal keamanan *Website* dengan menggunakan metode *OWASP* dan *tools OWASP ZAP*

	Menggunakan Metode *OWASP*		

Table 1 Penelitian Terkait

2.2.2 Posisi Penelitian

No.	*OWASP ZAP*	*OWASP*	*Laravel*
1	Adetya Putra Dewanto (UNIVERSITAS ISLAM INDONESIA) *PENETRATION TESTING* PADA *DOMAIN* UII.AC.ID MENGGUNAKAN *OWASP* 10		
2		Yunus, Moh (Universitas Gunadarma) ANALISIS KERENTANAN APLIKASI BERBASIS WEB MENGGUNAKAN KOMBINASI *SECURITY TOOLS PROJECT* BERDASARKAN *FRAMEWORK OWASP* VERSI 4	
3		Bahrun Ghozali, Kusrini, Sudarmawan (Universitas Amikom Yogyakarta) Mendeteksi Kerentanan Keamanan Aplikasi *Website*	

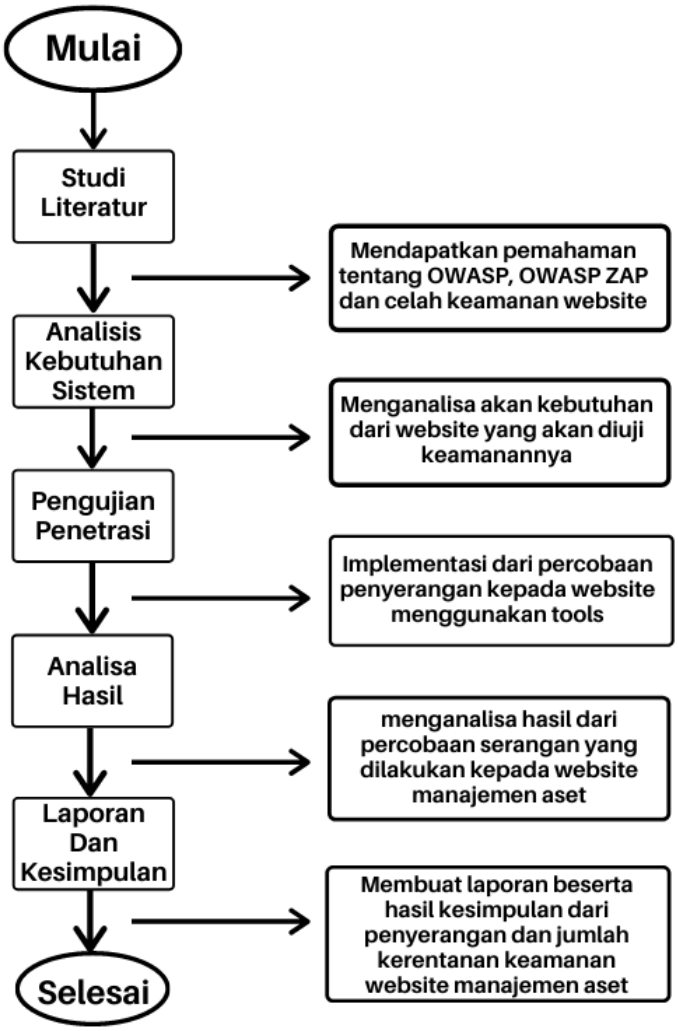
		Menggunakan Metode *OWASP (Open Web Application Security Project)* untuk Penilaian *Risk Rating*
4	Shidqi Anshori Robbani (Sekolah Tinggi Teknologi Terpadu Nurul Fikri) Analisis Kerentanan Keamanan Aplikasi Manajemen Aset Berbasis *Web* Menggunakan Metode *OWASP (Open Web Application Security Project)* Studi Kasus PT. XYZ	

Table 2 Posisi Penelitian

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Pada tahapan penelitian kali ini dengan langkah-langkah yang terstruktur, tujuannya adalah untuk memudahkan rangkaian analisa kerentanan sebuah aplikasi berbasis *Website*. Tahapan-tahapan disini adalah studi literatur, analisis kebutuhan sistem, pengujian penetrasi, analisa hasil, dan laporan dan kesimpulan.



Gambar 3 Tahapan Penelitian

1. Studi Literatur

   Pada tahap ini akan dilakukannya pencarian berbagai sumber referensi dan teori yang akan digunakan sebagai landasan dalam pembuatan tugas akhir dan melakukan penelitian. Beberapa teori yang mendukung dalam tahap ini adalah seperti pemahaman dari *OWASP*, teori *tools OWASP ZAP* dan sumber dari beberapa penelitian terkait terhadap tugas akhir ini. *Output* yang akan dihasilkan dari tahap ini adalah penulis akan mendapatkan beberapa teori yang berkaitan dengan tugas akhir ini untuk dapat melakukan implementasi.

2. Analisa Kebutuhan sistem

   Kemudian setelah melakukan studi literatur, maka tahap selanjutnya adalah dengan menganalisa dari kebutuhan sistem yang diperlukan, seperti kebutuhan dari *tools OWASP ZAP*, kelengkapan dari sistem aplikasi *Website*, dan kebutuhan penilaian pada analisa yang akan dilakukan. Pada tahap ini penulis akan mendapatkan *output* untuk hasil dari kebutuhan apa saja yang akan dan harus dilengkapi sebelum melakukan pengujian sistem keamanan *Website* manajemen aset.

3. Pengujian penetrasi

   Tahap ini merupakan impelementasi dalam tahap pengujian sebuah sistem keamanan *Website* manajemen aset untuk melihat kerentanan apa saja yang ditemukan pada saat pengajuan ini dilakukan, kemudia *output* yang akan ditemukan disini adalah beberapa kerentanan yang ditemukan ketika dilakukannya *penetration testing*.

4. Analisa dan Rancangan

   Pada saat *penetration testing* dilakukan, maka akan terlihat dan mendapatkan dari hasil kerentanan *Website* manajemen aset, setelah itu maka akan dilakukan analisa dari hasil yang didapatkan ketika melakukan *penetration testing. Output* dari tahap ini pun akan menghasilkan penilaian seberapa rentan keamanan *Website* manajemen aset.

5. Laporan dan Kesimpulan

   Tahap terakhir dari tugas akhir ini adalah membuat hasil dari analisa ketika sudah melakukan *penetration testing* dan akan dibuatnya sebuah kesimpulan dan laporan dari

tugas akhir ini. Output yang akan dihasilkan berupa sebuah kesimpulan kerentanan keamanan *Website* manajemen aset untuk dilakukannya perbaikan dan evaluasi.

3.2 Rancangan Penelitian

3.2.1 Jenis Penelitian

Jenis penelitian pada penulisan tugas akhir ini mencakup pada analisa aplikasi manajemen aset yang masih dalam tahap pengembangan, akan dilakukan pengujian kerentanan keamanan *Website* yang memungkinkan akan bisa dilakukan yang telah dibuat pada aplikasi manajemen aset PT.XYZ. Dengan menggunakan penelitian observasi memungkinkan dalam pengujian langsung terhadap *Website* yang ditentukan [9]. Menurut Kiki Joesyiana yang dituliskan pada jurnalnya bahwa kelebihan dari penelitian observasi adalah menyajikan media objek secara nyata, dan mudah dalam pelaksanaannya, dalam metode observasi inilah hasil yang didapat dari pengumpulan data dan uji coba secara langsung dapat dianalisa hasilnya [9].

3.2.2 Metode Pengumpulan Data

Pengumpulan data yang dilakukan dalam pengerjaan tugas akhir adalah dengan menggunakan teknik studi literatur dan observasi. Pada tahap ini dilakukan dengan mengumpulkan jurnal, buku, paper, dan juga artikel-artikel terkait yang berhubungan dengan tugas akhir ini [9], serta dengan adanya observasi akan dengan mudah menemukan langkah dan cara dalam pengerjaan dan penyelesaian tugas akhir ini.

3.2.3 Metode Analisis

Pada metode analisis tugas akhir ini akan berbasis pada studi literatur dengan menggunakan metode *OWASP* dan data-data yang sudah dikumpulkan sebagai pendukung untuk mendapatkan hasil dari analisis sebuah kerentanan pada aplikasi *Website* manajemen aset PT.XYZ memahami sebuah level dari tingkat kerentanan aplikasi *Website*. Hasil dari analisa inilah akan dilakukan setelah mendapatkan hasil dari pengujian dengan menggunakan metode *OWASP*.

3.2.4 Metode Pengujian

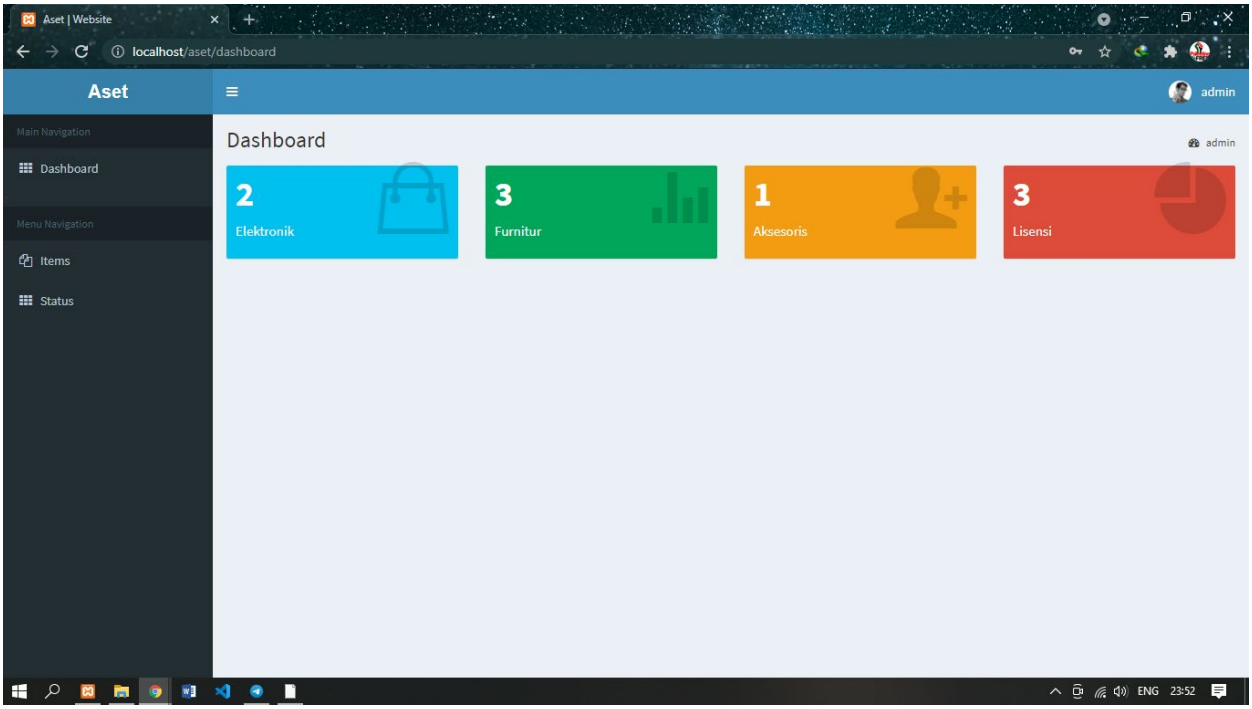
Pengujian Black box testing merupakan salah satu metode pengujian perangkat lunak yang dilakukan dengan melihat perangkat lunak sebagai sebuah kotak hitam

tanpa mengetahui detail dan cara kerja di dalamnya. Dalam metode ini, pengujian dilakukan berdasarkan input yang diberikan dan output yang dihasilkan oleh perangkat lunak tersebut. Tujuan dari black box testing adalah untuk mengetahui apakah perangkat lunak sudah berjalan sesuai dengan spesifikasi yang diinginkan dan mengidentifikasi potensi kesalahan atau bug yang mungkin terjadi.

Metode pengujian black box testing dapat dilakukan secara manual atau otomatis. Pada pengujian manual, tester akan melakukan simulasi penggunaan perangkat lunak seperti pengguna biasa. Sedangkan pada pengujian otomatis, pengujian dilakukan dengan menggunakan program komputer yang dapat menghasilkan input dan memeriksa output yang dihasilkan oleh perangkat lunak secara otomatis. Pengujian otomatis lebih efisien dan efektif dalam mengidentifikasi bug, namun pengujian manual tetap diperlukan untuk menguji aspek-aspek yang sulit diidentifikasi secara otomatis seperti usability dan user experience. Dalam pengujian perangkat lunak, black box testing sangat penting untuk memastikan kualitas dan keamanan perangkat lunak sebelum diluncurkan ke publik.

3.2.5 Objek Penelitian

Aplikasi manajemen aset adalah sebuah perancangan yang bertujuan untuk mengatur segala macam kebutuhan perusahaan, dengan adanya sistem manajemen aset maka sebuah perusahaan dapat dengan mudah mendata apa saja aset yang masuk, keluar bahkan dalam status sudah tidak terpakai.



Gambar 4 Website PT.XYZ

3.2.6 Alat dan Bahan

Penelitian ini akan dilakukan dengan membutuhkan perangkat keras yang akan digunakan untuk menjalankan software dari *OWASP* atau yang disebut juga *OWASP ZAP*, maka dibutuhkannya perangkat dengan spesifikasi laptop berikut,

Komponen	Spesifikasi yang digunakan
Processor	AMD A8-6410
RAM	12GB
Storage Memory	1TB
OS	Windows 10 64 bit

Table 3 Spesifikasi Perangkat

Kemudian untuk penggunaan software untuk penelitian ini adalah :

1. *Mozila Firefox*
2. *OWASP ZAP*
3. *Framework Laravel*
4. *SQL Map*

3.3 Waktu Penelitian

Tugas		Januari	Februari	Maret	April	Mei	Juni
Studi Literasi							
Penyusunan Proposal							
Seminar Proposal							
Analisa Kebutuhan Sistem							
Pengujian Penetrasi							
Penyusunan Seminar Hasil							

Gambar 5 Waktu Penelitian

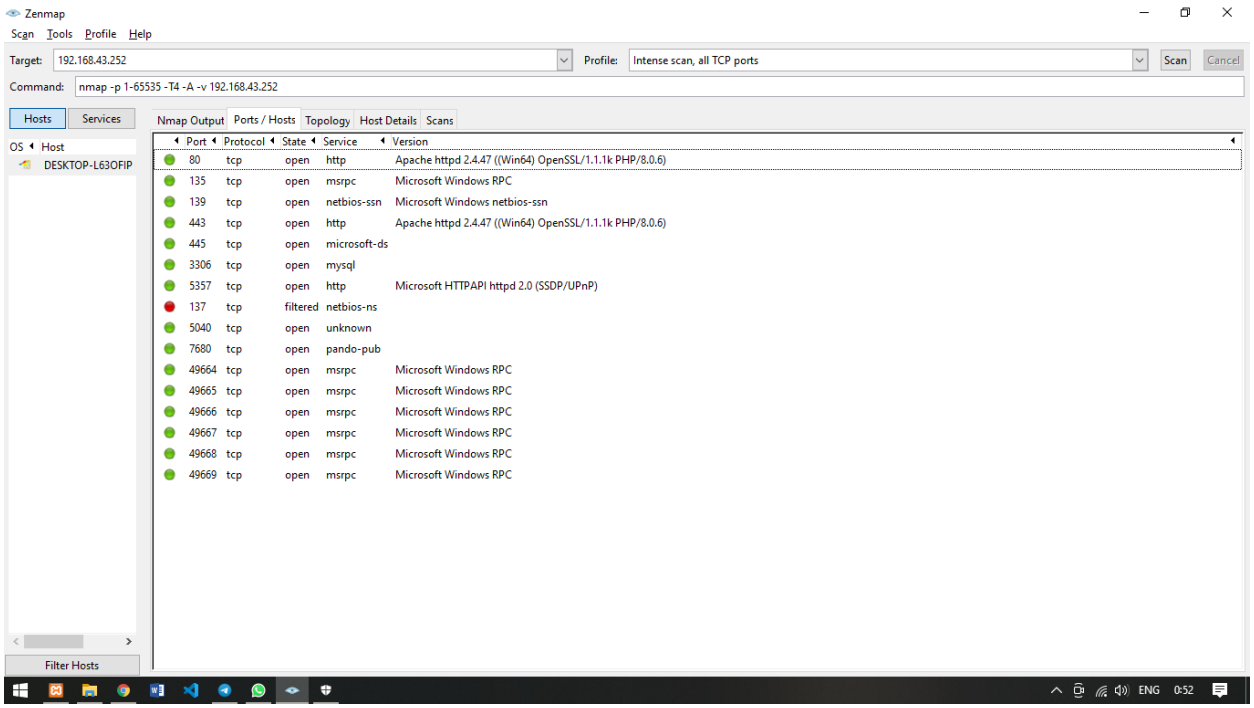
BAB IV

Analisa dan Tahapan

4.1 Analisa

4.1.1 Analisa Kebutuhan Sistem

Pada tahap ini dilakukan pencarian dan analisa pada *Website* manajemen aset yang akan di analisa, dengan menggunakan *Zenmap* maka akan ditemukan beberapa kebutuhan seperti dijalakan dengan menggunakan apa dan menggunakan *port* berapa pada aplikasi *Website* manajemen aset.



Gambar 6 Zenmap

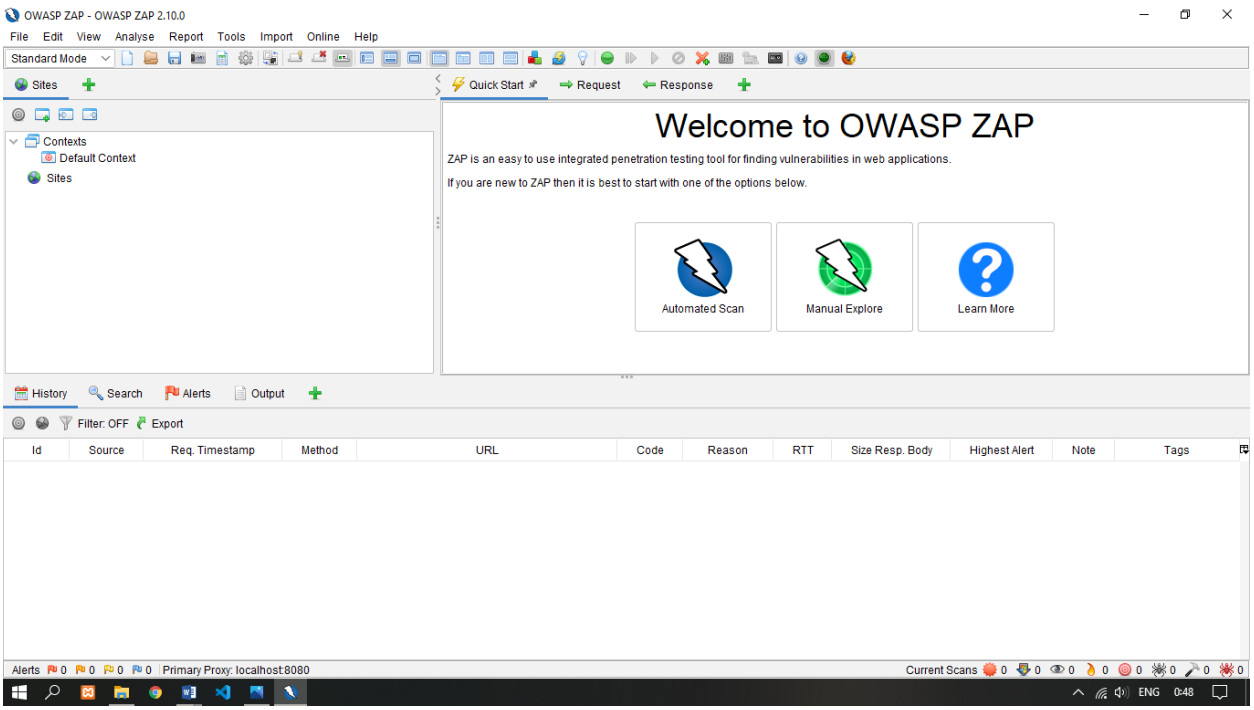
Nama *Website*	*IP Address*	Informasi *Website*				
		PHP Version	***Laravel Version***	***Hostname: Port***	***Status***	***Services***
Manajemen Aset	192.168.43.251	8.0.6	8.47.0	Localhost:80	Open SSL	http

Table 4 Hasil Implementasi Zenmap (informasi Website)

Dari hasil analisa kebutuhan sistem diatas dapat diambil data terkait penelitian ini bahwa *Website* manajemen aset dijalankan menggunakan localhost pada server local dengan *ip address* http://192.168.43.251 yang berjalan pada *port* 80 dengan status Open SSL dan service http. Dari penggunaan *framework* dan juga bahasa pemrograman menggunakan *laravel* versi 8 dan *php* versi 8, penulis menggunakan *default* pada *laravel* dari *auth* sampai beberapa dokumentasi untuk menilai dari sebuah kerentanan pada *Website* manajemen aset sebelum dirilisnya *Website* tersebut.

4.1.2 *Penetration Testing*

Pada pengujian penetrasi pada manajemen aset ini penulis menggunakan *tools* dari *OWASP ZAP* dengan hasil yang berlandaskan pada metode *OWASP* Top 10 dan *CWE*, yang dimana nantinya hasil dari pengujian akan menjadi perbandingan dari metode dan *default* penilaian dari 2 akses. Akses pertama akan dilakukan pengujian pada *user admin* dan akses kedua menggunakan akses *manager*.

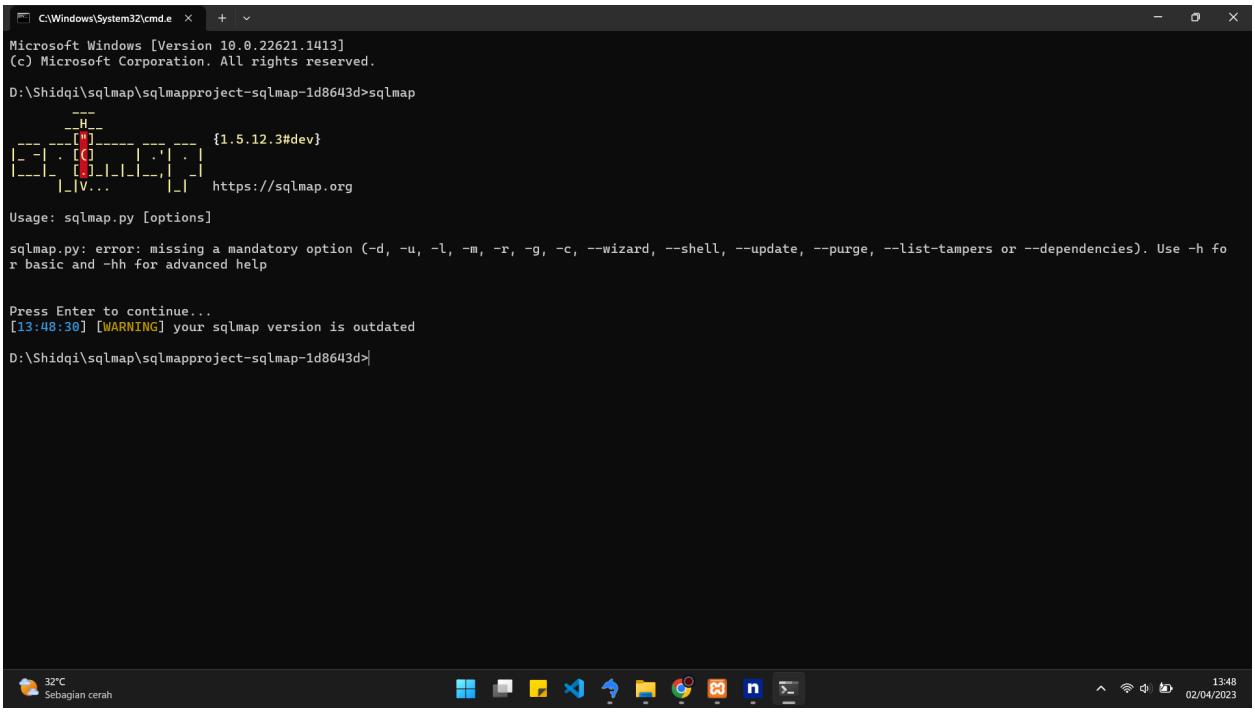


Gambar 7 OWASP ZAP

OWASP ZAP pertama menggunakan akan ditawarkan pada 2 pilihan, yang pertama adalah *automated scan* dan yang kedua adalah *manual scan*. Pada kedua bagian ini memiliki 2 perbedaan dalam memindai sebuah kerentanan *Website*.

4.1.3 *Manual testing*

Untuk menemukan sebuah kerentanan yang lainnya selain menggunakan tools dari *OWASP* yaitu *OWASP* ZAP, penelitian ini juga dilakukan sebuah *manual testing*. *Manual testing* adalah sebuah penetration testing secara terstruktur menggunakan metode khusus dari kerentanan yang dideklarasikan oleh *OWASP* top 10. Beberapa metode khusus yang digunakan untuk *manual testing* adalah mengambil SQL Injection dan Insecure deserilization



Gambar 8 SQL Injection

4.2 Tahapan

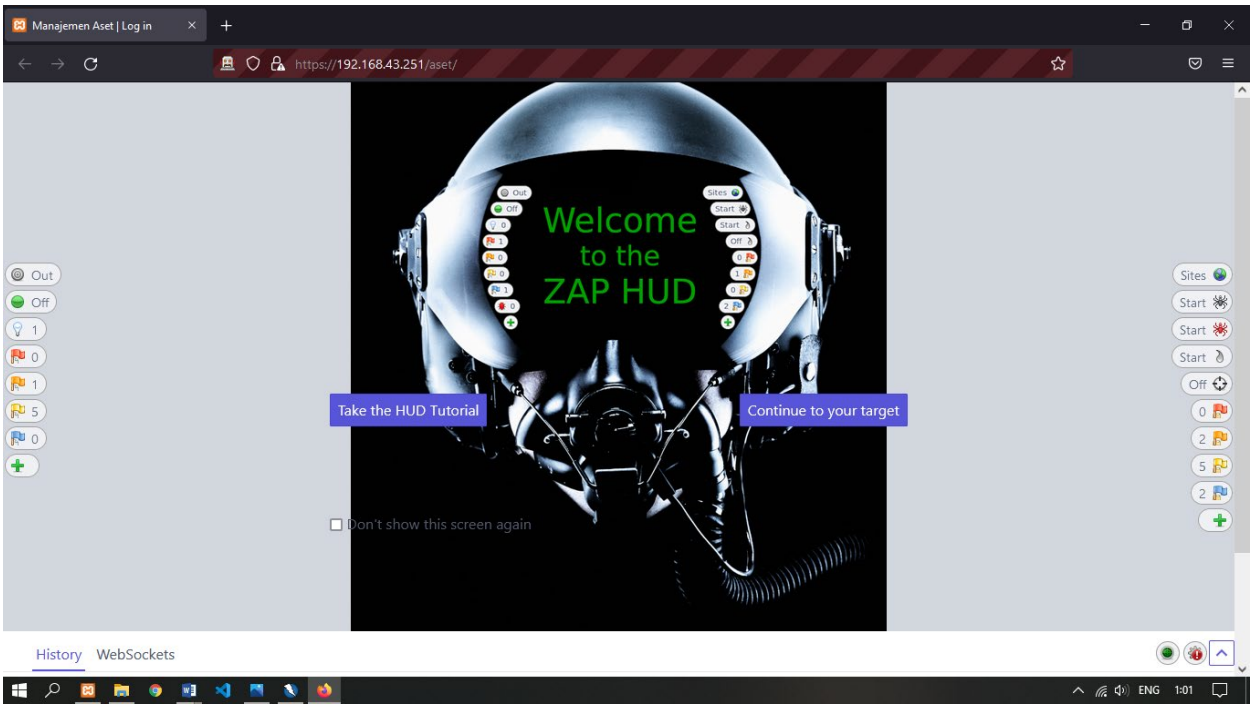
4.2.1 *Threat Identification*

Pada tahap ini dilakukan pencarian dan analisa pada *Website* manajemen aset yang akan di analisa, setelah menggunakan *Zenmap* maka akan ditemukan beberapa

kebutuhan seperti dijalakan dengan menggunakan apa dan menggunakan *port* berapa pada aplikasi *Website* manajemen aset. Maka selanjutnya adalah rancangan dari pengujian ini akan menggunakan *Scanning Website* dengan metode *Black Box* dan *manual testing*.

4.2.1.1 Scanning OWASP ZAP

1. *Manual scan* akan diberikan hak akses untuk melakukan pemindaian dari hasil pada apa yang sedang kita pilih atau akses pada *Website* yang tertampil. Banyak kondisi yang bisa dilakukan dengan menggunakan manual *Scanning* dikarenakan pada *OWASP ZAP* memberikan sebuah *HUD* yang tujuannya adalah untuk memberikan fitur seperti *scope*, *spider*, *scan* dan sebagainya untuk melakukan pemindaian pada lingkup yang dibutuhkan. Berikut merupakan hasil dari penetration testing yang menggunakan *manual Scanning* pada aplikasi manajemen aset.



Gambar 9 OWASP ZAP (Manual Scanning)

1. Pada pengujian *manual testing* yang pertama menggunakan SQL Map untuk menguji pada hasil dari *Scanning* SQL Injection pada *OWASP* ZAP. Disini penulis akan melakukan uji coba pada *Website* manajemen aset yang dijalankan pada localhost dan mengambil parameter id = 1, yang kemudian akan dinaikan tingkat pengujian menjadi level 3 dan dengan resiko menjadi level 5. Pada level disini SQL Map memberikan pengujian lebih pada sebuah *Website* dengan maksud menemukan celah atau resiko lebih mendalam agar ditemukannya celah jika pada default tidak ditemukan atau tidak berhasil.

Script
D:\Shidqgi\sqlmap\sqlmapproject-sqlmap-1d8643d>sqlmap D:\Shidqi\sqlmap\sqlmapproject-sqlmap-1d8643d>sqlmap.py —u https://aaea-103-144-175-204.ap.ngrok.io/index.php?id=1| --dbs

Table 5 Script Manual testing OWASP ZAP

BAB V
IMPLEMENTASI

5.1 Security Implementation

Security implementation adalah proses mengimplementasikan atau menerapkan langkah-langkah keamanan pada sistem atau aplikasi untuk melindungi informasi sensitif dan mencegah serangan atau ancaman keamanan. Hal ini melibatkan penggunaan teknologi dan praktik keamanan yang tepat untuk memastikan bahwa sistem dan data terlindungi dari akses yang tidak sah, manipulasi data, pencurian data, serta ancaman atau serangan lainnya. Langkah-langkah implementasi keamanan meliputi identifikasi kerentanan keamanan, pemilihan teknologi dan praktik keamanan yang tepat, penggunaan enkripsi, implementasi kebijakan akses, pemantauan keamanan, dan pelatihan pengguna. Implementasi keamanan yang efektif dapat membantu organisasi mengurangi risiko keamanan dan memastikan bahwa sistem dan data mereka terlindungi dengan baik.

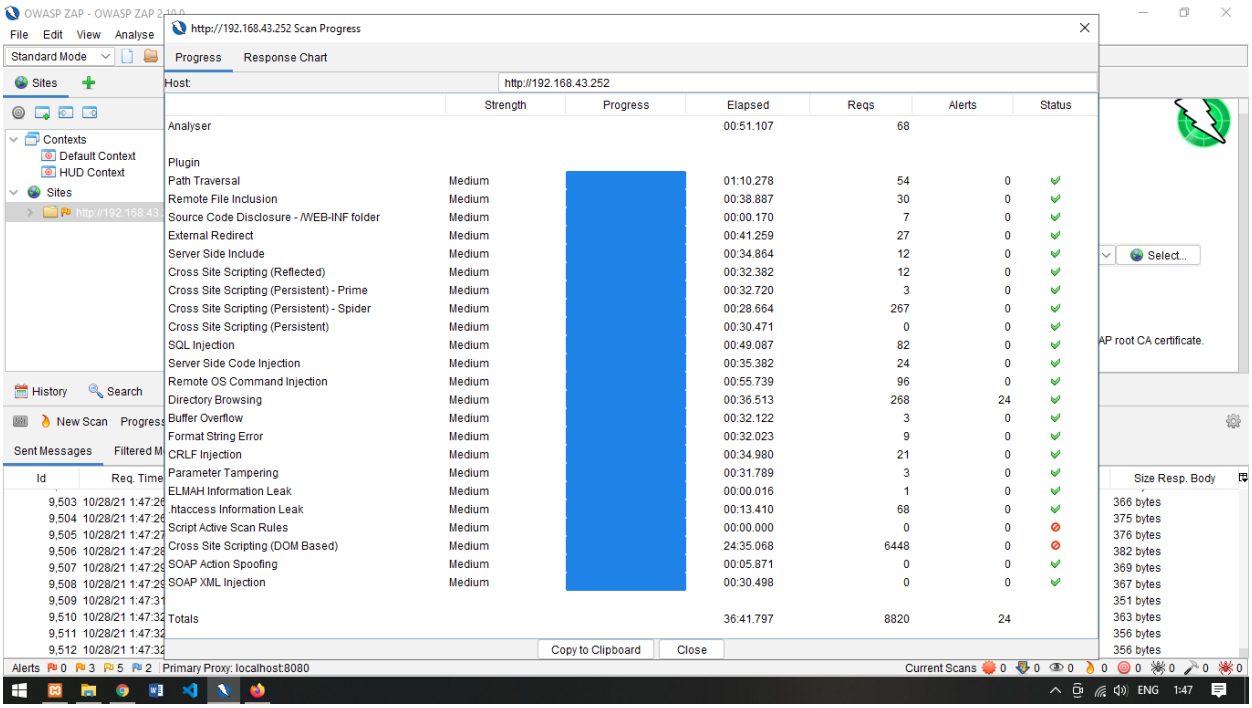
5.1.1 *Risk Estimation*

Pada tahap ini dilakukan pencarian dan analisa pada *Website* manajemen aset yang akan di analisa, setelah menggunakan *Zenmap* maka akan ditemukan beberapa kebutuhan seperti dijalakan dengan menggunakan apa dan menggunakan *port* berapa pada aplikasi *Website* manajemen aset. Maka selanjutnya adalah rancangan dari pengujian ini akan menggunakan *Scanning Website* dengan metode *Black Box* dan *manual testing* sebagai berikut :

1. *Scanning* User dengan *OWASP ZAP*

Pada tahap ini dilakukan pencarian dan analisa pada *Website* manajemen aset yang akan di analisa, dengan menggunakan *Zenmap* maka akan ditemukan beberapa kebutuhan seperti dijalakan dengan menggunakan apa dan menggunakan *port* berapa pada aplikasi *Website* manajemen aset.

1. Scanning user Admin



Gambar 10 Scanning Automatic OWASP ZAP

Pada fase ini dilakukan pemindaian pada *user admin. User admin* ini terdapat akses untuk :

1. *Create, Read, Update* dan *Delete* data aset
2. *Login username* dan *password*
3. *Logout*

Dari hasil pemindaian kerentanan keamanan *Website* untuk *user admin* pada manajemen aset maka dari beberapa fitur tersebut ditemukan hasil sebagai berikut:

Name	*Risk Level*	*Number of Instances*
absence of anti-csrf tokens laravel	*Low*	2
cookie no httponly flag laravel	*Low*	5

Table 6 Hasil Scanning Automatic OWASP ZAP User Admin

2. *Scanning user Manager*

Pada fase ini dilakukan pemindaian pada *user manager*. *User manager* ini terdapat akses untuk :

1. *View* data *user*
2. *Login username* dan *password*
3. *Logout*
4. *Download excel* untuk data user

Dari hasil pemindaian kerentanan keamanan *Website* untuk *user manager* pada manajemen aset maka dari beberapa fitur tersebut ditemukan hasil sebagai berikut :

Name	*Risk Level*	*Number of Instances*
absence of anti-csrf tokens laravel	*Low*	1
cookie no httponly flag laravel	*Low*	5

Table 7 Hasil Scanning Automatic OWASP ZAP User Manager

Dari tabel tersebut, terdapat dua nama risiko dan level risiko yang terkait dengan kerentanan pada aplikasi Laravel, yaitu "absence of anti-csrf tokens laravel" dan "cookie no httponly flag laravel", yang memiliki level risiko rendah (Low).

Kerentanan "absence of anti-csrf tokens laravel" merujuk pada kekurangan pada sistem keamanan yang tidak mempergunakan token CSRF (Cross-Site Request Forgery) pada formulir yang ada dalam aplikasi Laravel. Level risiko rendah berarti bahwa kerentanan ini dapat menyebabkan kerusakan yang minim pada sistem atau aplikasi, tetapi tetap perlu diperbaiki agar sistem dan data terlindungi dengan baik dari serangan CSRF.

Kerentanan "cookie no httponly flag laravel" merujuk pada pengaturan cookie pada aplikasi Laravel yang tidak memiliki flag "HttpOnly" yang dapat membatasi akses cookie dari skrip JavaScript. Level risiko rendah berarti bahwa kerentanan ini dapat memungkinkan serangan yang terbatas pada sistem atau aplikasi, tetapi tetap perlu diperbaiki agar sistem dan data terlindungi dengan baik dari serangan cookie-based.

2. *Manual testing*

1. Testing menggunakan SQL Map pada SQL Injection dengan menggunakan user id = 1

Script	User ID	Risk Level
D:\Shidqi\sqlmap\sqlmapproject-sqlmap-1d8643d>sqlmap.py -u https://aaea-103-144-175-204.ap.ngrok.io/index.php?id=1 --dbs	1 (Manager)	Warning

Table 8 Script untuk Implement SQL Injection user Manager

Pada pengujian ini dilakukan untuk mengetahui dan membuktikan pada kerentanan SQL Injection yang tidak ditemukan, SQL Map merupakan pengujian dengan menampilkan database atau mengambil data seperti tabel pada *Website* dengan hanya menuliskan URL target.

Dengan mengambil level pada tingkat 5 dan risk pada tingkat 3 membuat SQL Map akan mencari lebih dalam dengan metode yang lebih jauh, dengan begitu kerentanan akan ditemukan dengan cara yang sulit sekalipun. Pada penjelasan dan dokumentasi terhadap beberapa pengguna yang menyebutkan bahwa tingkat level menunjukkan SQL mampu dalam menemukan titik terjauh dan kelemahan pada database *Website*.

2. Testing menggunakan SQL Map pada SQL Injection dengan menggunakan user id = 2

Script	User ID	Risk Level
D:\Shidqi\sqlmap\sqlmapproject-sqlmap-1d8643d>sqlmap.py -u https://aaea-103-144-175-204.ap.ngrok.io/index.php?id=2 --dbs	2 (Admin)	Warning

Table 9 Script untuk Implement SQL Injection user Admin

5.2 Hasil Implementasi

1. *Absence of Anti-CSRF Tokens Laravel (CWE ID* 352 *)*

Name	*CWE ID*	*Risk Level*
absence of anti-csrf tokens laravel	352	Low

Table 10 Hasil Implementasi Scanning OWASP ZAP

Kerentanan yang terjadi ketika tidak ada anti-CSRF tokens pada framework Laravel adalah dimungkinkannya terjadinya serangan Cross-Site Request Forgery (CSRF). Pada serangan ini, seorang penyerang dapat memanfaatkan fakta bahwa pengguna telah login ke sebuah situs Web, untuk memaksa pengguna untuk melakukan tindakan yang tidak disengaja dan tidak diinginkan pada situs tersebut, misalnya mengubah password atau memasukkan data sensitif lainnya. Dengan adanya anti-CSRF tokens, maka aksi ini dapat dicegah karena akan memerlukan token yang hanya diketahui oleh pengguna yang melakukan aksi tersebut. Jadi, ketika tokens ini tidak ada, maka situs Web menjadi lebih rentan terhadap serangan CSRF dan penggunaannya menjadi lebih tidak aman.

2. *Cookie no Httponly Flag Laravel (CWE ID* 1004 *)*

Name	*CWE ID*	*Risk Level*
cookie no httponly flag laravel	1004	Low

Table 11 Hasil Implementasi Scanning OWASP ZAP

Kerentanan yang terjadi ketika cookie tidak memiliki HTTPOnly flag pada framework Laravel adalah dimungkinkannya terjadinya serangan Cross-Site Scripting (XSS). Pada serangan ini, seorang penyerang dapat memasukkan skrip berbahaya ke dalam halaman Web dan memperoleh akses ke informasi sensitif yang disimpan dalam cookie pengguna. Dengan adanya HTTPOnly flag pada cookie, maka cookie hanya dapat diakses melalui protokol HTTP dan tidak dapat diakses melalui JavaScript, sehingga dapat mencegah serangan XSS. Jadi, ketika flag ini tidak ada, maka cookie akan menjadi lebih rentan terhadap serangan XSS dan penggunaannya menjadi lebih tidak aman. Penting untuk memperhatikan hal ini dalam pengembangan aplikasi Web dengan framework Laravel dan mengatur HTTPOnly flag pada cookie sesuai dengan kebutuhan keamanan aplikasi.

1. *Manual testing* dengan SQL Map pada user manager

Script	***Result***	Page
D:\Shidqi\sqlmap\sqlmapproject-sqlmap-1d8643d>sq	[e2 [11/70] testing "Generic UNION query (random number) - 1 to 20 columns' [e2: [11/70] testing "Generic UNION query (NULL) - 21 to 46 columns'	Dashboard

lmap.py —u https://aaea-103-144-175-204.ap.ngrok.io/index.php?id=1 --dbs	[02 [11/70] testing "Generic UNION query (random number) - 21 to 46 columns' [02 [11170] testing "Generic UNION query (NULL) - 41 to 66 columns' [e2: [11/70] testing "Generic UNION query (random number) - 41 to 66 columns' [02 [11/70] testing "Generic UNION query (NULL) - 61 to 86 columns' [02 [11/70] testing "Generic UNION query (random number) - 61 to 86 columns' [e2: [11170] testing "Generic UNION query (NULL) - 81 to 166 columns' [02 [11170] testing "Generic UNION query (random number) - 81 to 166 columns' [02 [1170] checking if the injection point on Referer parameter 'Referer' is a false positive [e2: [WARNING] false positive or unexploitable injection point detected [02 [WARNING] parameter 'Referer' does not seem to be injectable [02 [11/70] testing if parameter 'Host' is dynamic	

	[e2: [WARNING] parameter ‘Host’ does not appear to be dynamic [02 [WARNING] heuristic (basic) test shows that parameter ‘Host’ might not be injectable [02 [11/70] testing for SOL injection on parameter ‘Host’ [e2: [11/70] testing "AND boolean-based blind - WHERE or HAVING clause [02 [11/70] testing "OR boolean-based blind - WHERE or HAVING clause’ [02 [WARNING] reflective value(s) found and filtering out [e2: [11/70] testing "OR boolean-based blind - WHERE or HAVING clause (NOT)' [02 [11/70] testing "AND boolean-based blind - WHERE or HAVING clause (subquery - comment)® [02 [11/70] testing "OR boolean-based blind - WHERE or HAVING clause (subquery - comment)’ [e2: [11/70] testing "AND boolean-based blind - WHERE or HAVING clause (comment)’	

	[02 [11/70] testing "OR boolean-based blind - WHERE or HAVING clause (Comment) [02 [11/70] testing "OR boolean-based blind - WHERE or HAVING clause (NOT - comment)® [e2: [11170] testing 'Boolean-based blind - Parameter replace (original value)' [02 [11170] testing "Boolean-based blind - Parameter replace (DUAL)' [02 [11/70] testing 'Boolean-based blind - Parameter replace (DUAL - original value)' [e2: [11170] testing "Boolean-based blind - Parameter replace (CASE)" [02 [11170] testing "Boolean-based blind - Parameter replace (CASE - original value)' [02 [11170] testing "HAVING boolean-based blind - WHERE, GROUP BY clause [e2: [11/70] testing "Generic inline queries' [02 [11/70] testing "Generic UNION query (NULL) - 1 to 16 columns' [02 [11170] testing "Generic UNION query (random number) - 1 to 10 columns'	

	[02:46:60] [WARNING] parameter 'Host' does not seem to be injectable [*] ending @ 62:46:00 /2022-01-01/	

Table 12 Hasil Implementasi Scanning SQL Injection

2. *Manual testing* dengan SQL Map pada user admin

Script	Result	Page
D:\Shidqi\sqlmap\sqlmapproject-sqlmap-1d8643d>sqlmap.py —u https://aaea-103-144-175-204.ap.ngrok.io/index.php?id=2 --dbs	[e2 [11/70] testing "Generic UNION query (random number) - 1 to 20 columns' [e2: [11/70] testing "Generic UNION query (NULL) - 21 to 46 columns' [02 [11/70] testing "Generic UNION query (random number) - 21 to 46 columns' [02 [11170] testing "Generic UNION query (NULL) - 41 to 66 columns' [e2: [11/70] testing "Generic UNION query (random number) - 41 to 66 columns'	Dashboard

	[02 [11/70] testing "Generic UNION query (NULL) - 61 to 86 columns’ [02 [11/70] testing "Generic UNION query (random number) - 61 to 86 columns’ [e2: [11170] testing "Generic UNION query (NULL) - 81 to 166 columns’ [02 [11170] testing "Generic UNION query (random number) - 81 to 166 columns’ [02 [1170] checking if the injection point on Referer parameter ‘Referer’ is a false positive [e2: [WARNING] false positive or unexploitable injection point detected [02 [WARNING] parameter ‘Referer’ does not seem to be injectable [02 [11/70] testing if parameter 'Host' is dynamic	

	[e2: [WARNING] parameter ‘Host’ does not appear to be dynamic [02 [WARNING] heuristic (basic) test shows that parameter ‘Host’ might not be injectable [02 [11/70] testing for SOL injection on parameter ‘Host’ [e2: [11/70] testing "AND boolean-based blind - WHERE or HAVING clause [02 [11/70] testing "OR boolean-based blind - WHERE or HAVING clause’ [02 [WARNING] reflective value(s) found and filtering out [e2: [11/70] testing "OR boolean-based blind - WHERE or HAVING clause (NOT)' [02 [11/70] testing "AND boolean-based blind - WHERE or HAVING clause (subquery - comment)® [02 [11/70] testing "OR boolean-based blind - WHERE or	

	HAVING clause (subquery - comment)’ [e2: [11/70] testing "AND boolean-based blind - WHERE or HAVING clause (comment)’ [02 [11/70] testing "OR boolean-based blind - WHERE or HAVING clause (Comment) [02 [11/70] testing "OR boolean-based blind - WHERE or HAVING clause (NOT - comment)® [e2: [11170] testing 'Boolean-based blind - Parameter replace (original value)' [02 [11170] testing "Boolean-based blind - Parameter replace (DUAL)' [02 [11/70] testing 'Boolean-based blind - Parameter replace (DUAL - original value)' [e2: [11170] testing "Boolean-based blind - Parameter replace (CASE)"	

	[02 [11170] testing "Boolean-based blind - Parameter replace (CASE - original value)' [02 [11170] testing "HAVING boolean-based blind - WHERE, GROUP BY clause [e2: [11/70] testing "Generic inline queries' [02 [11/70] testing "Generic UNION query (NULL) - 1 to 16 columns' [02 [11170] testing "Generic UNION query (random number) - 1 to 10 columns' [02:46:60] [WARNING] parameter 'Host' does not seem to be injectable [*] ending @ 62:46:00 /2022-01-01/	

Table 13 Hasil Implementasi Scanning SQL Injection

Pada hasil pengujian *manual testing* menggunakan SQL Map dikatakan pada warning menunjukkan bahwasannya *Website* manajemen aset tidak dapat dilakukan injeksi, bahkan dengan menambahkan level pencarian serta resiko pada pencarian ini pun masih dikatakan bahwa parameter id tidak dapat diinjeksi, kemudian pada user agent yang berperan untuk melakukan random injeksi pada database ini juga tidak dapat diinjeksi.

Beberapa alasan yang diketahui bahwa *Website* ini tidak dapat diinjeksi adalah

1. Laravel menyediakan proteksi pertahanan seperti csrf token yang dimiliki sehingga ketika ada serangan sseperti xsrf token yang dijalankan untuk memalsukan permintaan tidak valid akan ditolak.
2. Pada htaccess yang dimiliki oleh laravel membuat sebuah URL disembunyikan sehingga ketika akan melakukan injeksi pada URL tersebut tidak dapat dilakukan sehingga menyulitkan untuk melakukan Injection URL

1. *Scanning* dengan menggunakan XXS dengan menyusupkan script melalui user manager

Script	Result	Page
Localhost/manage/dashboard/dashboard.php?id=1 Dengan mencoba menyisipkan script <u>Dashboard</u>	Forbidden You don't have permission to access this resource.	Server at localhost Port 80

Table 14 Hasil Implementasi Scanning XXS

Pesan "Forbidden - You don't have permission to access this resource" dapat terjadi karena beberapa alasan, salah satunya adalah karena adanya serangan Cross-Site Scripting (XSS). Pada serangan ini, adanya pembatasan akses ke sumber daya atau halaman Web tertentu pada server. Hal ini dapat terjadi karena beberapa alasan, seperti kesalahan konfigurasi server, kesalahan hak akses pengguna, atau serangan keamanan seperti serangan Cross-Site Scripting (XSS). pengguna tidak dapat mengakses sumber daya atau halaman Web yang dimaksud dan menerima pesan error. Jika pesan error ini disebabkan oleh serangan keamanan seperti XSS, maka akibatnya dapat berdampak pada keamanan dan kerahasiaan data, bahkan dapat menyebabkan kerusakan sistem atau pencurian data sensitif.

Script yang diberikan adalah URL "localhost/manage-aset/dashboard.php?id=1" yang mengandung parameter "id" dengan nilai 1 dan tag HTML "u" yang menandakan bahwa teks

"Dashboard" akan diberi garis bawah. Script ini tidak terkait dengan serangan XSS (Cross-Site Scripting) secara langsung.

Namun, parameter "id" pada URL dapat menjadi titik masuk potensial untuk serangan XSS jika tidak divalidasi dengan benar atau di-filter dengan baik. Jika serangan XSS berhasil dilakukan pada parameter "id", maka serangkaian skrip JavaScript yang berbahaya dapat dimasukkan ke dalam halaman Web dan dieksekusi pada browser pengguna yang melihat halaman Web tersebut. Skrip JavaScript tersebut dapat memungkinkan serangan phishing, pencurian informasi, dan serangan malware, yang dapat membahayakan keamanan dan kerahasiaan data pengguna. Oleh karena itu, penting untuk selalu memastikan bahwa parameter input pada halaman Web yang digunakan oleh pengguna telah divalidasi dengan benar atau di-filter dengan baik, terutama jika parameter tersebut akan digunakan dalam operasi database atau memasukkan teks ke dalam halaman Web.

5.3 Evaluasi Sistem

Hasil	Testing *OWASP*	Level
all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more t ests	SQL Injection	Low
You don't have permission to access this resource.	Identification and Authentication Failures	Low

Table 15 Kesimpulan dari hasil Implementasi

1. SQL Injection

   Kesimpulan dari hasil pengujian SQL injection yang menyatakan "all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests" adalah sebagai berikut:

Hasil pengujian menunjukkan bahwa pada parameter-parameter yang telah diuji, tidak ditemukan adanya celah yang dapat dimanfaatkan untuk melakukan serangan SQL Injection. Ini menandakan bahwa aplikasi Web memiliki perlindungan yang cukup baik terhadap serangan jenis ini. Dalam situasi ini, peningkatan nilai untuk opsi '--level' atau '--risk' pada tool pengujian SQL Injection dapat membantu untuk melakukan pengujian lebih lanjut dengan tingkat risiko yang lebih tinggi. Hal ini dapat memungkinkan tool untuk mencoba teknik serangan yang lebih kompleks atau mengeksplorasi celah yang lebih dalam. Meskipun saat ini tidak ditemukan celah yang dapat dimanfaatkan, ini tidak menjamin bahwa aplikasi Web sepenuhnya bebas dari kerentanan SQL Injection. Ada kemungkinan bahwa celah dapat terletak pada parameter lain yang belum diuji atau memerlukan teknik serangan yang lebih canggih. Penting untuk memahami bahwa hasil pengujian ini hanya mencakup parameter-parameter yang telah diuji. Oleh karena itu, perlu dilakukan pengujian yang komprehensif untuk melihat secara menyeluruh keberadaan kerentanan SQL Injection pada aplikasi Web. Hasil yang menunjukkan tidak adanya celah yang dapat dimanfaatkan saat ini adalah indikasi positif tentang keamanan aplikasi Web. Namun, tidak boleh diabaikan atau dianggap sebagai jaminan keamanan mutlak. Upaya terus-menerus untuk meningkatkan keamanan dan melakukan pengujian berkala tetap penting. Pengujian keamanan lebih lanjut dengan metode yang lebih komprehensif dan menggunakan kombinasi opsi risiko dan level yang lebih tinggi dapat membantu mendeteksi celah keamanan yang belum terlihat sebelumnya dan meningkatkan keamanan aplikasi Web secara keseluruhan.

2. Identification and Authentication Failures

   Kesimpulan dari hasil pengujian yang menunjukkan "You don't have permission to access this resource" dalam konteks kegagalan identifikasi dan otentikasi (Identification and Authentication Failures) adalah sebagai berikut:

   Hasil pengujian menunjukkan bahwa terjadi kegagalan dalam proses identifikasi dan otentikasi pengguna yang mengakibatkan akses ditolak ke sumber daya yang diuji. Hal ini menunjukkan adanya masalah pada mekanisme identifikasi dan otentikasi yang digunakan

oleh aplikasi Web. Pesan "You don't have permission to access this resource" menandakan bahwa pengguna yang melakukan pengujian tidak memiliki hak akses yang diperlukan untuk mengakses sumber daya tersebut. Ini adalah respons yang diharapkan jika pengguna tidak dapat melewati proses identifikasi dan otentikasi yang diperlukan.

Kegagalan identifikasi dan otentikasi adalah kerentanan serius yang dapat memberikan akses tidak sah atau tidak terotorisasi ke sumber daya sensitif. Hal ini dapat membuka celah bagi penyerang untuk mencuri informasi, memanipulasi data, atau melakukan tindakan yang tidak diinginkan dalam aplikasi Web. Kegagalan identifikasi dan otentikasi dapat disebabkan oleh berbagai faktor, seperti penggunaan kata sandi yang lemah, kerentanan pada implementasi mekanisme identifikasi dan otentikasi, atau kegagalan dalam mengelola izin dan hak akses pengguna. Identifikasi dan otentikasi yang kuat merupakan komponen penting dalam menjaga keamanan aplikasi Web. Diperlukan praktik terbaik, seperti penggunaan kata sandi yang kompleks, mekanisme otentikasi dua faktor, dan pemantauan yang ketat terhadap hak akses pengguna, untuk melindungi sumber daya yang sensitif dan mencegah akses yang tidak sah. Meskipun hasil pengujian menunjukkan kegagalan dalam identifikasi dan otentikasi, ini bukan jaminan bahwa aplikasi Web sepenuhnya rentan. Pengujian keamanan yang lebih mendalam dan komprehensif diperlukan untuk mengevaluasi seluruh aspek keamanan aplikasi Web dan memastikan tidak ada celah yang dapat dimanfaatkan oleh penyerang.

BAB VI
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada kesimpulan dari penelitian kali ini membuktikan bahwa *Website* manajemen aset yang telah dilakukan *Scanning* vulnerability ditemukan beberapa kerentanan. Untuk menjaga keamanan website secara efektif, perlu dilakukan berbagai tindakan perbaikan. Hal ini meliputi perlindungan terhadap serangan *SQL Injection* dengan validasi input dan pengujian keamanan, peningkatan identifikasi dan otentikasi pengguna melalui kebijakan kata sandi kuat dan otentikasi dua faktor, perlindungan terhadap serangan *XSS* dan *CSRF* dengan sanitasi output dan penggunaan token *CSRF*, pemantauan dan pemeliharaan keamanan rutin.

Selain itu, melakukan audit keamanan secara teratur juga menjadi bagian penting dari strategi keamanan. Kesimpulannya, hasil pengujian menunjukkan bahwa tidak ditemukan parameter-parameter yang dapat dimasuki (*injectable*) tetapi meskipun hasil ini menunjukkan bahwa website telah melindungi dengan baik dari serangan, sangat penting untuk tetap waspada dan melakukan pengujian keamanan yang lebih mendalam untuk memastikan tidak ada celah yang dapat dimanfaatkan oleh penyerang.

5.2 Saran

Saran dari penulis, untuk meningkatkan keamanan *website*, beberapa saran perbaikan dapat dilakukan. Pertama, perlindungan dari serangan *SQL Injection* dapat ditingkatkan melalui validasi input pengguna, penggunaan parameter binding, dan pengujian keamanan secara berkala. Kedua, identifikasi dan otentikasi perlu diperkuat dengan kebijakan kata sandi kuat dan penggunaan otentikasi dua faktor. Ketiga, perlindungan terhadap *Cross-Site Scripting (XSS)* dapat dilakukan dengan sanitasi output dan implementasi kebijakan keamanan konten. Keempat, perlindungan terhadap *Cross-Site Request Forgery (CSRF)* dapat ditingkatkan dengan penggunaan token *CSRF*. Kelima, pemantauan dan pemeliharaan keamanan yang rutin penting untuk mengawasi aktivitas *website* dan melakukan pembaruan keamanan. Terakhir, pelatihan dan kesadaran keamanan bagi pengembang dan pengguna dapat membantu meningkatkan pemahaman dan kesadaran akan praktik keamanan yang perlu diterapkan. Dengan menerapkan saran-saran ini, keamanan *website* dapat ditingkatkan dan risiko serangan dapat dikurangi.

DAFTAR PUSTAKA

[1] Supanto, "Perkembangan Kejahatan Teknologi Informasi (Cyber Crime) Dan Antisipasinya Dengan Penal Policy," *Yust. J. Huk.*, vol. 5, no. 1, 2016, doi: 10.20961/yustisia.v5i1.8718.

[2] T. Dirgahayu, Y. Prayudi, and A. Fajaryanto, "Penerapan Metode ISSAF dan *OWASP* versi 4 Untuk Uji Kerentanan Web Server," *J. Ilm. NERO*, vol. 1, no. 3, pp. 190–197, 2015, [Online]. Available: http://nero.trunojoyo.ac.id/index.php/nero/article/download/29/27.

[3] H. Ardiyanti, "Cyber-Security Dan Tantangan Pengembangannya Di Indonesia," pp. 95–110, 1986.

[4] M. Yunus, "Analisis Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi Security Tools Project Berdasarkan Framework *OWASP* Versi 4," *J. Ilm. Inform. Komput.*, vol. 24, no. 1, pp. 37–48, 2019, doi: 10.35760/ik.2019.v24i1.1988.

[5] P. McDaniel and A. D. Rubin, *Web security*, vol. 48, no. 5. 2005.

[6] I. G. Handika and A. Purbasari, "Pemanfaatan Framework Laravel Dalam Pembangunan Aplikasi E-Travel Berbasis *Website*," *Konf. Nas. Sist. Inf. STMIK Atma Luhur Pangkalpinang*, pp. 1329–1334, 2018.

[7] G. Guntoro, L. Costaner, and M. Musfawati, "Analisis Keamanan Web Server Open Journal System (Ojs) Menggunakan Metode Issaf Dan *OWASP* (Studi Kasus Ojs Universitas Lancang Kuning)," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 5, no. 1, p. 45, 2020, doi: 10.29100/jipi.v5i1.1565.

[8] B. P. Zen, R. A. G. Gultom, and A. H. S. Reksoprodjo, "Analisis Security Assessment Menggunakan Metode Penetration Testing dalam Menjaga Kapabilitas Keamanan Teknologi Informasi Pertahanan Negara," *J. Teknol. Penginderaan*, vol. 2, no. 1, pp. 105–122, 2020.

[9] K. Joesyiana, "Penerapan Metode Pembelajaran Observasi Lapangan Pada Mata Kuliah Manajemen Operasional," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.

[10] Sucuri, *The Top 10 OWASP 2019*. Sucuri Corporation, 2019.

LAMPIRAN

Berikut adalah lampiran untuk hasil script testing lengkap untuk SQL Injection :

Script	Hasil
D:\Shidqi\sqlmap\sqlmapproject-sqlmap-1d8643d>sqlmap.py -u https://aaea-103-144-175-204.ap.ngrok.io/index.php?id=1 --dbs	C:\Windows\System32\cmd.e X X [14:02:03] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/phpmyadmin/index.php'. You can try to rerun by providing a valid value for option '--csrf-url', skipping to the next target [9/15] URL: GET https://localhost/phpmyadmin/js/messages.php?l=en&v=5.1.0&lang=en do you want to test this URL? [Y/n/q] > y [14:02:03] [INFO] testing URL 'https://localhost/phpmyadmin/js/messages.php?l=en&v=5.1.0&lang=en' [14:02:04] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/phpmyadmin/js/messages.php'. You can try to rerun by providing a valid value for option '-

	-csrf-url', skipping to the next target [10/15] URL: GET https://localhost/phpmyadmin/index.php?lang=en do you want to test this URL? [Y/n/q] > y [14:02:05] [INFO] testing URL 'https://localhost/phpmyadmin/index.php?lang=en' [14:02:06] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/phpmyadmin/index.php'. You can try to rerun by providing a valid value for option '--csrf-url', skipping to the next target [11/15] URL: GET https://localhost/phpmyadmin/index.php?route=/&lang=en do you want to test this URL? [Y/n/q] > y [14:02:07] [INFO] testing URL 'https://localhost/phpmyadmin/index.php?route=/&lang=en' [14:02:08] [ERROR] anti-CSRF token '_token' can't be found at

	'https://localhost:443/phpmyadmin/index.php'. You can try to rerun by providing a valid value for option '--csrf-url', skipping to the next target [12/15] URL: GET https://localhost/phpmyadmin/url.php?url=https://mariadb.com/kb/en/documentation/ do you want to test this URL? [Y/n/q] > y [14:02:11] [INFO] testing URL 'https://localhost/phpmyadmin/url.php?url=https://mariadb.com/kb/en/documentation/' got a 302 redirect to 'https://localhost:443/phpmyadmin/'. Do you want to follow? [Y/n] y [14:02:14] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/phpmyadmin/url.php'. You can try to rerun by providing a valid value fo r option '--csrf-url', skipping to the next target [13/15] URL: GET https://localhost/phpmyadmin/index.p

	hp?route=/server/databases&server=1&lang=en do you want to test this URL? [Y/n/q] > y [14:02:15] [INFO] testing URL 'https://localhost/phpmyadmin/index.php?route=/server/databases&server=1&lang=en' [14:02:16] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/phpmyadmin/index.php'. You can try to rerun by providing a valid value for option '--csrf-url', skipping to the next target [14/15] URL: [14:01:46] [WARNING] HTTP error codes detected during run: 405 (Method Not Allowed) - 1 times [5/15] Form: GET https://localhost/phpmyadmin/index.php?db=&table=&lang=en&token=3857393a4324304048302a2e517f2146&lang=sq do you want to test this form? [Y/n/q] > y Edit GET data [default:

	db=&table=&lang=en&token=3857393a4324304048302a2e517f2146&lang=sq]: do you want to fill blank fields with random values? [Y/n] y [14:01:50] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/phpmyadmin/index.php'. You can try to rerun by providing a valid value for option '--csrf-url', skipping to the next target [6/15] Form: POST https://localhost/phpmyadmin/index.php?route=/&server=1&lang=en POST data: tab_hash=&check_page_refresh=&lang=en&token=3857393a4324304048302a2e517f2146&submit_save=Navi&ShowDatabasesNavigationAsTree=on&NavigationLinkWith O&Navigation TreeDisplayItemFilterMinimum=30&NumRecent Tables=10&NumFavoriteTables=10&NavigationWidth=240&MaxNavigatio

	nItems=50&NavigationTreeEnableGrouping=0 n&NavigationTreeEnableExpansion=on&Navigation TreeShowTables=on&NavigationTreeShowViews=on&Navigation TreeShowFunctions=on&Navigation TreeShowProcedures=on&Navigation TreeShowEvents=on&Navigation TreeAutoexpandSingleDb=on&NavigationDisplayServers=on&DisplayServersList=on&NavigationTreeDisplayDbFilterMinimum=30&Navigation TreeDbSeparator=_&Navigation TreeDefaultTabTable=structure&Navigation TreeDefaultTabTable2=&Navigation TreeTableSeparator=__&Navigation TreeTableLevel=1 do you want to test this form? [Y/n/q] MainPanel=on&NavigationDisplayLogo=on&NavigationLogoLink=index.php&NavigationLogoLinkWindow=main&NavigationTreePointerEnable=on&FirstLevelNavigationItems=10 > y

	Edit POST data [default: tab_hash=&check_page_refresh=&lang=en&token=3857393a4324304048302a2e517f2146&submit_save=Navi&ShowDatabasesNavigationAsTree=on&Navi gationLinkWithMainPanel=on&NavigationDisplayLogo=on&NavigationLogoLink-index.php&NavigationLogoLinkWindow=main&Navigation TreePointerEnable=on&FirstLevelNavi gationItems=100&NavigationTreeDisplayItemFilterMinimum=30&NumRecent Tables=10&NumFavoriteTables=10&NavigationWidth=240&MaxNavigationItems=50&NavigationTreeEnableGrouping=on&NavigationTreeEnableExpansion=on&NavigationTreeShowTables=on&NavigationTreeShowViews=on&Navigation TreeShowFunctions=on&Navigation TreeShowProcedures=on&NavigationTreeShowEvents=on&Navigation

	TreeAutoexpandSingleDb=on&NavigationDisplayServers=on&DisplayServersList=on&Navigation TreeDisplayDbFilterMinimum=30&Navigation TreeDbSeparator=_&Navigation TreeDefaultTabTable=structure&Navigation TreeDefaultTabTable2=&Navigation TreeTable Separator=__&NavigationTreeTableLevel=1] (Warning: blank fields detected): do you want to fill blank fields with random values? [Y/n] y [14:01:55] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/phpmyadmin/index.php'. You can try to rerun by providing a valid value for option '--csrf-url', skipping to the next target [7/15] Form: POST https://localhost/phpmyadmin/index.php?route=/collation-connection&lang=en POST data: lang=en&token=3857393a43243040

	48302a2e517f2146&collation_connection= do you want to test this form? [Y/n/q] > y Edit POST data [default: lang=en&token=3857393a4324304048302a2e517f2146&collation_connection=] (Warning: blank fields detected): do you want to fill blank fields with random values? [Y/n] y [14:01:33] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)' [14:01:33] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' [14:01:34] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind' [14:01:34] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)' [14:01:34] [INFO] testing 'Oracle AND time-based blind' [14:01:34] [INFO] testing 'Generic UNION query (NULL) - 1 to 10

	columns' [14:01:34] [WARNING] GET parameter 'search' does not seem to be injectable [14:01:34] [ERROR] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more t ests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper-space2comment') and/or switch '--random-agent', skipping to the next target [4/15] Form: POST https://localhost/manage-aset/register/store POST data: _token=bGmljrGVDShvossIPvRbH6lkWt7NCne9FHceeFRU&name=&email=&password= do you want to test this form? [Y/n/q] > y Edit POST data [default: _token=bGmljrGVDShvossIPvRbH6lkWt7NCne9FHceeFRU&name=&email=&password=] (Warning: blank

	fields detected): do you want to fill blank fields with random values? [Y/n] y POST parameter '_token' appears to hold anti-CSRF token. Do you want sqlmap to automatically update it in further requests? [y/N] y [14:01:46] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/manage-aset/register/store'. You can try to rerun by providing a valid value for option '--csrf-url', skipping to the next target [14:01:46] [WARNING] HTTP error codes detected during run: 405 (Method Not Allowed) - 1 times [5/15] Form: GET https://localhost/phpmyadmin/index.php?db=&table=&lang=en&token=3857393a4324304048302a2e517f2146&lang=sq do you want to test this form? [Y/n/q] > y Edit GET data [default: db=&table=&lang=en&token=385739

	3a4324304048302a2e517f2146&lang=sq]: do you want to fill blank fields with random values? [Y/n] y [14:01:50] [ERROR] anti-CSRF token '_token' can't be found at 'https://localhost:443/phpmyadmin/index.php'. You can try to rerun by providing a valid value for option '--csrf-url', skipping to the next target [6/15] Form: POST https://localhost/phpmyadmin/index.php?route=/&server=1&lang=en POST data: tab_hash=&check_page_refresh=&lang=en&token=3857393a4324304048302a2e517f2146&submit_save=Navi&ShowDatabases NavigationAsTree=on&NavigationLinkWith MainPanel=on&NavigationDisplayLogo=on&NavigationLogoLink-index.php&NavigationLogoLinkWindow=main&NavigationTreePointerEnable=on&FirstLevelNavigationItems=10

	O&NavigationTreeDisplayItemFilterMinimum=30&NumRecent Tables=10&NumFavoriteTables=10&NavigationWidth=240&MaxNavigation Items=50&Navigation TreeEnableGrouping=0 n&NavigationTreeEnableExpansion=on&Navigation TreeShowTables=on&Navigation TreeShowViews=on&Navigation TreeShowFunctions=on&Navigation TreeShowProcedures=on&NavationTreeDbSeparator=_&Navigation TreeDefaultTabTable=structure&Navigation TreeDefaultTabTable2=&Navigation TreeTable Separator=__&NavigationTreeTable Level=1 igationTreeShowEvents=on&Navigation TreeAutoexpandSingleDb=on&NavigationDisplayServers=on&DisplayServersList=on&Navigation TreeDisplayDbFilterMinimum=30&Navig

Table 16 Lampiran Testing Manual