BAB III

METODOLOGI PENELITIAN

3.1 Ruang Lingkup Penelitian

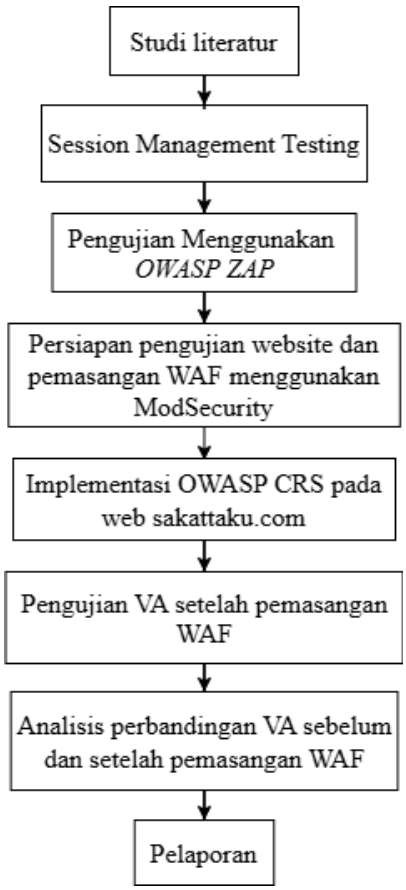
Metode yang dipilih peneliti adalah *VA* dan *WAF* untuk mengamankan *website* dari serangan siber. *Website* yang akan diteliti yaitu *website* Sakattaku yang dipegang oleh YSIC.

Gambar 3. 1 Tempat Studi Kasus Penelitian

3.2 Tahapan Penelitian

Gambar 3.2 menggambarkan pengujian keamanan web dengan *WAF* dan alat keamanan. Proses dimulai dengan *studi literatur,* dilanjutkan dengan perencanaan objek uji "Web Sakattaku". *WAF* diimplementasikan menggunakan ModSecurity, dan penilaian kerentanan dilakukan dengan *OWASP ZAP*. Kemudian, dilakukan pengujian penetrasi dengan serangan *XSS, SQL Injection*, dan *CSRF.* Simulasi serangan menggunakan *OWASP ZAP* menguji efektivitas *WAF*, dan hasil

pengujian dianalisis untuk menilai kinerja sistem. Proses diakhiri dengan analisis hasil untuk mengevaluasi efektivitas perlindungan *WAF.*



Gambar 3. 2 Alur Penelitian

3.2.1 *Studi Literatur*

Sebelum memulai pengujian, melakukan *studi literatur* untuk mengumpulkan informasi terkait dengan topik penelitian*, studi literatur* ini bertujuan untuk memahami konsep-konsep dasar keamanan *website*, jenis-jenis serangan yang umum terjadi, dan alat-alat yang dapat digunakan untuk melakukan pengujian. *Studi literatur* ini bisa didapat dari segala sumber seperti buku, internet, atau bahkan jurnal terkait dengan meninjau penelitian yang dilakukan sebelumnya

sehingga dapat dijadikan referensi serta melihat perbedaan antara penelitian yang sudah dilakukan sebelumnya dan yang akan dilakukan pada penelitian kali ini.

3.2.2 *Session Management Testing*

Pengujian ini dilakukan untuk mengidentifikasi kelemahan dalam pengelolaan sesi pengguna oleh *website*, dengan menggunakan *OWASP ZAP* dapat mengevaluasi apakah *website* rentan terhadap serangan *seperti SQL Injection, XSS, CSRF*, serta mengecek kelemahan autentikasi yang terjadi saat pengguna melakukan *login* atau aktivitas lainnya di *web* sakatta.

3.2.3 Pengujian Menggunakan *OWASP ZAP*

Tahap ini merupakan proses *VA* sebelum *WAF*. *Web* Sakattaku.com di-scan menggunakan *OWASP ZAP*, sebuah alat otomatis yang mengidentifikasi kerentanan dalam aplikasi web. Pengujian dilakukan dengan simulasi serangan seperti *XSS, SQL Injection,* dan *CSRF.* Hasil pengujian dikelompokkan berdasarkan tingkat risiko seperti *high, medium, low,* dan *informational.*

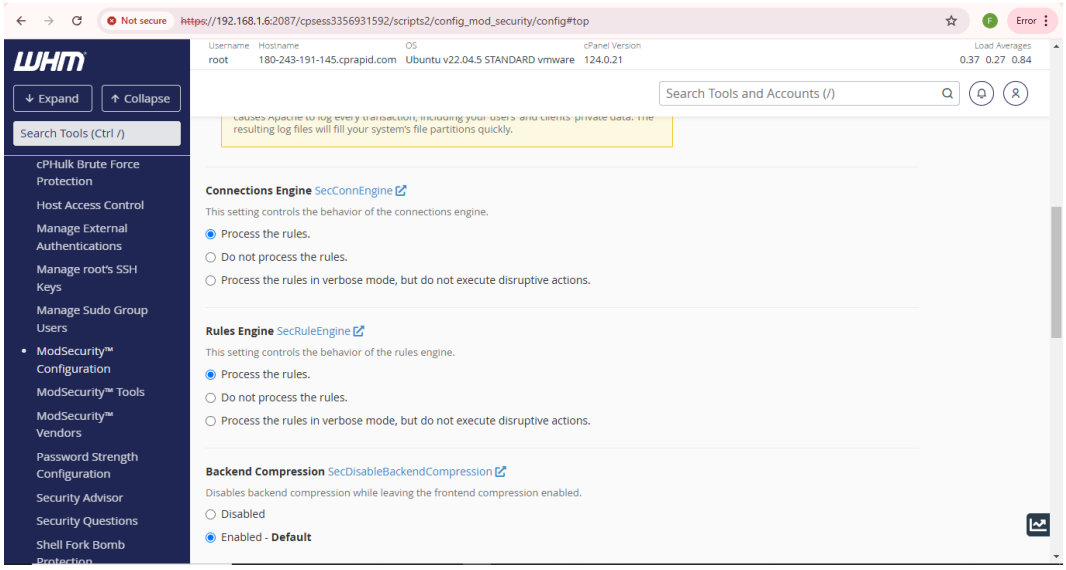
3.2.4 Persiapan Pengujian *Website* dan Pemasangan *WAF* Menggunakan *ModSecurity*

Tahap ini menyiapkan salinan virtual dari *web* Sakattaku.com di *server lokal* guna menjaga stabilitas *website* utama. Selanjutnya, dilakukan instalasi dan konfigurasi *WHM*, yang merupakan antarmuka administratif untuk *server.* Melalui *WHM, ModSecurity* sebagai komponen *WAF* diaktifkan dan disiapkan untuk melindungi *website.*

3.2.5 Implementasi *OWASP CRS* pada *Web* Sakattaku.com

Setelah virtualisasi w*eb* sakattaku.com dilakukan, tahap selanjutnya adalah pemasangan *WAF*. Aplikasi *WAF* yang digunakan pada penelitian ini adalah *ModSecurity*, dimana aplikasi tersebut merupakan aplikasi *WAF* yang bersifat *open source* dan dapat mengimplementasikan *rules* yang dapat memfilter lalu lintas paket antara *web* dan *client.*

Dikarenakan pada penelitian ini menggunakan *WHM* yang telah menyediakan layanan *ModSecurity*, maka aplikasi *ModSecurity* akan secara otomatis terinstall pada saat *WHM* dipasang pada *server*. Akan tetapi aplikasi *ModSecurity* perlu diaktifkan supaya dapat bekerja memfilter lalu lintas pada *web*. proses ini dijelaskan melalui Gambar 3.3



Gambar 3. 3 *ModSecurity* pada *WHM*

Setelah mengaktifkan ModSecurity, tahap selanjutnya memasang *CRS. CRS* dapat diintegrasikan pada *ModSecurity* ke dalam *server* melalui *WHM*. CRS menyediakan aturan keamanan untuk mendeteksi dan mencegah serangan seperti

SQL injection, *XSS*, dan ancaman berbasis *web* lainnya. Sementara *WHM* membantu mengelola *ModSecurity* dengan mengatur aturan dan memperbarui *CRS* secara teratur, *cPanel* memungkinkan pengguna mengelola situs *web* mereka dengan mudah tanpa mengorbankan keamanan. Selain itu, administrator dapat mengidentifikasi serangan potensial dan *false positives* melalui fitur log dan pemantauan yang tersedia. Administrator juga dapat menyesuaikan aturan sesuai dengan kebutuhan aplikasi *web* Sakattaku. Dengan cara ini, *OWASP CRS* tidak hanya melindungi situs *web* dari serangan berbasis aplikasi tetapi juga memastikan bahwa *web* aman, stabil, dan beroperasi dengan baik.

CRS mencakup beberapa kategori aturan yang berfokus pada mendeteksi dan mencegah ancaman yang umum terjadi pada aplikasi *web*. *CRS* menyediakan berbagai aturan untuk melindungi aplikasi *web* dari serangan umum yang sering digunakan oleh peretas.

3.2.6 Pengujian *VA* Setelah Pemasangan *WAF*

Pada tahap ini, pengujian keamanan diulangi terhadap *web* Sakattaku yang sudah dilindungi oleh *WAF. VA* dilakukan kembali menggunakan *OWASP ZAP* untuk mengetahui apakah masih ada kerentanan yang berhasil lolos meskipun *WAF* sudah diterapkan, jika ditemukan celah baru maka dilakukan penyesuaian atau penambahan *rules* khusus di *ModSecurity.*

3.2.7 Analisis Perbandingan *VA* Sebelum dan Setelah Pemasangan *WAF*

Menganalisis hasil dari pengujian yang telah dilakukan untuk mengevaluasi tingkat kerentanan suatu *website* terhadap ancaman keamanan. Proses analisis ini mencakup identifikasi celah keamanan yang dapat dieksploitasi oleh penyerang.

Selain itu, mencatat setiap jenis serangan yang berhasil menembus sistem keamanan, baik dari segi metode yang digunakan maupun kelemahan spesifik yang menjadi target serangan, dengan pendekatan ini dapat memberikan gambaran yang lebih jelas mengenai seberapa efektif sistem keamanan yang diterapkan pada *website* tersebut.

Lebih lanjut, analisis ini juga bertujuan untuk menilai dampak dari serangan yang dilakukan terhadap website yang diuji. Dampak tersebut dapat mencakup berbagai aspek, seperti penurunan performa sistem, potensi kebocoran data, hingga kemungkinan pengambilalihan kendali oleh pihak yang tidak berwenang, dengan memahami konsekuensi dari serangan yang terjadi dapat merumuskan rekomendasi yang lebih akurat dalam upaya meningkatkan keamanan *website*.

3.2.8 Pelaporan

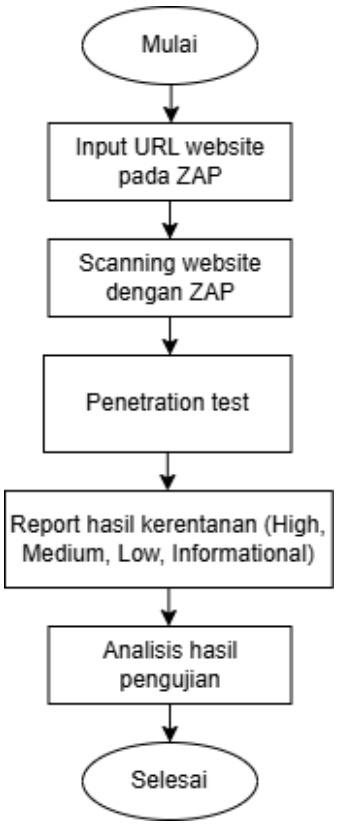
Tahap akhir adalah menyusun hasil penelitian ke dalam bentuk laporan. Laporan mencakup latar belakang, metodologi, hasil pengujian, analisis, serta kesimpulan dan saran. Peneliti juga memberikan rekomendasi konfigurasi dan pengamanan lanjutan berdasarkan hasil analisis untuk digunakan oleh pengelola website di masa mendatang.

3.1 Tahapan Pengujian

a. Alur pengujian *VA* sebelum *WAF*

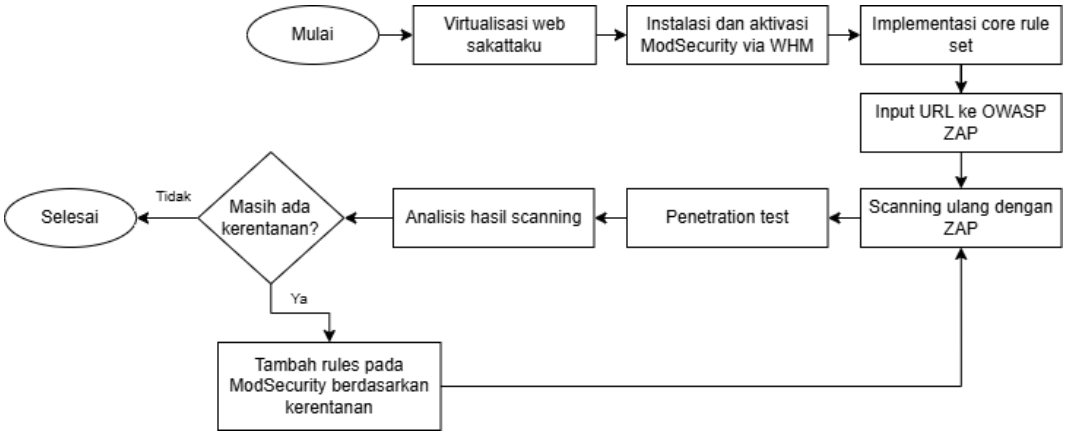
Gambar 3.4 menggambarkan tahapan proses pengujian *VA* pada website sakattaku.com sebelum diterapkannya *WAF*. Proses dimulai dengan tahap awal yaitu memasukkan *URL* website ke dalam aplikasi *OWASP ZAP. OWASAP ZAP* digunakan sebagai alat pemindai kemanan yang secara otomratis mengidentifikasi

celah keamanan pada aplikasi web. Setelah *URL* dimasukkan, dilakuakn proses scanning oleh *OWASP ZAP* untuk menelusuri semua struktur, halaman, dan parameter yang tersedia pada *web*. Selanjutnya*, OWASP ZAP* melakukan simulasi serangan terhadap *website*, seperti *SQL injection, XSS, dan CSRF*. Simulasi ini bertujuan untuk mengevaluasi seberapa rentan situs terhadap serangan yang terjadi di dunia nyata. Setelah proses simulasi selesai, *ZAP* akan menampilkan hasil dalam bentuk *alerts* yang mengelompokkan tingkat kerentanan ke dalam beberapa kategori, yaitu *high, medium, low*, dan *informational.* Tahap akhir yaitu melakukan analisis terhadap hasil pengujian, pada tahap ini meninjau jenis-jenis kerentanan yang ditemukan, menilai dampaknya terhadap keamanan web, serta mendokumentasikan hasil temuan sebagai dasar evaluasi.



Gambar 3. 4 *Flowchart* pengujian *VA* sebelum *WAF*

b. Alur pengujian *VA* setelah *WAF*



Gambar 3. 5 *Flowchart* pengujian *VA* setelah *WAF*

Gambar 3.5 merupakan *flowchart* yang menjelaskan langkah-langkah pengujian keamanan web dan implementasi *WAF* menggunakan *OWASP ZAP* dan *ModSecurity.* Proses dimulai dengan persiapan komponen, seperti *OWASP ZAP, web server*, dan salinan situs sakattaku.com. Salinan situs dikonfigurasi pada *virtual server* dan diuji aksebilitasnya. Setelah itu, *WAF* diimplementasikan dengan menginstal *ModSecurity* dan *CRS* melalui *WHM*. Selanjutnya, dilakukan *VA* menggunakan *OWASP* ZAP untuk memindai kerentanan dan menjalankan simulasi serangan. Hasil pengujian dianalisis dan didokumentasikan. Jika tidak ada kerentanan fatal, proses selesai. Namun, jika masih ada kerentanan, aturan tambahan pada *ModSecurity* diterapkan, dan proses diulang hingga semua kerentanan terselesaikan.

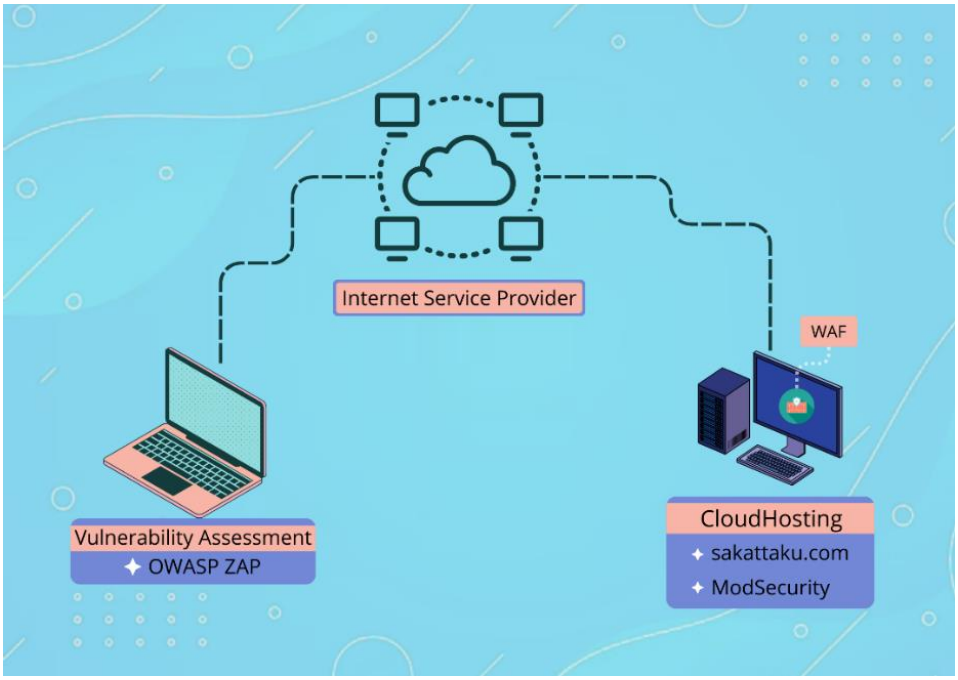
Pemasangan *WHM* dilakukan untuk memudahkan memanajemen *server* dan layanan-layanan yang dibutuhkan seperti *web server* (*Apache*), dan layanan keamanan seperti *ModSecurity* sebagai *WAF* yang dibutuhkan pada penelitian ini. selain itu, *web* sakattaku yang asli berlangganan dengan *Cloud Host*ing yang

menggunakan *WHM* serupa. *WHM* dapat diakses melalui browser melalui ip address *server* dengan port 2087 atau menggunakan tautan https://IP_Address_*server*:2087.

WHM menyediakan layanan yang menawarkan kontrol terperinci untuk *server* kepada administrator atau pengguna melalui fitur *cPanel*. Dimana *cPanel* adalah antarmuka berbasis *web* untuk mengelola *web* dalam penelitian ini.

3.2 Topologi Pengujian

Objek pengujian *VA* sebelum *WAF* diimplementasikan yaitu *web* sakattaku yang asli. Topologi menggambarkan sebuah skenario pengujian keamanan pada sebuah *website* Sakattaku, secara sederhana menggambarkan interaksi antara berbagai komponen yang terlibat dalam proses pengujian. Adapun topologi pengujian sebelum *WAF* diimplementasikan dapat dilihat pada Gambar 3.6



Gambar 3. 6 Topologi Pengujian *VA* sebelum *WAF* diimplementasikan

Komponen – komponen utama yang ada pada topologi pengujian sebagai berikut :

1. *VA*

Proses mengevaluasi kerentanan atau kelemahan keamanan pada suatu sistem. Dalam konteks gambar ini, *VA* dilakukan menggunakan *OWASP ZAP.*

2. *OWASP ZAP*

Alat *open source* yang populer digunakan untuk melakukan pengujian penetrasi pada aplikasi *web*. Alat ini akan mencoba menemukan celah keamanan *website* yang dapat dieksploitasi oleh *Hacker*.

3. *Internet Service ProVAider (ISP)*

Penyedia layanan internet yang menghubungkan berbagai perangkat ke internet. Dalam skenario ini, *ISP* bertindak sebagai jembatan antara perangkat penguji (Laptop) dan *server web*.

4. *Web Server*

Sistem komputer yang menjalankan *software* khusus untuk melayani permintaan halaman *web*. Dalam gambar, *web server* ini menjalankan *website* ”Sakattaku.com” dan dilengkapi *firewall* bernama *ModSecurity*.

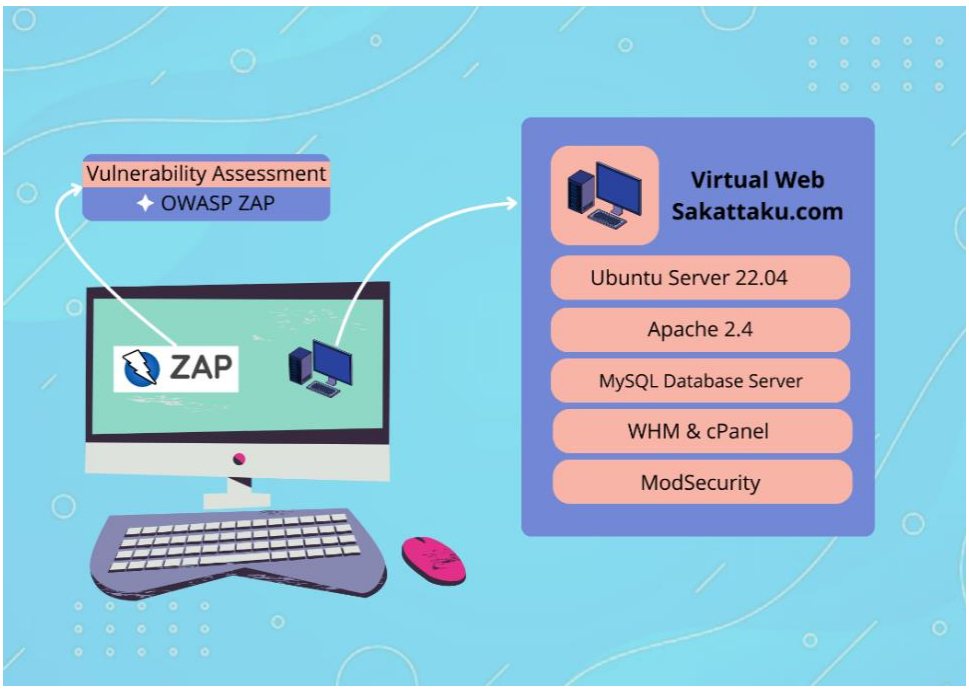
5. *ModSecurity*

Sebuah aplikasi *web server* yang berfungsi sebagai *WAF, WAF* ini berperan sebagai pertahanan pertama dalam melindungi aplikasi w*eb* dari berbagai jenis serangan. *ModSecurity* berfokus pada pemilahan data yang dikirim dan diterima oleh aplikasi *Web* dan berjalan pada layer 7 dari model osi.

6. *CloudHosting*

Server Sakattaku.com menggunakan layanan *CloudHosting,* dimana infrastruktur dari *server* dimiliki oleh penyedia layanan tersebut.

Adapun topologi pengujian setelah *WAF* diimplementasikan sedikit berbeda dikarenakan pada pengujian ini, objek yang digunakan adalah salinan atau virtualisasi dari *web* Sakattaku yang asli. Hal ini ditujukan untuk menjaga kestabilan *web* yang asli dari risiko yang ada, seperti adanya kesalahan dalam konfigurasi yang dapat membuat *web* menjadi *error.* Walaupun demikian pengujian melalui virtualisasi *web* tidak memiliki perbedaan dengan pengujian *web* yang asli. Hanya saja topologi nya menjadi berbeda. Adapun topologi *VA* sesudah *WAF* diimplementasikan, ditunjukkan pada Gambar 3.7



Gambar 3. 7 Toplogi Pengujian *VA* setelah *WAF* diimplementasikan

3.3 Metode Pengujian

Pengujian *VA* dengan *WAF* dan tanpa *WAF* memiliki tujuan yang sama, yaitu mengidentifikasi kerentanan dalam aplikasi *web*. Namun, tahapan pengujian berbeda karena *WAF* menambahkan lapisan keamanan tambahan. Tabel 3.1 adalah tahapan pengujian VA

Tabel 3. 1 Tahapan pengujian *VA* tanpa *WAF* dan dengan *WAF*

Tahapan Pengujian	Tanpa *WAF*	Dengan *WAF*
4.2.1 Perencanaan dan persiapan		
a. Menentukan Ruang Lingkup	*Asset* yang akan diuji: *web* Sakattaku. Jenis pengujian: *Scanning* atau *Penetration Testing.*	*Asset* yang akan diuji: *web* Sakattaku Jenis pengujian: *Scanning* atau *Penetration Testing..*
b. Menentukan Alat	Alat untuk *Scanning Vulnerability: OWASP ZAP.*	Alat untuk *Scanning Vulnerability: OWASP ZAP.* Alat *WAF*: *ModSecurity.*
4.2.2 Penemuan Kerentanan		
a. Simulasi Serangan	Melakukan simulasi serangan terhadap kerentanan yang diidentifikasi dengan *tools OWASP ZAP.*	Melakukan simulasi serangan terhadap kerentanan yang diidentifikasi dengan *tools OWASP ZAP.*
b. Teknik Penyerangan	Teknik penyerangan dilakukan dengan *tools Zed Attack Proxy (ZAP)* dengan mengedepankan metode *XSS*, *SQL injection*, dan *CSRF.*	Teknik penyerangan dilakukan dengan *tools Zed Attack Proxy (ZAP)* dengan mengedepankan metode *XSS*, *SQL injection*, dan *CSRF.*
c. Penilaian Dampak	Menilai dampak dari kerentanan yang berhasil dieksploitasi.	Mensimulasikan serangan untuk mengevaluasi kemampuan *WAF* dalam

Tahapan Pengujian	Tanpa *WAF*	Dengan *WAF*
		mendeteksi dan mencegah serangan.
4.2.3 Analisis dan penilaian		
a. Klasifikasi kerentanan	Mengkategorikan kerentanan berdasarkan tingkat keparahan, *eksploitabilitas,* dan dampak potensial.	Mengkategorikan kerentanan dengan mempertimbangkan kemampuan *WAF* dalam memitigasi eksploitasi.
b. Analisis Risiko	Mempertimbangkan faktor seperti kemungkinan serangan, dampak potensial, dan nilai aset yang terancam.	Mempertimbangkan kemampuan *WAF* untuk meminimalkan risiko.
c. Prioritas Perbaikan	Mengalokasikan sumber daya untuk mengatasi kerentanan dengan tingkat risiko tertinggi terlebih dahulu.	Memprioritaskan perbaikan kerentanan yang dapat dengan mudah dieksploitasi dan memiliki dampak signifikan. Dengan *rules* yang didapat.

Berikut perbedaan dari perbandingan *VA* dengan *WAF* dan tanpa *WAF*, disajikan pada tabel 3.2

Tabel 3. 2 Perbedaan VA dengan *WAF* dan tanpa *WAF*

Fitur	Tanpa *WAF*	Dengan *WAF*
Fokus	Mengidentifikasi kerentanan dalam aplikasi *Web*	Melindungi aplikasi *Web* dari serangan
Metode	Pemindaian otomatis, pengujian *penetrasi* manual, analisis kode *statis*	Pemindaian *pra-WAF*. Pemindaian *pasca-WAF*, Pengujian penetrasi *WAF*
Tujuan	Sebagai perbandingan untuk *Web* yang menggunakan *WAF* dan tidak dengan *WAF*	Mencegah serangan dan pelanggaran data

Fitur	Tanpa *WAF*	Dengan *WAF*
Kekurangan	Bersifat reaktif, tidak mendeteksi semua kerentanan	Memerlukan konfigurasi dan pemeliharaan yang berkelanjutan, dapat memblokir permintaan sah secara tidak sengaja
Keuntungan	Memberi wawasan mendalam tentang kerentanan aplikasi	Perlindungan *real-time* terhadap berbagai serangan

Berikut cakupan perbandingan dari *VA* dengan *WAF* dan tanpa *WAF*, disajikan pada Tabel 3.3

Tabel 3. 3 Cakupan Perbandingan *VA* dengan *WAF* dan tanpa *WAF*

Aspek	Tanpa *WAF*	Dengan *WAF*
Kerentanan	Semua kerentanan dalam aplikasi w*eb*	Kerentanan dalam aplikasi w*eb*, potensi *bypass WAF*, efektivitas *WAF*
Pengujian	Berfokus pada aplikasi w*eb*	Meliputi aplikasi w*eb*, fungsionalitas *WAF*, dan integrasi *WAF*-aplikasi
Temuan	Kerentanan dalam kode, konfigurasi, dan fungsionalitas aplikasi w*eb*	Kerentanan dalam aplikasi, potensi kelemahan *WAF*, dan celah *bypass*

Tahap akhir dari metodologi penelitian ini adalah pelaporan, yang merupakan kompilasi dari seluruh temuan dan analisis yang dilakukan selama penelitian. Laporan penelitian ini disusun dengan struktur yang sistematis, mencakup pendahuluan, metodologi, hasil dan pembahasan, dan kesimpulan. Setiap bagian dari laporan dirancang untuk memberikan gambaran yang jelas dan menyeluruh tentang proses penelitian, serta kesimpulan yang dapat ditarik dari hasil pengujian. Selain itu, laporan ini juga menyertakan rekomendasi untuk penerapan lebih lanjut dari *WAF* dalam meningkatkan keamanan aplikasi *web* berdasarkan temuan penelitian.