BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian ini akan terlaksana berdasarkan hasil dari peneliti-peneliti sebelumnya, yaitu oleh (Syaefuddin, 2018) dengan judul "Implementasi *Web Application Firewall* pada *Web Mytra Dashboard* dengan Menggunakan Modul *ModSecurity*" yang dimana dalam penelitian tersebut penulis melakukan pengujian serangan *SQL Injection* dan *XSS Attack* terhadap *web server Nginx* yang telah dilindungi oleh *WAF ModSecurity* menggunakan *rule Comodo Security*. Hasil yang didapatkan dari kedua serangan tersebut berbuah manis sesuai yang diharapkan ditandai dengan terdeteksinya serangan di *log* dan serangan berhasil dialihkan dengan tampilan pada *web* menjadi *403 Forbidden*.

Penelitian berikutnya dilakukan oleh (Suartana, Wahanani, & Sandy, 2015) dengan judul "Sistem Pengamanan *Web Server* Dengan *Web Application Firewall (WAF)*" yang dilakukan pengujian serangan *Cross Site Scripting* atau *XSS* dan *SQL Injection* terhadap *web server* dengan perlindungan dari *web application firewall* yaitu *ModSecurity* dengan pelaporan serangan menggunakan *Jwall Auditconsole* yang dimana *ModSecurity* akan bekerja memerika dan melakukan *filter request* yang akan dating meliputi *user, user angent, ip address* dan input yang akan diinput ke *web server* serta jenis *file* yang diminta, apakah berbahaya atau tidak sebelum mendapat akses ke *website*.

Selanjutnya penelitian yang telah dilakukan oleh (Jamain, Periyadi, & Ismail, 2015) dengan judul penelitian "Implementasi Keaman Aplikasi *Web* Dengan *Web Application Firewall*" yang dimana penulis akan menguji *web*

application firewall yaitu *Naxsi* dalam melindungi *web server* yang telah diimplementasikan pada *Linux Mint* terhadap serangan *Comman Execution, XSS dan SQL Injection*. Penulis melakukan pengujian dengan dan tanpa *WAF* yang dimana ketika *WAF* belum diaktifkan penyerang dapat mencuri data, sedangkan ketika *WAF* telah diaktifkan maka *Naxsi* akan melakukan *filtering* terhadap *HTTP Request* yang akan menuju server, jika mengandung tindakan penyerangan akan segera dihentikan.

Berikut adalah penelitian yang dilaksanakan oleh (Widianto & Azzam, 2018) yang berjudul "Analisis Upaya Peretasan *Web Application Firewall* dan Notifikasi Serangan Menggunakan *Bot Telegram* pada Layanan *Web Server*" yang dimana penulis akan melakukan pengujian terhadap *web server* dari berbagai serangan yaitu *Remote Command Execution, Remote File Inclusion, Local File Inclusion, XSS,* dan *SQL Injection* dan *Bot Telegram* berperan sebagai pemberitahu jika terjadi serangan yang telah terdeteksi oleh bantuan WAF *ModSecurity* pada syslog agar SysAdmin dapat mudah mengetahui apakah server sedang diserang secara jamak atau secara individual atau perorangan.

Penelitian selanjutnya dilakukan oleh (Pramaditya, 2016) yang berjudul "*Brite Force Password Cracking* Dengan Menggunakan *Graphic Processing Power*" yang dimana penulis mencoba membobol kata sandi secara *manual* dengan cara menebak semua abjad huruf atau angka serta simbol yang kemungkinan dapat diprediksi kata sandinya. Pada penelitian ini penulis menggunakan *software WinZip* dan *WinRAR* yang dapat menebak kata sandi sebanyak-banyaknya tanpa batas, dan membebankan terhadap spesifikasi komputer yang penulis gunakan yaitu *CPU (Intel Core i5-2500k)* dengan kecepatan sekitar 28 juta kata sandi per detik,

sedangkan pada pengujian dengan *GPU (Geforce GTX 460)* yang memakai *software Accent Password Recovery* dan ketika diuji dapat dinilai bahwa ada lonjakan performa yang sangat signifikan sebesar 500 juta kata sandi per detik.

Tabel 2.1 Tinjauan Pustaka

Peneliti	Tahun	Kesimpulan Penelitian
Rizky Nurachmad Syaeffudin	2018	*WAF ModSecurity* dengan *rule Comodo Security* dapat menangkal serangan *SQL Injection* dan *XSS* dengan baik serta mendeteksi dengan respon yang cepat
I Made, Henni & Aditya	2015	*ModSecurity* yang dibekali *Jwall Auditconsole* yang akan memeriksa dan melakukan *filter request* yang akan dating meliputi *user, user angent, ip address* dan input yang akan diinput ke *web server* serta jenis *file* yang diminta, apakah berbahaya atau tidak sebelum mendapat akses ke *website*.
Risma, Periyadi & Setia	2015	*Web Application Firewall* yaitu *Naxsi* juga efektif menangkal serangan-serangan seperti *SQL Injection*, XSS, *Remote Command Execution*, dan lain sebagainya.
Septian & Izzudin	2018	*ModSecurity* dapat melakukan kostumisasi dan penambahan *rules* dan notifikasi *Bot Telegram* sangat efektif untuk mengawasi *web server* secara *remote* dimana saja.
Himawan	2016	*Brute Force* bekerja dengan kemungkinan, semakin rendah keamanan *password* makan semakin efektif juga pembobolannya. Dan *Brute Force* juga membebani *resource* dari perangkat yang melakukan pembobolan yaitu *CPU* dan *GPU*.

Berdasar dari hasil penelitian yang telah dilakukan oleh para peneliti dengan topik yang sama, maka penulis akan menganalisa tingkat kemanan *web server* tersebut. Penelitian ini juga bermanfaat untuk mengetahui sejauh mana *web application firewall* dapat bekerja mengamankan *web server* dari berbagai serangan

eskternal yang memungkinkan membahayakan *web serber* itu sendiri seperti yang dilakukan oleh Rizky (2018), I Made, Henni & Adtiya (2015), Risma, Periyadi & Setia (2015), Septian & Izzudin (2018), dan Izzudin (2017) yang terbukti berhasil membuktikan dengan meneliti *web application firewall* dalam melindungi *web server*.

2.2 Landasan Teori

Untuk menganalisa dan merancang *web application firewall* pada *web server*, maka penulis merangkum beberapa poin sebagai landasan teori dalam penelitian ini. Landasan teori sendiri adalah kumpulan teori-teori yang digunakan sebagai landasan selama penelitian berlangsung. Berikut adalah teori-teori yang telah dirangkum:

2.2.1 Jaringan Komputer

Jaringan Komputer atau *Computer Network* adalah sebuah kumpulan yang meliputi beberapa perangkat atau komputer yang dihubungkan menjadi satu sehingga dapat bertukar data satu sama lain dan diolah menjadi sebuah informasi. Jaringan Komputer terbagi menjadi beberapa jenis sesuai dengan luas wilayah atau jarak cakupan jaringan komputer tersebut yakni Jaringan Pribadi atau *Personal Area Network (PAN)* yang hanya mencakup area pribadi yang dekat seperti antar perangkat nirkbal yang dimiliki pengguna, Jaringan Lokal atau *Local Area Network (LAN)* yang mencakup beberapa komputer di satu atau dua bangunan, Jaringan Metropolitan atau *Metropolitan Area Network (MAN)* yang mencakup beberapa daerah dalam satu kota, Jaringan Luas atau *Wide Area Network (WAN)* yang

mencakup area yang sangat luas meliputi antar kota, negara, maupun benua (Rahadjeng & Ritapuspitasari, 2018).

2.2.2 Keamanan Jaringan

Keamanan jaringan merupakan sebuah proses pencegahan dan identifikasi dari aktifitas penggunaan yang tidak sesuai serta dari jaringan komputer tersebut. Keamanan jaringan juga bertujuan untuk mengantisipasi ancaman dan resiko dalam bentuk logika maupun fisik yang mengganggu secara langsung dan tidak langsung (Ma'sum, Irwansyah, & Priyanto, 2017). Berikut ini adalah 3 konsep umum dalam keamanan jaringan:

1. *Risk*

   Resiko atau tingkat bahaya untuk menyatakan berapa besar sebuah kemungkinan jika jaringan komputer tersebut disusupi oleh *intruder* atau penyusup.

2. *Threat*

   Untuk menyatakan ancaman yang datang dari pihak yang ingin mengakses sistem jaringan komputer pengguna secara illegal.

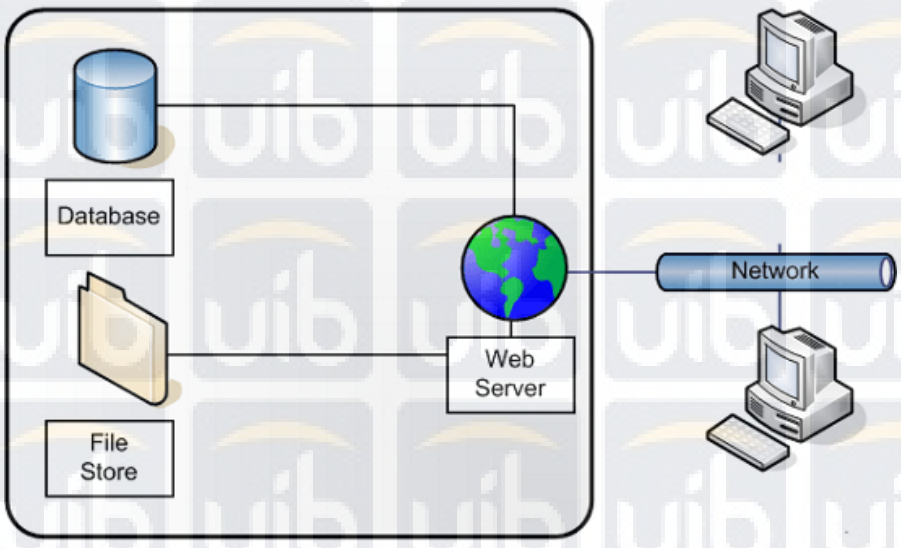
3. *Vulnerability*

   Untuk menyatakan seberapa tahan dan kuat sebuah sistem keamanan dari ancaman dan bahaya eksternal yang akan datang.

2.2.3 *Web Server*

Web Server sendiri merupakan wadah untuk menopang sebuah *website* yang berisikan informasi, konten, maupun akun surel atau *email* pengguna,

sekaligus sebagai penghubung antara *server* dan *client* dalam pengiriman dan perolehan informasi dan data serta mempercepat dan mengorganisasikannya secara terpusat. *Web server* sendiri menggunakan port 80 dan terdiri dari 2 komponen, yaitu perangkat komputer dan *software web server* yang dipakai, yang dimana *web server* inilah yang akan menopang *website* yang ditujukan kepada *client* atau pengguna untuk memberikan dan bertukar informasi. *Web server* juga terdapat beberapa jenis antara lain: *Nginx, Apache Tomcat, Apache Web Server, Microsoft Windows Server, Light HTTP, Internet Information Service (IIS), Sun Java System Web Server,* dan *Zerus Web Server*. Dan terkhusus untuk *Hosting Server*, hanya akan berfokus pada implementasi *web server* yang membahas bagaimana kinerja *web server* yang akan di implementasikan (Aziz & Tampati, 2015).

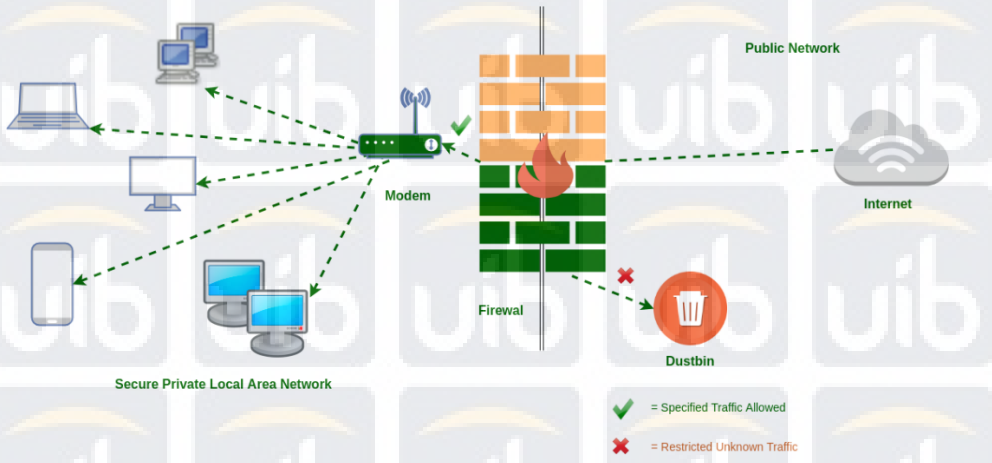


Gambar 2.1 Skema *Web Server*

2.2.4 *Firewall*

Firewall atau tembok api merupakan kumpulan komponen yang berfungsi untuk membatasi sebuah akses eksternal antar jaringan-jaringan dilindungi dan

internet yang melalui *network traffic*, dan berperan sebagai alat untuk mengimpementasikan *security policy* yang diciptakan berdasarkan antar implikasi keamanan dan fasilitas nya, semakin kompleks dan rumit sebuah *security policy* maka semakin rumit juga konfigurasi layanan dan fasilitas yang disediakan. Fungsi lain *firewall* juga mencegah *hacker* atau peretas untuk mencuri data dari sebuah perangkat yang terhubung ke internet. Firewall juga berfungsi untuk membatasi siapa saja yang dapat dan berhak untuk mengakses koneksi internet dalam sebuah jaringan, dan siapa saja yang mendapatkan izin untuk lewat dan mengakses koneksi jaringan internet hal ini juga biasa disebut dengan *filtering*, dan *firewall* juga berfungsi untuk memantau atau *monitoring traffic* sebuah jaringan. *Firewall* juga peka terhadap sebuah kegagalan dalam penerapan *policy* dan kesalahan konfigurasi sehingga diperlukannya sebuah tambahan peningkatan keamanan yang lainnya guna menyempurnakan fungsi *firewall* pada perangkat dan jaringan (Purwaningrum, Purwanto, & Darmadi, 2018).

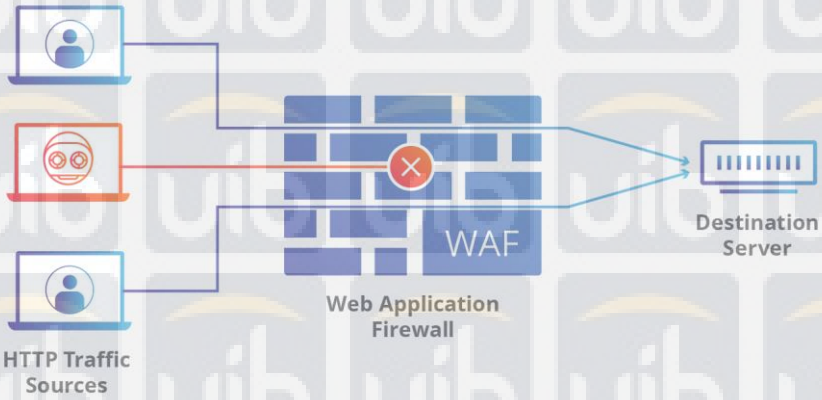


Gambar 2.2 Skema *Firewall*

2.2.5 *Web Application Firewal (WAF)*

Web Application Firewall atau WAF adalah sebuah aplikasi yang bertujuan untuk mengamankan dan mencegah sebuah *web* dari upaya penyerangan dari peretas untuk mendapatkan data dan informasi dan eksploitasi dalam jumlah yang besar yang dapat mempengaruhi kestabilan sebuah *web* hingga memnyebabkan *web* tidak dapat diakses atau sering disebut *down* (Syaefuddin, 2018).

Web Application Firewall berkerja dengan melakukan konfigurasi tambahan pada *web server* sehingga tidak perlu melakukan melakukan perubahan *script* pada applikasi tersebut, dan dapat diterapkan pada *web server* atau aplikasi yang sudah berjalan layaknya *firewall* yang melakukan *filtering* keluar dan masuknya data serta dengan tanggap menghentikan *traffic* yang diprediksi atau dianggap berbahaya berdasarkan *rule* yang telah dikonfigurasi dan ditetapkan. WAF juga memiliki beberapa fungsi antara lain: *Secure Directory, String Filtering, Traffic Network Monitoring,* dan proteksi untuk menangkal serangan-serangan seperti *Brute Force, Denial of Service* (DOS), *Distributed Denial of Service* (DDoS), SQL *Injection, Unrestricted File Upload*, serta *Cross-Site Scripting* (XSS) dan lain sebagainya (Jamain et al., 2015)



Gambar 2.3 Skema *Web Application Firewall* (WAF)

2.2.6 *ModEvasive*

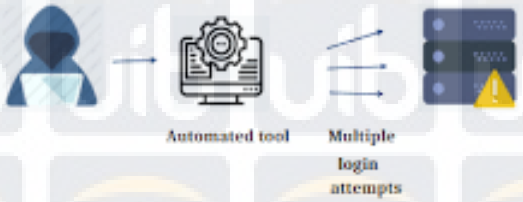
ModEvasive adalah sebuah *evasive maneuvers module web server* Apache yang berperan sebagai aksi pengelakan atau *evasive* ketika saat terjadi serangan terhadap HTTP DoS atau DDoS dan serangan *brute force*. *ModEvasive* juga berfungsi sebagai *network management tools* sekaligus pendeteksi, serta dapat dikonfigurasi secara mudah guna terdeteksi oleh *router*, *firewall*, dan *ipchains*. *ModEvasive* dapat melaporkan sebuah penyalahgunaan pada *web server* melalui *syslog* dan surel atau *e-mail*. Pendeteksian *ModEvsive* dilakukan dengan membuat *internal dynamic hash table* dari URLs dan IP, dan men-*deny* atau menolak alamat IP yang berasal dari: IP yang membuat banyak permintaan pada waktu yang bersamaan di-*blacklist* secara *temporary* pada *blocking list*, IP yang membuat permintaan lebih dari 50 *concurrent* pada *child* yang sama per detiknya, dan IP yang melakukan *request* halaman yang sama berulang kali pada kurun waktu tertentu per detiknya (Ma'sum et al., 2017).

Gambar 2.4 Ilustrasi ModEvasive pada *web server* Apache

2.2.7 *Brute Force*

Brute Force atau dalam Bahasa Indonesia yaitu Serang Brutal merupakan sebuah serangan terhadap sistem kemaman pada sebuah perangkat komputer, *web server, database* dan lain sebagainya yang memiliki kunci dengan percobaan kumpulan sandi atau *password*, kode, dan kombinasi. Kata sandi yang dibongkar dengan menggunakan program yang dinamakan *password cracker*, adalah sebuah program yang mencoba membobol sebuah kata sandi yang telah terenkripsi dengan algoritman tertentu dari berbagai kemungkinan percobaan, walaupun sangat sederhana dan terbilang memakan waktu yang cukup lama tergantung seberapa rumit kata sandi itu sendiri namun belum ada sistem di masa kini yang aman terhadap serangan sederhana seperti ini. *Brute Force* sangat bergantung pada kemungkinan atau probabilitas, semakin rumit dan Panjang sebuah kata sandi maka semakin banyak pula kata sandi yang ada untuk diperiksa dari masing-masing huruf, angka maupun symbol yang dipakai. Hal ini juga bergantung pada teori permutasi, yang berupa sususan angka dalam urutan-urutan tertentu. Layaknya anagram, jika diberi huruf c, b, dan a, berapa banyak susunan perintah berbeda yang bisa dibuat dengan berdasarkan tiga huruf tersebut, maka tiga huruf tersebut dalam dibuat dalam se-*set* enam permutasi dari himpunan {c, b, a}, yaitu [a, c, b], [a, b, c], [b, c, a], [b, a, c], [c, b, a], [c, a, b]. Namun kemungkinan pada kata sandi yang sederhana, pengulangan dibolehkan, sehingga rumus untuk jumlah kemungkinan-kemungkinan kata sandi **p** untuk ditebak adalah **p = x pangkat n** dimana **x** merupakan jumlah karakter kemungkinan, dan **n** adalah panjang dari kata sandi tersebut (Himawan, 2016). *Brute Force* juga dapat dilakukan dengan dukungan *tool* yang dapat memproses penebakan huruf secara cepat seperti: *John the Ripper*,

Aircrack-ng, *Cain and Abel*, *Rainbrow Crack*, *Opcrack*, *L0phtCrack*, *Hashcat*, *Crack*, *Ncrack*, *SAMInside*, *DaveGrohl* dan THC *Hydra* (Pramaditya, 2016).

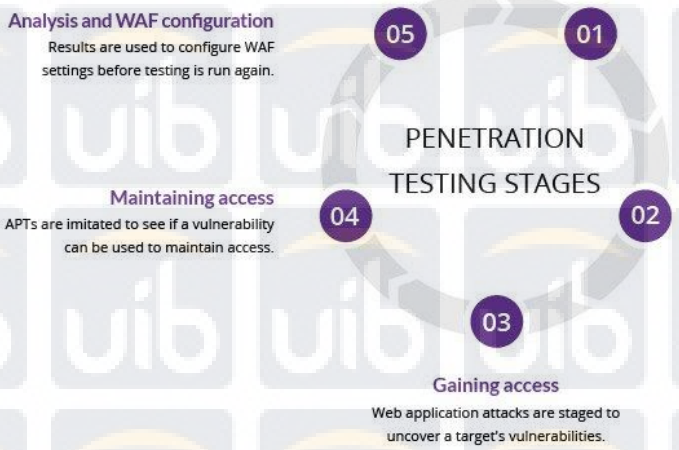


Gambar 2.5 Ilustrasi *Brute Force*

2.2.8 *Penetration Testing*

Uji Penetrasi atau *Penetration Testing* yang disingkat (*pentest)* merupakan sebuah rangkaian kegitatan yang dilakukan bertujuan untuk mengeksploitasi dan mengidentifikasi celah atau kerentanan keamanan sebuah sistem sebelum adanya keadaan dimana sistem tersebut dikendalikan oleh pihak yang tidak berwenang terhadap sistem tersebut. *Pentest* juga merupakan hal yang biasa digunakan oleh pihak IT sebuah perusahaan atau lembaga tertentu karena dinilai sebagai jaminan bernilai yang menguntungkan baik dari segi operasional dan bisnisnya, dari segi operasional *pentest* berfungsi untuk membantu dalam pembentukan strategi keamanan informasi melalui proses identifikasi yang akurat nan cepat; pelaksanaan tindakan korektif; dan penghapusan proaktif dari sebuah risiko yang telah diidentifikasi. *Pentest* juga dapat memberikan informasi terperinci tentang ancaman kemanan secara aktual, yang dapat dieksploitasi jika mencakup doktrin serta proses keamanan organisasi, yang dimana hal ini juga dapat membantu organisasi untuk mengidentifikasi dimana letak potensi celah dan kerentanan yang dikecualikan

secara cepat dan akurat (Tarigan, Kusyanti, & Yahya, 2017). Gambar dibawah ini menjelaskan alur penerapan *pentest* pada sebuah sistem:



Gambar 2.6 Alur *Penetration Testing*