Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438

Analisis Web Performance Load Test Setelah Menggunakan Azure WAF Studi Kasus Pada Aplikasi ERP

Denil Cristianto*, Indrastanti R Widiasari

Fakultas Teknologi Informasi, Teknik Informatika, Universitas Kristen Satya Wacana, Salatiga, Indonesia
Email: [1,*]672016144@student.uksw.edu, [2]Indrastanti.widiasari@uksw.edu
Email Penulis Korespondensi: 672016144@student.uksw.edu
Submitted: **21/03/2022**; Accepted: **31/03/2022**; Published: **31/03/2022**

Abstrak−Meningkatnya penggunaan aplikasi web tentu sebanding dengan meningkat nya serangan cybersecurity yang terjadi. Banyak perusahaan atau pun instansi yang kurang peduli terhadap keamanan aplikasi web yang digunakan sehingga aplikasi web tersebut dapat dengan mudah untuk diretas. Disisi lain ada banyak cara untuk melindungi website dari serangan hacker, salah satunya adalah mengimplementasikan waf,WAF adalah aplikasi atau perangkat yang memiliki kemampuan untuk melakukan filtering, monitoring, dan blocking traffic yang menuju ke suatu website. Waf yang digunakan pada penelitian ini adalah waf berbasis cloud yaitu azure application gateway. Berdasarkan hal tersebut, penelitian ini bertujuan untuk membahas tentang analisis web performa dan load test suatu website setelah mengimplementasikan waf. Tools yang digunakan untuk menguji Web Performance dan Load Test adalah GTMetrix dan Pingdom Tools, Dengan adanya penelitian ini diharapkan dapat menjadi acuan bagi individu ataupun instansi yang akan mengimplementasikan waf.

Kata Kunci: Waf; Web; Azure; Performa; Load

Abstract−The increasing use of web applications is certainly proportional to the increase in cybersecurity attacks that occur. Many companies or agencies are less concerned about the security of the web applications that are used so that these web applications can be easily hacked. On the other hand, there are many ways to protect websites from hacker attacks, one of which is implementing waf. WAF is an application or device that has the ability to filter, monitor, and block traffic that goes to a website. The waf used in this study is a cloud-based waf, namely azure application gateway. Based on this, this study aims to discuss web performance analysis and load testing of a website after implementing waf. The tools used to test Web Performance and Load Test are GTMetrix and Pingdom Tools. With this research, it is hoped that this research can be a reference for individuals or agencies that will implement waf.

Keywords: Waf; Website; Azure; Performance; Load

1. PENDAHULUAN

Seiring dengan berkembangnya teknologi jaringan, *website* pun juga turut berkembang dan menjadi populer di kalangan masyarakat, apalagi di situasi pandemi yang mengharuskan belajar atau bekerja dari rumah jumlah penggunaan aplikasi berbasis *website* pun meningkat [1][2]. Dengan meningkatnya penggunaan aplikasi web tentu sebanding dengan meningkat nya serangan *cybersecurity* yang terjadi, banyak perusahaan atau pun instansi yang kurang *aware* atau kurang peduli terhadap keamanan aplikasi web yang digunakan sehingga aplikasi *web* tersebut dapat dengan mudah untuk diretas, hal tersebut bisa dilihat pada situs arsip defaced *website* (www.zone-h.org) [3]. Maraknya aksi hacking tersebut cukup berbahaya karena jika *website* disusupi *hacker* bisa saja data pribadi atau data sensitif milik instansi atau perusahaan tersebut bocor [3] [4].

Dengan tidak ada nya perlindungan terhadap *server* yang menampung *web* tersebut maka *hacker* dapat dengan mudah untuk mendapatkan akses ke dalam server yang diserang. Di era teknologi sekarang ini terdapat banyak cara untuk melindungi *server* dari serangan serangan *hacker*, salah satunya adalah mengimplementasikan *WAF (Web Application Firewall)* [5].

WAF adalah aplikasi atau perangkat yang memiliki kemampuan untuk melakukan *filtering*, *monitoring*, dan *blocking traffic* [6]. *WAF* memiliki perbedaan dengan *firewall* pada umumnya karena *WAF* hanya memfilter *traffic* yang menuju aplikasi berbasis *web*[5]. Dengan adanya kemampuan menginspeksi *traffic HTTP*, hal tersebut dapat mencegah terjadinya serangan yang ditujukan ke aplikasi berbasis *web* tersebut seperti *SQL Injection*, *cross-site scripting*, dan *unrestricted file upload*[7]. Penelitian ini diharapkan akan membantu banyak pihak terutama pengelola *website* yang telah meng-implementasikan *waf* untuk mengetahu apakah *waf* yang telah dipasang atau diimplementasikan berpengaruh terhadap performa web[8].

Dalam penulisan penelitian ini penulis kaitkan dengan beberapa karya ilimiah terdahulu yang membahas mengenai *WAF*. Adapun karya ilmiah yang penulis maksud adalah sebagai berikut :

Penelitian terkait sebelumnya dilakukan oleh Bangkit Wiguna & Wahyu Adi Prabowo & Ridho Ananda. Penelitian ini membahas tentang maraknya serangan *SQL Injection* yang mengakibatkan dampak yang merugikan terhadap pengguna maupun pihak pengelola *website*. Dalam penelitian ini selain membahas mengenai *SQL Injection* di sini juga dijelaskan solusi apa yang dapat digunakan untuk mengatasi serangan *SQL Injection* yaitu dengan menggunakan *WAF* yang menggunakan metode *detection rules* yang telah ditetapkan untuk dapat memblokir akses bagi penyerang yang menginputkan permintaan berbahaya ataupun penyerang yang mengeksekusi serangan *SQL Injection* menggunakan tools kedalam *website* [9].

Dalam yang dilakukan oleh Rizky Nurachmad Syaeffudin, membahas tentang maraknya kejahatan cyber melalaui web application attack. Di penelitian ini juga membahas mengenai bagaimana upaya untuk membangun

Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438

suatu system keamanan yang mampu memantau dan mencegah terjadinya serangan menggunakan *WAF* Modsecurity yang diintegrasikan dengan *web server nginx*. Pada bagian akhir dari tulisan ini membuktikan bahwa system keamanan yang dibangun dapat mendeteksi serangan dan mencatat serangan yang terjadi kedalam log [10].

Dalam penelitian Paulus Miki Resa Gumilang dan Dian Widiyanto Chandra menyebutkan bahwa pada saat ini masih sangat marak aksi *hacker* luar negeri maupun dalam negeri yang dengan sengaja mengubah tampilan *website* secara illegal, setiap harinya terdapat belasan hingga puluhan situs penting di Indonesia berdomain (*go.id) yang terkena aksi serangan deface, hal tersebut dapat dilihat dari web arsip (zone-h.org). tidak semua serangan diarsipkan pada *website* zone-h sehingga jumlah total serangan bisa saja lebih besar, oleh karena itu pemerintah Indonesia harus lebih memperhatikan keamanan pada sistem maupun infrastruktur yang dimiliki [3].

Dalam penelitian yang dilakukan oleh Anggrahito dkk membahas mengenai maraknya serangan atau kasus peretasan yang banyak terjadi pada sektor pemerintahan. Di penelitian ini dibahas mengenai bagaimana menggunakan membangun *WAF* menggunakan *reverseproxy* dan *modsecurity*. Penelitian ini juga membuktikan bahwa dengan di implementasikan-nya *WAF* sangat efektif untuk mencegah serangan serangan terhadap aplikasi web[6]**.**

Penelitian yang dilakukan oleh Riska & Hendri Alamsyah membahas upaya mengamankan sebuah aplikasi web dengan memasang *WAF*. Di penelitian ini dijelaskan bahwa *WAF* dapat bekerja tanpa harus melakukan perubahan atas script default aplikasi, sehingga dapat diterapkan pada aplikasi yang sudah berjalan walaupun script tersebut belum sesuai dengan keinginan. Dalam hasil akhir penelitian ini membuktikan bahwa dengan dipasang nya *WAF* dapat melindungi server dari serangan *SQL Injection, Cross Site Scripting (XSS), dan Command Execution* [11].

Pada penelitian yang dilakukan oleh Jamie Karisma Anggreana, membahas mengenai tentang Keamanan pada aplikasi web kurang mendapat perhatian dari developer. Sehingga banyak serangan serangan yang ditujukan terhadap sebuah web maka dari masalah tadi diperlukan pengamanan khusus yakni dengan Web Application Firewall. Dalam penelitian ini juga diuji bagaimana efektivitas *WAF* dalam mengatasi setiap serangan serangan yang ditujukan ke sebuah web [12].

Pada penelitian Feri Setiyawan lebih menekankan implementasi firewall aplikasi web dengan menggunakan naxsi yang akan dikonfigurasikan pada web server nginx untuk mencegah serangan yang dilakukan dengan menggunakan teknik *SQL Injection*. Dalam penelitian ini dijelaskan bahwa implementasi Firewall Aplikasi Web Naxsi dapat digunakan untuk mencegah serangan *SQL Injection* baik yang dilakukan secara manual maupun menggunakan tools. Penelitian ini juga menjelaskan bahwasannya Firewall aplikasi naxsi yang dipasang pada web server nginx tidak terlalu berpengaruh terhadap kinerja dari web server nginx [13].

Penelitian yang dilakukan Anno Harsoyo membahas tentang bagaimana performa website milik kementrian di Indonesia, Sebagian besar performa website milik kementrian di Indonesia tergolong buruk, sedangkan dari segi response time dan broken link tergolong baik. Metode yang digunakan untuk melakukan pemeringkatan website adalah Metode Entropi dan Electre [2].

Penelitian Suliman tentang tentang Analisis Performa Website menggunakan Pingdom Tools Dan Gtmetrix di penelitian ini di jelaskan bahwa dengan dilakukan nya *performance test* dapat diketahui kiranya apa saja yang menjadi kekurangan dan mempengaruhi perfoma dari website yang diukur. Performa website pada perguruan tinggi harus menjadi perhatian masing-masing perguruan tinggi dikarenakan salah satu peranan penting website dalam menunjang kegiatan di perguruan tinggi. Terdapat beberapa acuan yang digunakan sebagai bahan pertimbangan untuk menentukan baik buruknya kualitas sebuah website. Sebagai bahan pertimbangan yaitu: kecepatan akses, isi mudah dibaca, dan tata letak atau desain yang konsisten [1].

Sedangkan Ilham Alamsyah melakukan penelitian tentang pemanfaatan JMETER untuk pengujian performa website, JMETER dapat digunakan di berbagai platform, bersifat open source dan merupakan perangkat lunak yang sangat populer untuk melakukan pengujian fungsionalitas suatu aplikasi. Dalam penelitian ini juga dijelaskan Hasil dari penelitian ini adalah grafik tingkat keberhasilan dan kegagalan akses berdasarkan jumlah pengguna yang mengakses website Teknik Informatika Universitas Pasundan. Hasil tersebut dapat dievaluasi apabila akan dilakukan pengembangan terhadap website [14].

Berdasarkan penelitian-penelitian sebelumya yang relevan dan terkait dengan *waf* dan *performance testing* dapat disimpulkan bahwa belum ada penelitian yang membahas apakah dengan diimplementasikan nya waf memiliki pengaruh terhadap performa website[7][15].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Metode Penelitian yang akan digunakan dalam Analisis Web Performance dan Load Test Setelah menggunakan Azure WAF Studi Kasus pada Aplikasi ERP milik PT PLOSS ASIA, dapat dilihat pada Gambar1.

Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438

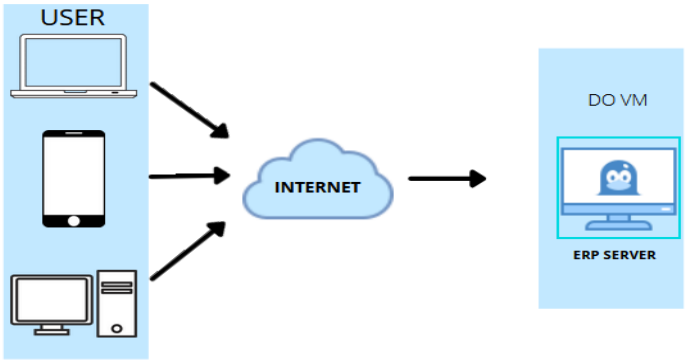
Gambar 1. Tahapan Penelitian.

Tahapan penelitian pada Gambar 1 dijelaskan sebagai berikut. **Tahap Identifikasi Masalah**: Pada tahap pertama ini akan dilakukan identifikasi terhadap permasalahan yang ada. **Tahap Observasi** : Pada Tahap ini penulis mengumpulkan penelitian penelitian yang sudah ada sebelumnya agar menjadi acuan perbandingan. **Tahap Persiapan Alat dan Bahan**: Pada tahap penulis akan mempersiapkan alat dan bahan yang dibutuhkan selama proses penelitian. **Tahap Penentuan Tools**: Pada tahap ini penulis menetukan tools apa yang akan digunakan pada proses performance testing. **Tahap Instalasi dan Konfigurasi *WAF***: Pada tahap yang kelima penulis akan melakukan instalasi dan konfigurasi *WAF* untuk selanjutnya di implementasikan pada server aplikasi web. **Tahap Pengujian dan Analisis** :Pada tahap akhir ini akan dilakukan penulisan laporan dari penelitian yang telah dilakukan sebagai bentuk dari dokumentasi serta *summary* rampungnya sebuah proyek penelitian.

2.2 Design Sistem

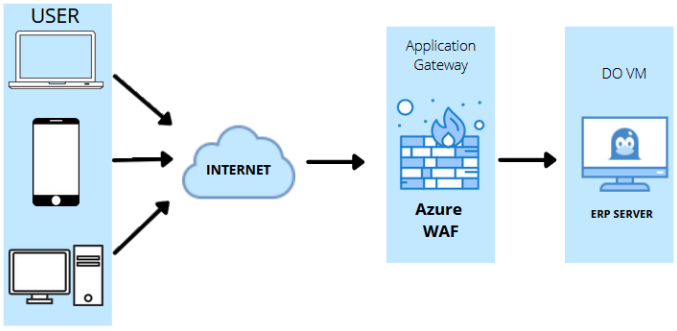
2.2.1 Topologi Jaringan

Sebelum melakukan penelitian terkait dengan implementasi *WAF*, terlebih dahulu akan dilakukan desain topologi dalam lingkungan *environment cloud* dari *system* atau infrastruktur yang akan diteliti. Adapun topologi jaringan yang akan digunakan adalah sebagai berikut :



Gambar 2. Topologi *Non waf*

Berdarkan Gambar 2, dapat dilihat bahwa user mengkases langsung menuju server aplikasi erp tanpa melalui filtering.



Gambar 3. Topologi Menggunakan Waf

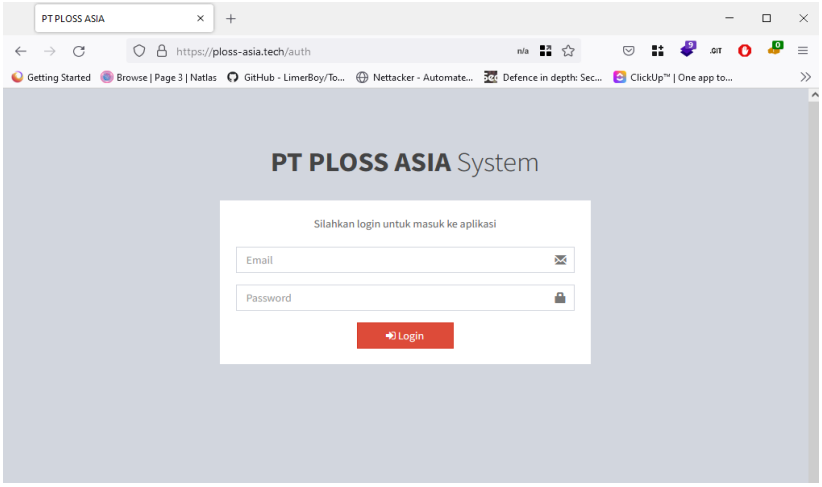
Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438

Berdasarkan gambar 3, dapat dilihat bahwa sebelum user mengkases server *website* atau aplikasi web, *trafic* dilewatkan terlebih melalui *Application gateway* yang di mana *Application Gateway* berfungsi sebagai *WAF* sehingga setiap trafik yang menuju server akan di filter dan di-sanitasi. *WAF* akan mendeteksi apakah traffic yang menuju ke server sebagai sebuah serangan atau bukan.

3. HASIL DAN PEMBAHASAN

3.1 Aplikasi Web

Aplikasi web yang akan digunakan adalah aplikasi ERP milik PT Ploss Asia. ERP adalah paket sistem dan software yang digunakan oleh perusahaan untuk mengelola kegiatan bisnis harian mereka, seperti pengelolaan keuangan, pengadaan, produksi, proyek, SDM, dan-lain-lain. Aplikasi ini berbasis PHP dengan basis data MySql yang akan dikonfigurasikan pada ubuntu server yang terdapat pada cloud milik *Digital Ocean* dengan alamat *ip public* 128.199.107.215 dan *domain* https://ploss-asia.tech.

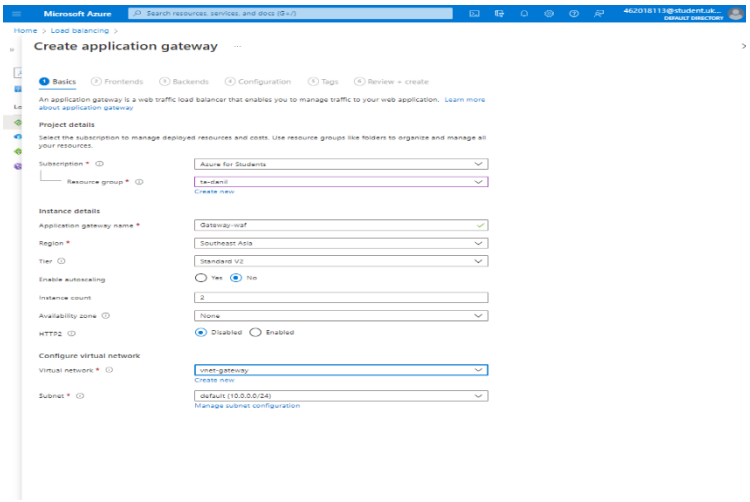


Gambar 4. Halaman depan *system* ERP

Gambar 4 merupakan aplikasi web yang akan digunakan sebagai studi kasus

3.2 Instalasi dan Konfigurasi *WAF*

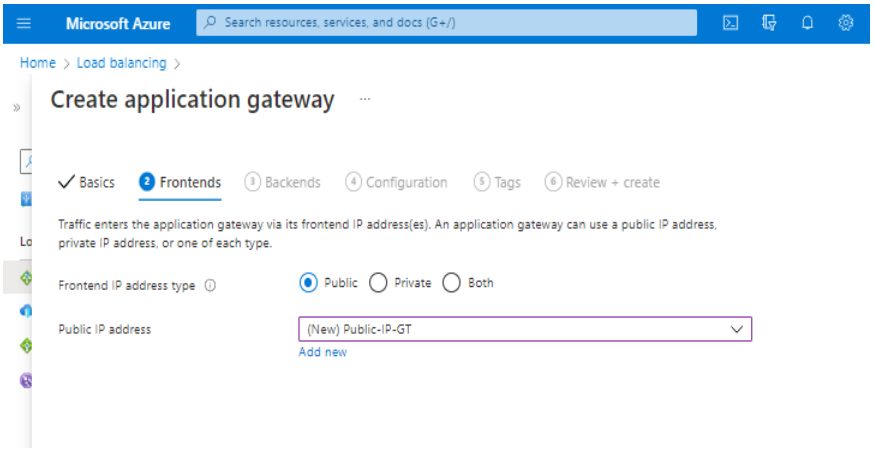
Untuk melakukan implementasi *WAF*, langkah pertama yang harus dilakukan adalah menambahkan fitur *Application Gateway* dari *Azure* yang di mana berfungsi sebagai *Load Balancer* dan juga *WAF*. Adapun untuk menambahkan fitur tersebut kita harus menyesuaikan dengan konfigurasi yang akan digunakan dan sesuaikan dengan konfigurasi server yang kita miliki.



Gambar 5. Konfigurasi Dasar *Application Gateway*

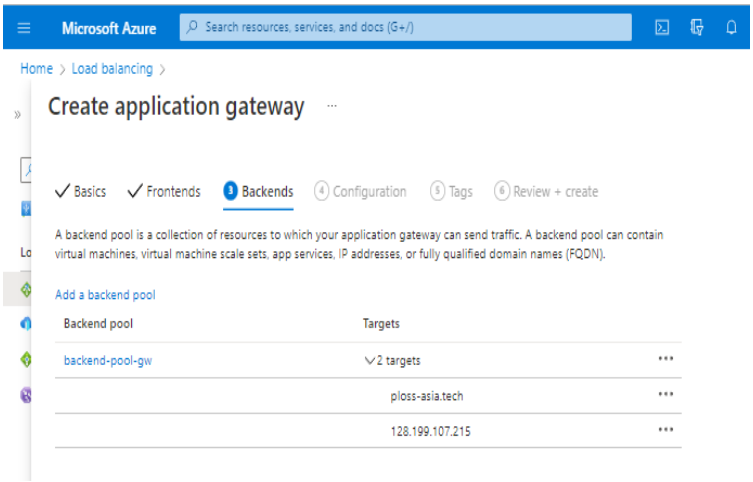
Setelah berhasil melakukan konfigurasi dasar dari *Application Gateway* langkah selanjutnya adalah mengkonfigurasikan *Frontends* dan *Backends*. *Frontends* sendiri berfungsi sebagai interface yang akan terhubung ke internet di mana *Frontends* memiliki alamat *IP* yang akan digunakan *user* untuk mengakses aplikasi *web*,sedangkan untuk *backends* sendiri adalah server asli yang menampung aplikasi *web* yang sudah dikonfigurasikan sebelumnya.

Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438



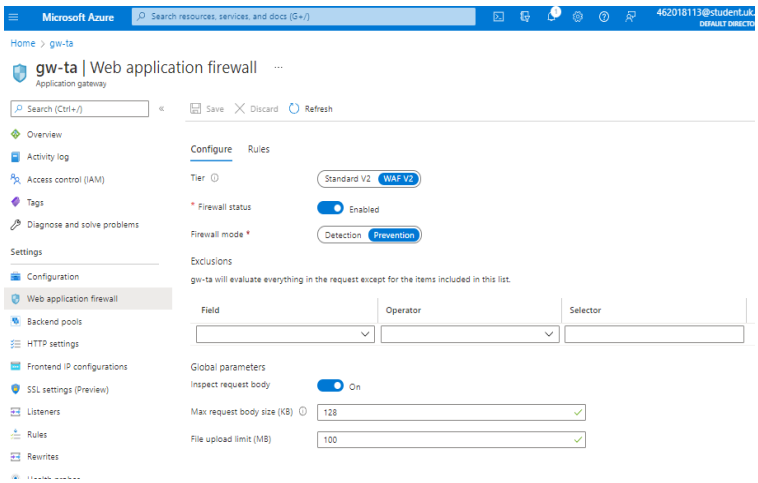
Gambar 6. Konfigurasi *Frontends Application Gateway*

Gambar 6 terdapat pilihan *IP Address* yang akan digunakan,karena aplikasi web akan diakses user melalui internet maka jenis *IP* yang akan digunakan *Public IP Adrress*.



Gambar 7. Konfigurasi *Backend Application Gateway*

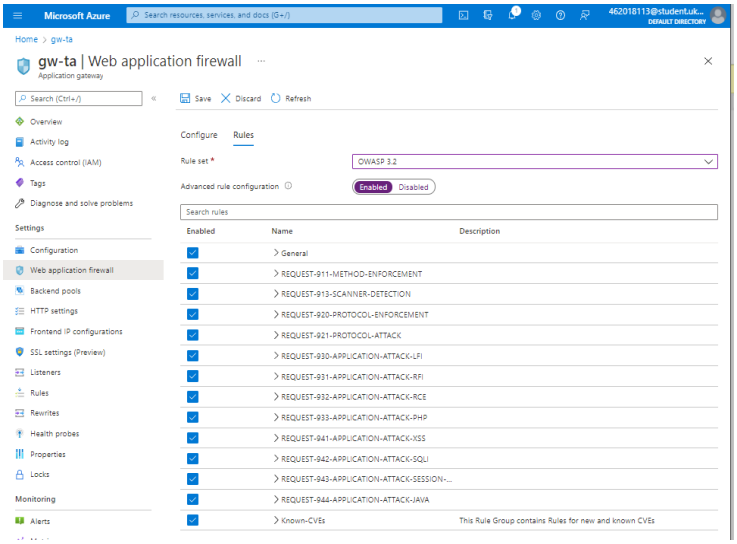
Gambar 7 menunjukkan *backend application gateway* menuju ke resource atau alamat ip dari server yang telah di konfigurasikan sebelumnya.



Gambar 8. Konfigurasi Web *Application Firewall*

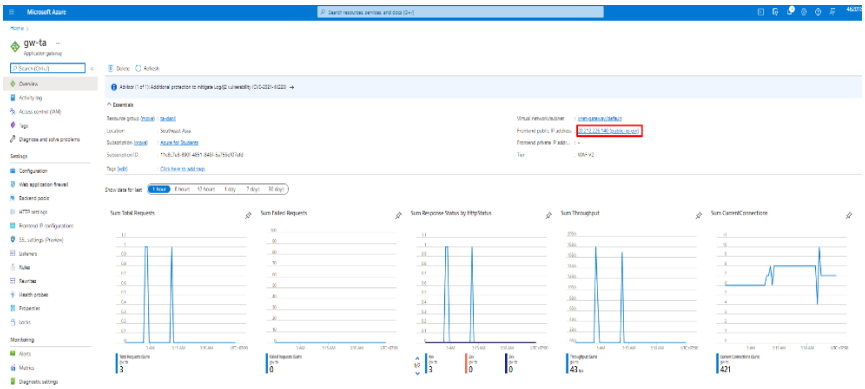
Gambar 8 terdapat isian untuk melakukan konfigurasi *waf* di mana *waf* dapat berjalan dalam dua mode yaitu *Detection* dan *Prevention.* Jika *waf* berjalan dalam mode *Detection* maka *waf* akan memonitor dan mencatat semua ancaman yang terdeteksi tetapi tidak memblokir serangan yang terdeteksi, Jika berjalan dalam mode *prevention* maka *waf* akan memblokir serangan yang dideteksi berdasarkan aturan yang telah di tentukan dan akan dicatat dalam *log.*

Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438



Gambar 9. Konfigurasi *Rule* yang digunakan Web *Application Firewall*

Gambar 9 menunjukkan *rules* yang telah disediakan oleh *Azure* di mana rules tersebut dapat mendeteksi serangan *Owasp Top 10.* Rules tersebut juga dapat di *custom* sesuai kebutuhan sistem. Sistem *web application firewall* akan memfilter setiap traffic lalu lintas data yang masuk, kemudian *web application firewall* akan menentukan apakah traffic yang masuk merupakan ancaman atau bukan. Ketika ancaman terdeteksi sesuai rule yang sudah di *define* maka sistem *waf* akan memblokir serangan dan mencatat dalam *Log Analytic*.

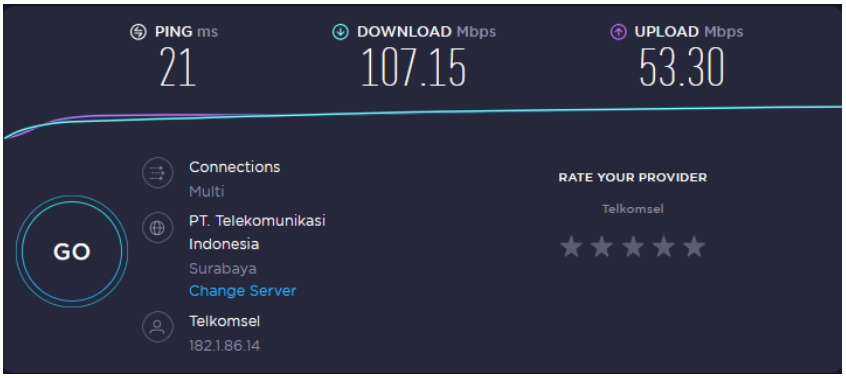


Gambar 10. *Final* konfigurasi waf

Gambar 10 menunjukkan bahwa waf sudah berhasil dipasang dengan alamat *ip public* 20.212.226.140 (public-ip-gw). Untuk memudahkan selama proses pengujian penulis menghubungkan alamat ip tersebut dengan domain *https://waf.ploss-asia.tech/*.

3.3 Pengujian

Sebelum melakukan pengujian, kecepatan akses internet diukur terlebih dahulu untuk memastikan selama proses pengujian mendapatkan hasil yang maksimal.



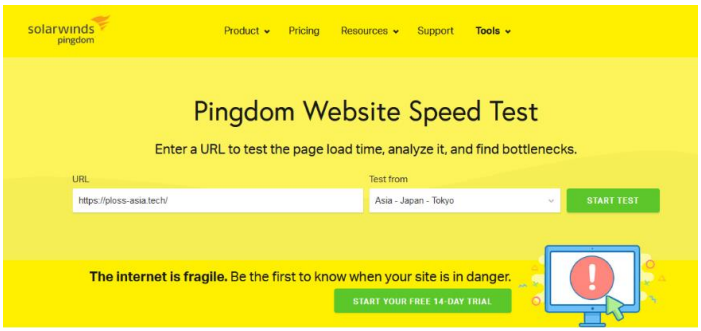
Gambar 11. Kecepatan internet selama proses pengujian

Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438

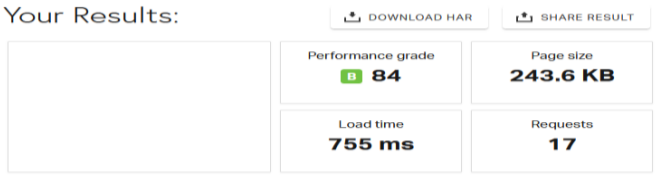
Proses pengujian dilakukan dengan memasukan alamat url atau domain dari website aplikasi erp pada form url yang terdapat pada halaman Pingdom Tools dan GTMetrix. Pada proses pengujian akan dilakukan dua kali pengujian, yaitu sebelum menggunakan waf dan sesudah menggunakan waf.

3.3.1 Pengujian Tanpa WAF

Pengujian akan dilakukan pada server aplikasi *web* pada alamat ip server 128.199.107.215 dengan domain *https://ploss-asia.tech/*

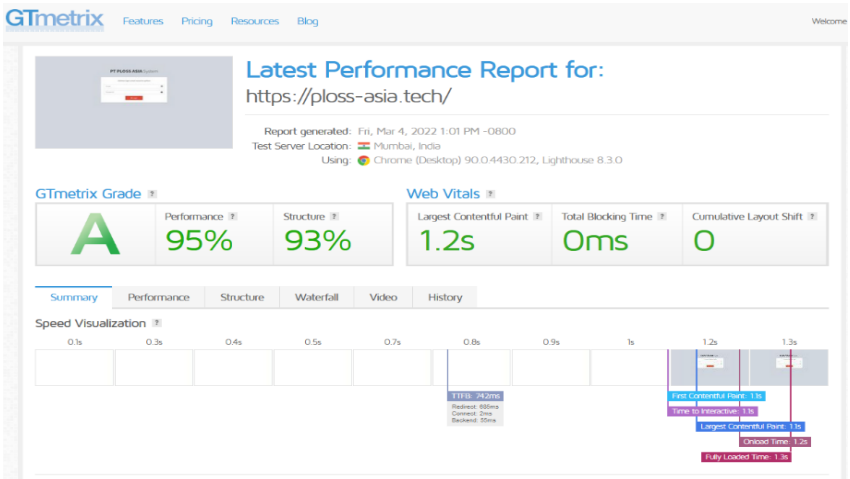


Gambar 12. Halaman depan *Pingdom*

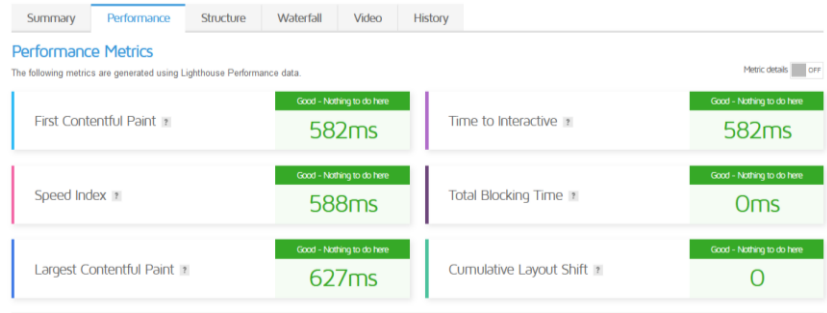


Gambar 13. Hasil Pengujian Pingdom tanpa waf

Dari hasil pengukuran menggunakan Pingdom didapatkan bahwa performance grade mendapatkan nilai 84 dengan grade B dengan loadtime sebesar 755 ms dengan 17 total request. Dengan melihat hasil tersebut dan grade yang didapatkan menunjukan bahwa halaman depan website erp sudah cukup baik.

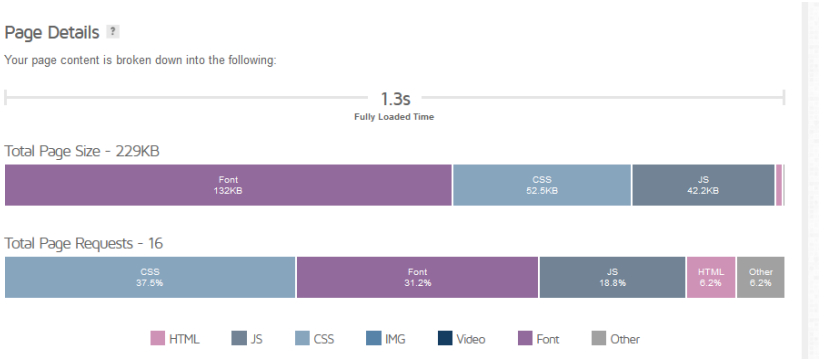


Gambar 14. Hasil Pengujian GTMetrix tanpa waf



Gambar 15. Hasil Performance Metrics tanpa waf

Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438

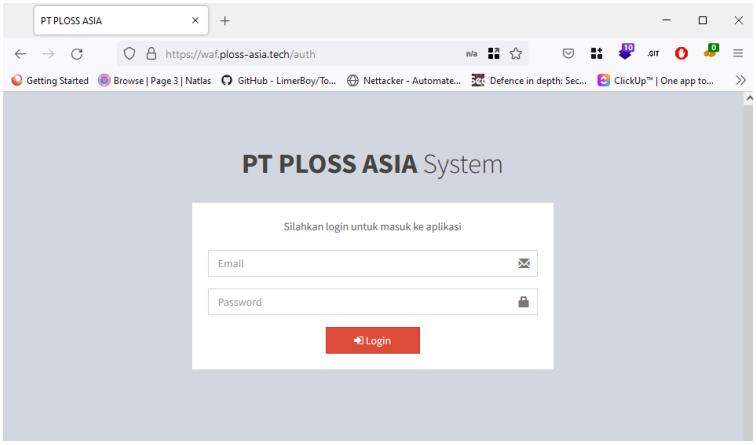


Gambar 16. Hasil Loadtime GTMetrix tanpa waf

Berdasarkan pengujian performa menggunakan GTMetrix website erp memperoleh grade yang sangat baik yaitu grade A dengan performance 95%. Kesimpulan yang dapat diambil dari pengujian menggunakan Pingdom dan GTMetrix adalah tanpa adanya *waf* website erp tersebut sudah memiliki performa yang cukup baik, baik dari segi response time dan performa. Namun belum tentu dengan hasil loadtime dan performance yang bagus menjamin keamanan website tersebut dari serangan *hacker*.

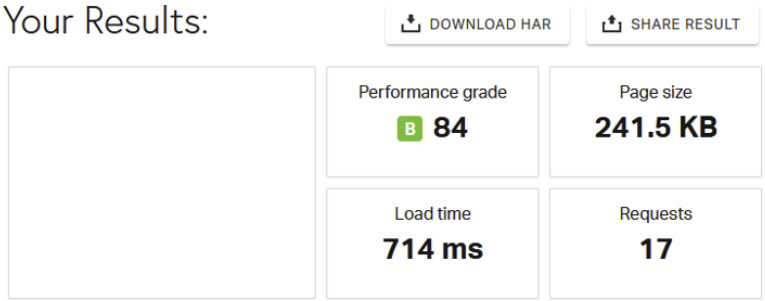
3.3.2 Pengujian Menggunakan WAF

Setelah *waf* diimplementasikan, maka semua *traffic* yang menuju ke alamat ip *128.199.107.215* atau domain https://ploss-asia.tech/ akan dialihkan melalui alamat ip 20.212.226.140 dengan domain https://wf.ploss-asia.tech/ dimana alamat ip tersebut adalah alamat ip *frontend application gateway* yang terhubung ke internet. Sehingga semua *traffic* yang masuk akan melalui *Application Gateway*,seperti pada gambar 4.10.



Gambar 17. Tampilan halaman depan web erp yang menggunakan WAF

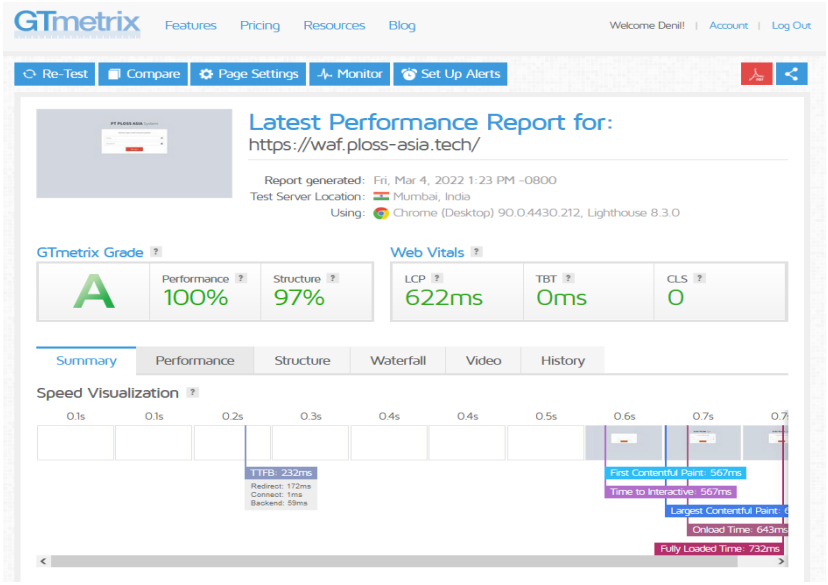
Sebagai perbandingan terhadap hasil yang akan didapatkan maka pengujianyang akan dilakukan sama seperti pada pengujian tanpa *WAF*.



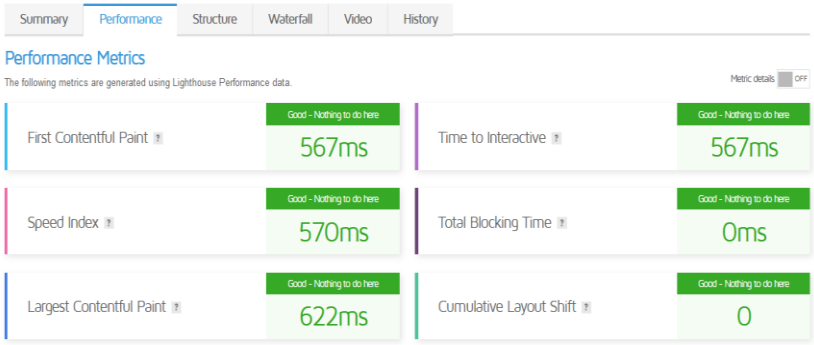
Gambar 18. Hasil Pengujian Pingdom dengan menggunakan waf

Dari hasil pengukuran menggunakan Pingdom didapatkan bahwa performance grade mendapatkan nilai 84 dengan grade B dengan loadtime sebesar 992 ms dengan 17 total request. Dengan melihat hasil pengujian tanpa waf hasil untuk performance grade dan page size cenderung sama, tetapi hasil loadtime yang didapatkan mengalami peningkatan menjadi 992ms dimana ketika menggunakan waf membutuhkan waktu load time yang lebih lama dibandingkan tanpa menggunakan *waf*.

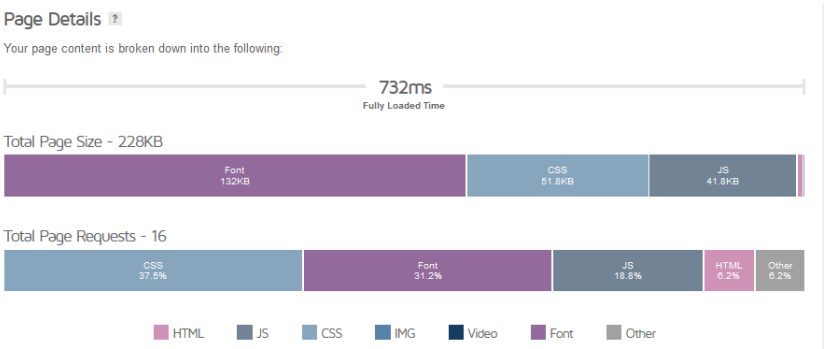
Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438



Gambar 19. Hasil Pengujian GTMetrix dengan menggunakan waf



Gambar 20. Hasil Performance Metrics dengan menggunakan waf



Gambar 21. Hasil l*oadtime* dengan menggunakan waf

Berdasarkan hasil pengujian performa website erp yang sudah menggunakan waf dengan tools Pingdom didapatkan hasil bahwa nilai loadtime pada Pingdom mengalami peningkatan dimana membutuhkan waktu lebih lama Ketika menggunakan waf, sedangkan pada hasil pengujian menggunakan tools GTMetrix ketika menggunakan waf menunjukan hasil yang lebih baik dimana nilai performance mencapai 100% dan nilai structure 97%. Pada nilai performance Metrics Ketika website sudah dipasang waf juga menunjukan hasil yang lebih baik dibandingkan tanpa menggunakan waf dimana Ketika menggunakan waf indicator indicator pada performance metrics menunjukan bahwa waktu yang dibutuhkan untuk meload website menjadi lebih cepat.

Tabel 1. Hasil Pengujian Pingdom

No.	Indikator	Keterangan	
		Tidak Menggunakan *WAF*	Menggunakan *WAF*
1	*Performance Grade*	84 (B)	84 (B)
2	*Page Size*	243.6 KB	241.5 KB

Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438

No.	Indikator	Keterangan	
		Tidak Menggunakan *WAF*	Menggunakan *WAF*
3	Load time	782 ms	714 ms
4	Request	17	17

Berdasarkan tabel 1, menunjukkan bahwa hasil load time Ketika menggunakan waf mengalami peningkatan yang cukup signifikan dimana dengan menerapkan *WAF* website dapat diakses lebih cepat. Sedangkan hasil Performance grade menunjukan hasil yang sama. Hasil Pingdom pada indikator page size ketika menggunakan *WAF* resource yang di download menjadi lebih sedikit karena sebagian besar konten statis seperti gambar, CSS, JavaScript, video, dan lain-lain disediakan dari lokasi server terdekat karena *waf* yang digunakan memiliki fitur *Content Delivery Network* atau *cdn. CDN* adalah jaringan server yang didistribusikan secara global agar membantu sebuah website memiliki kinerja lebih cepat, aman, via jaringan tersebut. Salah satu penggunaan CDN yang paling umum adalah untuk meningkatkan waktu muat sebuah halaman sehingga hal tersebut dapat meningkatkan kecepatan respon suatu website.

Tabel 2. Komparasi Gtmetrix

No.	Indikator	Keterangan	
		Menggunakan *WAF*	Tidak Menggunakan *WAF*
1	*GTmetrix Grade*	A	B
2	*Petformance Score*	95 %	100%
3	*Largest Contentful Pain*	94 %	97 %
4	*Total Blocking time*	1.2s	636ms
5	*Cumulative Layout Shift*	0	0

Berdasarkan tabel 2, menunjukkan bahwa Ketika menggunakan *waf* skor yang didapatkan pada semua indicator menunjukan hasil yang lebih baik, terlebih pada indicator performance score yang mencapai 100%. Pada indicator *LCP* hasil yang didapatkan Ketika menggunakan waf juga meningkat drastis dibandingkan tanpa menggunakan waf, *LCP* sendiri mengukur berapa lama waktu yang diperlukan untuk memuat elemen konten terbesar (misalnya, gambar pahlawan atau teks judul) di halaman..

Gambar 22. Grafik perbandingan waktu *load time*

Pada gambar 22, menunjukkan hasil komparasi GTMetrix yang menunjukan waktu yang dibutuhkan untuk memuat konten *static* dari website ketika tidak menggunakan waf dan menggunakan waf dimana pada bar sebelah kiri tidak menggunakan *waf* dan bar sebelah kanan menggunakan *waf*. Hasil tersebut menunjukan bahwa ketika menggunakan *waf* waktu yang dibutuhkan untuk memuat konten *static* menjadi lebih cepat, walaupun konten yang diakses sama namun karena menggunakan waf, dan waf tersebut memiliki fitur *cdn* yang menyimpan *cache* dari website maka sebagian besar konten diakses dari server terdekat.

4. KESIMPULAN

Dari hasil penelitian dan pengujian yang telah dilakukan, terhadap aplikasi erp yang tidak menggunakan dan menggunakan *WAF* dapat disimpulkan bahwa implementasi *WAF* memiliki pengaruh pada performa website dan loadtime website. Dalam pembuatannya, penulis menyadari bahwa penelitian ini tidak sepenuhnya terlepas dari kekurangan, jika dilihat dari hasil penelitian dan pembahasan, masih ada faktor-faktor lain yang mempengaruhi loadtime dan response time dari website yang diuji. Oleh karena itu penulis memiliki saran untuk peneliti selanjutnya agar dapat menguji faktor-faktor lain yang berpengaruh terhadap load time dan response time, bisa juga

Building of Informatics, Technology and Science (BITS)
Volume 3, No 4, Maret 2022 Page: 593−603
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v3i4.1438

membandingkan hasil loadtime dan response time menggunakan waf yang berbeda. Selain itu penulis sangat menerima kritik dan saran yang sifatnya membangun demi memaksimalkan penggunaan *WAF* yang telah penulis implementasikan.

REFERENCES

[1] Suliman, "Analisis Performa Website Universitas Teuku Umar Dan Universitas Samudera Menggunakan Pingdom Tools Dan Gtmetrix," *Simkom*, vol. 5, no. 1, pp. 24–32, 2020, doi: 10.51717/simkom.v5i1.47.

[2] A. HARSOYO, "ANALISIS WEBSITE PERFORMANCE MILIK KEMENTERIAN DI INDONESIA MENGGUNAKAN METODE PEMBOBOTAN ENTROPI DAN METODE PEMERINGKATAN ELECTRE," *Univ. NEGERI YOGYAKARTA*, 2017.

[3] P. M. R. Gumilang and D. W. Chandra, "Implementasi dan modifikasi WebShell untuk monitoring serangan berbasis website," *Aiti*, vol. 18, no. 1, pp. 54–68, 2021, doi: 10.24246/aiti.v18i1.54-68.

[4] indonesian cloud, "Content Delivery Network (CDN): Pengertian, Fungsi, dan Keuntungan," *indonesiancloud.com*. https://indonesiancloud.com/content-delivery-network-cdn/

[5] f5, "What is a Web Application Firewall (WAF)?," *f5*. f5.com/services/resources/glossary/web-application-firewall (accessed Dec. 01, 2021).

[6] E. M. Anggrahito, Ramadhan Ibrahim, Ahmad Fajri, "Implementasi Web Application Firewall Menggunakan ReverseProxy dan ModSecurity Sebagai Alternatif Pengamanan Aplikasi Web Pada Sektor Pemerintah," 2018, [Online]. Available: http://news.netcraft.com/archives/2018/02/13/february-2018-web-server-

[7] Vhorne, M. Asudbring, DCtheGeek, PRMerger10, and Tfitzmac, abshamsft. TravisCragg-MSFT, pareshverma91, winthrop28, "What is Azure Application Gateway?," *docs.microsoft.com*, 2021. https://docs.microsoft.com/en-us/azure/application-gateway/overview (accessed Jan. 02, 2022).

[8] D. Laksmiati, "IMPLEMENTASI CONTENT DELIVERY NETWORK (CDN) UNTUK OPTIMASI KECEPATAN AKSES WEBSITE," *Univ. Bina Sarana Inform.*, vol. 5, no. 1, pp. 49–56, 2020.

[9] Bangkit Wiguna, W. Adi Prabowo, and R. Ananda, "Implementasi Web Application Firewall Dalam Mencegah Serangan SQL Injection Pada Website," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 245–256, 2020, doi: 10.31849/digitalzone.v11i2.4867.

[10] R. Nurachmad Syaefuddin, "Implementasi Web Application Firewall pada Web Mytra Dashboard dengan Menggunakan Modul ModSecurity," *Tek. Inform. dan Komput. Politek. Negeri Jakarta*, no. April, 2018, doi: 10.13140/RG.2.2.15824.00006.

[11] R. Riska and H. Alamsyah, "Penerapan Sistem Keamanan Web Menggunakan Metode Web Aplication Firewall," *J. Amplif. J. Ilm. Bid. Tek. Elektro Dan Komput.*, vol. 11, no. 1, pp. 37–42, 2021, doi: 10.33369/jamplifier.v11i1.16683.

[12] J. K. Anggreana, "Simulasi keamanan pada aplikasi web dengan web aplication firewall," *Perpust. UNIKOM*, 2014.

[13] F. SETIYAWAN, "Implementasi Firewall Aplikasi Web Untuk Mencegah Sql Injection Menggunakan Naxsi," *Univ. Islam NEGERI SUNAN KALIJAGA*, 2014, [Online]. Available: https://digilib.uin-suka.ac.id/id/eprint/14398/

[14] I. Alamsyah, "Pemanfaatan Jmeter Untuk Pengujian Website Dengan Metode Performance Testing," *Ijns.org Indones. J. Netw. Secur.* -, vol. 1, no. 1, pp. 1–10, 2019.

[15] S. Borso, "Azure's Front Door," *www.sans.org*. https://www.sans.org/blog/azure-s-front-door/ (accessed Jan. 04, 2022).