

Original Research Paper

## Evaluasi Kinerja Web Server Apache menggunakan Protokol HTTP2

Riswandi<sup>1</sup>, Kasim<sup>1</sup>, Muh. Fajri Raharjo<sup>1</sup>

<sup>1</sup> Program Studi Teknik Komputer dan Jaringan, Jurusan Teknik Elektro, Politeknik Negeri Ujung Pandang, Indonesia.

### Article History

**Received:**  
05.01.2020

**Revised:**  
08.02.2020

**Accepted:**  
29.03.2020

**\*Corresponding Author:**  
Riswandi  
**Email:**  
riswandiddpsmc@gmail.com

This is an open access article,  
licensed under: [CC-BY-SA](#)



**Abstrak:** Penelitian ini dilakukan untuk mengukur kinerja dari protokol HTTP/2 dan cara untuk mengaktifkan protokol ini, sehingga protokol ini dapat diimplementasi dan mempercepat kinerja dari berbagai web server. Berdasarkan hasil penelitian ini, ditunjukkan bahwa pengimplementasian protokol ini dapat berjalan pada web server apache versi 2.4.17. Pada pengukuran kinerja yang dilakukan bahwa kecepatan response time yang dihasilkan HTTP/2 dalam mentransfer data lebih kecil dari pada HTTP/1 dapat dilihat dari hasil response time HTTP/1 web server apache sebesar 128,6 ms sedangkan pada HTTP/2 86,3 ms, dan pada web server nginx juga menunjukkan response time HTTP/2 lebih kecil dari pada HTTP/1 yang sebesar 184,3 ms dan 141,6 pada HTTP/2. Dari hasil pengujian response time HTTP/2 menunjukkan performa kecepatan yang lebih cepat jika diterapkan dalam web server apache dari pada menggunakan web server nginx.

**Kata Kunci:** Apache, HTTP/1, HTTP/2, Lama Tindak-Balas, Nginx.

### Apache Web Server Performance Evaluation using the HTTP2 Protocol

**Abstract:** This research was conducted to measure the performance of the HTTP/2 protocol and ways to activate this protocol, so that this protocol can be implemented and accelerate the performance of various web servers. Based on the results of this study, it was shown that implementing this protocol could run on the Apache web server version 2.4.17. In the performance measurement carried out that the speed of response time generated by HTTP/2 in transferring data is smaller than HTTP/1 can be seen from the results of the HTTP/1 web server response time of 128.6 ms while on HTTP/2 86.3 ms, and on the nginx web server also shows HTTP/2 response time is smaller than HTTP/1 which is 184.3 ms and 141.6 on HTTP/2. From the results of HTTP/2 response time testing shows the performance speed is faster if implemented in the Apache web server than using the Nginx web server.

**Keywords:** Apache, HTTP/1, HTTP/2, Nginx, Response-Time.



## 1. Pendahuluan

Saat Kebutuhan akan informasi semakin meningkat, dikarenakan hampir semua pekerjaan saat ini sudah berbasis online, hal ini membuktikan bahwa lintas trafik data pada dunia internet menjadi semakin padat [1]. Peningkatan tersebut tidak terlepas dari peran Transmission Control Protocol/Internet Protocol Suite (TCP/IP) dalam pertukaran informasinya. Hypertext Transfer Protocol (HTTP) salah satu protokol yang masih marak digunakan, yang merupakan protokol yang digunakan untuk request dan response antara client dan server [2]. HTTP/1 yang dirilis pada tahun 1999 yang masih banyak digunakan namun tidak bisa membendung peningkatan user karena setiap koneksi yang dibuka dan terhubung antara client dan server akan selalu ditutup, sehingga membutuhkan koneksi baru setiap request dari client maka akan mengakibatkan transfer data melambat [3]. Dari kondisi yang terjadi maka dikembangkan HTTP/2.

HTTP/2 dirilis pada tahun 2012, untuk mempercepat waktu loading halaman web [4] dengan mengembangkan beberapa fitur seperti mengurangi latensi dengan mengaktifkan permintaan penuh dan multiplexing response, meminimalkan overhead protokol melalui kompresi efisien bidang header HTTP, dan menambahkan dukungan untuk penentuan prioritas permintaan serta server push. Namun pada penerapan HTTP/2 ini masih kurang sehingga penelitian ini dilakukan untuk mengetahui penerapan dan mengukur kinerja dari protokol tersebut dalam pertukaran data antara server dan client menggunakan sistem operasi ubuntu pada web server apache dan nginx.

## 2. Landasan Teori

### 2.1. Server

Server merupakan sebuah tempat yang dipenuhi dengan berbagai macam informasi, dengan tugas utama untuk memberikan sebuah service atau layanan bagi para client yang terhubung dengannya. Terdapat berbagai macam jenis server dengan fungsi yang berbeda-beda, misalnya web server yang digunakan untuk menyimpan data dalam sebuah web, FTP server yang menangani perpindahan file (transfer file), dan mail server yang melayani urusan email para client, serta database server untuk menyimpan berbagai macam data atau file dan lain sebagainya.

### 2.2. Web Server

Web server adalah perangkat lunak yang menjadi tulang belakang dari world wide web (www). Web server menunggu permintaan dari client yang menggunakan browser seperti Netscape Navigator, Internet Explorer, Mozilla, dan program browser lainnya. Jika ada permintaan dari browser, maka web server akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke browser [5].

Salah satu web server yang bersifat open source ialah apache yang digunakan untuk melayani dan melakukan pengaturan fasilitas web, pada umumnya memiliki fungsi untuk memperoleh berkas berisi permintaan (request) dari client melalui web browser, kemudian apache akan memproses data tersebut dengan menghasilkan keluaran (output) yang diinginkan oleh client. Output didapatkan berdasarkan data yang tersimpan dalam database website tersebut [6].

### 2.3. Nginx

Nginx merupakan salah satu software web server yang powerful dan mempunyai performa tinggi, selain itu nginx juga di desain untuk server dengan sumber daya yang kecil. Nginx di rancang untuk mengatasi masalah C10K yang merupakan rancangan web server untuk menangani 10.000 koneksi, Nginx juga mampu melakukan hal ini melalui mekanisme event-based connection-handling, tetapi harus menggunakan sistem operasi yang mendukung mekanisme tersebut [7].

### 2.4. HTTP/2

Hypertext Transfer Protocol yang disingkat HTTP, merupakan suatu protokol yang digunakan oleh World Wide Web (WWW). HTTP mendefinisikan bagaimana suatu pesan bisa diformat dan dikirimkan dari server ke client. Prinsip dari HTTP ini apabila memasukkan sebuah URL pada browser, maka hal tersebut akan melakukan transfer URL ke browser, dan dari URL ini browser mengetahui server mana yang akan dihubungi dan file apa yang diminta kemudian web browser akan mengirimkan perintah HTTP ke web server. Web server selanjutnya akan menerima perintah ini dan melakukan aktivitas sesuai dengan perintah yang diminta oleh web browser. Hasil aktivitas tersebut akan dikirimkan kembali ke web browser untuk ditampilkan kepada user.

## 2.5. HTTP/2

HTTP/2 adalah versi berikutnya dari HTTP dan didasarkan pada Google SPDY, yang dirancang untuk mempercepat loading halaman web versi terbaru ini bertujuan untuk mengurangi latensi dengan mengaktifkan permintaan lengkap dan multiplexing respons, meminimalkan overhead protokol melalui kompresi efisien kolom header HTTP, dan menambahkan dukungan untuk penentuan prioritas permintaan serta push server [7]. Dari beberapa mekanisme yang dikembangkan oleh HTTP/2 ini maka pada penelitian ini akan mengukur kecepatan response time dari protokol ini.

## 2.6. Web Browser

Salah satu software yang digunakan untuk mengetahui keaktifan dari protokol HTTP/2 adalah web browser seperti firefox dan google chrome karena pada browser ini memiliki fitur yang dapat menampilkan protokol yang digunakan oleh web server yang dimuatnya [5]. Web browser ini juga digunakan sebagai user yang akan mengakses web server baik itu pada web server HTTP/1 maupun pada HTTP/2.

## 2.7. Apache J Meter

Apache Jmeter adalah aplikasi open source berbasis java yang dipergunakan untuk melakukan uji fungsional dan mengukur kinerja suatu server perangkat lunak (seperti aplikasi web) [9]. Awalnya tools ini dirancang untuk menguji aplikasi web tetapi diperluas ke fungsi uji lainnya seperti, aplikasi FTP, database server test, java objects, dan beberapa fungsi lainnya. Software ini dapat melakukan simulasi jumlah user yang digunakan dan dapat memberikan analisa dan laporan dari hasil pengujian.

## 2.8. Response Time

Performa yang baik dalam suatu web server juga dapat dilihat dari response time yang dihasilkan, semakin kecil response time yang dihasilkan maka semakin baik pula kinerja web server tersebut [10]. Response time merupakan lama user untuk menunggu sebuah web page untuk menampilkan pada saat pertama kali dikunjungi. Secara umum user akan menginginkan waktu yang cepat dalam mengakses website, namun kecepatan dari response time juga dipengaruhi oleh beberapa faktor seperti jumlah user yang mengakses dan performa jaringan.

- a) Jumlah user  
Semakin banyak jumlah pengguna maka akan semakin lambat. Meskipun demikian, di dalam waktu puncak (peak load periods) jumlah pengguna bisa melebihi rata-rata hasil estimasi tersebut sehingga dapat menurunkan performance.
- b) Performa jaringan/ kecepatan transmisi  
Kecepatan transfer data adalah jumlah data dalam bit yang melewati suatu medium dalam satu detik. Umumnya dituliskan dalam bit per detik (bit per second) dan disimbolkan bit/s atau bps bukan bits/s.

## 3. Metode Penelitian

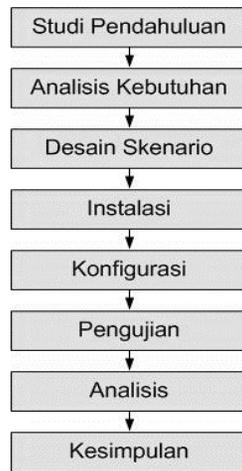
Metode yang dilakukan dalam penelitian ini adalah menggunakan tahapan yaitu studi pendahuluan, analisis kebutuhan, desain skenario, instalasi, konfigurasi, pengujian, analisis dan kesimpulan dapat disajikan dalam Gambar 1.

### 3.1. Studi Pendahuluan

Pada tahap studi pendahuluan dilakukan dengan tujuan untuk pengumpulan data yang akurat, relevan, dan reliable guna memperlancar proses perancangan sistem. Pengumpulan data dilakukan dengan cara membaca dan mengkaji dokumen atau literatur ilmiah yang berkaitan dengan tujuan penelitian.

### 3.2. Analisis Kebutuhan

Tahap kedua analisis kebutuhan adalah perencanaan jaringan yang dibuat serta menentukan hardware dan software yang digunakan dalam penelitian ini. Adapun perangkat keras yang digunakan adalah 2 laptop yaitu server dan client. Tabel 1 menunjukkan kebutuhan perangkat keras dalam membangun sistem HTTP.



Gambar 1. Tahapan Penelitian

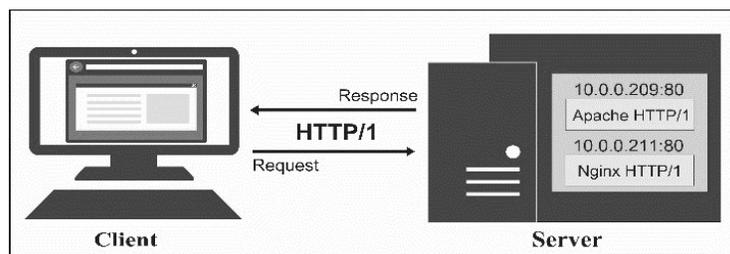
Tabel 1. Perangkat Keras

Perangkat	Processor	RAM	Harddisk
Laptop Asus A455L	Intel® Core™ i3-500U CPU @ 2.00GHz	4GB	500GB
Server PC	Intel® Core™ i5-7400 CPU @ 3.00Ghz	16GB	2TB

Perangkat lunak yang dibutuhkan Kebutuhan perangkat lunak menggunakan sistem operasi Ubuntu server, karena mendukung service untuk komputer server, dan juga karena dukungan integrase dari Apache2, Nginx, OpenSSL, XAMPP, PhpMyAdmin dan pada client perangkat lunak yang digunakan adalah google chrome, mozilla firefox, putty, winSCP dan Apache J Meter.

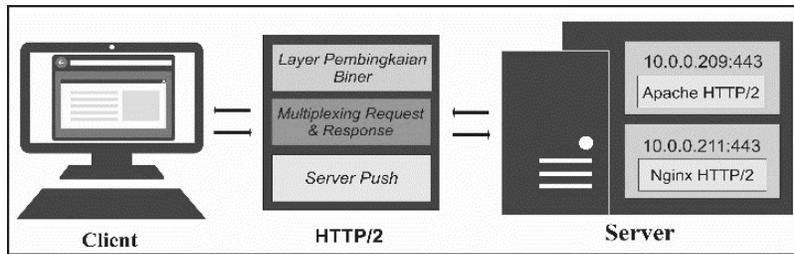
### 3.3. Desain Skenario

Tahap desain adalah tahapan untuk perancangan system yang dibangun dan juga merupakan tahap pendefinisian dari analisis kebutuhan sistem yang menggambarkan skenario yang akan diujikan. Pada penelitian ini terdapat dua skenario yang diujikan, hal ini dilakukan agar mendapatkan hasil pengujian yang dapat dijadikan sebagai bahan analisa. Skenario 1 masih menerapkan protokol HTTP/1 dalam berkomunikasi antara server dan client. Pada skenario 1 diterapkan pada 2 web server yang berbeda yaitu menggunakan apache dan Nginx. Gambar 2 menunjukkan desain skenario 1.



Gambar 2. Desain Skenario 1

Skenario 2 Skenario dua menerapkan protokol HTTP/2 dengan membentuk beberapa mekanisme dalam memaksimalkan kinerja HTTP yaitu layer pembungkaiian biner, multiplexing request/response, dan server push. Gambar 3 menunjukkan desain skenario 2.



Gambar 3. Desain Skenario 2

Skenario ini merupakan pengembangan dari skenario pertama, yang berarti protokol HTTP/2 telah diaktifkan. Sama halnya dengan skenario pertama, skenario ini juga terdapat dua kondisi yang berbeda yakni menggunakan web server apache dan web server nginx.

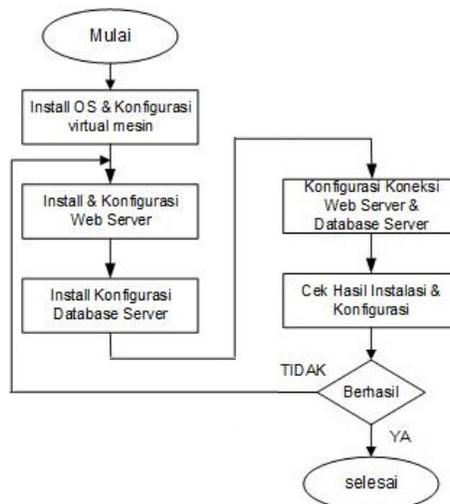
### 3.4. Instalasi dan Konfigurasi

Instalasi dan konfigurasi adalah Tahap yang merupakan lanjutan dari tahap desain dengan mengacu pada desain yang telah dirancang maka diimplementasikan dengan menggunakan perangkat keras (Hardware) dan perangkat lunak (Software). Di dalam tahap ini dilakukan penginstalan dan konfigurasi Apache2 dan Nginx yang digunakan sebagai web server. Gambar 4 menunjukkan urutan instalasi dan konfigurasi secara umum.



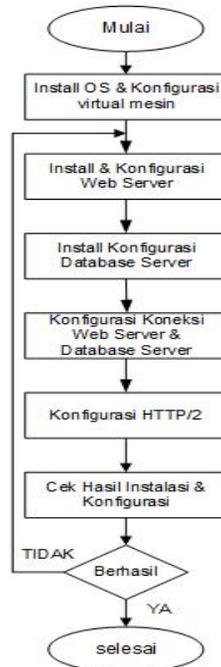
Gambar 4. Instalasi dan Konfigurasi

Pada skenario 1 yang menggunakan protokol HTTP/1 tidak memerlukan banyak instalasi dan konfigurasi. Dari Gambar 5 dapat dilihat flowchart tahapan penyelesaian pada skenario 1.



Gambar 5. Alur Skenario 1

Proses instalasi dan konfigurasi pada skenario 2 pada tahapannya sesuai dengan skenario 1 yaitu mengaktifkan terlebih dahulu HTTP/1 namun dalam pengaktifan HTTP/2 diperlukan tahapan lanjutan seperti mengaktifkan HTTP mode security atau HTTPS dan melakukan instalasi pendukung HTTP/2. Tahapan instalasi dan konfigurasi skenario 2 dapat dilihat pada Gambar 6.



Gambar 6. Alur Skenario 2

### 3.5. Pengujian

Tahapan ini bertujuan untuk mengecek dan memastikan apakah sistem telah berjalan. Dari hasil pengujian sistem, dilanjutkan dengan melakukan analisis terhadap data yang didapatkan dari setiap skenario pengujian sehingga diperoleh informasi yang dapat dievaluasi agar sesuai dengan tujuan yang ingin dicapai. Setiap skenario dilakukan 3 pengujian yaitu pengujian HTTP, pengujian response time dan pengujian latensi.

1. Pengujian keaktifan HTTP  
Pengujian yang dilakukan untuk mengetahui bahwa HTTP/2 yang dikonfigurasi telah aktif. Dalam pengujian dapat dilakukan dengan beberapa metode seperti menggunakan pengujian keycdn yang berbasis online, menggunakan phpinfo dan menggunakan web browser mozilla firefox. Namun pada pengujian ini hanya menggunakan mozilla firefox karena pada dasarnya sama yaitu untuk mengetahui aktif atau tidaknya protokol HTTP/2 dan dengan menggunakan mozilla firefox sudah mewakili tujuan dari pengujian ini.
2. Pengujian Response Time  
Response time diukur untuk mengetahui seberapa cepat web server memberikan respon ketika ada request dari client. Percobaan ini dilakukan pertama pada HTTP/1 dan selanjutnya dilakukan percobaan yang sama pada HTTP/2 untuk melihat perubahan nilai response time. Pengujian ini bertujuan untuk mengukur kinerja dari server push dan multipleksing respon yang dikembangkan dari protokol HTTP/2 ini.
3. Pengujian ini dilakukan menggunakan bantuan tools Apache Jmeter untuk mendapatkan nilai response time setiap skenario. Apache Jmeter mampu mensimulasikan jumlah user yang sangat besar agar dapat mengakses web secara bersamaan. Jumlah user yang diuji dimulai dari 1 user sampai 500 user. Pengujian dilakukan sebanyak 10 kali dengan mengambil nilai rata-rata dari tiap percobaan dengan jeda waktu 5-10 menit tiap percobaan.
4. Apache Jmeter menjadi pilihan tools yang digunakan karena tools ini dapat berjalan pada

sistem operasi manapun dengan Java, sehingga untuk proses instalasinya tergolong Selain itu, data recording yang dihasilkan juga dapat di export dalam berbagai jenis ekstensi file. Namun, nilai response time yang diperoleh melalui tools ini masih dipengaruhi oleh koneksi jaringan, penggunaan memori untuk membuka tools versi GUI yang cukup besar, spesifikasi perangkat yang diuji maupun yang digunakan untuk melakukan pengujian.

#### 5. Pengujian Latensi

Pengukuran latensi bertujuan untuk mengetahui berapa waktu yang dibutuhkan data untuk menempuh jarak dari server menuju client. Pada pengujian ini menggunakan apache jmeter untuk mengetahui delay yang dihasilkan oleh setiap skenario, sama halnya dengan pengukuran response time pada pengujian ini juga diperoleh dari skala 1 sampai 500 user, dengan melakukan 10 kali percobaan dari setiap jumlah user-nya.

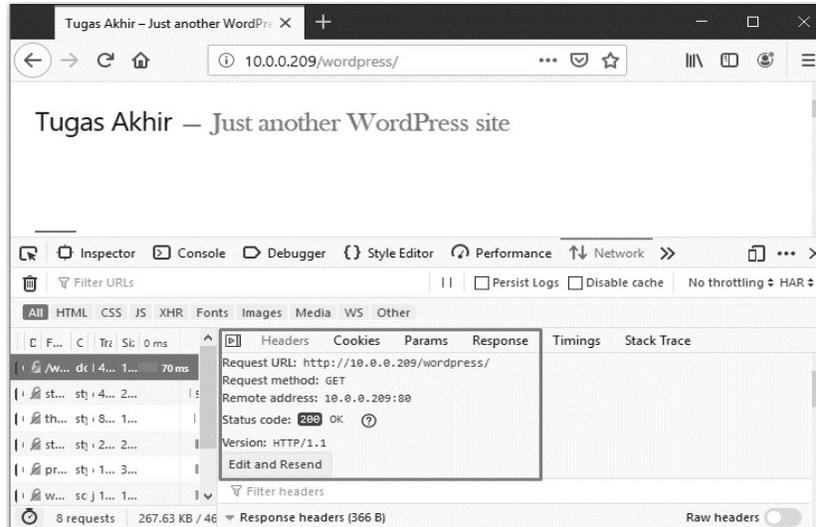
### 3.6. Analisis dan Kesimpulan

Pada tahapan ini dilakukan analisis terhadap hasil pengukuran dan pengujian tiap parameter dengan melihat referensi, selanjutnya dijadikan bahan dokumen produk akhir dalam penelitian ini. Masalah yang ditemukan dalam proses penelitian turut dijadikan sebagai data hasil penelitian dalam mengembangkan penelitian lebih lanjut.

## 4. Hasil dan Pembahasan

### 4.1. Skenario 1

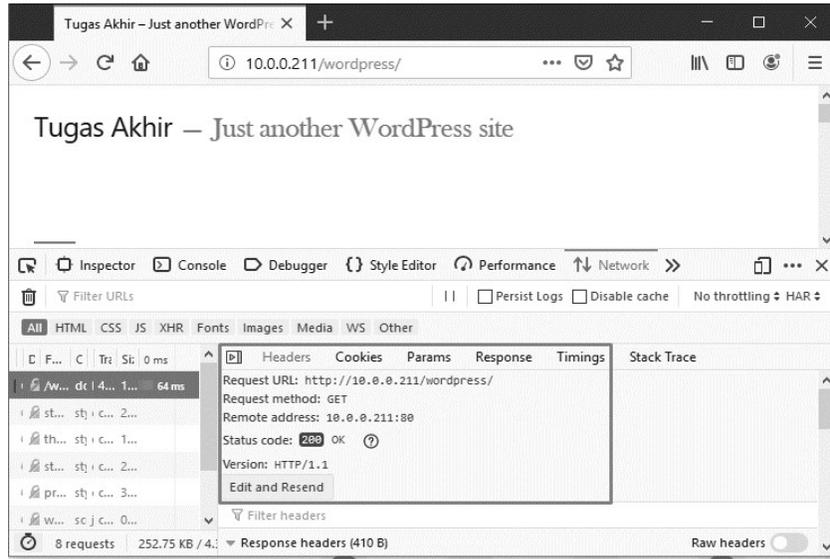
Pada skenario 1 web server apache dan nginx secara otomatis HTTP/1 akan aktif apabila telah melakukan proses instalasi web server baik pada apache maupun nginx. Dengan menggunakan web browser dapat memperlihatkan protokol yang digunakan pada suatu web server. Gambar 7 menunjukkan tampilan web server apache yang menggunakan port 80 yang merupakan port default dari HTTP/1 yang menandakan bahwa HTTP/1 telah aktif. Begitu pula dengan Gambar 8 juga menunjukkan tampilan web server nginx.



Gambar 7. Tampilan web server apache HTTP/1

Pengujian response time dilakukan sebanyak 10 kali didapatkan rata-rata nilai response time dari setiap user dan dituliskan dalam Tabel 2. Pengujian yang dilakukan menggunakan bantuan aplikasi Apache Jmeter. Dari hasil simulasi user tersebut didapatkan nilai response time pada rentang 1-10 user nilai response time sebesar 53,8 ms dan apache dan 72,8 ms dengan penambahan 1 user. Pada rentang 10-100 user response time mengalami kenaikan yang bervariasi, rata-rata kenaikan response time sebesar 18,7 ms pada apache dan 19,6 ms pada nginx dengan penambahan 10 user. Selang antara 100-500 user mengalami kenaikan response time yang cukup besar yaitu sejumlah 408,7 ms pada apache dan 453 pada nginx dengan penambahan 100 user. Tabel 2 menunjukkan rata-rata pengujian

setiap user pada HTTP/1.



Gambar 8. Tampilan Web Server nginx HTTP/1

Tabel 2. Response-Time HTTP/1

No.	User	Response time (ms)	
		Apache	Nginx
1.	1	43,8	72,6
2.	2	52,5	75,2
3.	3	54,3	70,6
4.	4	57,6	72,8
5.	5	55	72,2
6.	6	54,4	72,7
7.	7	55,1	72,6
8.	8	55,4	72,8
9.	9	55,6	73,6
10.	10	54,9	73,6
11.	20	37	79,8
12.	30	36,2	90,6
13.	40	36,6	101,4
14.	50	37,8	160
15.	60	40,6	177
16.	70	42	188,3
17.	80	50,8	211,8
18.	90	94,9	236,3
19.	100	125,3	250,6
20.	200	279,8	296,6
21.	300	332,5	353,4
22.	400	550,5	589
23.	500	755,5	775,6

Pengujian latensi ini bertujuan untuk mengetahui berapa waktu yang dibutuhkan data untuk menempuh jarak dari server menuju client. Sama halnya dengan pengukuran response time pengujian latensi ini dilakukan dengan 10 kali dari setiap user dan didapatkan hasil yang tertulis pada tabel 3. Pada rentang 1 sampai 10 user didapatkan rata-rata latensi sebesar 42,1 pada web server apache dan 51 pada nginx dengan ditambahkan 1 user setiap percobaannya. Rentang 10-100 user didapatkan rata-rata latensi sebesar 33,7 pada web server apache dan 102,7 pada web server nginx dengan penambahan 10 user, disini dapat dilihat pada penambahan 10 user nginx mengalami kenaikan latensi yang cukup besar yaitu 52 sedangkan dari web server apache mengalami penurunan latensi sebesar 8,4. Rentang 100-500 sebesar 297,5 pada web server apache sedangkan pada web server nginx mendapatkan 329 dengan penambahan 100 user. Rata-rata pengukuran latensi setiap user dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengukuran Latensi HTTP/1

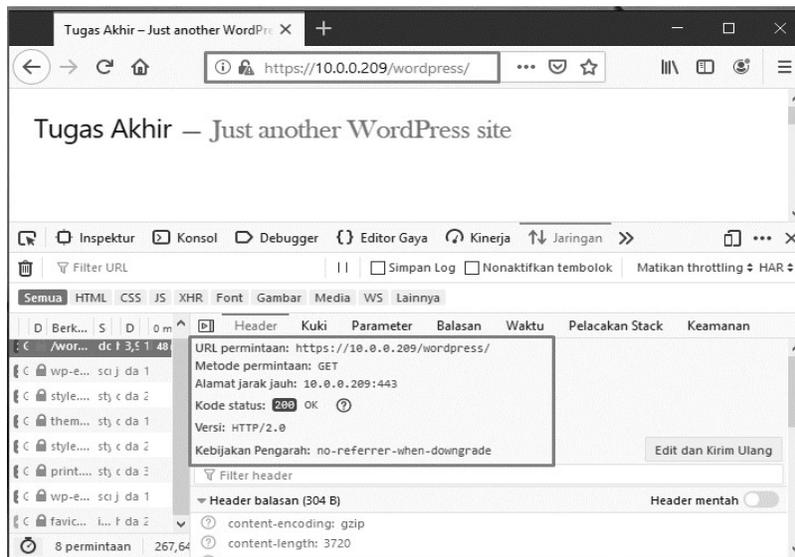
No.	User	Latensi	
		Apache	Nginx
1.	1	29,4	55,7
2.	2	38,6	57,4
3.	3	42,6	56
4	4	44,4	57,3
5	5	44,1	57
6	6	44,3	57,2
7	7	43,7	57,5
8	8	44,3	57,3
9	9	44,8	57,7
10	10	44,8	57,4
11	20	29,5	60,9
12	30	29,8	71,9
13	40	29,4	81,7
14	50	29,7	95,3
15	60	26,6	111,1
16	70	26,4	120,7
17	80	28,2	130,7
18	90	35,8	142,5
19	100	57,7	154,8
20	200	162,6	168
21	300	205	208,9
22	400	440	466,6
23	500	622,4	646,7

#### 4.2. Skenario 2

Skenario 2 ini protokol HTTP/2 sudah telah diimplementasikan, sehingga dalam menjalankannya diharuskan menggunakan Transport Layer Security (TLS). TLS ini merupakan penerus dari SSL. HTTP/2 dapat diakses melalui port 443 yaitu port HTTPS atau berjalan di atas HTTPS, maka untuk mengaktifkan protokol HTTP/2 terlebih dahulu mengaktifkan HTTPS atau melakukan instalasi SSL. Seperti pada skenario pertama pada skenario 2 ini menggunakan 2 web server yang berbeda yaitu web server apache dan web server nginx. Pada masing-masing web server telah terinstal wordpress sebagai beban web server serta aplikasi pendukung web server yaitu MariaDB dan PHP7.2. Aplikasi yang diterapkan pada skenario ini sama persis dengan skenario 1 yang membedakan hanya dari segi protokol, skenario 1 menerapkan protokol HTTP/1 sedangkan pada skenario 2 menerapkan protokol

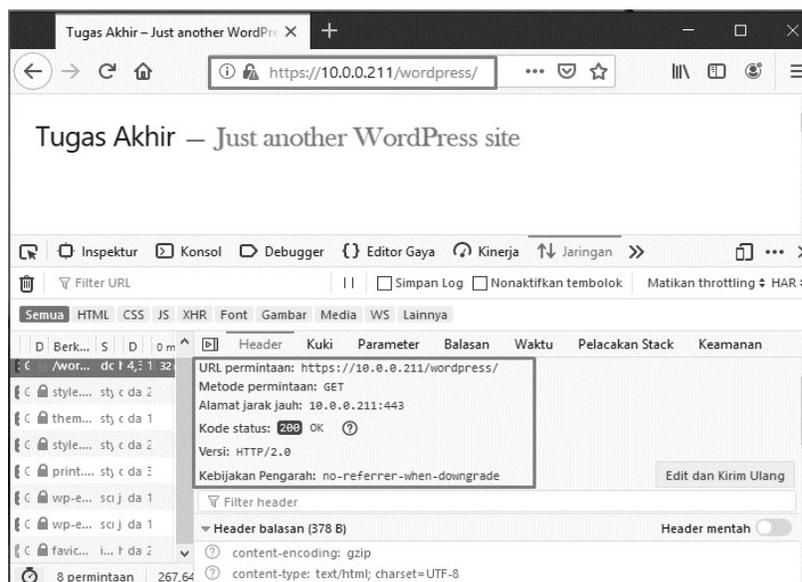
## HTTP/2.

Pada web server apache protokol HTTP/2 ini hanya akan berjalan pada web server apache versi 2.4.17 atau versi yang lebih terbaru, karena versi sebelumnya belum mendukung protokol HTTP/2. Adapun pada skenario ini web server yang digunakan merupakan web server yang sama pada skenario pertama. Pada pengimplementasian protokol HTTP/2 dapat dilihat dengan menggunakan web browser Mozilla Firefox pada menu header. Seperti pada Gambar 9 dalam menu header menampilkan alamat URL yaitu 10.0.0.209/wordpress/, menampilkan port yang digunakan yaitu port 443 yang merupakan port dari HTTPS, dan menunjukkan versi HTTP yang digunakan ialah HTTP/2.



Gambar 9. Tampilan Web Server Apache HTTP/2

Pada web server nginx versi yang digunakan mengharuskan versi 1.9.5 atau versi yang lebih terbaru, karena versi sebelumnya belum mendukung protokol HTTP/2. Pengimplementasian protokol HTTP/2 pada web server nginx dapat dilihat dengan menggunakan web server mozillah firefox.



Gambar 10. Tampilan Web Server Nginx HTTP/2

Seperti pada Gambar 10 menampilkan alamat URL yaitu 10.0.0.211/wordpress/, menampilkan port yang digunakan yaitu port 443 yang merupakan port dari HTTPS, dan menunjukkan versi HTTP yang digunakan ialah HTTP/2. Gambar 10 menunjukkan tampilan web yang menggunakan HTTP/2 dalam web server nginx.

Pengujian response time pada skenario 2 sama pada pengujian pada skenario 1 yaitu dengan mengambil rata-rata dari 10 kali pengujian dalam setiap usernya. Dari hasil simulasi didapatkan nilai response time pada rentang 1-10 user cukup bervariasi dengan rata-rata nilai response time sebesar 39,5 ms pada web server apache dan 76,5 ms pada nginx dengan penambahan 1 user. Pada rentang 10-100 user response time mengalami penurunan pada web server apache dengan nilai response time sebesar 29,4 sedangkan pada nginx mengalami kenaikan dengan sebesar 134,9 ms dengan penambahan 10 user. Selang antara 100-500 user mengalami kenaikan response time yang lebih besar yaitu sejumlah 237,4 ms pada web server apache dan 284,7 pada nginx dengan penambahan 100 user. Nilai rata-rata dari setiap pengujian dapat dilihat pada Tabel 4.

Tabel 4. Nilai Response-Time HTTP/2

No.	User	Response time (ms)	
		Apache	Nginx
1.	1	26,7	63,2
2.	2	38,4	78,3
3.	3	40,6	78,2
4	4	42,8	74,4
5	5	42,5	81,2
6	6	41,5	79,9
7	7	41,8	80,2
8	8	41,5	81,8
9	9	42,6	75,8
10	10	41	71,8
11	20	27	71,5
12	30	27,6	74,5
13	40	27,7	84,6
14	50	27,8	127,7
15	60	25,2	159,6
16	70	26,2	170
17	80	26,9	181,8
18	90	32,1	198,7
19	100	32,5	209,6
20	200	148,4	230,3
21	300	184,6	253,1
22	400	200,5	285,7
23	500	621	445,1

Pengujian latensi pada skenario 2 seperti pada skenario pertama, pengujian latensi ini dilakukan dengan 10 kali dari setiap user dan didapatkan hasil yang ditulis pada Tabel 5.

Pada pengujian rentang 1 sampai 10 user didapatkan rata-rata latensi sebesar 39,9 pada web server apache dan 55,5 pada nginx dengan penambahan 1 user setiap percobaannya. Selang antara 10 sampai 100 user didapatkan 29,4 pada apache disini mengalami penurunan jumlah latensi sebesar 10,5 dan mengalami kenaikan sebesar 101 pada nginx dengan penambahan 10 user. Rentang 100 sampai 500 user didapatkan rata-rata sebesar 237,4 pada apache dan 274,9 pada nginx dengan penambahan 100 user setiap percobaan.

Tabel 5. Hasil Pengukuran Latensi HTTP/2

No.	User	Latensi	
		Apache	Nginx
1.	1	26,7	49,3
2.	2	38,4	56,6
3.	3	40,6	55
4	4	42,8	55,7
5	5	42,5	55,8
6	6	41,5	56,6
7	7	41,8	56,5
8	8	41,5	56,3
9	9	42,6	56,9
10	10	41	56,4
11	20	27	58,8
12	30	27,6	70,5
13	40	27,7	80,8
14	50	27,8	93,4
15	60	25,2	109,4
16	70	26,2	118,4
17	80	26,9	129,7
18	90	32,1	141,1
19	100	32,5	151,6
20	200	148,4	165,9
21	300	184,6	206,5
22	400	200,5	206,5
23	500	622,4	646,7

## 5. Kesimpulan dan Saran

Berdasarkan proses perancangan, konfigurasi sampai pengujian sistem yang dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

- 1) Protokol HTTP/2 berjalan melalui HTTPS maka dalam mengaktifkan protokol HTTP/2 ini terlebih dahulu melakukan instalasi SSL pada web server yang digunakan. Jika protokol HTTP/2 diimplementasikan pada web server apache maka versi apache yang digunakan mengharuskan versi apache 2.4.17 karena versi sebelumnya tidak mendukung, begitu pula pada web server nginx versi yang digunakan ialah versi nginx 1.9.5.
- 2) Dari hasil pengujian response time yang dilakukan, terbukti bahwa kecepatan response time yang dihasilkan HTTP/2 dalam mentransfer data lebih kecil dari pada HTTP/1. HTTP/2 menunjukkan performa kecepatan response time yang lebih cepat jika diterapkan dalam web server nginx dari pada menggunakan web server apache.

Dalam perancangan sistem ini masih terdapat kekurangan sehingga dapat dilakukan perbaikan atau pengembangan selanjutnya.

- 1) Mengimplementasikan pada server langsung, tanpa menggunakan server virtual.
- 2) Pengujian setiap skenario dilakukan oleh user langsung tanpa bantuan tools.
- 3) Melakukan penelitian dengan menggunakan sistem operasi dan web server yang berbeda.

## Daftar Pustaka

- [1] A. Tantoni, "Analisis Komparasi Performa Jaringan Komputer pada Implementasi IPv4 dan

- IPv6,” *Jurnal Informatika & Rekayasa Elektronika*, vol. 1, no.2, pp. 55–61, 2018.
- [2] S. Sukaridhoto, “Jaringan Komputer I,” 2014. [Online]. Available <http://dphoto.lecturer.pens.ac.id/publications/book/2014/Dphoto-JaringanKomputer1>. [Accessed: Nov 7, 2019].
- [3] S. P. S. Engku, and W. Chandara, “*Analisis Transfer Rate View HTTP Versi 1.1 dan HTTP Analisis Transfer Rate View HTTP Versi 1.1 dan HTTP*”. Salatiga: Universitas Kristen Satya Wacana, 2015.
- [4] M. A. Abdillahi, U. Dossetov and A. Saqib, “Open Access Performance evaluation of HTTP/2 in Modern Web and mobile Devices,” *American Journal of Engineering Research (AJER)*,”vol. 1, no. 4, pp. 40–45, 2017.
- [5] D. Lukitasari, dan A. F. Oklitas, “Analisis Perbandingan Load Balancing Web Server Tunggal Dengan Web server Cluster Menggunakan Linux Virtual Server,” *Jurnal Generic*. Vol. 5, no.2 pp. 31-31, 2010.
- [6] I. Syafitri, “Pengertian Apache Beserta Fungsi, kelebihan dan kekurangan Apache yang perlu Anda Ketahui,” 2018. [Online] Available: <https://www.nesabamedia.com/pengertian-apache/>. [Accessed: Nov 7, 2019].
- [7] B. Parmaza, “Berkenalan dengan HTTP (Hypertext Transfer Protocol)”, 2015. [Online] Available: <http://itjambi.com/berkenalan-dengan-http-hypertext-transfer-protocol/>. [Accessed: Dec, 2019].
- [8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. BernersLee, “RFC 2068: Hypertext Transfer Protocol-HTTP1.1,” 1997, [Online] Available: <http://www.cis.ohio-state.edu/htbin/rfc/rfc2068.txt>. [Accessed: Dec 2019].
- [9] G. Carlucci, L. De Cicco, and S. Mascolo, “HTTP over UDP: an experimental investigation of quic”, *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pp. 609-614, 2015.
- [10] Y. Liu, T. Song, and L. Liao, “TPII: tracking personally identifiable information via user behaviors in HTTP traffic.” *Frontiers of Computer Science*, vol. 14, 2020.