

Perancangan dan Analisis Kinerja *Web Server* di Pusat Data Sistem Teknologi Informasi BP Batam

Haditia Ramadhan*, Arta Uly Siahaan, S.Pd, M.Pd **

* Teknologi Rekayasa Multimedia, Politeknik Negeri Batam

** Politeknik Negeri Batam

Article Info

Riwayat Artikel:

Diterima 29 Desember 2023

Direvisi 28 Februari 2024

Diterima 5 Maret 2024

Direvisi 9 Mei 2024

Disetujui 14 Mei 2024

Kata Kunci:

Merancang

Load testing

Kinerja

ABSTRAK

Pusat Data Sistem Teknologi Informasi (PDSI) BP Batam merupakan salah satu bagian dari BP Batam. Pada instansi tersebut terdapat pendataan pengunjung, pendataan ruangan, pendataan anggota, pendataan kegiatan, dan sebagainya. Namun pada pendataan pengunjung memiliki tempat penyimpanan yang berbeda. Pada meja resepsionis masih menggunakan kertas dan pena untuk mendata pengunjungnya. Dan pada lantai 2 menggunakan *website* yang versi lama untuk mendata pengunjung. Sehingga PDSI BP Batam membuat *website dashboard* yang belum dihosting dan *website* data pengunjung terbaru untuk menggantikan yang lama. Oleh karena itu, dirancanglah *web server* dengan metode *Network Development Life Cycle* (NDLC) dengan tujuan untuk menghosting kedua *website*. *Web server* dapat dikonfigurasi atau dilihat datanya oleh pegawai tertentu. *Web server* tersebut juga dianalisis performanya menggunakan metode *Load testing* untuk memastikan bahwa *web server* mampu menangani beban yang diterima.

1. PENDAHULUAN

Di instansi BP Batam (Badan Pengusaha) Batam, terdapat bagian Pusat Data Sistem Teknologi Informasi (PDSI) BP Batam. Pengelolaan data pengunjung di Pusat Data Sistem Teknologi Informasi BP Batam masih terbagi. Pada meja resepsionis, perekaman data pengunjung masih menggunakan metode mencatat di buku pengunjung. Hal ini menyebabkan data terpisah dan menumpuk ketika diarsipkan. Dan pada lantai 2 Pusat Data Sistem Teknologi Informasi BP Batam, juga terdapat sebuah *website* versi lama yang digunakan untuk merekam data pengunjung. Yang mana *website* versi lama ini menyebabkan komputer tersebut tidak dapat terhubung di internet. Oleh karena itu PDSI BP Batam membuat sebuah *website* data pengunjung yang baru. Dan PDSI BP Batam membuat sebuah *website dashboard* yang digunakan untuk memudahkan pekerjaan pegawai PDSI BP Batam. *Website* tersebut berguna untuk pendataan ruangan, kegiatan, nota dinas dan anggota oleh pegawai PDSI BP Batam.

Website dashboard dan *website* data pengunjung yang terbaru yang dibuat oleh PDSI BP Batam, sudah berhasil dijalankan menggunakan XAMPP. Dimana XAMPP berupa server lokal yang dapat menyimpan berbagai jenis data-data untuk *website* yang sedang dikembangkan. Dengan kata lain XAMPP menjalankan *website* tersebut seperti *web hosting* pada umumnya, namun *website* tersebut tidak dapat diakses selain dari perangkat itu sendiri. Sehingga para pegawai tidak dapat mengakses *website* untuk memasukkan data pengunjung, seperti yang terjadi pada *website* di Lantai 2. *Website* perlu dihosting supaya dapat dibuka oleh para pengguna. Tanpa dihosting, *website* tidak dapat dijalankan.

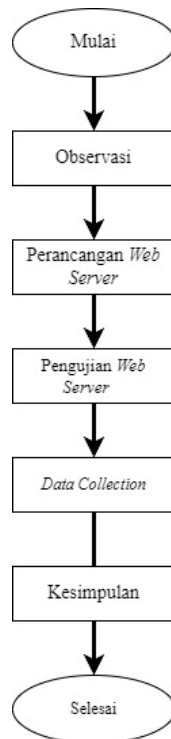
Oleh karena itu, dirancang sebuah *web server* untuk menghosting *website dashboard* dan *website* data pengunjung yang terbaru. Kedua *website* memiliki kebutuhan yang berbeda untuk dihosting didalam *web server*. Untuk mengurangi orang luar mengakses *website*, *web server* dikonfigurasi. Dimana hanya orang yang terhubung pada jaringan yang sama yang dapat mengakses *website*. *Web server* juga akan diberikan aplikasi tambahan untuk pegawai yang akan memantau dan mengkonfigurasi *web server* tanpa perlu mendatangi *web server* secara langsung. *Web server* sendiri adalah bentuk server khusus yang digunakan untuk menyimpan

halaman *website*[1]. *Web server* juga dapat dijelaskan bahwa *web server* sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang di kenal dengan web browser, dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML[2]. Untuk mengetahui bagaimana kinerja *web server*. Maka dilakukanlah analisis kinerja *web server* yang telah dikonfigurasi melalui aplikasi Jmeter. Karena faktor terpenting dalam menentukan sebuah kualitas suatu perangkat lunak yaitu dari performa aplikasi atau sistem tersebut[3]. Apache Jmeter sendiri merupakan aplikasi *open-source* yang dapat digunakan untuk pengujian kinerja seperti *load testing* dengan menyimulasikan *user* yang banyak dalam mengakses sebuah *website*[4].

Metode penelitian yang digunakan ialah metode *Network Development Life Cycle* untuk perancangan *web server* dan metode *load testing* untuk analisis kinerja *web server*. Metode ini umum digunakan dalam merancang sebuah server seperti pada penelitian (1) Konfigurasi *Hosting Server* Menggunakan Centos 7 pada Dinas Komunikasi dan Informatika Kabupaten Pesawaran[5]; (2) Perancangan Owncloud Storage Server Berbasis Ubuntu 20.04 pada PT.Harrisma Global Technologies Jakarta[6]. Pada metode analisis yang umum digunakan berupa metode *load testing* seperti pada penelitian (1) Analisis Kinerja *Website* Pemerintah Kota Pekalongan [4]; (2) Pengujian Aplikasi Menggunakan Metode *Load testing* dengan Apache Jmeter pada Sistem Informasi Pertanian[7].

2. METODE PENELITIAN

Pada penelitian ini, metode yang digunakan berupa metode *Network Development Life Cycle* (NDLC) untuk merancang *web server* dan menggunakan metode *load testing* untuk menguji kinerja *web server*. Metode NDLC adalah suatu siklus hidup pengembangan sistem jaringan komputer yang bersifat komprehensif dengan tingkat integritas yang kuat dari sejumlah tahapan yang harus dilakukan untuk mencapai sebuah keluaran yang akurat, valid dan memiliki produktivitas yang tinggi[6]. Dan metode *load testing* adalah metode untuk mengukur kinerja web terhadap tekanan kinerja ketika banyak pengguna atau pengunjung mengakses *website* secara bersamaan [4]. Tahapan pengerjaan penelitian ini seperti pada gambar 1.



Gambar 1. Tahap Penelitian

A. Tahap Observasi

Pada tahap ini, data dikumpulkan sesuai kebutuhan. Dimana mencari informasi *web server* apa yang cocok untuk menjalankan *website* yang diberikan. Dan juga melihat versi *software* apa yang dibutuhkan untuk kedua *website*. Spesifikasi *hardware* yang digunakan sesuai pada tabel 1.

Tabel 1. *Hardware*

<i>Hardware</i>	<i>Spesifikasi</i>
<i>Processor</i>	AMD Ryzen 7 3750h 1 core 2.3Ghz
<i>RAM</i>	4 GB DDR4
<i>Storage</i>	HDD 50 GB

B. Perancangan *Web server*

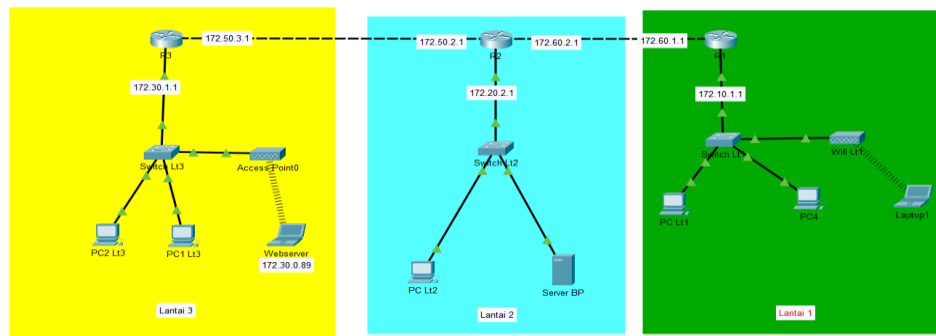
Tahap ini dimana *web server* dirancang. Metode yang digunakan untuk merancang *web server* adalah metode *Network Development Life Cycle* (NDLC). Tahapan metode NDLC terdiri dari analisis, desain, prototipe simulasi, implementasi, monitoring, dan *management*.

a. Tahap Analisis

Pada tahap ini data observasi dikumpulkan serta menentukan perangkat keras dan perangkat lunak yang digunakan. Tahap ini mencakup versi *software* apa yang digunakan oleh *website* yang akan dijalankan serta sistem operasi apa yang digunakan untuk menjalankan *web server*.

b. Desain dan Prototipe/Simulasi

Tahap ini dimana topologi jaringan yang akan digunakan dibuat serta percobaan menjalankan *website* menggunakan XAMPP untuk simulasi. Menggunakan XAMPP, dapat mempermudah dalam merancang *web server* ketika ada masalah dalam konfigurasi *website*.



Gambar 2. Topologi Jaringan

c. Implementasi

Pada tahap ini, *web server* dibuat dengan menginstall sistem operasi Linux pada Virtual Box. Linux yang digunakan ialah Ubuntu 22.04. Di tahap ini, *software* yang sudah ditentukan dipasang serta konfigurasi *web server* untuk menjalankan *website*. Hasil prototipe simulasi dari langkah sebelumnya dijadikan referensi dalam konfigurasi *web server*. *Software* yang dipasang termasuk PHP, MySQL Database, Apache, serta Webmin untuk memudahkan pegawai tertentu mengkonfigurasi *web server*.

d. *Monitoring* dan *Management*

Pada tahap ini *web server* yang sudah berhasil terbuat akan dipantau ketika berjalannya. Serta pada tahap *management* salah satu *user* dapat mengkonfigurasi *web server* sesuai kebutuhan melalui akses khusus. Seperti mengganti database dan konfigurasi *website*. Salah satu cara dalam melihat riwayat akses ialah membuka folder *access.log* sesuai pada gambar 3.

```

adminweb@web:/var/log/apache:$ cat access.log.1
172.18.29.26 - - [16/Mar/2023:09:08:53 +0700] "GET / HTTP/1.1" 200 3011 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36"
172.18.29.26 - - [16/Mar/2023:09:08:53 +0700] "GET /assets/css/custom-style.css HTTP/1.1" 200 6213 "http://172.30.0.89/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36"
172.18.29.26 - - [16/Mar/2023:09:08:53 +0700] "GET /assets/js/plugins-init.js HTTP/1.1" 200 3505 "http://172.30.0.89/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36"
172.18.29.26 - - [16/Mar/2023:09:08:53 +0700] "GET /assets/images/logo.jpg HTTP/1.1" 304 306 "http://172.30.0.89/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36"
172.18.29.26 - - [16/Mar/2023:09:08:53 +0700] "GET /assets/js/page-scripts.js HTTP/1.1" 200 7637 "http://172.30.0.89/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36"
172.18.29.26 - - [16/Mar/2023:09:08:53 +0700] "GET /assets/images/background.jpg HTTP/1.1" 304 307 "http://172.30.0.89/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36"
::1 - - [16/Mar/2023:09:09:01 +0700] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.52 (Ubuntu) OpenSSL/3.0.2 (internal dummy connection)"
::1 - - [16/Mar/2023:09:09:02 +0700] "OPTIONS * HTTP/1.0" 200 126 "-" "Apache/2.4.52 (Ubuntu) OpenSSL/3.0.2 (internal dummy connection)"
172.18.29.40 - - [16/Mar/2023:09:09:27 +0700] "GET / HTTP/1.1" 200 3010 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36"
172.18.29.40 - - [16/Mar/2023:09:09:27 +0700] "GET /assets/css/custom-style.css HTTP/1.1" 200 6213 "http://172.30.0.89/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36"

```

Gambar 3. Access.log

C. Pengujian *Web server*

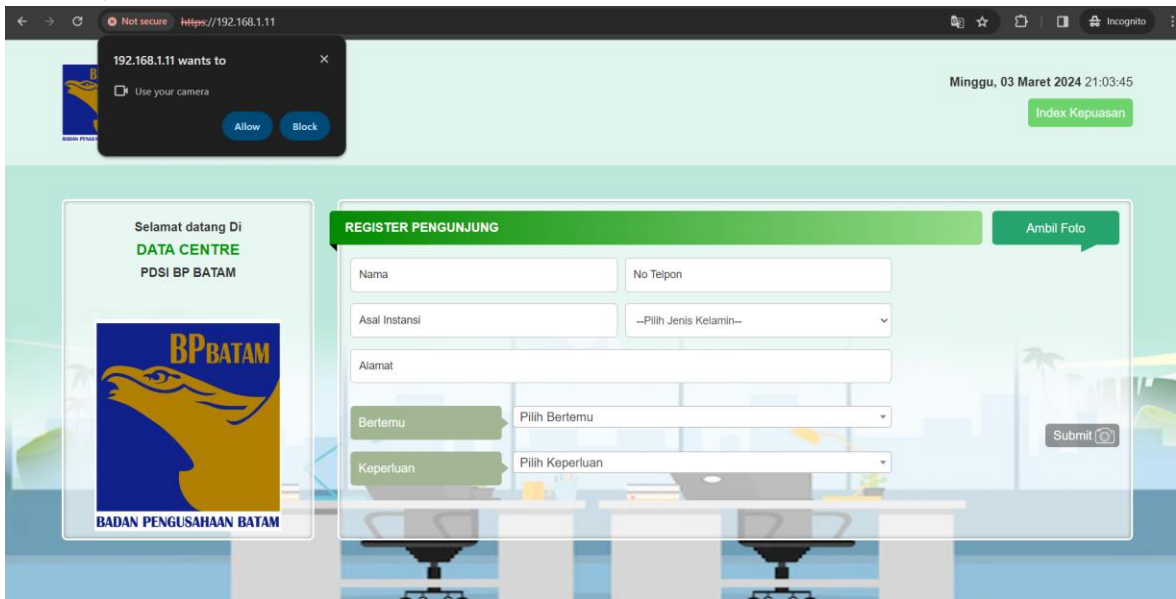
Pada tahap ini *web server* yang dirancang di analisis kinerjanya. Seperti seberapa cepat dan seberapa banyak jumlah *error* yang di alami oleh *web server* ketika diuji. Bagaimana *resource* yang digunakan *web server*. Metode pengujian yang digunakan ialah metode *load testing*. Pengujian tersebut menggunakan aplikasi Apache Jmeter dengan konfigurasi HTTP *Request* dengan metode GET dan POST. Pada pengujian menggunakan HTTP *Request* metode POST akan menggunakan sebuah gambar atau foto berukuran 100KB berformat PNG dan memasukkan data tamu kedalam *database*. Hasil Analisis tersebut akan ditarik kesimpulan dari data yang diambil. Data yang diambil dari pengujian berupa.

1. Jumlah *user* virtual berupa jumlah *request*. Jumlah user digunakan dari lebih kecil menuju lebih besar untuk melihat bagaimana *web server* menangani permintaan dari yang sedikit menjadi lebih banyak.
2. *Throughput* berupa kecepatan *request* yang ditangani dalam hitungan waktu.
3. *Average response time* yang mana berupa rata-rata waktu yang ditempuh oleh setiap *request*.
4. Standar Deviasi yang mana digunakan untuk melihat keakuratan nilai melalui proses matematis [3]. Jika standar deviasi lebih kecil dari *average*, maka variasi data lebih dekat dengan *average* [3].
5. *Error rate* berupa berapa persen *request* yang bermasalah atau gagal.

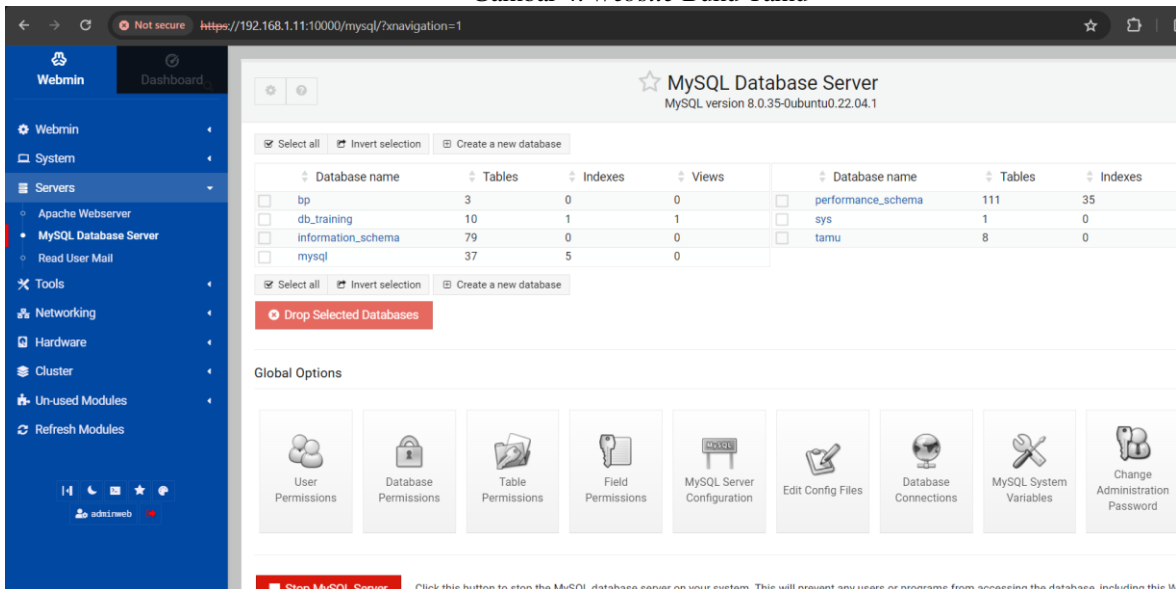
3. HASIL DAN PEMBAHASAN

3.1. Hasil Rancangan

Hasil dari perancangan berupa *web server* berbasis Linux Ubuntu 22.04 untuk *ghosting website* buku tamu terbaru seperti pada gambar 4 dan *website dashboard* sesuai yang disediakan. *Web server* dapat diubah konfigurasi melalui browser atau secara langsung. Webmin juga dipasang untuk memudahkan konfigurasi melalui browser dengan akun khusus seperti pada gambar 5 yang mana sudah tersambung kedalam *web server*.



Gambar 4. Website Buku Tamu



Gambar 5. Webmin

3.2. Analisis Kinerja

Web server yang sudah dijalankan diuji kinerjanya menggunakan Jmeter dengan metode *load testing* dimana metode tersebut mengirimkan jumlah *request* yang ditentukan kepada *website web server*. Pengujian yang dilakukan menggunakan *HTTP Request* yang mana dengan menggunakan metode yaitu GET dan POST. Pengujian *load testing* dengan GET akan dilakukan 5 kali untuk setiap jumlah *user* dengan *ramp-up* periode 5 detik.

3.2.1 Pengujian Load testing metode GET

Pengujian berikut menggunakan *HTTP Request* dengan metode GET yaitu metode untuk meminta atau memanggil halaman *website* dari *web server*. Pengujian ini dilakukan kepada kedua *website* yang dijalankan yaitu *website buku tamu* dan *website dashboard*. Berikut merupakan hasil pengujian kedua *website* pada Tabel 3 dan Tabel 4.

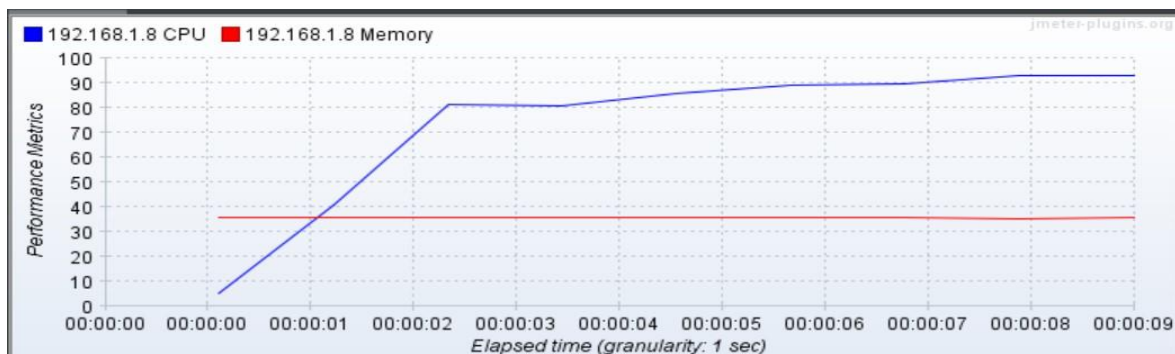
Tabel 2. Pengujian Website buku tamu

Total User	Throughput Req/sec	Average Response Time (ms)	Std. Deviasi	Error Rate
------------	-----------------------	-------------------------------	--------------	---------------

5	6.2	9	8.03	0
10	11	9	6.97	0
50	50.6	9	6.73	0
100	100.1	10	7.20	0
200	198.5	18	11.12	0
400	236.9	366	753.9	0

Tabel 3. Pengujian *Website Dashboard*

Total User	Throughput Req/sec	Average Response Time (ms)	Std. Deviasi	Error Rate
5	6.2	5	4.55	0
10	11.1	5	1.51	0
50	50.8	5	2.43	0
100	100.3	5	2.69	0
200	197.4	8	4.88	0
400	237.1	373	806	0

Gambar 6. *Resource CPU dan Memory* pada pengujian metode GET

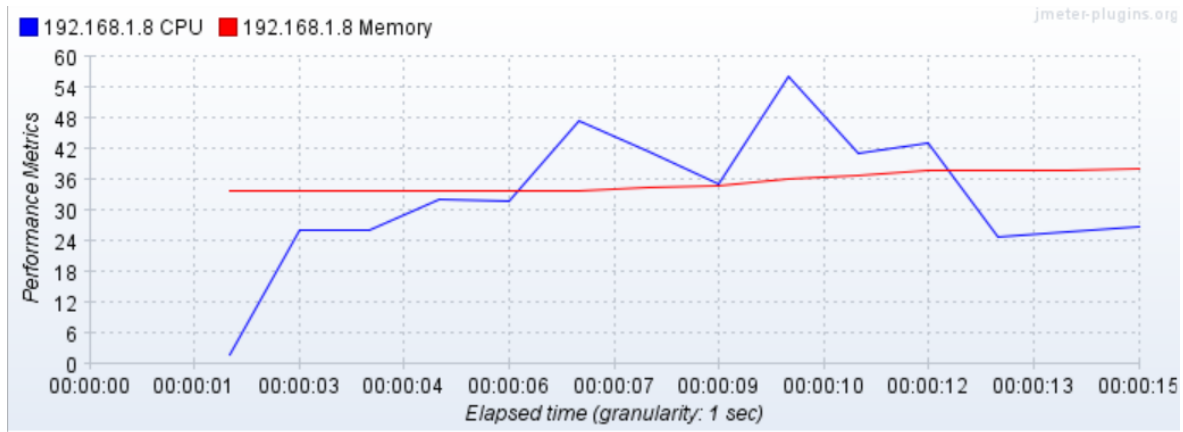
Sesuai dengan hasil pengujian *web server* memiliki *throughput* berkisar 236 *request/second* dalam menangani permintaan dari kedua *website*. Hal tersebut dapat berubah ketika jaringan lokal yang sibuk sehingga dapat mengurangi *throughput*-nya. Pada Gambar 6. Merupakan grafik *resource* yang digunakan oleh *web server* ketika dilakukan pengujian. Sesuai dengan grafik diatas, bahwa tidak ada terjadi penurunan mendadak dalam penggunaan *resource CPU* yang dapat mempengaruhi atau memperlambat hasil.

3.2.2 Pengujian *Load testing metode POST*

Pengujian ini dijalankan untuk melihat seberapa cepat *web server* dalam menangani *upload* sebuah file foto berukuran 100KB dengan format PNG serta data tamu pada *web server*. Konfigurasi yang digunakan khusus pengujian ini berupa *ramp-up* 1 detik dengan *loop* 5 kali.

Tabel 3. Pengujian *Upload*

Total User	Throughput Req/sec	Average	Std. Deviasi	Waktu Proses Pengujian	Error Rate
100	100.1	463	856.57	5	0
200	99.7	1118	1758	10	0
300	99.5	1661	2855	15	0



Gambar 7. Resource CPU dan Memory pada pengujian metode POST

Pengujian dilakukan langsung ke jumlah user yang tinggi untuk melihat seberapa jauh *web server* menanggapi input data yang berupa foto dan data tamu. Sesuai dengan hasil pengujian *web server* mampu menanggapi 100 permintaan POST perdetiknya maksimal. Dan dengan dilihat standar deviasinya kecepatan tiap permintaan jauh beragam karena jika standar deviasi lebih kecil dari *average*, maka variasi data mendekati *average*[3]. Berdasarkan Gambar 7. dimana penggunaan resource yang tidak stabil dipengaruhi dari *processing* data yang diupload ke *web server* dan *database* sehingga mempengaruhi kecepatan *throughput*. *Throughput* akan berubah atau menurun jika, jaringan lokal dan *web server* yang sibuk.

4. KESIMPULAN

Web server yang dirancang berhasil menjalankan kedua *website dashboard* dan *website* buku tamu. Data yang diupload berhasil tersimpan dalam *web server*. Pada saat pengujian, kinerja *web server* diuji menggunakan Apache Jmeter dengan HTTP Request dengan metode GET dan POST. Pengujian tersebut mengeluarkan hasil berbeda. Yang mana *Throughput* pada metode GET lebih tinggi dibandingkan metode POST. Hal tersebut dipengaruhi oleh data yang diupload pada metode POST perlu *processing* tambahan pada sisi *web server*. Sesuai dengan hasil pengujian pada metode GET mendapat *throughput* 236 permintaan per detik untuk kedua *website* secara bersamaan. Dan pada metode POST dengan mengupload sebuah gambar berukuran 100KB dan data tamu kedalam *database* mendapatkan *throughput* 99 atau 100 permintaan per detiknya. Namun *throughput* pada metode POST dapat lebih tinggi jika mengupload data yang berukuran lebih kecil. Hasil pengujian dapat berubah jika jaringan yang digunakan sibuk atau perangkat keras yang digunakan berubah.

DAFTAR PUSTAKA

- [1] S. Wahid Pramono, M. Andria Reza Effendi Putra, and G. Eka Yuliasuti, "Sistem Informasi Penerimaan Calon Peserta Didik Baru Berbasis Web di SMP Da'arul Muttaqien Surabaya," *Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi*, vol. 1, no. 2, 2022, doi: 10.31284/p.semtik.2022-1.3087.
- [2] K. A. Pratama, R. T. Subagio, M. Hatta, and V. Asih, "178~189 Implementasi Load Balancing Pada Web Server Menggunakan Apache Dengan Server Mirror Data Secara Real Time-(Kresna Adi Pratama)," 2021.
- [3] A. C. Barus, J. Harungguan, and E. Manulu, "PENGUJIAN API WEBSITE UNTUK PERBAIKAN PERFORMANSI APLIKASI DITENUN," *PENGUJIAN API WEBSITE UNTUK PERBAIKAN PERFORMANSI APLIKASI DITENUN*, 2021.
- [4] M. Reza Maulana, E. Budi Susanto, S. Wahyu Binabar, and S. Widya Pratama Pekalongan, "ANALISIS KINERJA WEBSITE PEMERINTAH KOTA PEKALONGAN," 2021. [Online]. Available: <https://pekalongankota.go.id/>.
- [5] I. Kurnia and D. Afriansyah, "KONFIGURASI HOSTING SERVER MENGGUNAKAN CENTOS 7 PADA DINAS KOMUNIKASI DAN INFORMATIKA KABUPATEN PESAWARAN," vol. 01, no. 01, pp. 26–32, 2020.
- [6] A. Idrus, "PERANCANGAN OWNCLOUD STORAGE SERVER BERBASIS UBUNTU 20.04 PADA PT. HARRISMA GLOBALTECHNOLOGIES JAKARTA," 2020.
- [7] D. I. Permatasari *et al.*, "Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian," vol. 8, no. 1, 2020.