



Tersedia Online : <http://e-journals.unmul.ac.id/>

ADOPSI TEKNOLOGI DAN SISTEM INFORMASI (ATASI)

Alamat Jurnal : <http://e-journals2.unmul.ac.id/index.php/atasi/index>



Pengujian Kinerja Web Server Atas Penyedia Layanan *Elastic Cloud Compute (EC2)* Pada *Amazon Web Services (AWS)*

Fandy ^{1*)}, Rosmasari ²⁾, Gubtha Mahendra Putra ³⁾

Program Studi Informatika, Fakultas Teknik, Universitas Mulawarman

E-Mail : fandy.file@gmail.com ^{1*)}

ARTICLE INFO

Article history:

Received : 24 Mei 2022

Revised : 14 Juni 2022

Accepted : 17 Juni 2022

Available online : 30 Juni 2022

Keywords:

Web Server,
Virtual Private Server ,
Amazon Web Service,
Elastic Cloud Compute,
Apache JMeter

Kata Kunci :

Web Server,
Virtual Private Server ,
Amazon Web Service,
Elastic Cloud Compute,
Apache JMeter

APA style in citing this article:

Fandy, Rosmasari, & Putra, G. M. (2022). Pengujian Kinerja Web Server Atas Penyedia Layanan Elastic Cloud Compute (EC2) Pada Amazon Web Services (AWS). *Adopsi Teknologi Dan Sistem Informasi (ATASI)*, 1(1), 21 - 35. <https://doi.org/10.30872/atasi.v1i1.45>

ABSTRACT

Making a strong Web Server in terms of reliability, speed and performance is mandatory in handling all user requests. Virtual Private Server (VPS) is a server that is divided into several parts that have their own operating system. The favorite VPS provider and often recommended for VPS service choices is Amazon Web Services. The advantage of having an Enterprise SSD or the use of an SSD (Solid State Drive) in its storage hardware. Various opinions have come about regarding the performance or performance of the provider regarding which VPS service is suitable to be used as a formidable Web Server. Load Testing Testing on a Web Server built on elastic cloud compute services on Amazon Web Services, several test parameters such as Throughput, Response Time, Latency and Resource Utilization are used as an assessment of the performance or performance of the Amazon Web Services Web Server. Testing also uses the Performance Testing application from Apache JMeter. Based on the test results on Amazon Web Services with the test type Load Testing and the type of HTTP Request testing, the Web Server built on the Elastic Cloud Compute service has better performance or Web Server performance than other service providers.

ABSTRAK

Pembuatan Web Server yang tangguh dalam segi kehandalan, kecepatan maupun performa menjadi hal yang wajib dalam menangani semua permintaan pengguna. Virtual Private Server (VPS) merupakan server yang dibagi menjadi beberapa bagian yang mempunyai system operation sendiri. Provider VPS yang menjadi favorit dan sering menjadi rekomendasi pilihan layanan VPS adalah Amazon Web Services. Keunggulannya memiliki Enterprise SSD atau penggunaan SSD (Solid State Drive) dalam hardware penyimpanannya. Berbagai pendapat datang mengenai kinerja atau performa dari provider tersebut mengenai layanan VPS mana yang sesuai dijadikan sebagai sebuah Web Server yang tangguh. Pengujian Load Testing pada Web Server yang dibangun atas layanan elastic cloud compute pada Amazon Web Services, beberapa parameter pengujian seperti Throughput, Response Time, Latency dan Resource Utilization digunakan sebagai penilaian kinerja atau performa dari Web Server Amazon Web Services. Pengujian juga menggunakan aplikasi Performance Testing dari Apache JMeter. Berdasarkan hasil pengujian pada Amazon Web Services dengan jenis pengujian Load Testing dan tipe pengujian HTTP Request, Web Server yang dibangun pada layanan Elastic Cloud Compute memiliki performa atau kinerja Web Server lebih baik dari penyedia layanan lain.

2022 Adopsi Teknologi dan Sistem Informasi (ATASI) with CC BY NC SA license.

1. PENDAHULUAN

Semakin pesatnya pertumbuhan internet saat ini, berdampak pada meningkatnya akses pengguna yang terhubung di dalamnya. Hal tersebut berpengaruh pada kebutuhan terhadap mesin penyedia layanan, seperti halnya *server web server*. Hadirnya teknologi cloud saat ini, sangat membantu para pengelola *web server* dalam melakukan

*) Corresponding Author

<https://doi.org/10.30872/atasi.v1i1.45>

2022 Adopsi Teknologi dan Sistem Informasi (ATASI) with CC BY SA license.

manajerial *Web server* khususnya terhadap mesin server yang digunakan. Namun dalam upaya untuk meningkatkan kinerja sebuah web server yang mengalami peningkatan akses oleh penggunanya, Sehingga beban koneksi yang masuk untuk meminta layanan web server tersebut dapat dilayani oleh beberapa server yang menyediakan konten *web server* yang sama. Sehingga dengan hal tersebut, maka kinerja dari sebuah web server dapat terus dipertahankan bahkan mungkin ditingkatkan. (Riskiono & Darwis, 2020).

Web Server yang tangguh dalam segi kehandalan, kecepatan maupun performa sudah menjadi hal yang wajib dimiliki oleh sebuah Web Server dalam menampung semua permintaan pengguna atau pengunjung. Bahwasannya terdapat 3 jenis pilihan sebuah server yang biasanya dapat digunakan sebagai *Web Server*. Yaitu: *Shared Hosting*, *VPS (Virtual Private Server)* dan *Cloud Hosting*. *Shared hosting* digunakan secara bersama-sama oleh banyak pengguna sekaligus. Para pengguna layanan *shared hosting* tersebut berbagi resource & kapasitas server bersama-sama namun ini merupakan layanan paling murah dan paling ekonomis, *Virtual Private Server (VPS)* adalah tipe *web hosting* yang sumber dayanya berasal dari satu server fisik, yang kemudian terbagi lagi menjadi beberapa *server virtual* yang lebih kecil. Dalam hal ini, masing-masing *server virtual* berjalan seperti *dedicated server*. Sedangkan *Cloud hosting* adalah tipe *hosting* dengan sumber daya (*resource*) yang berasal dari beberapa *server* fisik terpusat yang berjalan secara bersamaan atau yang biasanya disebut dengan *cluster*.

Dari ketiga pilihan server yang biasa dijadikan *Web Server* tersebut solusi terbaik adalah VPS atau *Virtual Private Server*, dimana user dapat me-manage sendiri dengan harga yang relatif murah. Banyak sekali berbagai pendapat mengenai kinerja atau performa dari penggunaan *Virtual Private Server* untuk dijadikan sebagai sebuah *Web Server* yang unggul demi menampung *traffic visitor* yang tinggi. Dilihat dari permasalahan dimana belum bisa memastikan bahwa *provider VPS* dapat menampung *traffic* dalam jumlah yang sangat besar, maka solusinya adalah dilakukan pengujian kinerja *web server* pada sebuah VPS atau yang berasal dari perusahaan *Amazon Web Services (AWS)* (Riza, 2018).

Amazon Web Services (AWS) adalah anak perusahaan dari raksasa *e-commerce* dunia Amazon, menyediakan layanan *cloud computing platform*. Produk layanan *cloud* dari AWS sangat beragam jenisnya, seperti *virtual server*, *scaleable storage*, *database*, *analytics*, *blockchain*, *machine learning*, *internet of things*, *content delivery network*, dan masih banyak lagi. AWS ini adalah sebuah *platform cloud* paling komprehensif dan digunakan secara luas di dunia, menawarkan lebih dari 200 layanan unggulan yang lengkap dari pusat data secara global. (AWS, 2021)

VPS yang ada di *Amazon Web Services* biasa disebut dengan istilah *Elastic Compute Cloud (EC2)*. Beberapa produk AWS telah didesain untuk memudahkan pengguna, contohnya *Elastic Compute Cloud* atau EC2. Pada dasarnya EC2 adalah sebuah server virtual. Kemudahan yang ditawarkan oleh EC2 adalah yaitu jika kita bandingkan dengan *server* tradisional, penggunaan EC2 ini lebih efektif dan efisien. *Amazon EC2 (Elastic Compute Cloud)* digunakan sebagai infrastruktur (kapasitas pemrosesan, memori, dan harddisk) yang menyediakan layanan yang dibutuhkan oleh pengguna.

Atas kecanggihan dari *web server* sebagai sebuah alat teknologi yang dapat memudahkan web bagi perorangan maupun perusahaan, maka dalam hal ini penulis ingin meneliti sebuah penelitian yang mengkaji tentang *web server* yakni adalah penyedia layanan *Elastic Cloud Compute (EC2)* (AWS EC2, 2021).

Oleh sebab itu, penelitian ini mengangkat judul “Analisis Pengujian Kinerja *Web Server* Atas Penyedia Layanan *Elastic Cloud Compute (EC2)* Pada *Amazon Web Services (AWS)*”. Hal ini bertujuan agar dapat memastikan kualitas performa layanan dari *Elastic Cloud Compute (EC2)* dengan instance *Type General Purpose (t-family: t3.small)* yang ada di wilayah Indonesia yang berlokasi di Jakarta dalam menangani beban dari permintaan client kepada *web server* dengan melakukan dua kali pengujian beban untuk website *beritasamarinda.com* dimana pengujian pertama server akan di uji beban ringan dan pengujian kedua *server* akan di uji beban berat. Pengujian terdiri dari *parameter Throughput*, *Sample/Respon Time*, dan *Latency* menggunakan tools *Apache JMeter*.

2. TINJAUAN PUSAKA

A. Amazon Web Service

Amazon Web Services adalah sekumpulan layanan-layanan berbasis *Cloud Computing* yang disediakan oleh *amazon* sejak tahun 2002. *Amazon.com* sebelumnya lebih terkenal dengan toko buku *online*-nya. Meski demikian, beberapa tahun yang lalu (sekitar tahun 2005), *Amazon* mengembangkan dirinya menjadi *Amazon Web Services (AWS)* yang menyediakan layanan komputasi awan, dimana disetiap fungsi yang ada di dalamnya bisa diakses dengan panggilan *Web Services* (Sugiyanta & Nurahma, 2017). Pelanggan yang bekerjasama dengan AWS beberapa diantaranya adalah *Tokopedia*, *Traveloka*, *Gojek*, *Lion Air*, *Kompas*.

B. Server

Server dapat diartikan sebagai pusat, baik pusat data, pusat data base, pusat sistem. Jadi server adalah komputer yang bertugas melayani semua permintaan yang dilakukan oleh komputer klien selanjutnya hasil dari permintaan akan dikirimkan kembali kepada komputer klien. Server merupakan piranti khusus dalam jaringan komputer yang menjadi tempat bagi semua *nodes* di dalam jaringan untuk bisa melakukan *resource sharing*. *Server* melayani semua *nodes*, jika *nodes* membutuhkan. *Server* ada beberapa macam, yaitu: *printer server*, *file server*, *disk server*, dan *database server*. *Server* bisa bersifat *dedicated*, artinya *server* tidak bisa dipergunakan sebagai *nodes* untuk komunikasi, ada juga yang bersifat *non-dedicated*, yaitu selain berfungsi sebagai *server* juga dapat dipergunakan sebagai titik masuk untuk berkomunikasi di dalam jaringan. Cara seperti ini populer dengan istilah *client-server* (Nugroho, A., & Mustofa, T. K. 2012).

C. Apache Jmeter

Aplikasi *Apache Jmeter* adalah perangkat lunak *open source*, 100% aplikasi *Java* murni dirancang untuk memuat tes perilaku fungsional dan mengukur kinerja. Ini pada awalnya dirancang untuk pengujian Aplikasi *Web* tetapi sejak diperluas untuk fungsi tes lainnya. *Apache Jmeter* adalah aplikasi *open source* berbasis *Java* yang dapat dipergunakan untuk *performance test*. Bagi seorang *QA Engineer Jmeter* bisa digunakan untuk melakukan *load/stress testing Web Application, FTP Application* dan *Database server test*. *Apache JMeter* dapat digunakan untuk menguji kinerja baik pada sumber daya statis dan dinamis (*web services (SOAP / REST), Web bahasa dinamis - PHP, Java, ASP, NET, File, dll -, Java Objects*, Pangkalan Data dan Pertanyaan, *FTP Server* dll). Hal ini dapat digunakan untuk mensimulasikan beban berat pada *server*, sekelompok *server*, jaringan atau objek untuk menguji kekuatan atau untuk menganalisis kinerja secara keseluruhan di bawah jenis beban yang berbeda. (Ahmad & Setiawan, 2011)

D. Putty

PuTTY adalah sebuah program *open source* yang dapat digunakan untuk melakukan protokol jaringan *SSH, Telnet* dan *Rlogin*. Protokol ini dapat digunakan untuk menjalankan sesi *remote* pada sebuah komputer melalui sebuah jaringan, baik itu LAN, maupun internet. Program ini banyak digunakan oleh para pengguna komputer tingkat menengah ke atas, yang biasanya digunakan untuk menyambungkan, mensimulasi, atau mencoba berbagai hal yang terkait dengan jaringan. Program ini juga dapat digunakan sebagai tunnel di suatu jaringan (Kurnia, 2020).

E. Cloud Computing

Cloud Computing atau secara kata bila diterjemahkan kedalam Bahasa Indonesia dapat berbunyi “Komputasi Awan”, adalah gabungan pemanfaatan teknologi komputer (komputasi) dan pengembangan berbasis *Internet* (awan) teknologi ini menjadikan internet sebagai pusat pengelolaan data dan aplikasi, di mana pengguna diberikan hak akses. *Cloud computing* adalah sebuah konsep pemahaman dalam rangka pembuatan kerangka kerja komputasi secara *online* lokal (*LAN*) maupun *global (internet)* dimana terdapat beragam aplikasi maupun data dan media penyimpanan yang dapat diakses dan digunakan secara berbagi (*shared service*) dan bersamaan (*simultaneous access*) oleh para pengguna yang beragam mulai dari perorangan sampai kepada kelas pengguna korporasi atau perusahaan (Sajjad, Muhammad. 2018)

F. Cloud Flare

Cloudflare adalah salah satu dari *Content Delivery Network (CDN)* yang dapat membuat kecepatan pemuatan situs web lebih cepat. Fungsi utama CDN adalah untuk menjaga situs web, pertahanan pertama melawan serangan *hacker*, serangan *DDoS* serta perlindungan dari ancaman lain. *Cloudflare* memiliki beberapa situs server di seluruh dunia. Situs ini mendistribusikan salinan data yang kepada klien untuk mengurangi latensi dan meningkatkan ketersediaan. Proses menyimpan data situs web di pusat datanya juga memiliki efek positif pada banyak hal selain keamanan dan kecepatan. *Cloudflare* melindungi dan mempercepat situs web *online*. Sehingga dengan menggunakan *cloudflare* situs web perusahaan lebih baik dalam hal kecepatan akses bagi pengguna dan keamanan dari serangan *hacker* dan lainnya (Jayanti et al., 2020)

3. METODE PENELITIAN

A. Tahap Pelaksanaan Penelitian

Tahap pelaksanaan penelitian dilakukan dengan melaksanakan identifikasi masalah yang menjadi fokus dalam penelitian. Setelah itu dilaksanakan perumusan masalah agar tidak meluas dan keluar dari batasan penelitian yang akan dilaksanakan. Untuk memudahkan dan memperkuat pelaksanaan penelitian maka dilakukan studi literatur yang berhubungan dengan penelitian ini dari baik dari Jurnal, Artikel, Buku hingga ke website yang berhubungan langsung dengan penelitian.

B. Pengumpulan Data

Pengumpulan data dilakukan, untuk mendukung penelitian mengenai Analisis Pengujian Kinerja *Web Server* Atas Penyedia Layanan *Elastic Cloud Compute (EC2)* Pada *Amazon Web Services (AWS)* dengan melaksanakan studi literatur dilakukan untuk mencari beberapa data mengenai tingkat performa yang baik untuk mendukung hasil kesimpulan mengenai efektivitas *Web Server*. Dalam hal ini peneliti mengumpulkan dan melakukan analisis terhadap beberapa literatur yang berasal dari Buku-buku referensi, Jurnal, maupun internet yang terkait dengan penelitian dan performa kinerja yang baik dari web server.

C. Perancangan Proses

Proses pengumpulan data yang diperoleh peneliti akan dikumpulkan dan dianalisis sesuai dengan judul yang diambil oleh peneliti yakni Analisis Pengujian Kinerja *Web Server* Atas Penyedia Layanan *Elastic Cloud Compute (EC2)* Pada *Amazon Web Services (AWS)*. Pengambilan data akan dilakukan dengan menggunakan aplikasi *software Apache JMeter* pada websites <https://beritasamarinda.com>.

Beberapa tahapan dari proses analisis akan dijabarkan sebagaimana berikut:

- a. Tahapan pertama, membuat akun AWS pada halaman <https://portal.aws.amazon.com/billing/signup#/> dengan mengisi data diri secara lengkap dan wajib menggunakan kartu debit/kredit sebagai metode pembayaran utama. Setelah akun berhasil dibuat, pengguna memilih layanan EC2 yang ada di halaman utama kemudian terdapat beberapa Langkah-langkah dalam pembuatan EC2 yaitu:
 1. Memilih Sistem Operasi yang akan digunakan, disini pengguna akan menggunakan sistem operasi *CentOS 7* yang berlokasi di Asia Tenggara (Jakarta).
 2. Memilih *Tipe EC2* yang akan digunakan, disini pengguna akan memilih *Tipe General Purpose (t-family: t3.small)* dengan *vCPU 2x* dan memori *2 GB*.
 3. Konfigurasi *EC2*, pengguna tidak ada mengubah format konfigurasi yang ada.
 4. Menambah penyimpanan, pengguna menambahkan penyimpanan sebesar *30 GB* dengan tipe *General Purpose SSD (gp2)*.
 5. Menambah *Tag*, pengguna tidak ada menambah *tag*.
 6. Konfigurasi *Firewall*, pengguna membuat aturan akses untuk semua pengunjung dalam berbagai *port* mulai dari *port 0-65535* dan untuk semua *IP* dapat mengaksesnya.
 7. Tinjauan, pengguna meninjau kembali detail yang telah dibuat sebelumnya, dan pengguna bisa mengubahnya kembali untuk setiap bagian.
- b. Melakukan pemasangan *AaPanel* menggunakan *Software PuTTY* untuk mengelola sebuah *website* layaknya sebuah *hosting*. Semua aktivitas termasuk dalam proses pembuatan *WEB, Database, Email, DNS, FTP, SSL, Backup*, mengubah *PHP* dan lain sebagainya.
- c. Setelah membuat *web* pada *AaPanel* selanjutnya pengguna bisa langsung melakukan pemasangan *WordPress* melalui menu *installer* yang telah disediakan oleh *AaPanel* dan jika ingin membuat *website* lain pengguna bisa mengunggah *script* di file manager secara manual.
- d. Pengguna melakukan pembelian *domain https://beritasamarinda.com* di *NameCheap* dan mengarahkan *NameServer domain* ke *CloudFlare*, Disini pengguna menggunakan *CloudFlare* untuk mengelola *DNS* karena banyak fitur-fitur yang pengguna butuhkan di *CloudFlare* diantaranya anti *DDOS* dan *SSL* gratis untuk keamanan *website* dari *hacking* dan menyembunyikan *IP asli server*.
- e. Mengatur *DNS Server domain* menggunakan *CloudFlare* dengan cara menambahkan *DNS Record*, memasukkan *IP Address server* ke *DNS Record Tipe A* dan melakukan *CNAME subdomain WWW* ke *domain* utama.

D. Perancangan Proses Pengujian

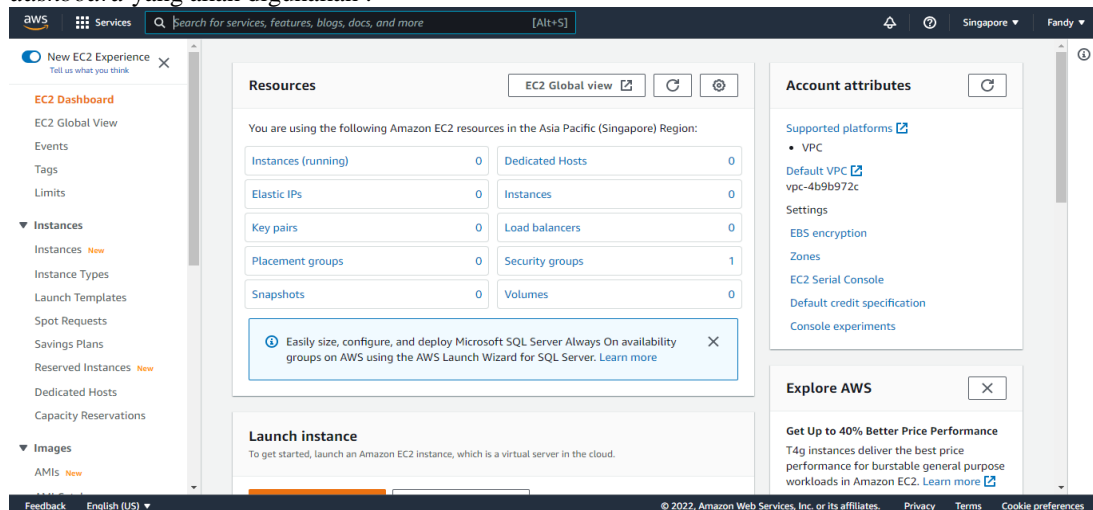
Dalam upaya analisis Pengujian Kinerja *Web Server* Atas Penyedia Layanan *Elastic Cloud Compute (EC2)* Pada *Amazon Web Services (AWS)* beberapa proses yang akan dilakukan adalah:

1. Melakukan Implementasi *Website*

Tahapan ini akan melakukan implementasi *websites https://beritasamarinda.com*, yang mana akan didaftarkan pada tanggal 23 Februari 2022.

2. Melakukan Implementasi *Web Server* pada *Elastic Cloud Compute (EC2)*

Setelah melakukan implementasi *website*, peneliti kemudian akan mengimplementasikan-nya kepada *Web Server Virtual Private Server (VPS)* pada *Amazon Web Services (AWS)* yang biasa disebut dengan istilah *Elastic Compute Cloud (EC2)*. Berikut merupakan gambaran *Elastic Compute Cloud (EC2) dashboard* yang akan digunakan :



Gambar 1. Elastic Compute Cloud (EC2) dashboard

3. Melakukan uji performa kinerja Web Server

Uji performa kinerja *Web Server Virtual Private Server (VPS)* pada *Amazon Web Services (AWS)* yang biasa disebut dengan istilah *Elastic Compute Cloud (EC2)* akan dilakukan dengan menggunakan bantuan dari aplikasi *software Apache JMeter*. Pengujian dilakukan dengan menggunakan:

- a. `$ apache-jmeter-5.4.1/bin/jmeter;`
- b. Kemudian, klik *Add > Threads (Users) > Thread Group*; isi data;
- c. Lalu, membuat sampel dengan *Add* pada (*Tread Group*) > *Sampler > HTTP Request*; isi data;
- d. Pengatur waktu dengan *Add* (pada *HTTP Request*) > *Timer > Constant Timer*; lalu tambahkan pengatur waktu;
- e. Pendengar dengan *Add* (pada *HTTP Request*) > *Listener > {pilih jenis laporan; Results in Table, Results Tree, Response Times Over Time, Transaction per Second, Statistical Aggregate Report dan Summary Report}*
- f. Jalankan *Performance Test*.

E. Perancangan Pengujian

Pengujian akan dilakukan dengan menggunakan bantuan software aplikasi *Apache Jmeter*. Pengujian dimaksudkan untuk melihat beberapa hal berikut:

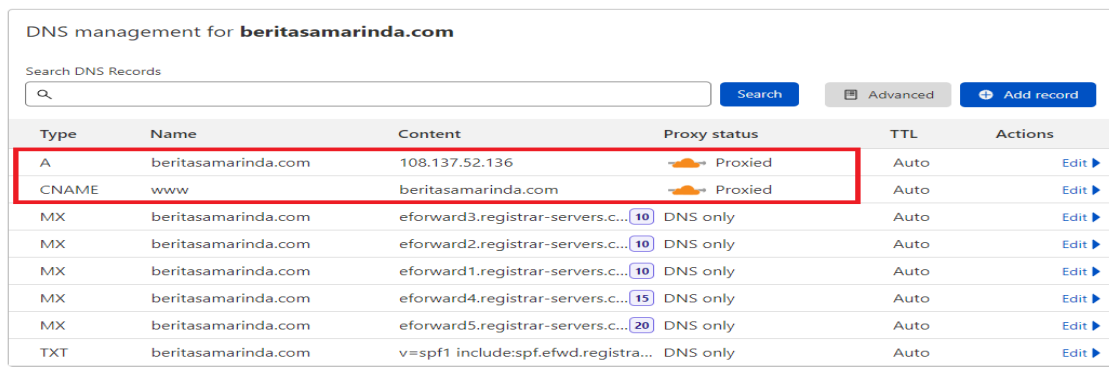
1. *Throughput*, yaitu jumlah permintaan per menit yang diproses server
2. *Sample Time/Response Time*, yaitu perbedaan antara waktu ketika permintaan dikirim dan waktu ketika respon telah diterima sepenuhnya.
3. *Latency*, yaitu waktu yang dibutuhkan untuk permintaan untuk mencapai server dan untuk menanggapi perjalanan balik (dari server ke klien) atau perbedaan antara waktu ketika permintaan dikirim dan waktu ketika respon sudah mulai diterima.

Data dari hasil *performance* tersebut akan dibandingkan dengan data yang didapatkan peneliti pada literatur mengenai performa terbaik untuk *Web Server*. Setelahnya, peneliti kemudian dapat mengambil kesimpulan untuk mendapatkan *Web Server* dengan kinerja terbaik, handal serta tangguh sesuai dengan kebutuhan penggunaan.

4. HASIL DAN PEMBAHASAN

A. Hasil Pengujian

Pengujian dilakukan dengan 2 tahap tes. Sebelum dilakukan pengujian pada *DNS Record Cloudflare* pada *IP Address* dirubah menjadi *IP Address Amazon Web Service*, karena pengujian *Web Server* untuk *Amazon Web Service*.



| Type | Name | Content | Proxy status | TTL | Actions |
|-------|---------------------|--|--------------|------|---------|
| A | beritasamarinda.com | 108.137.52.136 | Proxied | Auto | Edit |
| CNAME | www | beritasamarinda.com | Proxied | Auto | Edit |
| MX | beritasamarinda.com | eforward3.registrar-servers.c... ¹⁰ | DNS only | Auto | Edit |
| MX | beritasamarinda.com | eforward2.registrar-servers.c... ¹⁰ | DNS only | Auto | Edit |
| MX | beritasamarinda.com | eforward1.registrar-servers.c... ¹⁰ | DNS only | Auto | Edit |
| MX | beritasamarinda.com | eforward4.registrar-servers.c... ¹⁵ | DNS only | Auto | Edit |
| MX | beritasamarinda.com | eforward5.registrar-servers.c... ²⁰ | DNS only | Auto | Edit |
| TXT | beritasamarinda.com | v=spf1 include:spf.efwd.registra... | DNS only | Auto | Edit |

Gambar 2. Merubah DNS Record IP Address Amazon Web Service

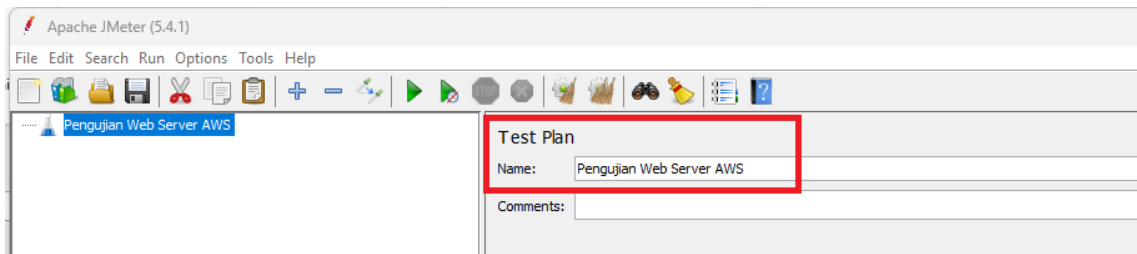
Setelah *IP Address* sesuai dengan *IP Address Amazon Web Service*, maka dibawah ini dilakukan pengujian *Web Server Amazon Web Service*.

1. Pengujian Web Server Amazon Web Service Tes 1

Pengujian pertama *Load Testing* pada *Web Server Amazon Web Service* dengan *Apache JMeter* dengan format Tes sebagai berikut:

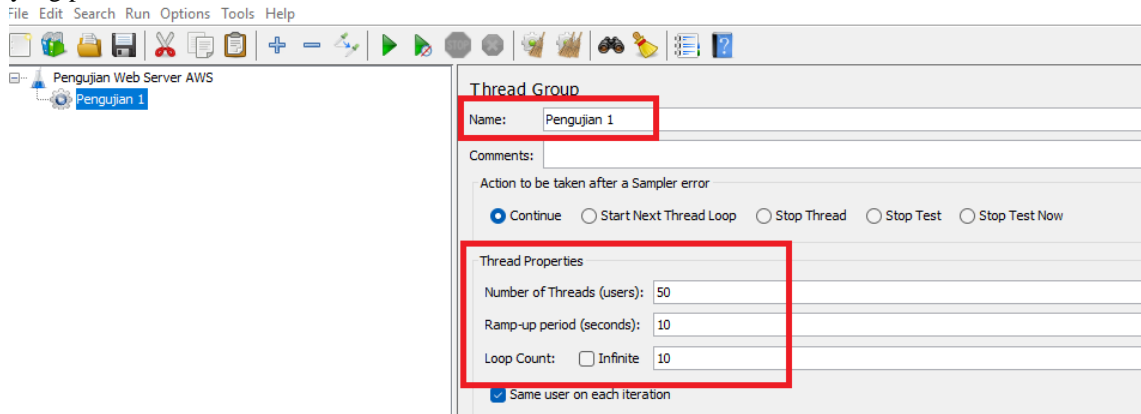
- a. Jenis Pengujian *HTTP Request*.
- b. *Thread / User* 50.
- c. *Ramp-Up (in Seconds)* 10.
- d. *Loop Count* 10.
- e. *Total HTTP Request* 500.

Berikut langkah *Test Plan* dari Pengujian pertama *Web Server Amazon Web Service*: Pertama, Pembuatan *Test Plan* dengan name “Pengujian Web Server Amazon Web Service”.



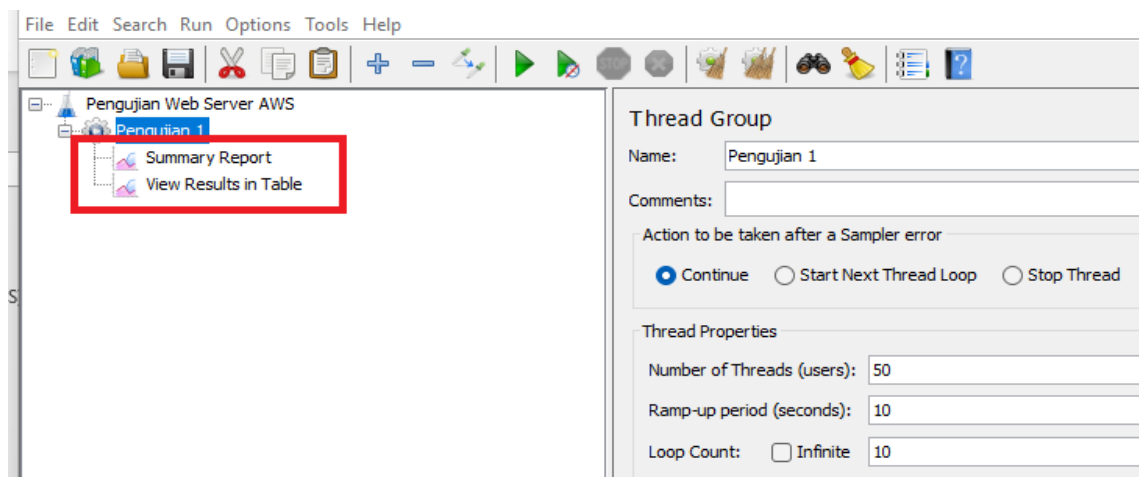
Gambar 3. Test Plan Web Server Amazon Web Service

Kedua, membuat *Thread Group* dengan name “Pengujian 1” dan mengisi *Thread Properties* sesuai dengan tes kasus yang pertama.



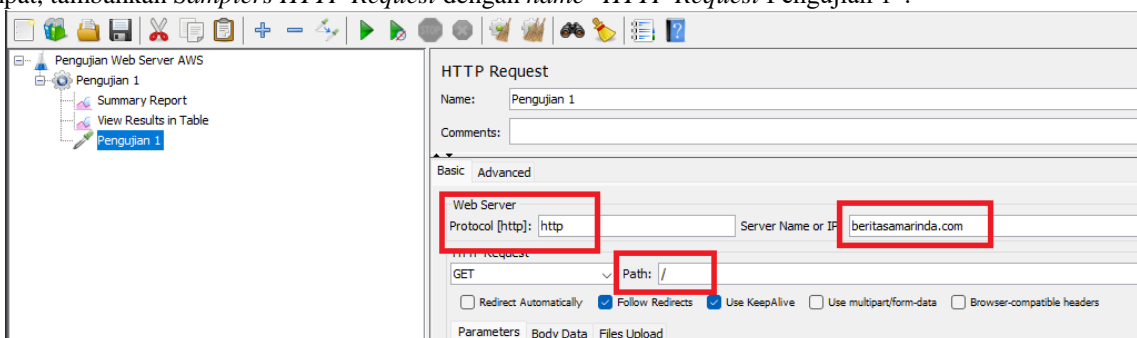
Gambar 4. Thread Group Pengujian 1 Web Server Amazon Web Service

Ketiga, tambahkan *listener* untuk melihat hasil dari parameter pengujian pada *Thread Group*. *Listener* yang ditambahkan adalah *Summary Report* (contoh gambar 4) dan *View Results in Table* (contoh gambar 5).



Gambar 5. Listener Pada Thread Group Pengujian 1 Web Server Amazon Web Service

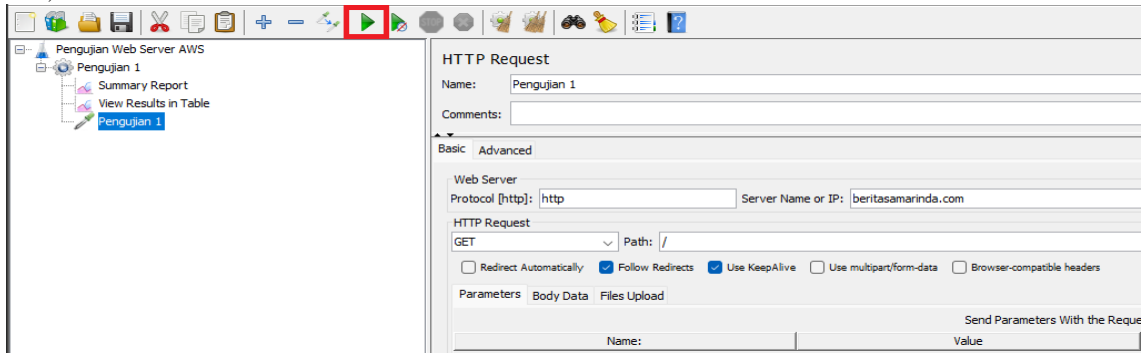
Keempat, tambahkan *Samplers HTTP Request* dengan name “HTTP Request Pengujian 1”.



Gambar 6. HTTP Request Pengujian 1 Web Server Amazon Web Service

Penjelasan pada Web Server diisi dengan *Domain Website* “beritasamarinda.com”, sedangkan pada Path diisi “/” karena *Load Testing* dengan *User* buatan atau *Thread* akan mencoba mengakses halaman utama / awal dari Website *beritasamarinda.com*.

Kelima, *Run Test*.



Gambar 7. Run Test Penguji 1 Web Server Amazon Web Service

Hasil pengujian pertama dari Web Server Amazon Web Service adalah sebagai berikut:

a. *Throughput*

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|-----------|-----------|---------|-----|-------|-----------|---------|------------|-----------------|-------------|------------|
| Penguji 1 | 500 | 12122 | 697 | 15447 | 2428.11 | 0.00% | 3.8/sec | 107.31 | 0.46 | 28914.0 |
| TOTAL | 500 | 12122 | 697 | 15447 | 2428.11 | 0.00% | 3.8/sec | 107.31 | 0.46 | 28914.0 |

Gambar 8. Summary Report Penguji 1 Amazon Web Service

Hasil *Throughput* dari *samples HTTP Request* dalam pengujian pertama Web Server Amazon Web Service adalah 3.8/sec dengan *error samples* 0.0%.

b. *Sample Time / Response Time*

| Sample # | Start Time | Thread Name | Label | Sample Time(ms) | Status | Bytes | Sent Bytes | Latency | Connect Time(ms) |
|----------|--------------|----------------|-----------|-----------------|--------|-------|------------|---------|------------------|
| 1 | 23:27:00.481 | Penguji 1 1-1 | Penguji 1 | 697 | ✓ | 29004 | 124 | 614 | 79 |
| 2 | 23:27:00.682 | Penguji 1 1-2 | Penguji 1 | 1063 | ✓ | 29004 | 124 | 922 | 75 |
| 3 | 23:27:00.885 | Penguji 1 1-3 | Penguji 1 | 945 | ✓ | 29004 | 124 | 853 | 83 |
| 4 | 23:27:01.089 | Penguji 1 1-4 | Penguji 1 | 1933 | ✓ | 29004 | 124 | 1612 | 74 |
| 5 | 23:27:01.179 | Penguji 1 1-1 | Penguji 1 | 1931 | ✓ | 29004 | 124 | 1722 | 74 |
| 6 | 23:27:01.252 | Penguji 1 1-5 | Penguji 1 | 2016 | ✓ | 29004 | 124 | 1812 | 85 |
| 7 | 23:27:01.496 | Penguji 1 1-6 | Penguji 1 | 2428 | ✓ | 29004 | 124 | 2143 | 83 |
| 8 | 23:27:01.746 | Penguji 1 1-2 | Penguji 1 | 2746 | ✓ | 29004 | 124 | 2439 | 79 |
| 9 | 23:27:01.685 | Penguji 1 1-7 | Penguji 1 | 2887 | ✓ | 29004 | 124 | 2554 | 139 |
| 10 | 23:27:01.830 | Penguji 1 1-3 | Penguji 1 | 2925 | ✓ | 29004 | 124 | 2547 | 92 |
| 11 | 23:27:01.886 | Penguji 1 1-8 | Penguji 1 | 3035 | ✓ | 29004 | 124 | 2686 | 76 |
| 12 | 23:27:02.089 | Penguji 1 1-9 | Penguji 1 | 3268 | ✓ | 29004 | 124 | 2887 | 72 |
| 13 | 23:27:02.292 | Penguji 1 1-10 | Penguji 1 | 3649 | ✓ | 29004 | 124 | 3130 | 69 |
| 14 | 23:27:02.495 | Penguji 1 1-11 | Penguji 1 | 4542 | ✓ | 29004 | 124 | 3907 | 85 |
| 15 | 23:27:02.682 | Penguji 1 1-12 | Penguji 1 | 4677 | ✓ | 29004 | 124 | 3817 | 73 |
| 16 | 23:27:02.883 | Penguji 1 1-13 | Penguji 1 | 5166 | ✓ | 29004 | 124 | 4499 | 138 |
| 17 | 23:27:03.022 | Penguji 1 1-4 | Penguji 1 | 5235 | ✓ | 29004 | 124 | 4416 | 82 |
| 18 | 23:27:03.082 | Penguji 1 1-14 | Penguji 1 | 5573 | ✓ | 29004 | 124 | 4706 | 76 |
| 19 | 23:27:03.111 | Penguji 1 1-1 | Penguji 1 | 6151 | ✓ | 29004 | 124 | 5515 | 83 |
| 20 | 23:27:03.683 | Penguji 1 1-17 | Penguji 1 | 8078 | ✓ | 29004 | 124 | 6857 | 71 |
| 21 | 23:27:03.284 | Penguji 1 1-15 | Penguji 1 | 8477 | ✓ | 29004 | 124 | 7256 | 70 |
| 22 | 23:27:03.923 | Penguji 1 1-6 | Penguji 1 | 7932 | ✓ | 29004 | 124 | 6855 | 95 |
| 23 | 23:27:03.308 | Penguji 1 1-15 | Penguji 1 | 8579 | ✓ | 29004 | 124 | 7401 | 92 |
| 24 | 23:27:03.484 | Penguji 1 1-16 | Penguji 1 | 8414 | ✓ | 29004 | 124 | 7207 | 77 |
| 25 | 23:27:03.881 | Penguji 1 1-18 | Penguji 1 | 8225 | ✓ | 29004 | 124 | 7091 | 90 |

Gambar 9. Sample Time / Response Time Penguji 1 Web Server Amazon Web Service

Pada gambar 9 merupakan hasil dari *sample time / response time*, nilai yang diambil untuk *sample time / response time* adalah *average* (*View Results in Table* Lengkap pada Lampiran A.1), karena *average* dihasilkan dari keseluruhan nilai *sample time / response time* dibagi jumlah keseluruhan *sample*. Maka nilai untuk *sample time / response time* pada tes pertama Web Server Amazon Web Service adalah 12122/ms (*millisecond*).

c. *Latency*

Nilai *latency* pada masing-masing *sample* (*View Results in Table* Lengkap pada Lampiran A.1) bisa dilihat pada gambar 5.78, untuk mencari nilai rata-rata *latency* dari keseluruhan *sample* harus dihitung secara *manual* dengan menghitung Rata-rata

$$Latency = \frac{\text{Jumlah Seluruh Nilai Latency}}{\text{Jumlah Sample}}$$

maka nilai dari rata-rata *latency* pada pengujian, yaitu :

pertama adalah $\frac{800718}{500} = 1601/ms$ (*millisecond*).

d. *Resouce Utilization*

Pengujian dilakukan dari *sample* nomor 1 mulai berjalan pada pukul 23:27:00 (gambar 5.79), sampai sample terakhir atau sample nomor 500 berakhir pada pukul 23:29:06 (gambar 5.80). Maka dari itu server monitoring yang diperhatikan pada penggunaan *resouce utilization* dalam interval waktu tersebut.

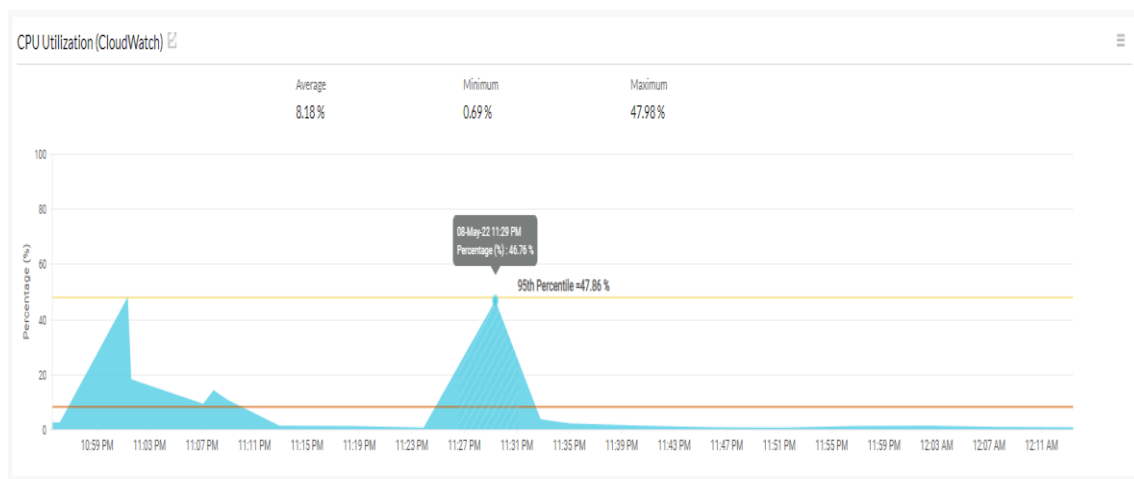
| Sample # | Start Time | Thread Name | Label | Sample Time(ms) | Status |
|----------|--------------|-----------------|-------------|-----------------|--------|
| 1 | 23:27:00.481 | Pengujian 1 1-1 | Pengujian 1 | 697 | ✓ |
| 2 | 23:27:00.682 | Pengujian 1 1-2 | Pengujian 1 | 1063 | ✓ |
| 3 | 23:27:00.885 | Pengujian 1 1-3 | Pengujian 1 | 945 | ✓ |
| 4 | 23:27:01.089 | Pengujian 1 1-4 | Pengujian 1 | 1933 | ✓ |
| 5 | 23:27:01.179 | Pengujian 1 1-1 | Pengujian 1 | 1931 | ✓ |
| 6 | 23:27:01.292 | Pengujian 1 1-5 | Pengujian 1 | 2016 | ✓ |
| 7 | 23:27:01.496 | Pengujian 1 1-6 | Pengujian 1 | 2428 | ✓ |
| 8 | 23:27:01.746 | Pengujian 1 1-2 | Pengujian 1 | 2746 | ✓ |
| 9 | 23:27:01.685 | Pengujian 1 1-7 | Pengujian 1 | 2887 | ✓ |
| 10 | 23:27:01.830 | Pengujian 1 1-3 | Pengujian 1 | 2925 | ✓ |
| 11 | 23:27:01.886 | Pengujian 1 1-8 | Pengujian 1 | 3055 | ✓ |

Gambar 10. Waktu Mulai Pengujian 1 Web Server Amazon Web Service

| | | | | | |
|-----|--------------|------------------|-------------|------|---|
| 489 | 23:29:05.364 | Pengujian 1 1-42 | Pengujian 1 | 6305 | ✓ |
| 490 | 23:29:05.016 | Pengujian 1 1-39 | Pengujian 1 | 6666 | ✓ |
| 491 | 23:29:05.154 | Pengujian 1 1-40 | Pengujian 1 | 6599 | ✓ |
| 492 | 23:29:06.010 | Pengujian 1 1-47 | Pengujian 1 | 5780 | ✓ |
| 493 | 23:29:05.424 | Pengujian 1 1-37 | Pengujian 1 | 6448 | ✓ |
| 494 | 23:29:06.744 | Pengujian 1 1-43 | Pengujian 1 | 5158 | ✓ |
| 495 | 23:29:06.098 | Pengujian 1 1-45 | Pengujian 1 | 5824 | ✓ |
| 496 | 23:29:06.248 | Pengujian 1 1-44 | Pengujian 1 | 5724 | ✓ |
| 497 | 23:29:06.478 | Pengujian 1 1-50 | Pengujian 1 | 5532 | ✓ |
| 498 | 23:29:06.423 | Pengujian 1 1-49 | Pengujian 1 | 5592 | ✓ |
| 499 | 23:29:06.287 | Pengujian 1 1-46 | Pengujian 1 | 5730 | ✓ |
| 500 | 23:29:06.333 | Pengujian 1 1-48 | Pengujian 1 | 5712 | ✓ |

Gambar 11. Waktu Selesai Pengujian 1 Web Server Amazon Web Service

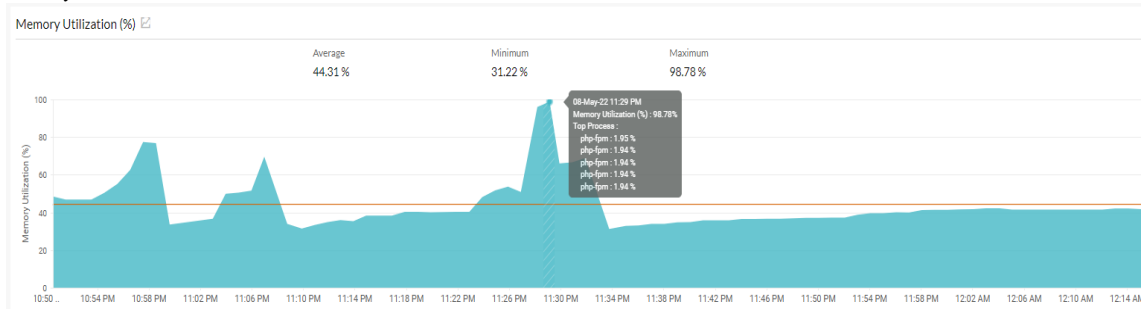
a. *CPU Utilization*



Gambar 12. CPU Utilization Pengujian 1 Web Server Amazon Web Service

Penggunaan *CPU Utilization* paling tinggi dalam interval pengujian yang sudah disebutkan diatas adalah 46.76%.

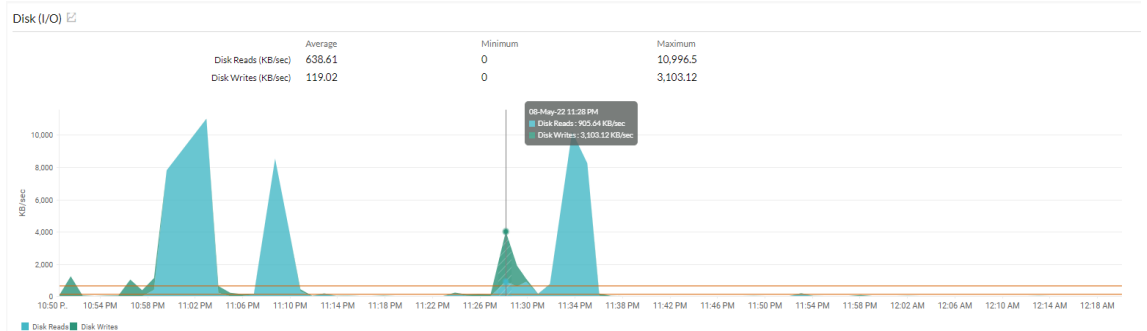
b. *Memory Utilization*



Gambar 13. Memory Utilization Pengujian 1 Web Server Amazon Web Service

Penggunaan *Memory Utilization* sebesar 98.78%

c. *Disk I/O Utilization*



Gambar 14. Disk I/O Utilization Pengujian 1 Web Server Amazon Web Service

Penggunaan *Disk I/O Utilization* menunjukkan untuk *Disk Reads*: 905.64 kb/sec dan *Disk Writes*: 3.103 kb/sec.

2. Pengujian Web Server Amazon Web Service Tes 2

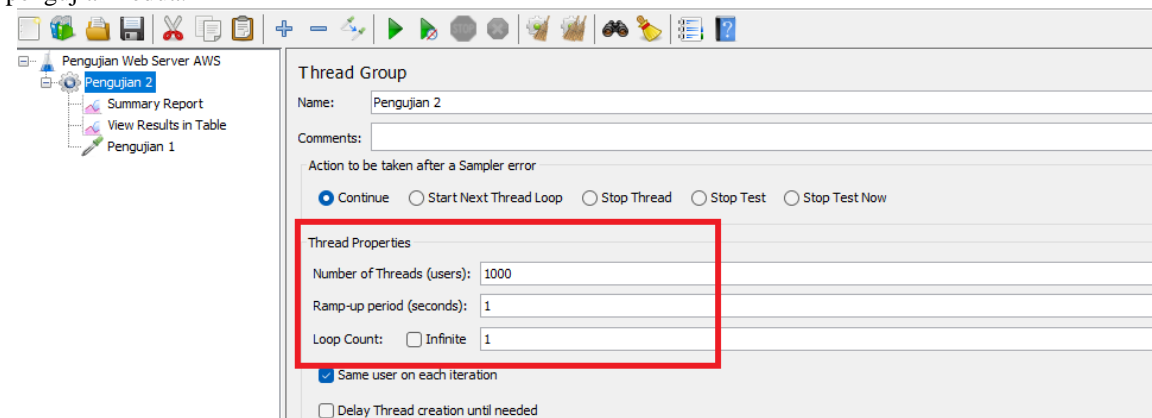
Pengujian kedua *Load Testing* pada Web Server Amazon Web Service dengan Apache JMeter dengan format Tes sebagai berikut:

- a. Jenis Pengujian *HTTP Request*.
- b. *Thread / User* 1000.
- c. *Ramp-Up (in Seconds)* 1.
- d. *Loop Count* 1.

a. *Total HTTP Request* 1000.

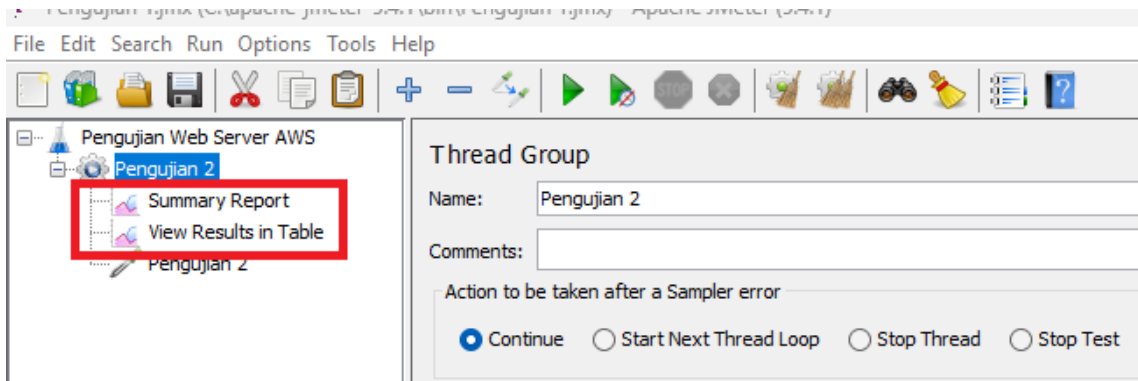
Berikut pengujian kedua Web Server Amazon Web Service.

Menambah *Thread Group* baru dengan name “Pengujian 2” dan mengisi *Thread Properties* sesuai pengujian kedua.



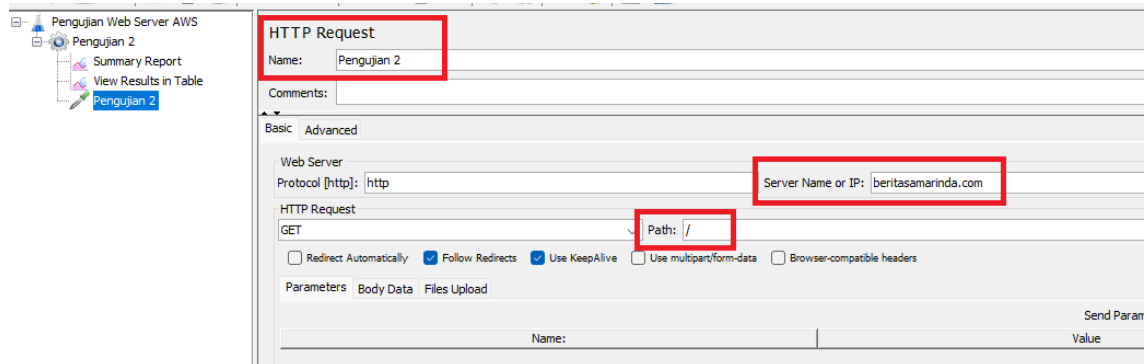
Gambar 15. Thread Group Pengujian 2 Web Server Amazon Web Service

Menambahkan *listener* untuk melihat hasil dari parameter pengujian pada *Thread Group*. *Listener* yang ditambahkan adalah *Summary Report* dan *View Results in Table*



Gambar 16. Listener Penguujian 2 Web Server Amazon Web Service

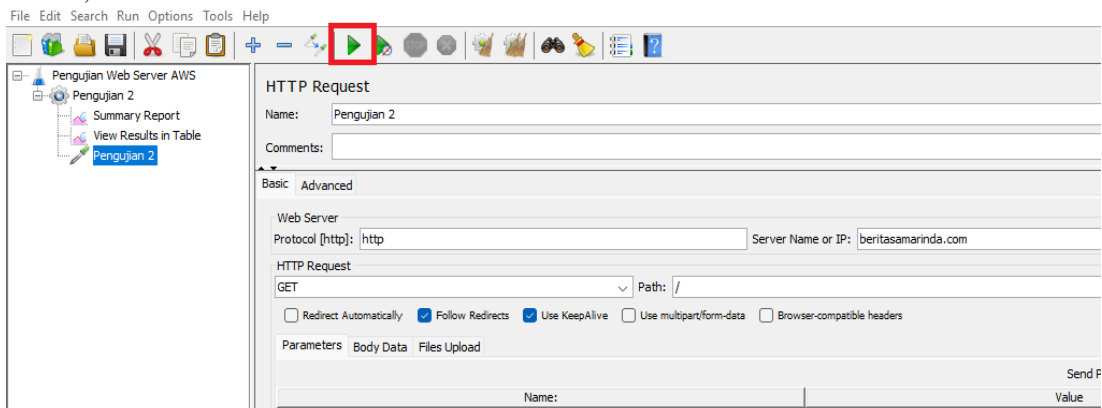
Menambahkan *Samplers HTTP Request* dengan name “*HTTP Request Penguujian 2 Web Server Amazon Web Service*”.



Gambar 17. Sampler HTTP Request Penguujian 2 Web Server Amazon Web Service

Penjelasan pada Web Server diisi dengan *Domain Website “beritasamarinda.com”*, sedangkan pada Path diisi “/” karena *Load Testing* dengan *User* buatan atau *Thread* akan mencoba mengakses halaman utama / awal dari Website *beritasamarinda.com*.

Kelima, *Run Test*.



Gambar 18. Run Test Penguujian 2 Web Server Amazon Web Service

Hasil pengujian kedua dari Web Server Amazon Web Service adalah sebagai berikut:

1. *Throughput*

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput |
|--------------|-----------|---------|------|--------|-----------|---------|------------|
| Penguujian 1 | 1000 | 84101 | 1945 | 114864 | 28009.31 | 64.90% | 8.6/sec |
| TOTAL | 1000 | 84101 | 1945 | 114864 | 28009.31 | 64.90% | 8.6/sec |

Gambar 19. Throughput Penguujian 2 Web Server Amazon Web Service

Hasil *Throughput* dari *samples HTTP Request* dalam pengujian pertama Web Server Amazon Web Service adalah 8.6/sec dengan *error samples* 0.00%.

2. *Sample Time / Response Time*

| Sample # | Start Time | Thread Name | Label | Sample Time(ms) | Status | Bytes | Sent Bytes | Latency |
|----------|--------------|------------------|-------------|-----------------|--------|-------|------------|---------|
| 1 | 01:39:06.802 | Pengujian 1 1-11 | Pengujian 1 | 1945 | ✓ | 28652 | 124 | 1938 |
| 2 | 01:39:06.801 | Pengujian 1 1-10 | Pengujian 1 | 1964 | ✓ | 28650 | 124 | 1951 |
| 3 | 01:39:06.801 | Pengujian 1 1-9 | Pengujian 1 | 1977 | ✓ | 28655 | 124 | 1972 |
| 4 | 01:39:06.796 | Pengujian 1 1-3 | Pengujian 1 | 2012 | ✓ | 28648 | 124 | 1982 |
| 5 | 01:39:06.793 | Pengujian 1 1-2 | Pengujian 1 | 2016 | ✓ | 28658 | 124 | 2008 |
| 6 | 01:39:06.803 | Pengujian 1 1-12 | Pengujian 1 | 2010 | ✓ | 28667 | 124 | 2005 |
| 7 | 01:39:06.798 | Pengujian 1 1-5 | Pengujian 1 | 4667 | ✓ | 28652 | 124 | 4662 |
| 8 | 01:39:06.799 | Pengujian 1 1-8 | Pengujian 1 | 4694 | ✓ | 28652 | 124 | 4688 |
| 9 | 01:39:06.814 | Pengujian 1 1-22 | Pengujian 1 | 4773 | ✓ | 28650 | 124 | 4761 |
| 10 | 01:39:06.806 | Pengujian 1 1-15 | Pengujian 1 | 4788 | ✓ | 28660 | 124 | 4787 |
| 11 | 01:39:06.797 | Pengujian 1 1-4 | Pengujian 1 | 4891 | ✓ | 28658 | 124 | 4891 |
| 12 | 01:39:06.799 | Pengujian 1 1-6 | Pengujian 1 | 4963 | ✓ | 28655 | 124 | 4963 |
| 13 | 01:39:06.793 | Pengujian 1 1-1 | Pengujian 1 | 5005 | ✓ | 28650 | 124 | 4999 |
| 14 | 01:39:06.805 | Pengujian 1 1-14 | Pengujian 1 | 5171 | ✓ | 28656 | 124 | 5165 |
| 15 | 01:39:06.807 | Pengujian 1 1-16 | Pengujian 1 | 8439 | ✓ | 28658 | 124 | 8431 |
| 16 | 01:39:06.812 | Pengujian 1 1-21 | Pengujian 1 | 8782 | ✓ | 28652 | 124 | 8776 |
| 17 | 01:39:06.804 | Pengujian 1 1-13 | Pengujian 1 | 8947 | ✓ | 28648 | 124 | 8921 |
| 18 | 01:39:06.799 | Pengujian 1 1-7 | Pengujian 1 | 9008 | ✓ | 28653 | 124 | 8996 |
| 19 | 01:39:06.821 | Pengujian 1 1-30 | Pengujian 1 | 12419 | ✓ | 28660 | 124 | 12407 |
| 20 | 01:39:06.822 | Pengujian 1 1-31 | Pengujian 1 | 12450 | ✓ | 28648 | 124 | 12449 |
| 21 | 01:39:06.811 | Pengujian 1 1-19 | Pengujian 1 | 12518 | ✓ | 28654 | 124 | 12517 |
| 22 | 01:39:06.809 | Pengujian 1 1-18 | Pengujian 1 | 12601 | ✓ | 28644 | 124 | 12595 |
| 23 | 01:39:06.811 | Pengujian 1 1-20 | Pengujian 1 | 12612 | ✓ | 28648 | 124 | 12606 |
| 24 | 01:39:06.819 | Pengujian 1 1-28 | Pengujian 1 | 13665 | ✓ | 28650 | 124 | 13659 |
| 25 | 01:39:06.835 | Pengujian 1 1-44 | Pengujian 1 | 13742 | ✓ | 28656 | 124 | 13741 |

Gambar 20. Sample Time / Response Time Pengujian 2 Web Server Amazon Web Service

Hasil dari *sample time / response time*, nilai yang diambil untuk *sample time / response time* adalah *average*, karena *average* dihasilkan dari keseluruhan nilai *sample time / response time* dibagi jumlah keseluruhan *sample*. Maka nilai untuk *sample time / response time* pada tes kedua Web Server Amazon Web Service adalah 84101/ms (*millisecond*).

2. *Latency*

Nilai *latency* pada masing-masing *sample* bisa dilihat pada gambar 5.90, untuk mencari nilai rata-rata *latency*, dari keseluruhan *sample* harus dihitung secara *manual* dengan menghitung Rata-rata

$$Latency = \frac{\text{Jumlah Seluruh Nilai Latency}}{\text{Jumlah Sample}}$$

maka nilai dari rata-rata *latency* pada pengujian, yaitu :
pertama adalah $\frac{68.455.941}{1000} = 68.455/ms$ (*millisecond*).

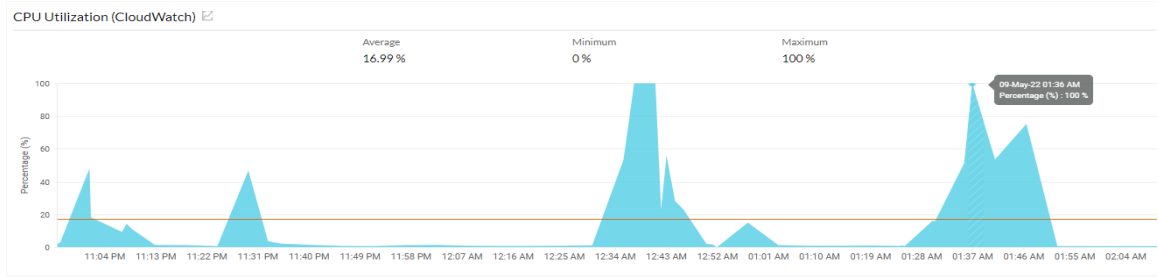
3. *Resource Utilization*

| Sample # | Start Time | Thread Name | Label | Sample Time(ms) | Status |
|----------|--------------|-------------------|-------------|-----------------|--------|
| 1 | 01:39:06.802 | Pengujian 1 1-11 | Pengujian 1 | 1945 | ✓ |
| 2 | 01:39:06.801 | Pengujian 1 1-10 | Pengujian 1 | 1964 | ✓ |
| 3 | 01:39:06.801 | Pengujian 1 1-9 | Pengujian 1 | 1977 | ✓ |
| 4 | 01:39:06.796 | Pengujian 1 1-3 | Pengujian 1 | 2012 | ✓ |
| 5 | 01:39:06.793 | Pengujian 1 1-2 | Pengujian 1 | 2016 | ✓ |
| 6 | 01:39:06.803 | Pengujian 1 1-12 | Pengujian 1 | 2010 | ✓ |
| 7 | 01:39:06.798 | Pengujian 1 1-5 | Pengujian 1 | 4667 | ✓ |
| 8 | 01:39:06.799 | Pengujian 1 1-8 | Pengujian 1 | 4694 | ✓ |
| 9 | 01:39:06.814 | Pengujian 1 1-22 | Pengujian 1 | 4773 | ✓ |
| 990 | 01:39:07.542 | Pengujian 1 1-752 | Pengujian 1 | 114600 | ✗ |
| 991 | 01:39:07.576 | Pengujian 1 1-785 | Pengujian 1 | 114572 | ✗ |
| 992 | 01:39:07.585 | Pengujian 1 1-795 | Pengujian 1 | 114569 | ✗ |
| 993 | 01:39:07.579 | Pengujian 1 1-788 | Pengujian 1 | 114703 | ✗ |
| 994 | 01:39:07.530 | Pengujian 1 1-739 | Pengujian 1 | 114752 | ✗ |
| 995 | 01:39:07.574 | Pengujian 1 1-783 | Pengujian 1 | 114714 | ✗ |
| 996 | 01:39:07.583 | Pengujian 1 1-793 | Pengujian 1 | 114711 | ✗ |
| 997 | 01:39:07.569 | Pengujian 1 1-779 | Pengujian 1 | 114725 | ✗ |
| 998 | 01:39:07.555 | Pengujian 1 1-765 | Pengujian 1 | 114739 | ✗ |
| 999 | 01:39:07.570 | Pengujian 1 1-780 | Pengujian 1 | 114788 | ✗ |
| 1000 | 01:39:07.553 | Pengujian 1 1-763 | Pengujian 1 | 114864 | ✗ |

Gambar 22. Run Test Selesai Pengujian 2 Web Server Amazon Web Service

Pengujian dilakukan dari *sample* nomor 1 mulai berjalan pada pukul 01:39:06 (gambar 24), sampai *sample* terakhir atau *sample* nomor 700 berakhir pada pukul 01:39:07 (gambar 25). Maka dari itu server monitoring yang diperhatikan pada penggunaan *resource utilization* dalam interval waktu tersebut.

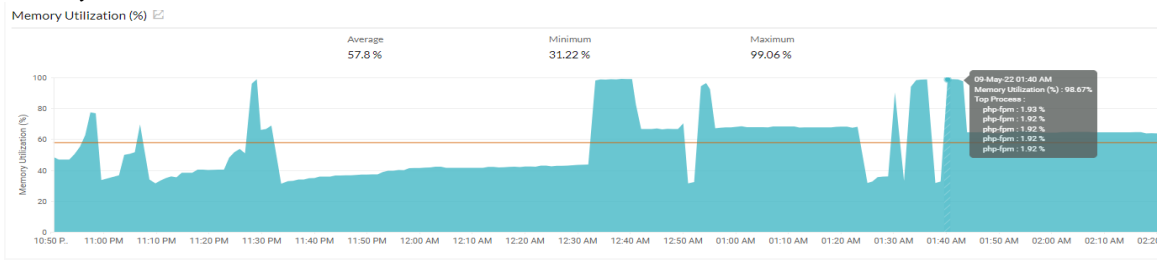
a. *CPU Utilization*



Gambar 23. CPU Utilization Pengujian 2 Web Server Amazon Web Service

Menunjukkan penggunaan *CPU Utilization* mengalami lonjakan pada angka 100%.

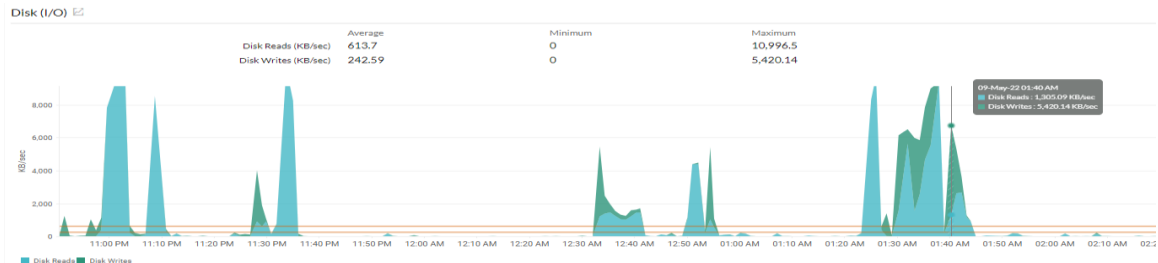
b. *Memory Utilization*



Gambar 24. Memory Utilization Pengujian 2 Web Server Amazon Web Service

Menunjukkan penggunaan *Memory Utilization* pengujian 2 Web Server Amazon Web Service sebesar 98.67%.

c. *Disk I/O Utilization*



Gambar 25. Disk I/O Utilization Pengujian 2 Web Server Amazon Web Service

Penggunaan *Disk I/O Utilization* menunjukkan untuk *Disk Reads*: 1305 kb/sec dan *Disk Writes*: 5420 kb/sec.

Pengujian telah dilakukan pada Web Server Amazon Web Service terdiri dari 2 tahap pengujian dengan *thread / user* yang berbeda-beda. Maka hasil pengujiannya adalah sebagai berikut:

Tabel 1. Hasil Pengujian Web Server Amazon Web Service

| Test | Throughput (second) | Sample/ Response Time (Millisecond) | Latency (Millisecond) | Resource Utilization | | | | Sample (User) | Error Sample (%) |
|------|------------------------|--|--------------------------|----------------------|---------------|------------------|-------------------|------------------|------------------------|
| | | | | CPU (%) | Memory (%) | Disk I/O | | | |
| | | | | | | Read (kb/sec) | Write (kb/sec) | | |
| 1 | 3.8 | 12122 | 1.601 | 46.76 | 98.78 | 905.64 | 3.103 | 500 | 0 |
| 2 | 8.6 | 84101 | 68.455 | 100 | 98.6 | 1305 | 5420 | 1000 | 64.90 |

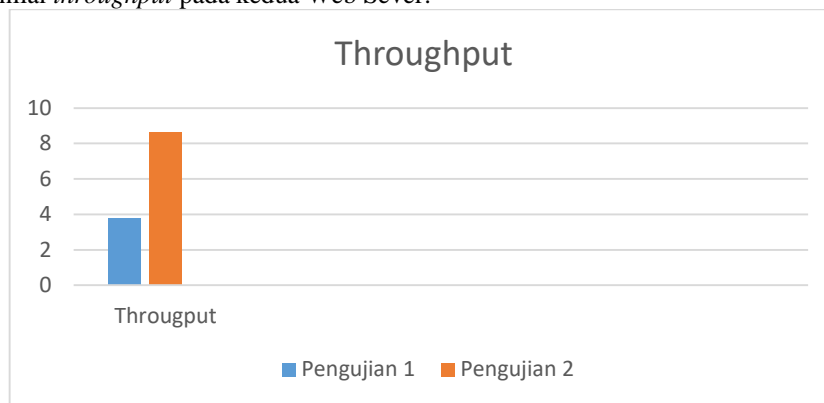
Pengujian pertama *sample / user* mengalami *error* sebesar 0% dari total *sample* 500 *user*, sehingga dari 500 *sample / user* tidak ada *sample* yang mengalami kegagalan. Sedangkan pada pengujian kedua *sample / user* mengalami *error* sebesar 64.90% dari total *sample* 1000, sehingga dari 1000 *sample / user* ada 649 *sample* yang mengalami kegagalan.

B. Pembahasan

Hasil yang didapatkan saat pengujian pada Web Server Amazon Web Service akan dilakukan analisis nilai yang didapatkan pada setiap parameter. Berikut analisis pengujian berdasarkan parameter yang diujikan:

1. Throughput

Berikut diagram nilai *throughput* pada kedua Web Sever:

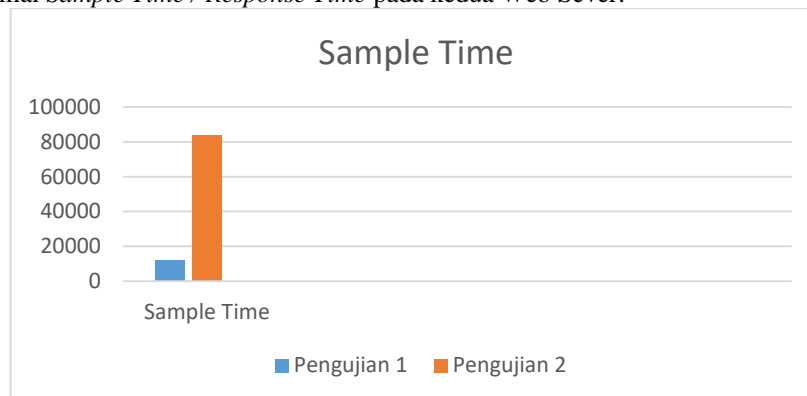


Gambar 26. Diagram Throughput Kedua Web Server

Pada diagram (gambar 29) diatas pada pengujian 1 Amazon Web Service unggul dengan *throughput* 3,8/second. Dan pada *throughput* pada pengujian 2 sebesar 8,6/second. Sehingga dengan hasil tersebut Pengujian 2 lebih unggul dalam parameter *throughput*.

2. Sample Time / Response Time

Berikut diagram nilai *Sample Time / Response Time* pada kedua Web Sever:

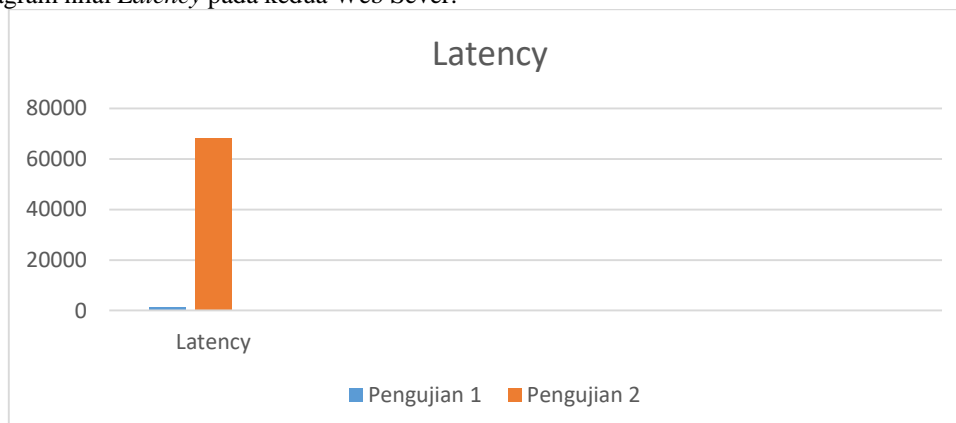


Gambar 27. Diagram Sample Time / Response Time Kedua Web Server

Pada diagram (gambar 30) diatas pada pengujian 1 dengan *Sample Time / Response Time* 12122/ms, dan pada pengujian 2 dengan *Sample Time / Response Time* 84101/ms, Sehingga dengan hasil tersebut Pengujian 2 lebih unggul dalam parameter *Sample Time / Response Time*.

3. Latency

Berikut diagram nilai *Latency* pada kedua Web Sever:



Gambar 28. Diagram Latency Kedua Web Server

Pada diagram (gambar 31) diatas pengujian 1 dalam *latency* dengan nilai 1601/ms, pada pengujian 2 dengan nilai 68455/ms. Sehingga dengan hasil tersebut Pengujian 2 lebih unggul dalam parameter *latency*.

5. KESIMPULAN

Berdasarkan hasil pengujian terhadap web beritasamarinda.com dengan metode load testing ini maka dapat diambil kesimpulan Secara keseluruhan, sebagian besar skenario yang telah dilakukan pengujian telah mencapai goals dimana pengujian pertama server masih berjalan lancar setelah dikirim 500 user dalam periode waktu 10 detik. Sedangkan pada pengujian kedua server sudah tidak sanggup/down setelah dikirim 1000 user dalam periode 1 detik.

6. DAFTAR PUSTAKA

- Alamsyah, I. (2019). Pemanfaatan Jmeter Untuk Pengujian Website Dengan Metode Performance Testing. *Ijns.Org Indonesian Journal on Networking and Security -*, 1(1), 1–10.
- Perkasa, M. I., & Setiawan, E. B. (2018). Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token. *Jurnal ULTIMA Computing*, 10(1), 19–26. <https://doi.org/10.31937/sk.v10i1.838>
- Agustika, C. P., Saputra, W. S., & Idhom, M. (2021). Pengujian Aplikasi Greenwallet dengan Metode Load Testing dan Apache Jmeter. *Jurnal Informatika Dan Sistem Informasi (JIFoSI)*, 2(2), 190–195.
- Qustoniah, A., & Darwanto. (2011). Manajemen Bandwidth Jaringan Komputer Menggunakan Metode Hierarchical Token Bucket (HTB) Pada PC Router Berbasis Linux.
- AWS EC2. (2021). Fitur Amazon EC2. <https://aws.amazon.com/id/ec2/features/>
- AWS. (2021). Apa Itu AWS. In *Aws*. <https://aws.amazon.com/id/what-is-aws/>
- Riskiono, S. D., & Darwis, D. (2020). Peran Load Balancing Dalam Meningkatkan Kinerja Web Server Di Lingkungan Cloud. *Krea-TIF*, 8(2), 1. <https://doi.org/10.32832/kreatif.v8i2.3503>
- Rahmatulloh, A., & MSN, F. (2017). Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 3(2), 241–248. <https://doi.org/10.25077/teknosi.v3i2.2017.241-248>
- Apriantsani Budiman, D., Maulana Nugraha, D., & Mardira Indonesia, S. (2019). Aplikasi Raport Online Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus Di SMK Angkasa 1 Margahayu). *Jurnal Computech & Bisnis*, 13(2), 112–121.
- Riskiono, S. D., & Darwis, D. (2020). Peran Load Balancing Dalam Meningkatkan Kinerja Web Server Di Lingkungan Cloud. *Krea-TIF*, 8(2), 1. <https://doi.org/10.32832/kreatif.v8i2.3503>
- Apriantsani Budiman, D., Maulana Nugraha, D., & Mardira Indonesia, S. (2019). Aplikasi Raport Online Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus Di Smk Angkasa 1 Margahayu). *Jurnal Computech & Bisnis*, 13(2), 112–121.
- Jayanti, D. E., Umar, R., & Riadi, I. (2020). Implementation of Cloudflare Hosting for Access Speed on Trading Websites. *Sisfotenika*, 10(2), 227. <https://doi.org/10.30700/jst.v10i2.962>
- Sibarani, D. (2019). Sistem Informasi Makanan Menggunakan CMS Wordpress. 3–10. <http://eprints.akakom.ac.id/id/eprint/8758>
- Tuara, H. A., Maridyah, N., & Khaerudin, K. (2021). Implementasi CDN (Content Delivery Network) Menggunakan Cloudflare terintegrasi Dengan Docker Container. *Journal of Mechatronic and Electrical Engineering*, 1(1), 42–51.
- Zakaria, F. S., Farid S, M., & Megiyanto R, G. (2020). Infrastruktur Jaringan Menggunakan Server Web Hosting Centos 6 Sebagai Server Aplikasi Monitoring Perkebunan. *Prosiding The 11th Industrial Research Workshop and National Seminar*, October, 5.
- Lukman, & Yudhiastari, M. (2021). Analisis Kinerja Web Server Apache Dan Litespeed Menggunakan Httpperf Pada Virtual Private Server (Vps). *Respati*, 16(2), 24–32.
- Ezzat, P., & Ahmad, A. (2019). Cloud Computing In Amazon Web Services, Microsoft Windows Azure, Google App Engine and IBM Cloud Platforms: A Comparative Study.
- Yamani, S. Q., Saedudin, R. R., & Almaarif, A. (2020). Analisis Dan Benchmarking Disk Performance Pada Windows Azure Virtual Machine (Vm) Dan Amazon Web Service EC2 Analysis and Benchmarking Disk Performance on Windows Azure Virtual Machine (VM) and Amazon Web Service EC2. *EProceedings ...*, 7(2), 6619–6633. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/12474>
- Chandra, A. Y. (2019). Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request. *Jurnal Sistem Dan Informatika (JSI)*, 14(1), 48–56. <https://doi.org/10.30864/jsi.v14i1.248>
- Intan Permatasari, D., & Santoso, B. (2019). Pengukuran Throughput Load Testing Menggunakan Test Case Sampling Gorilla Testing. *Seminar Nasional Sistem Informasi*, 4, 2008–2014.
- Riza, F. (2018). Analisis Pengujian Kinerja Web Server Yang Dibangun Atas Penyedia Layanan Virtual Private Server (Vps) Antara Digital Ocean Dan Vultr.
- Muhammad, F., Saedudin, R., & Almaarif, A. (2020). Analisis Performansi Metrik CPU Dan Memory Pada Windows Azure Virtual Machine (VM) dan Amazon Web Service Elastic Compute Cloud (EC2). *EProceedings ...*, 7(2), 6975–6983. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/12660>
- Sudaryono, S., Aryani, D., & Ningrum, I. T. (2012). Cloud Computing: Teori Dan Implementasinya Dalam Dunia Bisnis Dan Pemasaran. *CCIT Journal*, 5(2), 145–167. <https://doi.org/10.33050/ccit.v5i2.147>
- Permatasari, D. I. (2020). Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 8(1), 135. <https://doi.org/10.26418/justin.v8i1.34452>
- Susilo, I., & Nugraha, G. K. (2012). Pembangunan Web Server Menggunakan Debian Server Untuk Media Pembelajaran Di Sekolah Menengah Kejuruan (SMK) Negeri 1 Sragen. *Indonesian Journal on Networking and Security (IJNS)*, 2(1), 22–27.

- Kurnia, I. (2020). Konfigurasi Hosting Server Menggunakan Centos 7 Pada Dinas Komunikasi dan Informatika Kabupaten Pesawaran. *Jurnal Informatika Software Dan Network*, Vol 01, No(01), 26–32. <http://jurnal.dccpringsewu.ac.id/index.php/ji/article/view/13>
- Qamal, M., Hamdhana, D., & Pratomo, R. (2019). Website Media Pembelajaran Online. 3, 319–327.
- Kamarudin, Kusri, S. A. (2018). Uji kinerja sistem web service pembayaran mahasiswa menggunakan apache jmeter (studi kasus: Universitas amikom yogyakarta). *Teknologi Informasi*, XIII(1), 44–52.
- Sugiyanta, L., & Nurahma, W. A. (2017). Analisis Perbandingan Antara Colocation Server Dengan Amazon Web Services (Cloud) Untuk Usabilitas Portal Swa.co.id Di PT. Swa Media Bisnis). *PINTER : Jurnal Pe*
- Rahmatulloh, A., & MSN, F. (2017). Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 3(2), 241–248. <https://doi.org/10.25077/teknosi.v3i2.2017.241-248>
- Ahmad, A., & Setiawan, H. (2011). Cloud Computing : Solusi ICT ? *Sistem Informasi*, 3(6), 1–5. <https://ejournal.unsri.ac.id/index.php/jsi/article/view/736>