

LOAD BALANCING WEB SERVER BERBASIS CLOUD DENGAN MENGGUNAKAN ALGORITMA ROUND-ROBIN PADA SAMPOERNA UNIVERSITY

Febby Triadi Adidrajat¹; Astriana Mulyani²

Program Studi Teknik Informatika¹, Program Studi Teknik Informatika²

STMIK Nusa Mandiri¹, STMIK Nusa Mandiri²

febby.adidrajat@gmail.com¹, astriana.atm@nusamandiri.ac.id²

Abstrak

Abstrak – Sampoerna University terdapat banyak aplikasi dan *website* salah satunya Sistem Informasi Penerimaan Mahasiswa Baru. Sistem Informasi Penerimaan Mahasiswa Baru adalah *website* yang melayani pendaftaran bagi calon mahasiswa baru. *Website* tersebut menggunakan *server biznet cloud*. Pada *web server website* tersebut tidak menggunakan *load balancer*. Hal ini menyebabkan terjadi *down time* pada *web server* ketika *website* diakses oleh banyak *user*. Karena daya tampung sebuah *web server* terhadap *request* (permintaan) menjadi tolak ukur kehandalan sebuah *website*. Terutama pada *website* dengan jumlah pengunjung yang sangat banyak seperti *website* Penerimaan Mahasiswa Baru membagi *traffic request* (beban) diperlukan *web server* tambahan, dimana masing-masing *server* bekerja dalam sebuah konsep *load balancing* dengan algoritma *round-robin*. Dengan teknologi *cloud computing* akan sangat banyak membantu dari segi kecepatan dan mengurangi/menghilangkan *down time* sebuah *web server* yang dialami oleh Sampoerna University.

Kata Kunci: *Website, web server, load blancing,, Round-Robin*

I. PENDAHULUAN

Latar belakang

Web server sebagai induk penyedia *service website* berperan penting dalam sebuah *website*. Seiring berjalannya waktu dan perkembangan *internet*, saat ini jumlah pengunjung *website* meningkat tajam. Tidak hanya *public website*, tetapi hal ini berlaku juga pada aplikasi atau sistem informasi berbasis *website*. Kemampuan sebuah *web server* untuk menampung, melayani, dan memproses *request* dari *client* menjadi tolak ukur kehandalan sebuah *website* tujuannya dapat mengurangi atau menghilangkan *down time*.

Pada Sampoerna University terdapat banyak aplikasi dan *website* salah satunya Sistem Informasi Penerimaan Mahasiswa Baru. Sistem Informasi Penerimaan Mahasiswa Baru adalah *website* yang melayani pendaftaran bagi calon mahasiswa baru. *Website* tersebut menggunakan *server biznet cloud*. Pada *web server website* tersebut tidak menggunakan *load balancer*. Hal ini dikhawatirkan terjadi *down time* pada *web server* ketika *website* diakses oleh banyak *user*.

Setelah terhubung secara fisik, *protocol* TCP/IP (*networking protocol*) yang memungkinkan semua komputer dapat berkomunikasi antar satu dengan lainnya. Pada saat aplikasi *browser* meminta data *web page* ke *server* maka instruksi permintaan data oleh *browser* tersebut dikemas dalam TCP yang merupakan *protocol transport* dan dikirim ke alamat yang merupakan *protocol* berikutnya yaitu *hyper text transfer protocol* (HTTP). Data yang diparsing dari *browser* ke *web server* disebut sebagai HTTP *request* yang meminta halaman *web* dan kemudian *web server* akan mencari data HTML yang dibutuhkan dan dikemas dalam TCP *protocol* kemudian dikirim kembali ke *browser*. Data yang dikirim dari *server* ke *browser* disebut sebagai HTTP *response*. Jika data yang diminta oleh *browser* tidak ditemukan pada *web server* maka akan menampilkan *error* pada *web page* yaitu *Error: 404 Page Not Found.*"

Masalah

Pada tahun ajaran 2019 diprediksi calon mahasiswa akan meningkat, tentunya dibutuhkan *web server* yang dapat menampung *request* dalam jumlah banyak.

Batasan

penyediaan *web server* berbasis *cloud*, penerapan *load balancing* pada *server* menggunakan algoritma *Round-*

Robin, hingga tahap pengujian menggunakan *apache benchmark* pada Sampoerna University.

Tujuan

Dalam penelitian ini bertujuan untuk merancang suatu *web server load balancer* pada Sampoerna University.

Manfaat

Membantu pihak universitas agar web server tidak down dengan membagi akses pada servernya

II. LITERATUR DAN METODE

Tentang Cloud Computing, menurut (Husain, Zarlis, Nasution, Sihotang, & Wahyuni, 2018) menyatakan :

“*Cloud Computing* adalah suatu ilmu sub komputasi atau sebuah metoda komputasi di mana kapabilitas tentang teknologi informasi tersaji sebagai suatu layanan (*as a service*), sehingga pengguna dapat mengaksesnya lewat Internet tanpa mengetahui apa yang ada didalamnya yang mempunyai kendali terhadap sebuah infrastruktur teknologi yang dapat membantunya”.

Menurut (Novianta & Setyaningsih, 2015) menyatakan bahwa : “*Web server* merupakan *software* yang memberikan layanan data, berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML, konsep *web server* antara lain:

a) *Web server* merupakan mesin aplikasi atau *software* yang beroperasi dalam mendistribusikan *web page* ke *user*, tentu saja sesuai dengan permintaan *user*.

b) Hubungan antara *web server* dan *browser internet* merupakan gabungan atau jaringan komputer yang berada diseluruh dunia.

Mengenai beban *web server*, (Rahmatulloh & MSN, 2017) berpendapat : “Saat ini Universitas Siliwangi memiliki populasi pengguna Sistem Informasi Akademik sebesar 12829 orang yang terdiri dari mahasiswa, karyawan dan dosen. Dengan arsitektur server tunggal saat ini, sering terjadi overload jika banyak request user secara bersamaan seperti pada kegiatan pengisian kartu rencana studi mahasiswa. Hal ini menimbulkan kondisi server down karena matinya aplikasi web server dan database server sehingga banyak user yang tidak dapat dilayani dengan kondisi server tunggal”.

A. Konsep dasar jaringan

Menurut [Novrianda, 2017] “Jaringan komputer (*computer network*) adalah suatu himpunan interkoneksi sejumlah komputer autonomus”. Para ahli kemudian membagi jaringan komputer berdasarkan beberapa klasifikasi, diantaranya:

1. Local Area Network

Menurut [Pratiwi & Akbi, 2019] “Local Area Network (LAN) adalah jaringan yang mempunyai sifat internal seperti hanya milik pribadi dan area jangkauan terbatas”. Suatu LAN mempunyai karakteristik khusus diantaranya, yaitu: Ruang lingkup geografis yang terbatas, kecepatan pengiriman data relatif tinggi (1-100 Mbps), pemilikan dan pengoperasian dilakukan oleh perusahaan yang bersangkutan.

2. Metropolitan Area Network

Pendapat [Wongkar, Sinsuw, & Xaverius, 2015] menjelaskan bahwa “MAN singkatan dari metropolitan area network. Jenis jaringan komputer MAN ini adalah suatu jaringan komputer dalam suatu kota dengan transfer data berkecepatan tinggi yang menghubungkan suatu lokasi seperti sekolah, kampus, perkantoran dan pemerintahan”.

MAN memiliki jarak jangkauan antara 10 KM sampai 50 KM. Wilayah jangkauan MAN dapat mencakup sebuah wilayah kota, yang didalam nya terdapat banyak gedung dan pemukiman. Ini berarti di dalam sebuah MAN telah terintegrasi banyak LAN yang berasal dari gedung dan pemukiman yang ada.

3. Wide Area Network

Menurut Aditya dalam [Idrus, 2016] “WAN (Wide Area Network) adalah kumpulan dari LAN dan/atau Workgroup yang dihubungkan dengan menggunakan alat komunikasi modem dan jaringan Internet, dari/ke kantor pusat dan kantor cabang, maupun antar kantor cabang”. Dengan sistem jaringan ini, pertukaran data antar kantor dapat dilakukan dengan cepat serta dengan biaya yang relatif murah.

B. Cloud Computing Network

Dikutip dari (Adi, 2019) “Komputasi awan atau *Cloud Computing* mengacu pada paradigma dalam teknologi informasi di mana komponen perangkat keras dan perangkat lunak di-*outsourcing* ke sistem di luar infrastruktur TI *in-house*. *Cloud Computing* memungkinkan akses, pengeditan, dan pembagian data pada sumber daya terdistribusi yang membentuk awan”.

C. Topologi Jaringan

Menurut (Halawa, 2016:3) “Topologi adalah suatu aturan/*rules* bagaimana menghubungkan komputer (*node*) satu sama lain secara fisik dan pola hubungan antara komponen-komponen yang berkomunikasi melalui media/peralatan jaringan, seperti : *server, workstation, hub/switch*, dan pengabelannya, sedangkan jaringan

merupakan sebuah sistem yang terdiri atas komputer, perangkat komputer, tambahan dan perangkat jaringan lainnya yang saling berhubungan dengan menggunakan media tertentu dengan aturan yang sudah ditetapkan”.

Dari sekian banyak topologi, saat ini yang *familiar* pada umumnya berkisar pada empat bentuk topologi jaringan komputer, yaitu :

1. Topologi Bus

Menurut (Rofii, Hunaini, & Sholawati, 2018) “Topologi bus adalah jenis topologi yang menggunakan saluran tunggal sebagai media transmisinya yang menghubungkan semua client dengan server”. Topologi bus hanya menggunakan sebuah jalur koneksi yang digunakan secara bersama-sama oleh beberapa komputer dan perangkat jaringan komputer terhubung lainnya.

2. Topologi Ring

Menurut (Pratama & Dharmesta, 2018) ”Topologi Ring adalah sebuah topologi jaringan dimana setiap komputer yang terhubung membentuk suatu lingkaran”. Topologi jaringan ini hanya menghubungkan setiap komputer satu per satu, sehingga membentuk sebuah rangkaian menyerupai cincin (*ring*).

3. Topologi Star

Menurut (Pratama & Dharmesta, 2018) “Topologi *star* merupakan sebuah topologi yang setiap komputer pada jaringan tersebut ke sebuah pusat jaringan. Pusat jaringan ini bisa berupa switch, hub atau komputer lainnya. Masing-masing komputer yang terlibat tidak saling terhubung sehingga semua komunikasi harus melewati pusat jaringan terlebih dahulu.

4. Topologi Tree

Menurut (Pratama;Marlinda, 2015) “Topologi tree adalah topologi yang merupakan generalisasi dari topologi bus, media transmisi berupa kabel yang bercabang tanpa loop tertutup. Topologi tree selalu dimulai pada titik yang disebut *headend*. Satu atau beberapa kabel berasal dari *headend*”.

D. Perangkat Lunak Jaringan

Pada jaringan komputer terdapat banyak sekali tool/software pendukung. Masing-masing software memiliki fungsi yang berbeda. Kebanyakan software tersebut berfungsi untuk memonitor sebuah jaringan.

1. Microsoft Software Monitor

Microsoft Network Monitor merupakan *software* yang dapat digunakan untuk mengetahui lalu lintas data yang sedang dikirim dan diterima melalui jaringan komputer maupun dari file data yang diambil sebelumnya sehingga dapat dilakukan analisa. Software ini menyediakan pilihan penyaringan untuk analisis kompleks mengenai jaringan data.

2. Dude

[The Dude adalah](#) sebuah *software* yang memudahkan seorang admin jaringan memonitoring jaringan dan mendukung berbagai protokol jaringan, seperti SNMP, ICMP, DNS, dan [TCP](#). Cara kerja The Dude adalah scan otomatis termasuk perangkat jaringan berbasis subnet, memetakan jaringan computer, peringatan otomatis pada saat ditemukan *error* maupun saat *troubleshooting*.

3. Angry IP Scanner

Software Angry IP Scanner paling banyak digunakan oleh network engineer. Selain ringan software ini dapat digunakan pada berbagai flatform seperti windows, linux, dan Macintosh. Fungsi Angry IP Scanner adalah melakukan scan port dan IP, mendeteksi serangan peretas dan lain-lain.

E. Manajemen Jaringan

Menurut Wasita dalam (Sufian, 2017) manajemen jaringan adalah sebuah pekerjaan untuk memelihara seluruh sumber jaringan dalam keadaan baik, karena saat ini jaringan sangat kompleks, dinamik dan terdiri atas komponen yang tidak dapat diandalkan sepenuhnya, peralatan yang baik diperlukan untuk mengelola jaringan tersebut.

a. TCP/IP dan Subnetting

1. TCP/IP

TCP/IP pada awalnya dikembangkan oleh suatu Departmen Pertahanan (*Departmen of Defence / DOD*) di amerika. Dalam risetnya mampu merancang hubungan antar jaringan yang berbeda. Seperti halnya *Refensi Model OSI* TCP/IP juga mempunyai urutan standarisasi layer tersendiri, untuk lebih jelasnya urutan tersebut adalah sebagai berikut:

Dalam TCP/IP dikenal 3 alamat yakni: physical address, IP address dan port address.

a. *Physical*

Physical address kerap disebut sebagai *MAC address*, yaitu alamat yang dimiliki oleh *NIC (LAN card)* besarnya 32 digit angka Heksadesimal, *Physical address* mempunyai sifat tetap dan tidak berubah-ubah.

b. *IP Address*

IP address terdiri atas dua bagian yaitu *network ID* dan *host ID*, dimana *network ID* menentukan alamat jaringan komputer, sedangkan *host ID* menentukan alamat host (komputer, *router*, *switch*). Oleh sebab itu *IP address* memberikan alamat lengkap suatu *host* beserta alamat jaringan di mana *host* itu berada. *IP address* sendiri dibagi dalam tiga kelas:

1) *IP address* kelas A diberikan untuk jaringan dengan jumlah host yang sangat besar. *Range IP* 1.xxx.xxx.xxx. – 126.xxx.xxx.xxx, terdapat 16.777.214 (16 juta) *IP address* pada tiap kelas A. *IP address* kelas A diberikan untuk jaringan.jumlahhost yang sangat besar.

2) *IP address* kelas B biasanya dialokasikan untuk jaringan berukuran sedang dan besar. Pada *IP address* kelas B, *network ID* ialah 16 bit pertama, sedangkan *host ID* ialah 16 bit berikutnya. Dengan panjang *host ID* 16 bit, *network* dengan *IP address* kelas B dapat menampung sekitar 65000 host. *Range IP* 128.0.xxx.xxx – 191.155.xxx.xxx.

3) *IP address* kelas C awalnya digunakan untuk jaringan berukuran kecil (LAN). *Host ID* ialah 8 bit terakhir. Dengan konfigurasi ini, bisa dibentuk sekitar 2 juta *network* dengan masing-masing *network* memiliki 256 *IP address*. *Range IP* 192.0.0.xxx – 223.255.255.x.

c. Port Address

Port adalah mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya dan program di dalam jaringan. *Port* dapat mengidentifikasi aplikasi dan layanan yang menggunakan koneksi di dalam jaringan TCP/IP. Sehingga, *port* juga mengidentifikasi sebuah proses tertentu di mana sebuah server dapat memberikan sebuah layanan kepada klien atau bagaimana sebuah klien dapat mengakses sebuah layanan yang ada dalam server.

2. Subnetting

Dengan semakin pesatnya perkembangan teknologi informasi, penggunaan internet tidak hanya digunakan oleh perusahaan besar atau lembaga pemerintahan saja yang menggunakan jasa internet namun perusahaan kecil, lembaga-lembaga pendidikan baik formal maupun non formal juga sudah menggunakan *internet*, oleh sebab itu pula persediaan IP Address semakin berkurang. Sebuah *Netmask* membantu mengetahui bagian mana dari alamat mengidentifikasi jaringan dan bagian mana dari alamat mengidentifikasi node.

Kelas A , jaringan B , dan C memiliki *Defaut mask*, seperti yang ditunjukkan di sini :

Class A: 255.0.0.0 Class B: 255.255.0.0 & Class C: 255.255.255.0

Alamat IP pada jaringan kelas A yang belum disubnet akan memiliki alamat / netmask pasangan mirip dengan : 8.20.15.1 255.0.0.0. Untuk melihat bagaimana netmask membantu dalam mengidentifikasi jaringan dari *alamat IP* , yaitu dengan cara mengkonversi alamat dan netmask menjadi bilangan biner.

8.20.15.1 = 00001000.00010100.00001111.00000001

255.0.0.0 = 11111111.00000000.00000000.00000000

Setelah memiliki alamat dan netmask diwakili dalam biner , maka identifikasi jaringan dan *host ID* menjadi lebih mudah. Alamat bit yang telah sesuai netmask pada bit set ke 1 mewakili ID jaringan. Dan bit yang sesuai netmask pada bit set 0 mewakili *node ID* .

8.20.15.1 =

00001000.00010100.00001111.00000001

8.20.15.2 255.0.0.0 =

11111111.00000000.00000000.00000000

netid = 00001000 = 8

hostid = 00010100.00001111.00000001 = 20.15.1

Subnetting memungkinkan untuk membuat beberapa jaringan logis yang ada dalam Kelas A, B, atau C jaringan tunggal. Jika tidak disubnet, kita hanya dapat menggunakan satu jaringan dari Kelas A, B, atau C, yang tidak realistis. Setiap data yang menghubungkan pada jaringan harus memiliki ID jaringan yang unik, dengan setiap simpul pada link menjadi anggota jaringan yang sama. Jika kita membagi jaringan utama (Kelas A, B, atau C) menjadi subnetwork yang lebih kecil, memungkinkan untuk membuat jaringan interkoneksi subnetwork.

Setiap data link pada jaringan ini akan memiliki ID jaringan atau subnetwork yang unik. Setiap perangkat, atau *gateway*, yang menghubungkan jaringan dan subnetwork memiliki alamat IP yang berbeda, satu untuk setiap jaringan atau subnetwork bahwa interkoneksi. Dalam rangka untuk subnet jaringan, memperpanjang netmask alami dengan beberapa bit dari *host ID* bagian dari alamat untuk membuat ID subnetwork. Misalnya, diberikan jaringan Kelas C dari 204.17.5.0 yang memiliki *netmask* 255.255.255.0

F. Konsep Penunjang Usulan

1. Google Cloud Platform

Google Cloud Platform (GCP) merupakan salah satu *cloud computing services* yang menyediakan *cloud server*. Salah satu keunggulan dari GCP adalah gratis penggunaan untuk 365 hari. Pengguna dapat menginstall

operating system berbayar seperti *windows server* maupun *open sources* seperti *Linux* pada *compute engine*. *Compute engine* dapat difungsikan sebagai *web server*, maka pengguna tak lagi harus menyewa *web hosting*.

Compute engine GCP *web server* usulan akan dibuat sebagai pembandingan dengan *web server existing*.

2. Load Balancing

Menurut Herlambang dalam (Sukendar, Toni, Ikhsan, 2017) *Load balancing* adalah suatu teknik pendistribusian beban trafik atau pengalihan beban trafik berdasarkan jaringan pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap. Konsep *load balancing* selanjutnya akan diterapkan pada *web server* usulan, sebelumnya *web server existing* tidak menggunakan *load balancing*.

3. Apache Benchmark

Menurut (Edriyansyah & Freddy, 2017) “Apache Benchmark adalah salah satu software yang di gunakan untuk mengukur performance *web server* apache, dengan memberikan gambaran seberapa banyak request yang dapat dilayani oleh *web server* yang telah di rancang atau di bangun.”

Selanjutnya *apache benchmark* akan digunakan pada *web server existing* dan *web server* usulan, sehingga didapat data yang pasti mengenai daya tampung *request* masing-masing *web server*.

4. Algoritma Round Robin

Menurut (Gani et al., 2015) “Round Robin adalah salah satu algoritma yang digunakan pada metode *load balancing* untuk membagi beban kerja. Dengan algoritma ini proses dibagi secara merata pada semua server yang saling terhubung. Setiap proses baru yang ditugaskan pada server akan masuk antrian dan urutannya disusun berdasarkan proses yang sedang berlangsung”.

III. METODE

Metode Penelitian adalah salah satu cara bagaimana memahami suatu pembahasan, permasalahan dan pemecahannya. Metode penelitian yang penulis gunakan adalah:

1. Metode Pengumpulan Data

Pengumpulan data dan informasi dilakukan selama 1 (satu) bulan yaitu dimulai sejak tanggal 26 Maret 2019 hingga 26 April 2019 dengan menggunakan beberapa metode sebagai berikut:

a) Observasi

Metode observasi ini dilakukan langsung pada IT *data center* Sampoerna University untuk melakukan pengamatan, pencatatan informasi mengenai penggunaan *web server*.

b) Wawancara

Wawancara dilakukan langsung penulis dengan penanggungjawab IT *Data Center* pada Sampoerna University yaitu Pak Hesra.

c) Studi Pustaka

Studi pustaka diambil dari berbagai jurnal penelitian, buku-buku referensi yang berhubungan dengan pembahasan, dan situs website remi.

2. Analisa Penelitian

Sebelum melakukan perancangan *load balancer web server*, penulis melakukan analisa yang terbagi menjadi beberapa bagian yaitu :

a. Analisa Kebutuhan

Dalam tahap ini penulis melakukan analisis kebutuhan apa saja yang digunakan untuk merancang *load balancing cloud web server*.

b. Desain

Pada tahap ini penulis memulai merancang skema multi *cloud web server*, *load balancing* dengan algoritma *round-robin*.

c. Testing

Untuk mengetahui kehandalan atau daya tampung *web server* digunakan *tool benchmark* yaitu *apache benchmark*.

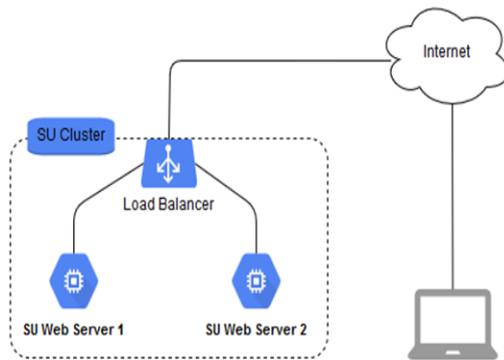
d. Implementasi

Di tahap ini akan diterapkan semua yang telah di rencanakan dan di rancang sebelumnya. Tahap penerapan implementasi ini merupakan tahap yang sangat menentukan dari berhasil atau gagal *project* yang akan di bangun.

IV. HASIL DAN PEMBAHASAN

a. Topologi Jaringan

Penulis akan menggunakan Cloud IAAS (Infrastructure as a Services) dari Google Cloud Platform. Terdapat 2 (dua) *compute engine* yang akan dijadikan *web server*. Metode HTTP *load balancing* diterapkan pada *group/cluster vm instances*.



Sumber : Dokumen pribadi.
Gambar 1 Topologi Jaringan *load balancer*

Semua HTTP *request* dari *user* diterima oleh *Cluster*, pada bagian inilah diterapkan pengaturan *load balancing round robin*. Sehingga sejumlah HTTP *request* dari *client* pada detik pertama dilayani oleh *Web Server 1*, sedangkan pada detik kedua sejumlah HTTP *request* dari *client* dilayani oleh *Web Server 2*. Skema yang sama berlaku pada detik berikutnya. Hal ini menerangkan bahwa Masing-masing *Web Server* melayani *Request* per detik

1. Keamanan Jaringan

Keamanan jaringan disediakan oleh *Google Cloud* sebagai penyedia layanan *cloud engine* dengan hanya mengaktifkan port-port yang di *request* oleh *customer*. Pada *firewall rules* terdapat izin akses http port 80 untuk *healthy check* sebagai salah satu persyaratan *backend configuration*.

2. Rancangan Jaringan

a. Pembuatan Instance VM Compute Engine

Telah dibuat dua buah VM Instances yang difungsikan sebagai web server dengan spesifikasi sebagai berikut :

- CPU : f1-micro, 1 vCPU
- RAM : 614 mb
- Hardisk : 10 gb *Standard persistent disk*
- Region : Asia-Southeast1 (Singapore)
- Zone : Asia-Southeast1-a
- OS : Linux Centos 6

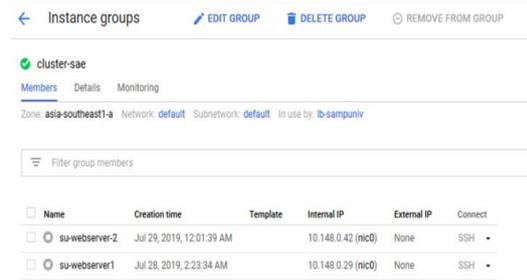
b. Instalasi Software Pendukung

Compute engine vm instances tersebut berfungsi sebagai web server pertama dengan software *httpd services*.

4. Manajemen Jaringan

a. VM Instances Grouping

Sebagai salah satu syarat *load balancing* maka perlu dibuat *group* atau *cluster* dari kedua *vm instances*.

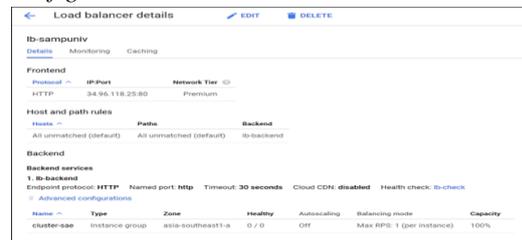


Sumber : Dokumen pribadi
Gambar 2 *Group/Cluster vm instance*

b. Pengaturan HTTP Load Balancing

Pada pengaturan inilah konsep pembagian *http request* dengan algoritma *round robin* dibagi rata pada kedua *server* menggunakan *RPS (Request per second) per server*.

Load balancer Google Cloud Flatform menggunakan *ip static* yang telah di-*reserv* sebelumnya. Pada pengaturan ini terdapat 3 (tiga) persyaratan yaitu *beckend configuration, host & path rules, dan frontend configuration*.



Sumber : Dokumen pribadi
Gambar 3 Pengaturan *Load Balancer*

c. Backend Services



Sumber : Dokumen pribadi
Gambar 4 Pengaturan *Backend Services*

Pada gambar pengaturan *backend services* diatas menggunakan *balancing mode rate*, dimana request HTTP dibagi rata 1 (satu) request per 1 (satu) VM Instance.

d. Front End Services

Pada frontend services menggunakan IP static dan dibuka port 80. Hal tersebut memungkinkan request HTTP dari user pada website www.sampuniversty.com akan di-point ke IP load balancer diteruskan pada port 80. Selanjutnya load balancer akan membagi HTTP request pada kedua VM Instance.

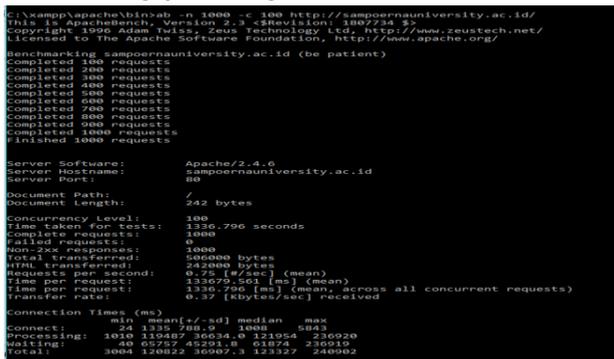


Sumber : Dokumen pribadi
Gambar 5 Frontend Configuration

e. Pointing Domain ke IP Load Balancer
Sebagai bahan perbandingan penulis membeli sebuah domain yaitu sampuniversty.com. Pada DNS godaddy di-pointing ke IP load balancer.

5. Pengujian Jaringan

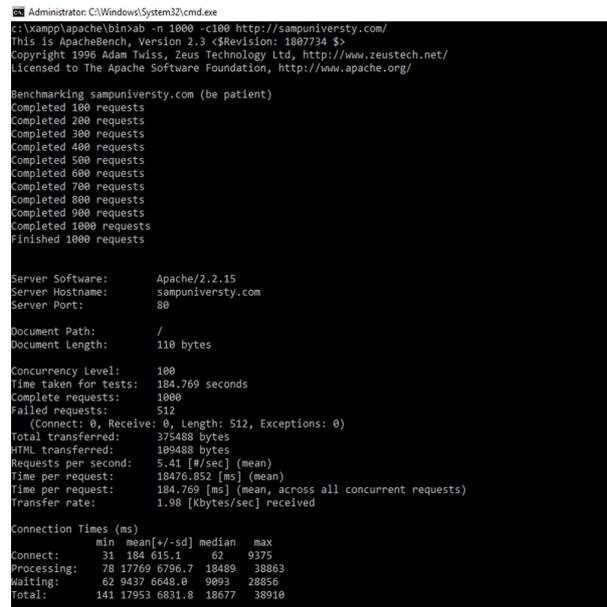
a. Pengujian Jaringan awal



Sumber : Hesra – IT Data Center Sampoerna University
Gambar 6 Benchmarking pada sampoernauniversity.ac.id

Dari gambar diatas diperoleh Request per second 0.75 ini berarti dalam satu detik web server sampoernauniversity.ac.id mampu menampung sebanyak 0.75 pengguna. Dengan kata lain satu orang pengguna dapat dilayani selama 1.3 detik.

b. Pengujian Jaringan Akhir



Sumber ; Dokumen Pribadi
Gambar 7 Benchmarking pada sampuniversty.com

Dari gambar diatas, telah dilakukan benchmarking pada sampuniversty.com diperoleh hasil request per second : 5.41. Hal tersebut menyatakan bahwa dalam satu detik web server sampuniversty.com mampu menampung sebanyak 5.41 pengguna.

V. KESIMPULAN

Setelah melakukan analisa dan perbandingan antara sampoernauniversity.ac.id dengan sampuniversty.com maka penulis berkesimpulan sebagai berikut:

- 1.Web server sampoernauniversity.ac.id dikhawatirkan tidak mampu mengakomodir jumlah pengguna yang sangat banyak.
2.Spesifikasi Web server sampoernauniversity.ac.id tidak optimal, terlalu banyak spek tinggi namun kinerjanya kurang.
3.Spesifikasi hardware web server pembanding tidak lebih tinggi dari web server sampoernauniversity.ac.id namun terbukti mampu menampung request user lebih banyak.
4.Load balancer merupakan salah satu solusi untuk untuk web server yang diakses oleh banyak pengguna.

REFERENSI

- [1] Husain, Zarlis, M., Nasution, Z., Sihotang, H. T., & Wahyuni, S. (2018). Filsafat Ilmu Komputer dan Cloud Computing secara Etimologis. *Jurnal Mantik Penusa*, 2(2), 15–21.
- [2] Novianta, M. A., & Setyaningsih, E. (2015). Sistem Informasi Monitoring Kereta Api Berbasis Web Server Menggunakan layanan GPRS. *Momentum*, 17(2), 58–67.
- [3] Rahmatulloh, A., & MSN, F. (2017). Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi. *Jurnal Teknologi Dan Sistem Informasi*, 3(2), 241. <https://doi.org/10.25077/teknosi.v3i2.2017.241-248>
- [4] Novrianda, R. (2017). Rancang Bangun Keamanan Jaringan Wireless pada STIPER Sriwigama Palembang dengan Radius Server. *Jurnal Maklumatika*, 4(1), 19–29.
- [5] Pratiwi, D. N., & Akbi, D. R. (2019). Penerapan Metode Filtering Video Streaming dan Malware Pada Jaringan Local Area Network. *Techno.Com*, 17(4), 384–394. <https://doi.org/10.33633/tc.v17i4.1820>
- [6] Wongkar, S., Sinsuw, A., & Xaverius, N. (2015). Analisa Implementasi Jaringan Internet Dengan Menggabungkan Jaringan LAN Dan WLAN Di Desa Kawangkoan Bawah Wilayah Amurang II. *E-Journal Teknik Elektro Dan Komputer*, 4(6), 62–68.
- [7] Idrus, A. (2016). Sistem Monitoring Jaringan PT. Exhibition Network Indonesia Dengan THE Dude Berbasis Mikrotik. *Informatics for Educators and Professionals*, 1(1), 84–93
- [8] Adi, H. (2019). *Konsep Dasar Komputasi Awan atau Cloud Computing*.
- [9] Halawa, S. (2016). *Topologi Bab 2.4*. 66–71.
- [10] Pratama, I. P. A. E., & Dharmesta, P. A. (2018). Implementasi Teknik Deep Packet Inspection Dengan Menggunakan Wireshark Pada Sistem Operasi Ubuntu. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, 1(2), 79–85. <https://doi.org/10.31598/jurnalresistor.v1i2.274>
- [11] Pratama;Marlinda. (2015). Perancangan Jaringan Komputer Menggunakan Aplikasi Vhp Online Reporting System. *Teknik Komputer AMIK BSI*, 1(1), 106–113.
- [12] Sufian, I. (2017). Rancang Bangun Manajemen Jaringan Dengan Memanfaatkan SNMP Pada Metropolitan Area Network (Man) Pemerintah Kota Batam. *Teknik Ibnu Sina*, 2(2), 41–47.
- [13] Sukendar, Toni, Ikhsan, S. (2017). Menjaga konektifitas Internet Agar Selalu Up Dengan Metode Fail Over Berbasis Mikrotik Pada SMA Darussalam Jakarta. *Jurnal Teknik Komputer, Vol. III(2)*, 48–52.
- [14] Edriyansyah, & Freddy, W. (2017). *Rancang Bangun Load Balancing Apache Webserver Dan Database Cluster (Studi Kasus : Pada Bagian Administrasi Pembangunan Dan PDE Kota Pekanbaru)*. 6(1), 40–49.
- [15] Gani, T. A., Arafat, A., & Melinda, . (2015). Analisis Kinerja MySQL Cluster Menggunakan Metode Load Balancing. *Jurnal Rekayasa Elektrika*, 11(4), 129. <https://doi.org/10.17529/jre.v11i4.2358>
- [16] Rofii, F., Hunaini, F., & Sholawati, S. (2018). Kinerja Jaringan Komunikasi Nirkabel Berbasis Xbee pada Topologi Bus, Star dan Mesh. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 6(3), 393. <https://doi.org/10.26760/elkomika.v6i3.393>