

PERANCANGAN VPS DAN *MULTI-REGION LOAD BALANCER* UNTUK *WEBSITE* PT SCHLEMMER AUTOMOTIVE INDONESIA MENGGUNAKAN LAYANAN *GOOGLE CLOUD PLATFORM*

Grevalby ^{1*}, Anggi Elanda ², Karya Suhada ³

^{1*,2,3} Program Studi Teknik Informatika, STMIK ROSMA, Kabupaten Karawang, Provinsi Jawa Barat, Indonesia.

Email: grevalby.ti2019@mhs.rosma.ac.id ^{1*}

Histori Artikel:

Dikirim 6 Juni 2023; *Diterima dalam bentuk revisi* 5 Juli 2023; *Diterima* 8 Agustus 2023; *Diterbitkan* 10 September 2023. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIK Indonesia Banda Aceh.

Abstrak

PT Schlemmer Automotive Indonesia sebagai perusahaan multinasional di bawah Schlemmer Group membutuhkan website yang dapat diakses dengan lancar oleh pengguna dari berbagai negara, seperti Indonesia, Jerman, dan China. Agar website dapat diakses dengan cepat dan stabil di masing-masing negara tersebut, diperlukan infrastruktur yang handal dan efisien. Penelitian ini bertujuan untuk merancang sebuah sistem VPS dan load balancer untuk website PT Schlemmer Automotive Indonesia yang dapat membagi beban trafik di antara beberapa server atau instance VPS dan mengarahkan trafik untuk mengakses server yang berada di lokasi terdekat dengan pengguna. Metode yang digunakan dalam penelitian ini adalah metode prototyping dengan tahapan yaitu analisa kebutuhan sistem, mendesain sistem, testing sistem, dan implementasi. Hasil pengujian menunjukkan bahwa sistem yang dirancang dapat berjalan dengan baik dan membuktikan penggunaan VPS dan load balancer dapat meningkatkan kinerja dan kestabilan website ketika terjadi lonjakan trafik. Dalam kesimpulannya, penelitian ini menunjukkan bahwa penggunaan VPS dan load balancer pada website PT Schlemmer Automotive Indonesia menggunakan layanan Google Cloud Platform menghasilkan kinerja website yang baik ketika diakses di masing-masing negara.

Kata Kunci: Cloud Computing; Load Balancer; Virtual Private Server; Website.

Abstract

PT Schlemmer Automotive Indonesia as a multinational company under the Schlemmer Group requires a website that can be accessed smoothly by users from various countries, such as Indonesia, Germany and China. In order for the website to be accessed quickly and stably in each of these countries, a reliable and efficient infrastructure is required. This study aims to design a VPS system and load balancer for the PT Schlemmer Automotive Indonesia website that can divide the traffic load between several servers or VPS instances and direct traffic to access servers that are in the closest location to the user. The method used in this study is the prototyping method with the stages of system requirements analysis, system design, system testing, and implementation. The test results show that the designed system can run well and proves that the use of VPS and load balancers can improve website performance and stability when traffic spikes occur. In conclusion, this study shows that the use of VPS and load balancers on the PT Schlemmer Automotive Indonesia website using the Google Cloud Platform service results in good website performance when accessed in each country.

Keyword: Cloud Computing; Load Balancer; Virtual Private Server; Website.

1. Pendahuluan

Situs *web* adalah kunci keberhasilan bisnis, bertindak sebagai saluran komunikasi antara bisnis dan pelanggannya [1]. Kecepatan dan keandalan website menjadi salah satu dimensi kualitas *website* selain dari desain, keamanan, dan layanan pelanggan, semua dimensi kualitas tersebut memiliki pengaruh yang signifikan terhadap kepercayaan pelanggan [2]. PT. Schlemmer Automotive Indonesia dibawah Schlemmer Group sebagai perusahaan multinasional membutuhkan *website* yang dapat diakses dengan cepat dan stabil oleh pelanggan dari berbagai negara, diantaranya Indonesia, Jerman, dan China. Dalam konteks ini, perbedaan geografis antara pengunjung dan *server* serta pertumbuhan jumlah pengunjung *website* yang tinggi dapat menyebabkan performa *website* menjadi lambat dan tidak stabil. Oleh karena itu, diperlukan infrastruktur yang handal untuk memastikan bahwa *website* dapat diakses dengan cepat dan stabil.

Dalam menghadapi kebutuhan yang terus berkembang dan dinamis, PT Schlemmer Automotive Indonesia menyadari pentingnya memiliki infrastruktur *website* yang fleksibel dan dapat diskalakan. Infrastruktur ini akan memungkinkan perusahaan untuk menyesuaikan kapasitas dan sumber daya secara efisien sesuai dengan permintaan dan pertumbuhan pengguna yang meningkat. Dengan adanya infrastruktur *website* yang dapat diskalakan, PT Schlemmer Automotive Indonesia akan dapat dengan mudah menambah atau mengurangi kapasitas *server* dan sumber daya lainnya sesuai dengan kebutuhan yang sedang berlangsung. Hal ini memungkinkan perusahaan untuk mengantisipasi lonjakan lalu lintas *website* di waktu tertentu, serta mengoptimalkan penggunaan sumber daya pada saat permintaan menurun.

Untuk dapat menerbitkan sebuah *website* maka dibutuhkan penyedia layanan *hosting*. *Hosting* adalah tempat fisik dimana semua isi atau *content* sebuah *website* disimpan di dalamnya. *Hosting* dapat diakses dan digunakan oleh pemilik *website* [3]. Dalam upaya membangun sebuah *website* dengan kinerja tinggi, penelitian sebelumnya telah dilakukan oleh Yosli (2021)[3], yang membahas cara meningkatkan kinerja *website* dengan menambah kapasitas *hosting website* berbayar pada layanan *web hosting* konvensional. Selain itu, Kurniansyah & Sinurat (2020)[4] juga melakukan penelitian yang membahas sistem pendukung keputusan dalam memilih *server hosting* terbaik untuk *web server*, dengan mempertimbangkan kriteria lokasi *server*, kecepatan *server*, kredibilitas perusahaan, pelayanan pelanggan, dan keamanan *server*. Sayangnya, penelitian ini tidak mempertimbangkan kriteria skalabilitas dan fleksibilitas *server* yang sangat penting bagi PT Schlemmer Automotive Indonesia. Oleh karena itu, peneliti mencari alternatif lain untuk membangun *website* dengan kinerja tinggi.

Pada saat yang sama, layanan *cloud computing* telah menjadi pilihan yang paling populer untuk merancang dan mengelola infrastruktur *website*, elastisitas merupakan faktor penting yang membuat *cloud computing* berbeda dari bentuk komputasi terdistribusi lainnya, ini memungkinkan sistem alokasi dan distribusi sumber daya bisa dilakukan secara otomatis dan dinamis [5]. *Cloud computing* merupakan solusi yang memungkinkan PT Schlemmer Automotive Indonesia memenuhi kebutuhan mereka akan *website* yang handal dan dapat diakses dengan cepat dan stabil di tiga negara basis pelanggannya. *Cloud computing* memberikan kemudahan dalam mengimplementasikan *load balancing* yang berfungsi untuk mengoptimalkan penggunaan sumber daya secara efisien dan membagi beban kerja komputasi di berbagai wilayah (*region*) [6]. Google Cloud Platform adalah salah satu penyedia layanan *cloud computing* yang menawarkan berbagai solusi untuk membuat dan mengelola infrastruktur *website* [7].

Oleh karena itu, dalam penelitian ini akan dirancang *Virtual Private Server* (VPS) untuk *hosting website* PT Schlemmer Automotive Indonesia dan *load balancer* untuk membagi beban trafik di antara beberapa *server* atau *instance* VPS dan mengarahkan trafik untuk mengakses *server* yang berada di lokasi terdekat dengan pengguna, dalam penelitian ini akan menggunakan layanan Google Cloud Platform sebagai penyedia layanan *cloud computing*. Penelitian ini diharapkan dapat memberikan kontribusi pada pengembangan teknologi informasi dan dapat dijadikan acuan bagi perusahaan lain dalam merancang infrastruktur *website* yang handal dan efisien.

2. Metode Penelitian

2.1 Alat dan Bahan

Penelitian ini akan menggunakan beberapa software dan layanan dari Google Cloud Platform sebagai berikut:

Tabel 1. Alat dan Bahan Penelitian

No.	Resource	Spesifikasi	Keterangan
1.	Compute Engine: Instance Template dan Managed Instance Group di 3 region(Jakarta, Changhua, dan Frankfurt)	e2-medium 2vCPU, 4GB memory Debian11 OS	Auto scaling
2	Firewall rule	allow-http dan allow-health-check	
3	Sistem Operasi untuk Compute Engine	Debian 11	
4	HTTP(S) Load Balancer	Target CPU utilization 80%, 3 backend service	
5	Apache2	-	Web server
6	Apache Benchmark	-	Load testing tool

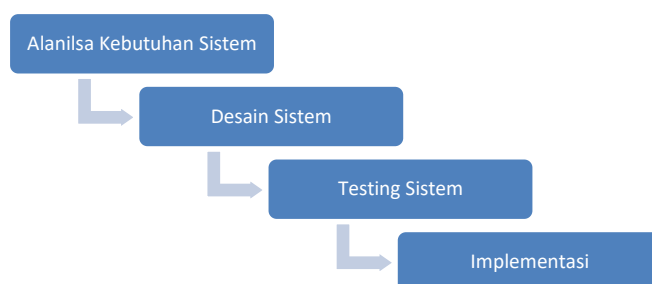
Penelitian ini menggunakan metode *prototyping*. *Prototyping* adalah proses pengembangan yang berupa model kerja fisik dari sistem dan berfungsi sebagai bentuk awal dari sistem yang akan dikembangkan [8].

Terdapat 4 metode utama *prototyping*, yaitu:

- 1) Ilustrasi, menghasilkan contoh berupa tampilan layer.
- 2) Simulasi, mensimulasikan alur kerja sistem tetapi tidak menggunakan data nyata.
- 3) Fungsional, mensimulasikan alur kerja sistem yang sebenarnya dan menggunakan data nyata.
- 4) Evolusioner, menghasilkan model baru yang menjadi bagian dari proses kerja.

Sedangkan dalam penelitian ini akan menggunakan metode *prototyping* fungsional. Langkah-langkah dalam *prototyping* adalah sebagai berikut [9]:

- 1) Analisa kebutuhan sistem, dimana analisis dilakukan untuk memperhitungkan macam-macam komponen yang akan digunakan oleh sistem, antara lain perangkat keras, perangkat lunak, jaringan dan sumber daya manusia.
- 2) Desain sistem, desain sistem terdiri dari aktivitas desain yang menghasilkan spesifikasi fungsional.
- 3) Testing sistem, pengujian sistem bertujuan untuk menemukan kesalahan yang terjadi pada sistem dan melakukan revisi sistem. Tahapan ini dibutuhkan untuk memastikan bahwa sistem bebas dari kesalahan.
- 4) Implementasi, setelah prototipe diterima, maka pada tahap ini adalah implementasi sistem yang siap dioperasikan.



Gambar 1. Metode *Prototyping*

3. Hasil dan Pembahasan

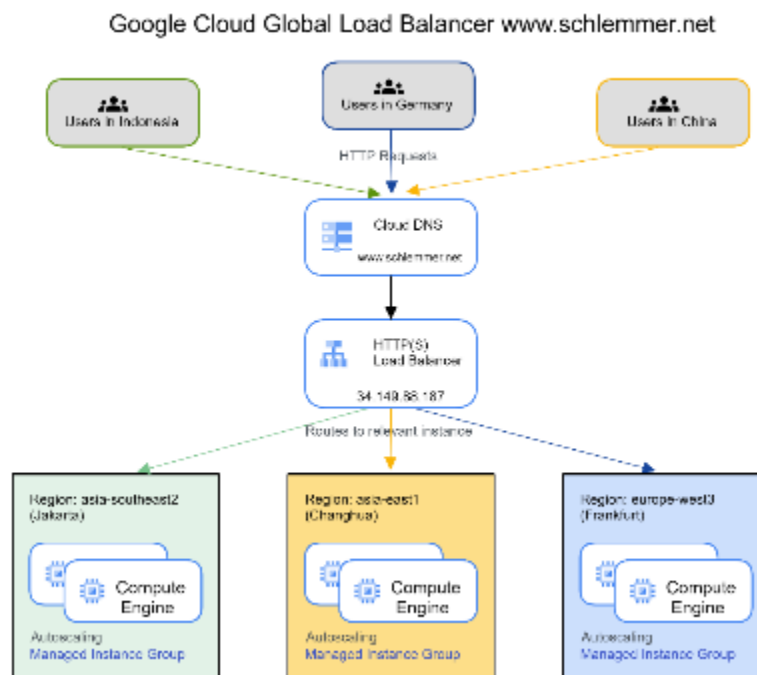
3.1 Analisa Kebutuhan Sistem

Ditahap analisa kebutuhan sistem dilakukan proses evaluasi untuk memahami kemampuan sistem tersebut kemudian menentukan kebutuhan apa yang harus dipenuhi sistem [10]. Untuk membangun infrastruktur website yang dapat membagi beban trafik dan mengarahkan trafik untuk mengakses server yang berada di lokasi terdekat dengan pengguna, maka dibutuhkan *Virtual Private Server* dan *Load Balancer*, dalam penelitian ini menggunakan layanan *cloud computing* dari Google Cloud Platform. Dengan kebutuhan sebagai berikut:

- 1) *Instance Template* dan *Managed Instance Group* yang akan dipakai untuk *autoscaling* ketika terjadi lonjakan trafik.
- 2) *Google Compute Engine* di tiga *region* yaitu *asia-southeast2* (Jakarta), *asia-east1* (Changhua), dan *europa-west3* (Frankfurt) sebagai *Virtual Private Server* untuk *hosting web server*.
- 3) *Firewall Rule* untuk mengizinkan akses HTTP dan *health check*.
- 4) Debian 11 sebagai sistem operasi *compute engine*.
- 5) Apache2 untuk *web server*.
- 6) HTTP(S) *load balancer* untuk membagi beban trafik dan mengarahkan trafik kepada *server* terdekat.
- 7) Apache Benchmark untuk melakukan *load testing*.

3.2 Desain Sistem

Desain sistem menentukan bagaimana sistem akan berfungsi, desain sistem terdiri dari aktivitas desain yang menghasilkan spesifikasi fungsional [11]. Berikut adalah desain sistem yang akan dibangun.



Gambar 2. Arsitektur VPS dengan *Load Balancer*

Salah satu keunggulan utama cloud computing adalah elastisitas, yang memungkinkan untuk memperluas atau mengurangi jumlah sumber daya yang disewa dengan cepat untuk beradaptasi dengan variasi beban, menjamin kualitas layanan yang diinginkan[12]. Gambar diatas menunjukkan bagaimana *load balancing* terjadi pada arsitektur yang akan dibangun. *Load Balancing* adalah metode untuk mengoptimalkan sumber daya *Virtual Machine* di lingkungan *Cloud Computing*. *Load balancing* di

lingkungan *cloud* adalah salah satu teknik penting yang digunakan untuk memastikan adanya distribusi beban kerja yang merata dan dinamis serta penggunaan sumber daya yang efisien. Keseimbangan beban kerja yang efisien menghasilkan kepuasan pengguna yang lebih tinggi dan alokasi sumber daya yang lebih baik [13].

3.3 Pengujian

Pengujian merupakan proses verifikasi untuk penilaian kualitas suatu sistem untuk melihat apakah sistem tersebut memenuhi proses dan arahan yang diharapkan atau tidak [14].

Tabel 2. Skenario Pengujian

No	Skenario Pengujian	Hasil yang diharapkan
1	User dari Indonesia mengakses IP <i>load balancer</i>	Mendapat respon dari <i>server</i> Jakarta
2	User dari China mengakses IP <i>load balancer</i>	Mendapat respon dari <i>server</i> Changhua
3	User dari German mengakses IP <i>load balancer</i>	Mendapat respon dari <i>server</i> Frankfurt
4	Lonjakan trafik, CPU utilization mencapai 80%	<i>Server</i> akan melakukan <i>autoscaling</i>

3.4 Implementasi

3.4.1 Membuat *Firewall Rule*

Buat *Firewall Rule* yang mengizinkan trafik HTTP dan *Health Check*.

Firewall adalah paket perangkat lunak yang dirancang untuk mencegah pengguna mengakses komputer, jaringan, atau *endpoint* [12].



Gambar 3. *Firewall Rule*

3.4.2 Membuat *Instance Template*

Buat *instance template* untuk tiga region (Jakarta, Changhua, dan Frankfurt) sesuai spesifikasi yang dibutuhkan, untuk kasus ini menggunakan tipe mesin e2-medium dan OS Debian, tambahkan juga *startup script* untuk menginstal Apache *web server* dan *clone website* dari repositori Github.

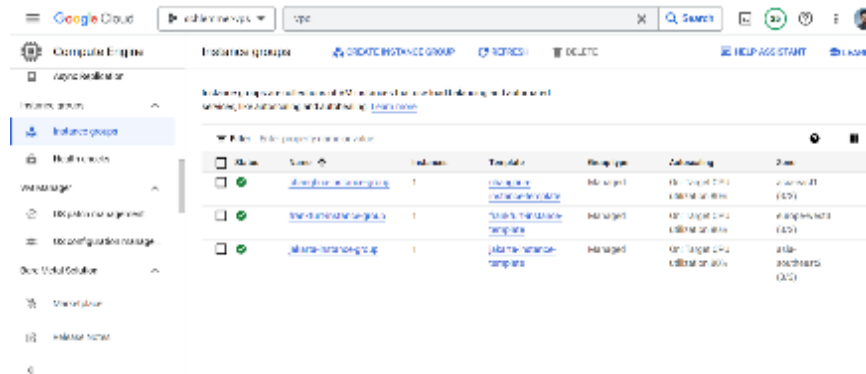


Gambar 4. *Instance Template*

3.4.3 Membuat *Instance Group*

Buat *instance group* untuk tiga *region* (Jakarta, Changhua, dan Frankfurt) dengan target auto scaling

ketika CPU *utilization* mencapai 80%.



Gambar 5. Instance Group

Setelah *instance group* dibuat maka secara otomatis VM *compute engine* juga akan disediakan sesuai *instance group*.



Gambar 6. VM Instance

Masing-masing VM sudah memiliki *external IP* untuk diakses public dan karena pada *instance template* sudah ditambahkan *startup script* untuk menginstal Apache *web server* dan *clone website* dari repository Github maka *website* sudah bisa diakses melalui masing-masing *IP address*, tetapi bukan ini yang diharapkan karena sistem yang dirancang akan menggunakan *IP address load balancer* untuk mengakses *website* di ketiga *region*, trafik akan diarahkan tergantung dari *server* mana yang terdekat dengan pengguna.

3.4.4 Mengkonfigurasi Load Balancer

Buat HTTP(S) *global external load balancer* dan tambahkan tiga *instance group* yang sudah dibuat sebelumnya sebagai *backend service*.

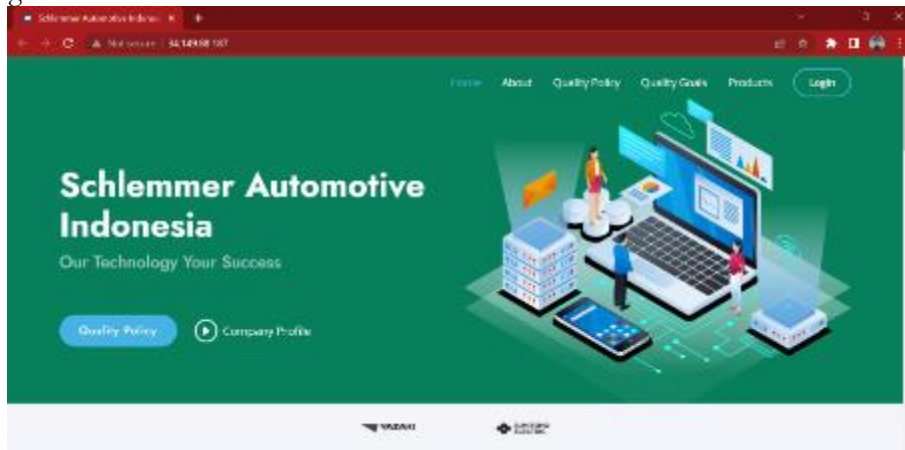


Gambar 7. HTTP(S) Global Load Balancer

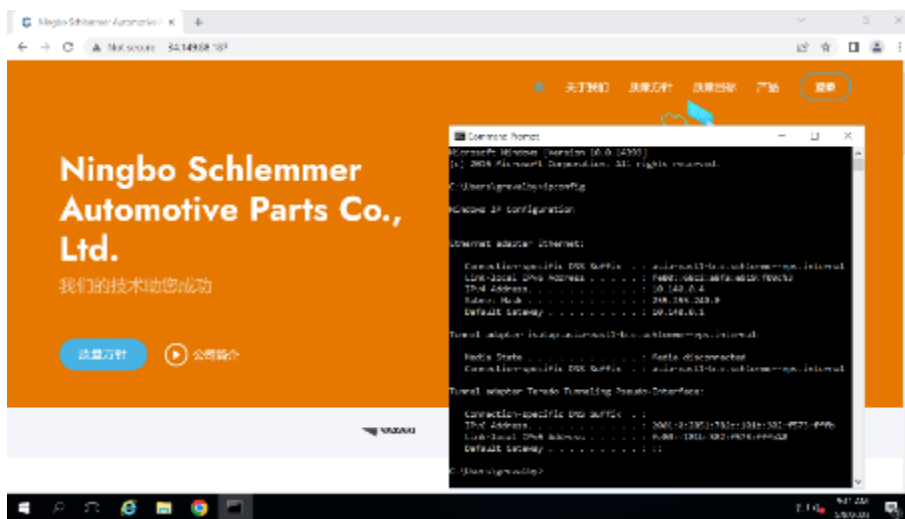
User sudah bisa mengakses *website* melalui satu IP address yaitu IP address *load balancer* dan akan dilayani oleh *server* dengan jarak terdekat dengan *user*. *Load Balancer* juga akan melakukan *autoscaling* ketika *CPU utilization server* mencapai 80% kemudian membagi beban trafik kepada *server* tersebut.

3.4.5 Testing user mengakses *website* dari ketiga *region*

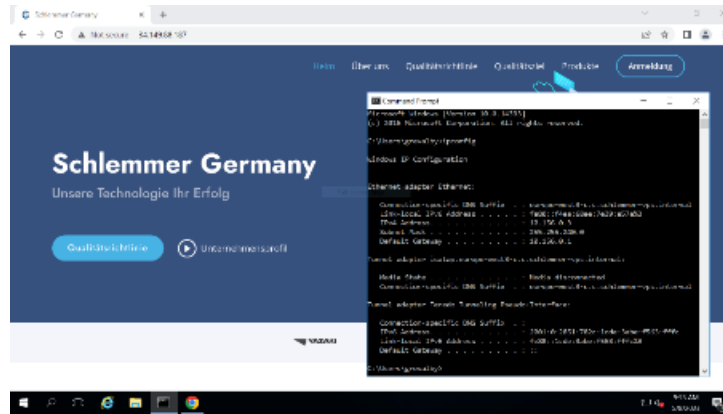
Untuk server China dan German peneliti akan membuat *Compute Engine VM Instance* dari *region* tersebut untuk melakukan *testing*. Berikut adalah hasil *testing* ketika user mengakses *website* dari masing-masing negara:



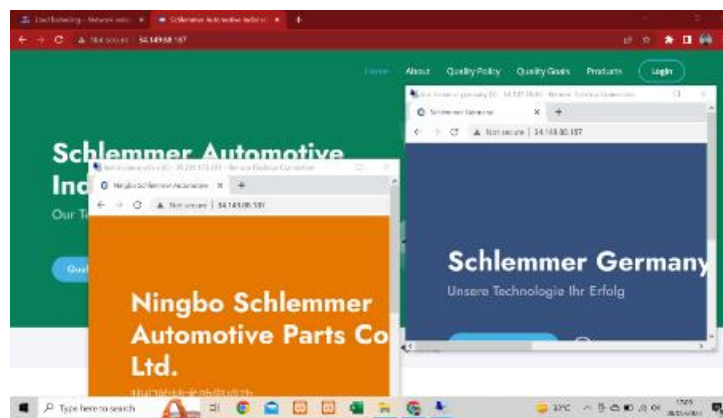
Gambar 8. User dari Indonesia ketika mengakses IP *load balancer*



Gambar 9. User dari China ketika mengakses IP *load balancer*



Gambar 10. User dari German ketika mengakses IP *load balancer*



Gambar 11. *Global load balancer* sudah berfungsi dengan baik

3.4.6 Load Testing

Untuk mengetahui apakah *load balancer* melakukan *auto scaling* ketika penggunaan CPU mencapai 80%, maka dilakukan *load test* pada *website*. *Load test* dilakukan untuk menguji kinerja situs *web* pada load level tertentu dengan meningkatkan jumlah pengguna *virtual* dan *request* hingga mencapai beban yang diinginkan [13]

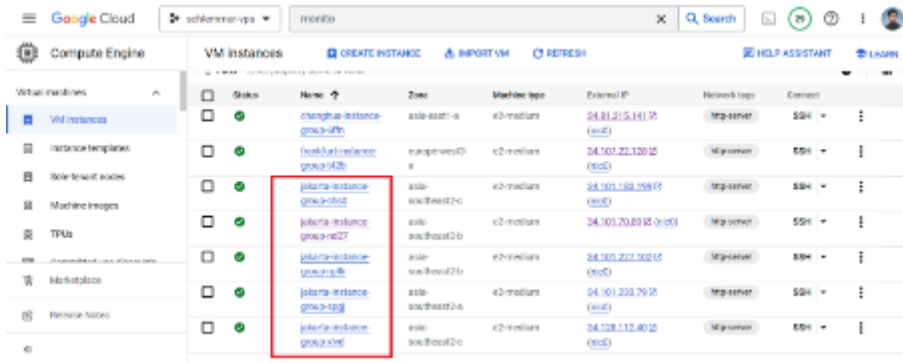
```

Failed requests:          2363
  (Connect: 0, Receive: 0, Length: 2363, Exceptions: 0)
Non-2xx responses:      3
Total transferred:      270460220 bytes
HTML transferred:      267740620 bytes
Requests per second:    263.42 [#/sec] (mean)
Time per request:       949.046 [ms] (mean)
Time per request:       3.796 [ms] (mean, across all concurrent requests)
Transfer rate:          6957.55 [Kbytes/sec] received

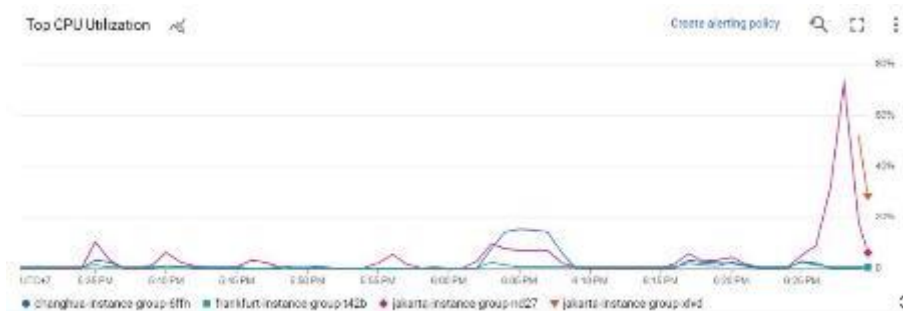
Connection Times (ms)
  min  mean[1/ sd] median  max
Connect:    0    1  1.1    1    11
Processing: 17  922 1128.5 422  7979
Waiting:    16  877 1131.1 370  7976
Total:      16  923 1128.4 422  7979

Percentage of the requests served within a certain time (ms)
 50%    422
 66%    913
 75%   1090
 90%   1017
 95%   2976
 98%   3086
 99%   4028
 99%   4920
100%   7979 (longest request)
-----s@cloudshell:~ (schlemmer-vps) $
  
```

Gambar 12. *Load testing* menggunakan *Apache Benchmark*



Gambar 13. VM Instance autoscaling ketika terjadi lonjakan trafik pada server Jakarta



Gambar 14. Grafik CPU Utilization dari masing-masing VM

Load test dilakukan kepada server Jakarta, Gambar 13 dan 14 memperlihatkan ketika penggunaan CPU pada server jakarta-instance-group-nd27 mencapai 80% maka load balancer melakukan autoscaling dengan menambah instance jakarta-instance-group-xlvd dan melakukan pembagian beban.

4. Kesimpulan

Dalam kesimpulannya, sistem yang dirancang dapat berjalan sesuai dengan yang diharapkan, yaitu website yang dijalankan di Virtual Private Server dapat diakses oleh publik, serta kemampuan sistem load balancer untuk mengarahkan lalu lintas pengguna ke server terdekat, serta melakukan autoscaling dan membagi beban di antara beberapa server ketika terjadi lonjakan trafik. Penelitian ini menunjukkan bahwa penggunaan VPS dan load balancer pada website PT Schlemmer Automotive Indonesia menggunakan layanan Google Cloud Platform dapat menghasilkan kinerja website yang baik dan stabil.

5. Daftar Pustaka

[1] Qalati, S. A., Vela, E. G., Li, W., Dakhan, S. A., Hong Thuy, T. T., & Merani, S. H. (2021). Effects of perceived service quality, website quality, and reputation on purchase intention: The mediating and moderating roles of trust and perceived risk in online shopping. *Cogent Business & Management*, 8(1), 1869363.

[2] Lee, T. S., Ariff, M. S. M., Zakuan, N., Sulaiman, Z., & Saman, M. Z. M. (2016, May). Online sellers' website quality influencing online buyers' purchase intention. In *IOP Conference Series: Materials Science and Engineering* (Vol. 131, No. 1, p. 012014). IOP Publishing.

- [3] Yosli, R. (2021). Meningkatkan Kapasitas Hosting, Mengelola Content Management System Untuk Kenyamanan Memakai Website Berbayar. *Jurnal Vokasi Informatika*, 31-38.
- [4] Kurniansyah, M. I., & Sinurat, S. (2020). Sistem Pendukung Keputusan Pemilihan Server Hosting dan Domain Terbaik Untuk WEB Server Menerapkan Metode VIKOR. *Jurnal Sistem Komputer dan Informatika (JSON)*, 2(1), 14-24. DOI: <http://dx.doi.org/10.30865/json.v2i1.2450>.
- [5] Ghobaei-Arani, M., Souri, A., Baker, T., & Hussien, A. (2019). ControCity: an autonomous approach for controlling elasticity using buffer Management in Cloud Computing Environment. *IEEE Access*, 7, 106912-106924. DOI: 10.1109/ACCESS.2019.2932462.
- [6] Nazir, J., Iqbal, M. W., Alyas, T., Hamid, M., Saleem, M., Malik, S., & Tabassum, N. (2022). Load balancing framework for cross-region tasks in cloud computing. *Computers, Materials & Continua*, 70(1), 1479-1490.
- [7] Wankhede, P., Talati, M., & Chinchamalature, R. (2020). Comparative study of cloud platforms-microsoft azure, google cloud platform and amazon EC2. *J. Res. Eng. Appl. Sci*, 5(02), 60-64.
- [8] Purnomo, D. (2017). Model prototyping pada pengembangan sistem informasi. *JIMP (Jurnal Informatika Merdeka Pasuruan)*, 2(2). DOI: <http://dx.doi.org/10.37438/jimp.v2i2.67>.
- [9] Satyawati, E., & Cahjono, M. P. (2017). Development of accounting information system with Rapid Application Development (RAD) method for micro, small, and medium scale enterprises. *Review of Integrative Business and Economics Research*, 6, 166.
- [10] Catillo, M., Rak, M., & Villano, U. (2020). Auto-scaling in the cloud: Current status and perspectives. In *Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 14th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2019) 14* (pp. 616-625). Springer International Publishing.
- [11] Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2022). Load balancing techniques in cloud computing environment: A review. *Journal of King Saud University-Computer and Information Sciences*, 34(7), 3910-3933.
- [12] Yulistina, S. R., Nurmala, T., Supriawan, R. M. A. T., Juni, S. H. I., & Saifudin, A. (2020). Penerapan Teknik Boundary Value Analysis untuk Pengujian Aplikasi Penjualan Menggunakan Metode Black Box Testing. *Jurnal Informatika Universitas Pamulang*, 5(2), 129-135.
- [13] Menascé, D. A. (2002). Load testing of web sites. *IEEE internet computing*, 6(4), 70-74. DOI: 10.1109/MIC.2002.1020328.