

Pemanfaatan Teknologi *Pound* Untuk Pemerataan Beban Server Web

Wagito

Program Studi Teknik Informatika
STMIK Akakom, Jl. Raya Janti Karang Jambe Yogyakarta
Email : wagito@akakom.ac.id

Abstrack - Web access is a service that is widely available on the Internet network. Server load can be very high when users access it very much and simultaneously. One way that servers can serve high access is the use of multiple servers. Access loads are distributed on multiple servers. The research aims to implement load balancing, develop methods to observe server load and analyze the distribution of load access to the server for a certain period of time. Server load is distributed to several servers using load balancing techniques. The balancing process is carried out by Pound software. Web access services are carried out by Apache software. The research method is done by observing the amount of access received by the server. Observation of web access is carried out for six months. The results of recording web access are processed to determine the different load of access received by the web server. The results of research processed based on daily, weekly and monthly access shows that Pound software can function well for even distribution of web server load. Web server access load was found to be somewhat less evenly distributed with a difference in the number of daily access by 5.5%. Weekly access and monthly access result in a difference of 3.18%. The research found that there was a slight difference in the results of daily, weekly and monthly access due to a disturbance on one server on a certain date.

Keywords: Load, balancing, web, access

Intisari - Akses *web* merupakan layanan yang banyak disediakan pada jaringan Internet. Beban *server* bisa sangat tinggi pada saat diakses pengguna yang sangat banyak dan secara serentak. Salah satu cara yang dilakukan supaya *server* dapat melayani akses tinggi adalah penggunaan beberapa *server*. Beban akses didistribusikan pada beberapa *server*. Penelitian bertujuan untuk menerapkan pemerataan beban, menyusun metode untuk mengamati beban *server* dan menganalisis pembagian beban akses terhadap *server* selama kurun waktu tertentu. Beban *server* didistribusikan pada beberapa *server* menggunakan teknik pemerataan beban. Proses pemerataan beban dilakukan oleh perangkat lunak Pound. Layanan akses *web* dilakukan oleh perangkat lunak Apache. Metode penelitian dilakukan dengan pengamatan terhadap jumlah akses yang diterima oleh *server*. Pengamatan akses *web* dilakukan selama enam bulan. Hasil pencatatan akses *web* diolah untuk menentukan beda beban akses yang diterima *server web*. Hasil penelitian yang diolah berdasarkan akses harian, mingguan dan bulanan menunjukkan bahwa perangkat lunak Pound dapat berfungsi secara baik untuk pemerataan beban *server web*. Beban akses *server web* ditemukan agak kurang terbagi merata dengan perbedaan jumlah akses harian sebesar 5,5%. Akses mingguan dan akses bulanan menghasilkan angka perbedaan sebesar 3,18%. Pada penelitian ditemukan ada sedikit perbedaan hasil akses harian, mingguan dan bulanan yang disebabkan karena adanya gangguan pada salah satu *server* pada tanggal tertentu.

Kata Kunci : Pemerataan, beban, akses, web

I. PENDAHULUAN

Jaringan Internet sudah berkembang sedemikian rupa, sehingga sudah masuk dalam sebagian besar ruang kehidupan manusia. Jaringan Internet melayani

berbagai macam layanan mulai dari web, email, chat, teleconference, multimedia dan sebagainya. Kegiatan lain yang dahulunya belum melibatkan Internet, sekarang sudah melibatkan Internet seperti melihat acara televisi, *mendengarkan* musik, melakukan

panggilan telepon dan sebagainya. Perkembangan menuntut perusahaan komunikasi seperti televisi, radio maupun telepon menyediakan fitur yang melibatkan Internet untuk menarik pelanggan.

Layanan yang disediakan oleh jaringan Internet dilakukan oleh server. Tiap server melayani jenis layanan yang berbeda. Salah satu layanan pada Internet adalah web. Web merupakan layanan yang paling banyak dimanfaatkan pengguna jaringan Internet baik sebagai sarana untuk menyebarkan informasi maupun mencari informasi. Layanan web digunakan protokol HTTP atau HTTPS

Layanan HTTP dan HTTPS dilakukan oleh server web. Dari sekian banyak perangkat lunak server web, Apache merupakan server web yang paling populer dan paling banyak digunakan. Apache sebagai perangkat lunak layanan web harus mampu menangani beban akses web.

Apabila akses web pada suatu situs cukup tinggi, maka Apache menjadi perangkat lunak yang harus bekerja ekstra. Supaya dapat melayani beban akses yang cukup tinggi, perlu dilakukan distribusi terhadap beban tersebut. Beban tinggi akses web dapat terjadi pada saat KRS Online. Beban tinggi akses web sistem informasi akademik diteliti dan dipublikasikan pada jurnal IJCCS [1].

Salah satu cara yang dipakai untuk mengurangi beban sever *adalah* melibatkan beberapa server web. Akses web distribusi pada beberapa server secara merata. Teknik yang dipakai adalah pemerataan beban (load balancing). Penerapan load balancing pada server web jaringan membutuhkan perangkat keras atau perangkat lunak yang membagi lalu-lintas antar server. Jaringan yang menerima sejumlah lalu-lintas akses data yang tinggi bahkan mungkin memiliki satu atau lebih server yang didedikasikan untuk menyeimbangkan beban akses antara server dan perangkat jaringan yang lain dalam sistem jaringan komputer. Server tersebut sering disebut sebagai pemerata beban (load balancer). Perancangan dan implementasi load balancing reverse proxy

menggunakan perangkat lunak HAProxy pernah diteliti dan dihasilkan hasil pemerataan beban ketika semua server dalam keadaan hidup dan hasil pemerataan beban apabila salah satu server mengalami kegagalan [2]. Penelitian lain berkaitan dengan implementasi load balance pada jaringan multihoming menggunakan router dengan metode round robin. Penelitian membahas penggunaan router untuk proses load balance pada jaringan yang punya beberapa jalur sumber Internet [3].

Penelitian tentang perbandingan kinerja beberapa load balancer pernah dilakukan perangkat lunak Apache, Nginx, Inlab, Pound, HAProxy dan Lighttpd [4]. Pilihan teknologi Pound yang dipakai dalam penelitian dengan pertimbangan bahwa perangkat lunak Pound khusus didedikasikan untuk *pemerata beban*, sederhana namun fitur lengkap dan free software. Hasil uji kinerja Pound juga cukup baik, tidak yang tertinggi namun juga tidak terendah. Karena kesederhanaan Pound, bisa dikembangkan pada mesin khusus *yang melayani pemerataan beban*.

Tujuan yang ingin dicapai dari hasil penelitian yang dilakukan adalah menerapkan pemerata beban Pound sebagai perangkat lunak untuk membagi beban akses server web, menyusun metode untuk mengamati beban akses server web berdasarkan catatan akses yang dihasilkan server web, menganalisis pembagian beban akses terhadap server web selama kurun waktu tertentu serta menganalisis beban akses server web secara periode waktu harian, mingguan dan bulanan.

II. SIGNIFIKASI STUDI

A. Studi Literatur

1. Pemerataan beban (Load balancing)

Pemerataan beban adalah pendistribusian lalu-lintas jaringan yang masuk secara efisien kepada sekelompok *server back-end*, juga dikenal sebagai *server farm* atau *server pool*. Situs modern yang punya lalu-lintas data tinggi harus mampu melayani ratusan ribu, jika tidak

dikatakan jutaan, permintaan bersamaan dari pengguna atau klien dan mengembalikan teks, gambar, video, atau data aplikasi yang benar, semuanya dengan cara yang cepat dan handal. Berdasar skala biaya-efektif untuk memenuhi volume tinggi, komputasi modern, umumnya praktik terbaik adalah menggunakan penambahan beberapa *server* [6].

Sebuah pemerata beban bertindak sebagai "polisi lalu-lintas" yang diletakkan di depan *server* dan permintaan *routing* klien menuju semua *server* serta mampu memenuhi permintaan tersebut dengan cara yang memaksimalkan kecepatan dan pemanfaatan kapasitas serta memastikan bahwa tidak ada satu *server* yang terlalu banyak bekerja, sehingga dapat menurunkan kinerja. Jika salah satu *server* mati, pemerata beban mengalihkan lalu-lintas ke *server* lain yang *online*. Ketika *server* baru ditambahkan pada kelompok *server*, pemerata beban secara otomatis mulai mengirim permintaan kepada *server* baru tersebut [6].

Dengan cara demikian, pemerata beban melakukan fungsi-fungsi sebagai berikut: mendistribusikan permintaan klien atau beban jaringan secara efisien di beberapa *server*, memastikan ketersediaan dan keandalan yang tinggi dengan mengirimkan permintaan hanya untuk *server* yang sedang *online* serta memberikan fleksibilitas untuk menambah atau mengurangi *server* sesuai perintah permintaan.

Ada beberapa algoritme *load balancing*. Algoritme *load balancing* yang berbeda memberikan manfaat yang berbeda; pilihan metode *load balancing* tergantung pada kebutuhan: [6] *Round Robin* - Permintaan didistribusikan pada seluruh kelompok *server* secara berurutan, *Least Connections* – Sebuah permintaan baru dikirim pada *server* dengan koneksi sesaat paling sedikit kepada klien. Kapasitas komputasi relatif masing-masing *server* adalah faktor dalam menentukan mana yang memiliki koneksi paling sedikit serta *IP Hash* - Alamat IP dari klien

digunakan untuk menentukan *server* mana yang menerima permintaan tersebut.

Ada beberapa manfaat penggunaan *server* pemerataan beban [7]. Skalabilitas (*scalability*): *Server* pemerataan beban memungkinkan penambahan secara mudah *server* dan peningkatan skala dari aplikasi. Kemudahan menambahkan *server* dan peningkatan skala aplikasi dengan *load balancing* sejumlah *server* dari satu URL dan alamat IP. Kinerja (*performance*): *Server* pemerataan beban mengurangi waktu respon dan waktu transaksi untuk *web*, aplikasi dan layanan *cloud*. Pengurangan waktu respon dan waktu transaksi dilakukan dengan *caching*, kompresi dan akselerasi SSL sampai lima kali kinerja aplikasi yang lebih baik. Ketersediaan (*availability*): Pemerataan beban memastikan aplikasi tetap aktif dan berjalan meskipun satu atau lebih *server* menjadi tidak aktif. Pemerataan beban untuk mendistribusikan lalu-lintas, melakukan pemeriksaan kesehatan dan menjaga sesi persisten pada beberapa *server* untuk memastikan aplikasi tetap aktif dan berjalan dalam hal satu atau lebih *server* menjadi tidak aktif. Efisiensi (*efficiency*): *Offloading* SSL dan pengolahan TCP dapat menyediakan hingga sepuluh kali efisiensi *server* yang lebih besar. Efisiensi *server* yang diperoleh sampai sepuluh kali yang lebih besar untuk ROI yang lebih besar dan kinerja aplikasi yang lebih baik dengan *offloading* SSL dan pengolahan TCP. Fleksibilitas (*flexibility*): Tersedia sebagai peralatan khusus atau pemerata beban *server virtual* untuk VMware, Citrix, Hyper-V, OpenXen dan KVM. Sekumpulan larik pemerata beban yang ada dalam sistem dalam perusahaan pusat data dan *cloud* publik atau privat. Fitur paritas pada seluruh *platform* dan kinerja tak tertandingi, fleksibilitas dan integrasi manajemen membuat larik pemerata beban aplikasi menjadi solusi ideal untuk penawaran layanan *cloud* publik dan privat.

2. Pound

Perangkat lunak Pound adalah *proxy reverse*, pemerata beban dan HTTPS *front-end* untuk satu atau beberapa *server web*. Pound dikembangkan untuk memungkinkan penyebaran beban antara beberapa *server web* dan untuk memungkinkan pembungkus SSL yang bagus bagi beberapa *server web* tersebut yang tidak dicakup secara native. Pound didistribusikan di bawah GPL - tidak ada garansi, gratis untuk digunakan, menyalin dan memberikan secara sembarang [5].

Ada beberapa fitur yang dapat dilakukan perangkat lunak Pound[5]. *Reverse-proxy*: melewati permintaan dari *browser* klien untuk satu atau lebih *server back-end*. Pemerata beban: mendistribusikan permintaan dari *browser* klien kepada beberapa *server* yang berlaku sebagai *back-end*, sekaligus menjaga sesi informasi. pembungkus SSL: dekripsi permintaan protokol HTTPS dari *browser* klien dan melewatkannya sebagai permintaan protokol HTTP biasa pada *server back-end*. HTTP/HTTPS *sanitizer*: verifikasi permintaan untuk tingkat kebenaran dan hanya menerima dalam bentuk terbaik. *Server failover*: ketika *server back-end* gagal, Pound akan mencatat bukti dan menghentikan pelewatan lalu-lintas data lewat permintaan sampai *server* tersebut pulih. *Request redirector*: permintaan dapat didistribusikan di antara *server* sesuai dengan URL yang diminta.

3. Konfigurasi Pound

Pengaturan kerja perangkat lunak Pound dilakukan dengan cara mengatur *file* konfigurasi `/etc/pound.cfg` yang berupa *file* teks. Isi *file* konfigurasi Pound yang berkaitan dengan penelitian yang dilakukan adalah sebagai berikut [8] **User**: *user* sistem yang menjalankan Pound. **Group**: *group* untuk *user* sistem. **ListenHTTP**: pengaturan untuk menentukan pengaturan protokol HTTP. Pada pengaturan juga dikonfigurasi beberapa bagian. **Address**: IP *server balancer*. **Port**: *port* kerja protokol

HTTP, biasanya 80. **ListenHTTPS**: pengaturan untuk menentukan protokol HTTPS. **Address**: IP *server Balancer*. **Port**: *port* kerja protokol HTTP, biasanya 443. **Service**: layanan *server back-end* yang disediakan. Opsi **Backend** digunakan untuk mengatur hal berkaitan dengan *server web*. Opsi ini mengatur dua hal yaitu **Address** (IP *server web back-end*) dan **Port** (*port* kerja *server web back-end*).

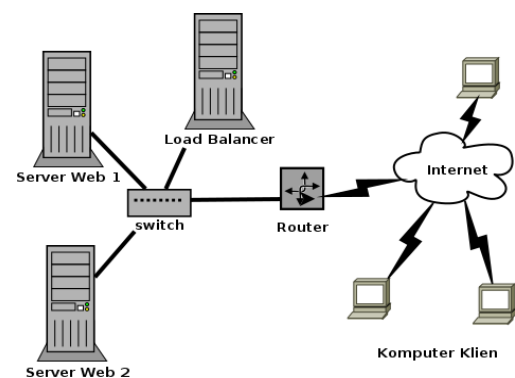
B. Pemodelan

1. Peralatan

Perangkat lunak yang digunakan dalam penelitian adalah Sistem Operasi Linux Mageia 2015, server web Apache, pemerata beban Pound, Synchronizer Rsync, database SQLite, SQLite3 tools, server SSH, SSH client, Mikrotik Router OS, Sshpass dan Midnight commander. Perangkat keras yang digunakan dalam penelitian meliputi *server* prosesor Intel(R) Xeon® X5650 @2.67GHz, tipe sistem 64 bit, RAM 32 GB, harddisk 20 GB dan dukungan Gigabit Ethernet.

2. Topologi Jaringan

Pada penelitian terlibat tiga mesin utama yaitu satu mesin pemerata beban dan dua mesin *server web*. Mesin pemerata beban digunakan sebagai *server* penerima akses *web* (*front-end*) dan mendistribusikan beban kepada dua *server web*. Dua *server web* berfungsi sebagai *server* akhir penerima akses *web* dan menanggapi permintaan akses *web*. *Server web* berlaku sebagai *back-end*.



Gambar 1. Topologi Jaringan

Untuk mencoba akses dapat digunakan sembarang komputer yang dilengkapi dengan *browser* serta fasilitas akses jaringan untuk akses *web*. Akses jaringan dapat dilakukan menggunakan sembarang media transmisi baik jaringan kabel maupun jaringan *wireless*.

Akses langsung dari jaringan Internet diterima oleh Router. Router memilah jenis akses dan meneruskan kepada server yang bertugas untuk melayani. Salah satu akses web yang diuji pada penelitian ditujukan pada server web pertama dan server web kedua.

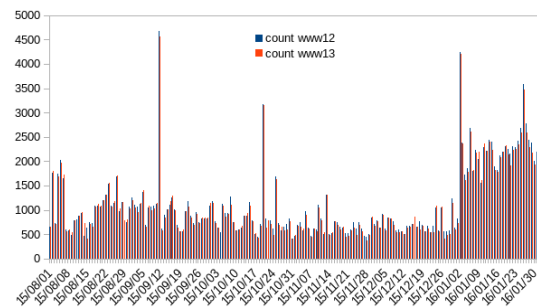
Akses web diteruskan kepada server pemerata beban yang selanjutnya akan diatur untuk diteruskan pada server web. server pemerata beban berfungsi untuk mengatur beban akses web yang akan diterima oleh server web. Server web pertama dan kedua menerima akses HTTP yang merupakan hasil dari penerusan akses yang semula diterima oleh server pemerata beban. server web berfungsi untuk menerima beban akses yang sesungguhnya dan menanggapi permintaan akses. Pada penelitian digunakan dua server web identik baik perangkat keras maupun perangkat lunak yang dipasang.

Komputer berfungsi untuk menguji akses menuju server web. Komputer klien dapat berupa komputer *desktop*, *notebook*, *smartphone*, atau peralatan lain yang dapat digunakan untuk akses layanan web. Letak komputer klien dapat berada dalam jaringan lokal maupun jaringan publik. Koneksi komputer klien dapat menggunakan media transmisi yang umum dipakai seperti kabel, wifi, GPRS dan sebagainya.

III. HASIL DAN PEMBAHASAN

Uji coba load balancing dilakukan selama enam bulan dari bulan Agustus sampai bulan Januari. Pengamatan dilakukan pada berapa jumlah akses pada masing-masing server. Jumlah akses pada masing-masing server dicatat menggunakan sekrip PHP yang disisipkan pada halaman aplikasi situs. Data akses disimpan dalam database SQLite3.

Sekrip akan mencatat jumlah akses yang diterima server web setiap hari dari jam 00:00 sampai jam 24:00. Catatan harian akses web selama enam bulan dalam bentuk angka lengkap, ditampilkan dalam halaman Lampiran. Hasil olahan catatan data harian dalam bentuk grafik untuk beda jumlah akses antara dua server web yang diamati ditampilkan pada Gambar 2.



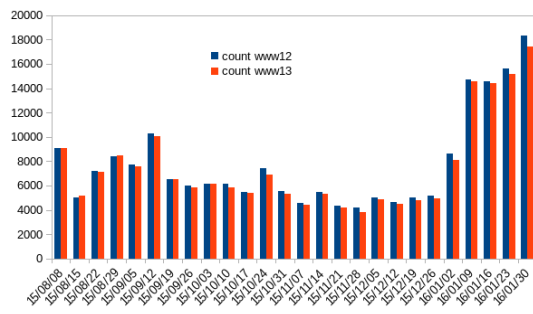
Gambar 2. Perbedaan Akses Dua Server Web Secara Harian

Hasil tampilan grafiknya kurang bisa membedakan beda cacah beban akses antara dua server web. Hal demikian disebabkan karena ukuran grafik dan jumlah hari pencatatan yang cukup besar.

Rata-rata perbedaan jumlah beban akses yang diterima oleh dua server web sebesar 5,5% dihitung dari besarnya perbedaan dibanding rata-rata beban akses pada dua server web. Nilai *terendah* perbedaan jumlah beban akses pada dua server web sebesar 0% yang artinya tidak ada perbedaan jumlah beban akses pada dua server pada hari tersebut. Nilai *terbesar* perbedaan akses dua server web sebesar 44%. Secara umum, nilai perbedaan beban akses cukup kecil. Dengan demikian apabila beberapa hari tertentu nilai perbedaan cukup besar, hal demikian disebabkan karena salah satu server web mengalami gangguan. Gangguan dapat terjadi baik pada server web pertama maupun pada server web kedua. Penyebab gangguan belum bisa diungkapkan dalam penelitian.

Selanjutnya dicoba untuk menganalisis perbedaan beban akses antara server pertama dan server kedua secara kumulatif mingguan. Pengolahan secara kumulatif

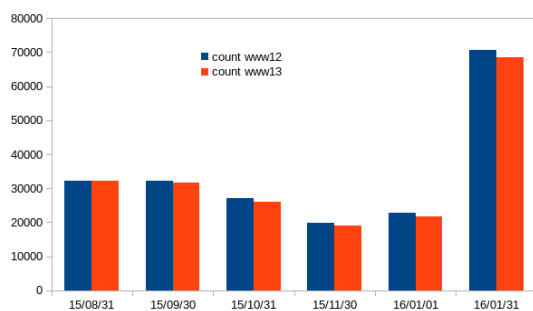
dimaksudkan supaya hasil grafik yang ditampilkan dapat dilihat beda beban akses secara lebih jelas. Data dijadikan kumulatif dari Senin sampai Ahad kemudian dilihat perbedaan jumlah beban akses yang diterima. Perbedaan beban akses harian tidak terlihat secara jelas, namun pada kumulatif mingguan beda jumlah beban akses dapat dilihat lebih jelas.



Gambar 3. Perbedaan Akses Dua Server Web Secara Mingguan

Perbedaan jumlah beban akses pengguna pada dua server yang dijadikan kumulatif secara mingguan serta ditunjukkan pada Gambar 3

Secara umum, dapat ditemukan bahwa Pound menghasilkan sedikit tidak rata dalam pembagian beban akses pada server web pertama dan kedua, meskipun tidak signifikan. Rata-rata perbedaan jumlah beban akses yang diterima oleh dua server web secara mingguan sebesar 3,18%. Nilai *terendah* perbedaan jumlah beban akses pada dua server web sebesar 0,19% dan nilai terbesar perbedaan beban akses dua server web sebesar 10,58%. Dapat dilihat, bahwa nilai terbesar beda beban akses terjadi pada saat terjadinya gangguan pada salah satu server.



Gambar 4. Perbedaan Akses Dua Server Web Secara Bulanan

Perbedaan jumlah beban akses dicoba dijadikan kumulatif secara bulanan supaya grafik yang ditampilkan menjadi lebih jelas. Grafik nilai kumulatif jumlah beban akses bulanan ditampilkan pada Gambar 4.

Rata-rata perbedaan jumlah beban akses yang diterima oleh dua server web sebesar 3,18%. Nilai *terendah* perbedaan jumlah beban akses pada dua server web sebesar 0,54% dan nilai terbesar perbedaan beban akses dua server web sebesar 5,33%. Dengan demikian nilai perbedaan jumlah akses antara dua server cukup kecil. Hasil menunjukkan bahwa proses pemerataan beban antara dua server dapat berjalan secara baik.

Hasil yang diperoleh pada penelitian menunjukkan bahwa proses load balancing antara dua server web yang diamati dapat berjalan secara baik. Dengan demikian perangkat lunak Pound dapat menjalankan fungsi proses load balancing dua server web dengan tingkat perbedaan yang cukup kecil.

Penelitian bisa dikembangkan membandingkan kestabilan beberapa server web yang digunakan untuk melayani akses informasi. Antara dua server web yang diamati, server mana yang sering mengalami gangguan selama kurun waktu tertentu. Selanjutnya dapat dianalisis penyebab terjadinya gangguan sedemikian, sehingga dapat dilakukan metode pencegahan terhadap gangguan yang terjadi pada server. Penyebab terjadinya gangguan dapat dianalisis dari hasil catatan log layanan yang tersedia pada server

Pada penelitian, dua server yang dipakai adalah identik, baik perangkat keras maupun perangkat lunak. Bisa dibandingkan apabila dibuat perbedaan pada perangkat keras atau pada perangkat lunak. Perbedaan perangkat keras dapat dilakukan dengan membedakan prosesor, RAM atau perangkat keras lain. Perbedaan perangkat lunak dapat dilakukan dengan membedakan perangkat lunak server web yang dipakai. Beberapa perangkat lunak server web bisa dipakai, misalnya server

pertama pakai Apache dan yang kedua pakai Nginx. Apakah terjadi perbedaan jumlah beban akses apabila ada perbedaan perangkat lunak yang digunakan untuk menjalankan aplikasi web.

Penelitian dapat dikembangkan untuk jumlah server web yang dipakai. Pada penelitian digunakan dua server web yang bisa ditambah jumlahnya. Jika jumlah server web ditambah, maka perubahan pengaturan dilakukan pada server balancing untuk mengatur konfigurasi Pound.

IV. KESIMPULAN

Teknologi perangkat lunak Pound dapat diterapkan untuk pemerataan beban akses *server web* Apache. Beban akses *server web* ditemukan agak kurang merata terbagi dengan perbedaan beban akses harian sebesar 5,5%. Beban akses mingguan dan bulanan menghasilkan angka perbedaan sebesar 3,18%. Pada penelitian ditemukan ada sedikit perbedaan hasil persentase jumlah beban akses harian, mingguan dan bulanan yang disebabkan adanya gangguan salah satu *server* pada tanggal tertentu.

REFERENSI

- [1] Alimuddin dan kawan-kawan, 2016, Peningkatan Kinerja Siakad Menggunakan Metode Load Balancing dan Fault Tolerance Di Jaringan Kampus Universitas Halu Oleo, jurnal IJCCS, UGM
- [2] Ari Budi Noviyanto dan kawan-kawan, 2015, Perancangan dan implementasi load balancing reverse proxy menggunakan HAProxy pada aplikasi web, jurnal JARKOM, AKPRIND, Yogyakarta
- [3] I Made Widhi Wirawan dan kawan-kawan, 2011, Implementasi load balance pada jaringan multihoming menggunakan router dengan metode round robin, Jurnal Ilmu Komputer, Universitas Udayana
- [4] Piórkowski, A., dkk, 2010, Load Balancing for Heterogeneous Web Servers, Computer networks. 17th conference, CN 2010, Ustroń, Poland, June 15–19, 2010. Proceedings
- [5] Apsis, IT Security, 2016, Pound - Reverse-Proxy and Load-Balancer, <http://www.apsis.ch/pound/>
- [6] Nginx, 2017, What Is Load Balancing?, <https://www.nginx.com/resources/glossary/load-balancing/>
- [7] Array Networks, 2017, Server Load Balancing for Application High-Availability, <https://www.arraynetworks.com/solutions-server-load-balancing.html>
- [8] Segall, R., 2010, Pound - HTTP/HTTPS Reverse-Proxy and Load-Balancer, <https://linux.die.net/man/8/pound>