Analisis Mekaniseme *Multi Server Load Balancing* pada Server SIAKAD Universitas Brawijaya

R. Arief Setyawan, Adharul Muttaqin, Angger Abdul Razak, Lastono Risman

***Abstrak*—Salah satu permasalahan utama dalam pelaksanaan sistem administrasi akademi secara *online* adalah proses KRS. Mahasiswa memilih secara mandiri matakuliah yang akan diambil pada semester tersebut. Meskipun jangka waktu pelaksanaan KRS online adalah 2 minggu, tapi pada kenyataannya hampir 80% mahasiswa akan melakukan KRS *online* pada hari pertama. Hal ini menyebabkan beban akses yang diterima oleh *server* Siakad menjadi sangat tinggi. Dengan jumlah total mahasiswa sebanyak 40000, maka sistem harus siap untuk menangani sekitar 30 ribu akses pada hari pertama KRS online. Sehingga proses ini tidak mungkin hanya di tangani oleh 1 *server*.**

Dalam penelitian ini dibuat suatu mekanisme membagi beban akses KRS *online* menjadi beberapa server dengan standar dan data yang valid. Mekanisme dilakukan dengan menggunakan *server load balancing* sebagai pengatur beban serta mencegah terjadinya penumpukan akses di salah satu server saja. Dengan demikian diharapkan proses KRS online di Universitas Brawijaya menjadi lancar dan tidak ada gangguan.

Dari hasil penelitian menunjukkan bahwa penggunaan *load balancing* telah meningkatkan jumlah mahasiswa yang dapat di layani oleh SIAM. Pada puncak akses dalam 1 jam *server* dapat menangani sebanyak 23.986 *request*. Puncak akses terjadi pada tanggal 8 Agustus 2012 sebanyak 43.245 *request* dalam 1 hari. Dengan demikian lonjakan akses dapat tertangani melalui mekanisme ini.

Kata Kunci*— *load balancing* dan *RoundRobin

I. PENDAHULUAN

PERKEMBANGAN teknologi komputer dewasa ini telah memasuki seluruh aspek kehidupan. Demikian pula teknologi informasi yang semakin meningkatkan efisiensi berbagai proses administrasi yang ada. Pada khususnya proses administrasi akademik mahasiswa di Universitas Brawijaya. Seluruh data telah terpusat sehingga meningkatkan efisiensi proses administrasi. Proses administrasi ini ditunjukkan dalam Gambar 1.

Proses administrasi mahasiswa diawali melalui SELMA, sebagai sistem untuk menerima calon mahasiswa. Seluruh jalur masuk UB baik mandiri maupun nasional di catat di sistem ini. Proses seleksi akan menggunakan data dari SELMA untuk melakukan proses seleksi. Setelah data seluruh calon mahasiswa di seleksi, maka calon mahasiswa yang diterima akan secara otomatis tercatat di SIREGI sebagai sistem registrasi mahasiswa. Baik registrasi awal, maupun bagi mahasiswa lama yang melakukan daftar ulang. Mahasiswa yang telah melakukan pembayaran awal akan tercatat di SIAKAD untuk melaksanakan proses akademik selama seorang mahasiswa terdaftar di UB. Mahasiswa yang telah dinyatakan lulus akan diproses melalui SIUDA untuk melaksanakan proses Wisuda. Dan bagi alumni UB akan tercatat di sistem JPC dan Treasure Study. Seluruh sistem ini telah terintegrasi dan membantu mahasiswa, dosen, dan staf administrasi dalam melaksanakan tugasnya.

Gambar 1 Sistem Informasi Mahasiswa

Salah satu permasalahan utama dalam pelaksanaan sistem *online* adalah proses dalam Sistem Informasi Akademik (SIAKAD). Semakin meningkatnya jumlah mahasiswa menyebabkan beban akses dari *server* meningkat sangat drastis. Hal ini terjadi terutama saat pelaksanaan KRS online. Pada proses KRS online setiap mahasiswa memilih secara mandiri matakuliah yang akan diambil pada semester tersebut. Proses ini terkadang menimbulkan persaingan antar mahasiswa. Mahasiswa berusaha untuk berebut kelas tertentu yang mereka inginkan, sedangkan jumlah kursi dalam satu kelas terbatas. Meskipun jangka waktu pelaksanaan KRS online adalah 2 minggu, tapi pada kenyataannya hampir 80% mahasiswa akan melakukan KRS online pada hari pertama. Hal ini menyebabkan beban akses yang diterima oleh server SIAKAD menjadi sangat tinggi. Dengan jumlah total mahasiswa sebanyak 40000, maka sistem harus siap untuk menangani sekitar 30 ribu akses pada hari pertama KRS *online*. Sehingga proses ini tidak mungkin hanya di tangani oleh 1 server. Untuk itu diperlukan suatu mekanisme pengaturan beban server untuk mencegah terjadinya beban lebih (*overload*) di server yang menyebabkan layanan server

R. Arief Setyawan adalah dosen jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya (email : rarief@ub.ac.id)

Adharul Muttaqin, Angger Abdul Razak adalah dosen jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya.

Lastono Risman adalah Alumni jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya

tidak aktif (down).

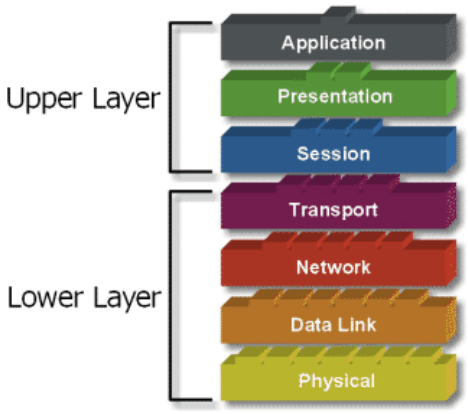
Dalam penelitian ini ditunjukan untuk menganalisa dan menerapkan sistem load balancing pada akses server siakad UB. Sehingga diperoleh metode terbaik dalam menangani akses yang berlebihan dalam satu saat dengan menggunakan banyak server dengan layanan yang sama.

Penelitian ini di titik beratkan untuk mencari mekanisme pembagian beban server secara seimbang berdasarkan jumlah akses pada satu saat yang sama. Setiap aplikasi memiliki karakteristik tersendiri dalam melakukan pembagian beban. Dengan mengetahui karakteristik Sistem Informasi Akademik, diharapkan diperoleh teknik pembagian beban server yang tepat, dengan efisiensi tinggi

II. TINJAUAN PUSTAKA

A. *OSI Layer*

OSI layer pada awalnya dikemukakan oleh Hubert Zimermann [1] pada tahun 1980 dan dalam perkembangannya di tetapkan oleh ITU-T sebagai standar yang di acu oleh seluruh desainer komunikasi elektronik. Hal ini dalam rangka mempermudah interaksi antar perangkat network. OSI Berdasarkan standar RFC 1122 tentang lapisan-lapisan interaksi jaringan. Lapisan OSI terdiri dari 7 lapis seperti ditunjukkan dalam Gambar 2.



Gambar 2. OSI Layer

Lapisan Atas (Upper Layer) merupakan layer aplikasi, yang menekankan pada pengolahan data. Sedangkan lapisan bawah (Lower Layer), berisi lapisan-lapisan yang bekerja sama guna menghantarkan data dari pengirim sampai tujuan. Dalam implementasinya, komunikasi data dan suara melibatkan seluruh lapisan tersebut. Penelitian ini di fokuskan pada lapisan bawah yang bertanggung dalam menghantarkan data dari pengirim sampai ke tujuan tanpa kesalahan sedikitpun.

B. *Transmission Control Protocol/Internet Protocol (TCP/IP)*

TCP/IP merupakan protokol data yang digunakan dalam jaringan internet. Dalam TCP/IP terdapat beberapa lapisan (layer) yang masing-masing menangani protokol-protokol yang berbeda-beda. Dalam TCP/IP ada empat lapisan, yang ditunjukkan pada Gambar 3.

C. *Transmission Control Protocol (TCP)*

Application Layer
Transport Layer
Internet Layer
Network Interface Layer

Gambar 3. TCP/IP

TCP menyediakan komunikasi yang berorientasi pada koneksi (*connection-oriented*) dan dapat diandalkan (*reliable*). Transfer menggunakan TCP lebih lambat daripada UDP dan biasanya digunakan untuk transfer data dalam jumlah besar, untuk memastikan bahwa data tersebut tidak perlu dikirimkan ulang.

Berorientasi pada koneksi (*connection oriented*) mengandung arti bahwa sebuah koneksi dibentuk saat komunikasi dimulai. Sepanjang koneksi tersebut, sejumlah informasi dipertukarkan.

Dapat diandalkan (*reliable*) mengandung arti bahwa sebuah "tanda terima" (acknowledgement) akan dikirimkan kembali ke pengirim sebagai verifikasi bahwa paket-paket yang bersangkutan telah diterima. Setiap kali sebuah segmen data diterima di tempat tujuan, sebuah *acknowledgement* dikirimkan kepada pengirim dalam periode waktu tertentu. Jika *acknowledgement* tidak terkirim dalam periode waktu tersebut, maka pengirim akan mengirimkan ulang data tersebut. Jika penerima mendapatkan data tersebut dalam kondisi rusak, paket yang rusak tersebut akan langsung dibuang. Penerima tidak akan mengirimkan *acknowledgement* untuk paket yang rusak, dan karena pengirim tidak menerima acknowledgement, maka data tersebut akan dikirimkan ulang [2].

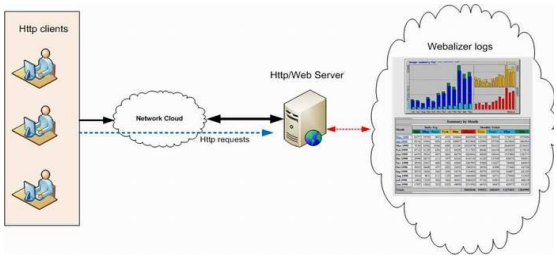
D. *Website*

Secara terminologi website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam World Wide Web (WWW) di Internet. WWW terdiri dari seluruh situs web yang tersedia kepada publik. Halaman-halaman sebuah situs web (web page) diakses dari sebuah URL yang menjadi "akar" (root), yang disebut homepage (halaman induk; sering diterjemahkan menjadi "beranda", "halaman muka"), URL ini mengatur web page untuk menjadi sebuah hirarki, meskipun hyperlink-hyperlink yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan.

E. *Hypertext Transfer Protocol (HTTP)*

Hypertext Transfer Protocol (HTTP) merupakan protokol yang digunakan untuk jenis layanan World Wide Web (WWW) pada jaringan TCP/IP. Pengembangan HTTP dikoordinasi oleh konsorsium WWW dan IETF (Internet Engineering Task Force) dan dipublikasikan melalui kumpulan RFC (Request For

Comments). RCF 2616 mendefinisikan HTTP/1.1 yang merupakan versi HTTP yang saat ini umum digunakan [3].



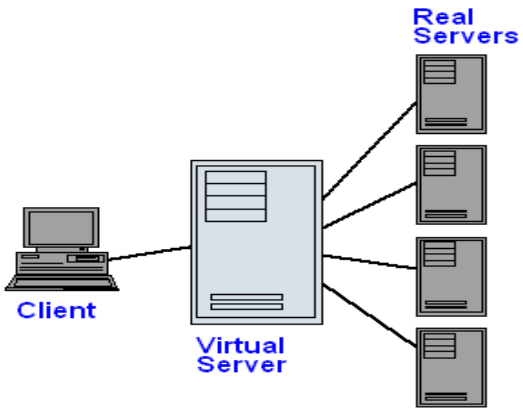
Gambar 4. Web Server

Sebuah HTTP client menginisiasi request dengan membuat koneksi TCP (Transmission Control Protocol) menuju server (umumnya adalah port 80). Sedangkan HTTP server menunggu adanya pesan request pada port yang telah ditentukan. Setelah menerima request dari client, server kemudian mengirimkan status line antara lain "HTTP/1.1 200 OK". Setelah itu dilanjutkan dengan mengirimkan file yang diinginkan client beserta pesan kesalahan atau informasi lainnya. HTTP diidentifikasi menggunakan Uniform Resource Identifier (URI) dengan format penulisan tertentu.

Protokol HTTP bersifat request-respon, yaitu dalam protokol ini client menyampaikan pesan request ke web server, dan web server kemudian memberikan respon yang sesuai dengan request tersebut. Request dan respon dalam protokol HTTP disebut sebagai request chain dan respon chain. Hubungan HTTP yang paling sederhana adalah terdiri atas hubungan langsung antara client agent dengan server asal.

F. Load balancing

Layanan Load Balancing memungkinkan pengaksesan sumber daya dalam jaringan didistribusikan ke beberapa host lainnya agar tidak terpusat sehingga unjuk kerja jaringan komputer secara keseluruhan bisa stabil.



Gambar 5. Load Balancing
Sumber: [4]

Ketika sebuah sebuah server sedang diakses oleh para pengguna, maka sebenarnya server tersebut sebenarnya sedang terbebani karena harus melakukan proses permintaan kepada para penggunanya. Jika penggunanya banyak maka prosesnyapun banyak.

Sesi-sesi komunikasi dibuka oleh server tersebut untuk memungkinkan para pengguna menerima servis dari server tersebut. Jika satu server saja terbebani, tentu server tersebut tidak bisa banyak melayani para penggunanya karena kemampuan melakukan proses ada batasnya.

Solusi yang paling ideal adalah dengan membagi-bagi beban yang datang ke beberapa server. Jadi yang melayani pengguna tidak hanya terpusat pada satu perangkat saja. Teknik ini disebut Teknik Load balancing. [4]

Dalam Load balancing terdapat beberapa algoritma penjadwalan diantaranya :

a. Least-Connection
   Least-connection mengalokasikan koneksi ke real-Server dengan jumlah koneksi paling sedikit[4].
b. Weighted Least-Connection (wlc)
   Weighted Least-Connection (wlc) hampir sama dengan agoritma least-connection, namun ini algoritma yang mempertimbangkan bobot. Bobot diberikan kepada masinng-masing real-Server. Untuk koneksi baru akan dipilih Server yang paling sedikit kapasitasnya. Kapasitas dihitung dengan membagi bobot suatu real-Server dibagi dengan bobot seluruh real-Server yang terhubung dengan virtualServer[5].
c. Round-Robin (rr)
   Round-Robin (rr) menempatkan semua real-Server pada antrian yang melingkar dan mengalokasikan koneksi bergantian untuk setiap turn [4].
d. Weighted Round-Robin (wrr)
   Penjadwalan ini memperlakukan real-Server dengan kapasitas proses yang berbeda. Masing-masing realServerdapat diberi bobot bilangan integer yang menunjukkan kapasitas proses, dimana bobot awal adalah 1[4].
e. Locality-Based Least-Connection (lblc)
   Mendistribusikan permintaan lebih ke Server dengan koneksi yang aktif lebih sedikit dibandingkan dengan IP tujuan mereka.Algoritma ini akanmeneruskan semua request kepada real-Server yang memilikikoneksi kurang aktif tersebut sampai kapasitasnya terpenuhi [5].
f. Locality-Based Least-Connection Scheduling with Replication
   Mendistribusikan permintaan lebih ke Server dengan koneksi yang aktif lebih sedikit dibandingkan dengan IP tujuan. Algoritma ini juga dirancang untuk digunakan dalam sebuah clusterServer proxy-cache. Ini berbeda denganpenjadwalan Locality-Based Least-Connection,dengan pemetaan alamat IP target untuk subset dari Nodereal-Server. Permintaan ini kemudian diteruskan ke Server dalam subset dengan jumlah koneksi paling sedikit. Jika semua Node untuk IP tujuan di atas kapasitas, dilakukan replikasi Server baru untuk alamat IP tujuan dengan menambahkan realServer dengan koneksi

yang sedikit dari keseluruhan real-Server pool untuk subset dari real-Server sebagai IP tujuan request. Node yang bermuatan lebihdikeluarkan dari subset real-Server untuk mencegah over-replikasi[5].

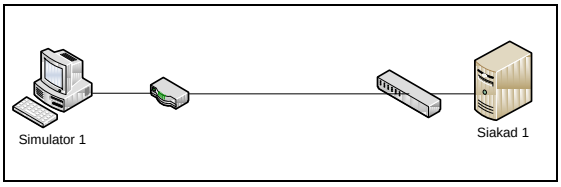
g. Destination Hash Scheduling
Menggunakan hash statis dari alamat IP tujuan untuk mengalokasikan koneksi[5].

h. Source Hash Scheduling
Mendistribusikan permintaan ke kumpulanreal-Server dengan melihat sumber IP dalam tabel hash statis[5].

i. Shortest Expected Delay (SEDR)
Mengalokasikan koneksi ke Server yang akan melayani permintaan dengan delay terpendek [5].
.

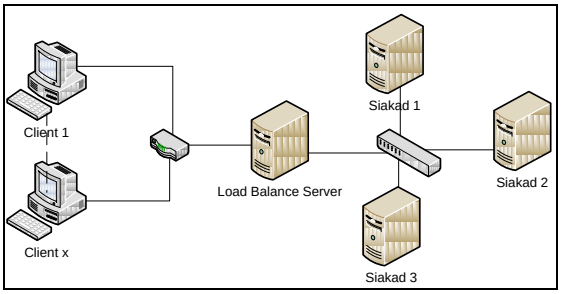
III. METODE PENELITIAN

Untuk menguji teori dan implementasi mekanisme load balancing dibuat topologi jaringan sesuai dengan kenyataan yang ada di ruang data center Universitas Brawijaya. Layout pertama ditunjukkan dalam Gambar 6. Pengujian dilakukan dengan mencoba membuat profil dari kemampuan suatu server dalam menangani banyak akses secara bersamaan. Pada proses ini diharapkan diperoleh jumlah akses maksimum sesuai dengan parameter standar webserver siakad UB.



Gambar 6. Layout Pengujian Profil Siakad

Pengujian kedua dilakukan dengan menggunakan load balancing server menggunakan parameter round-robin. Topologi server disusun seperti dalam Gambar 7



Gambar 7. Topologi Pengujian pembagian beban menggunakan Load Balance Server

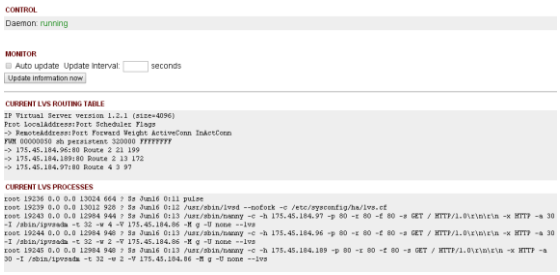
Pada tahapan ini disimulasikan akses diatas kemampuan satu server yang diperoleh dari pengujian pertama. Dari proses itu dianalisis pula jumlah akses tiap servernya. Mekanisme proses pembagian beban di ukur. Jumlah akses terus di tingkatkan hingga seluruh server tidak dapat menangani permintaan akses.

Setelah diperoleh data jumlah maksimum akses pada mekanisme round-robin, digunakan mekanisme least connection. Pada proses ini dicatat kembali jumlah akses yang dapat di tangani sebelum salah satu server jatuh (down).

Implementasi yang dilakukan dapat dibagi menjadi beberapa bagian yaitu:

1. Konfigurasi jaringan pada director
2. Konfigurasi jaringan pada masing-masing real server
3. Installasi LVS pada director
4. Installasi web server pada director
5. Installasi webserver pada masing-masin real server
6. Konfigurasi pada masing-masing real server

Konfigurasi jaringan pada director meliputi konfigurasi alamat ip, gateway serta DNS. Untuk dapat mengkonfigurasi harus sebagai root dari sistem. File yang perlu diubah dari konfigurasi adalah file-file yang berada di /etc/sysconfig/network-script/ifcfg-eth0 untuk konfigurasi alamat ip, /etc/sysconfig/network untuk konfigurasi default gateway, /etc/resolve.conf untuk konfigurasi DNS. File-file tersebut harus dikonfigurasi sesuai desain perancangan yang telah dijelaskan sebelumnya.



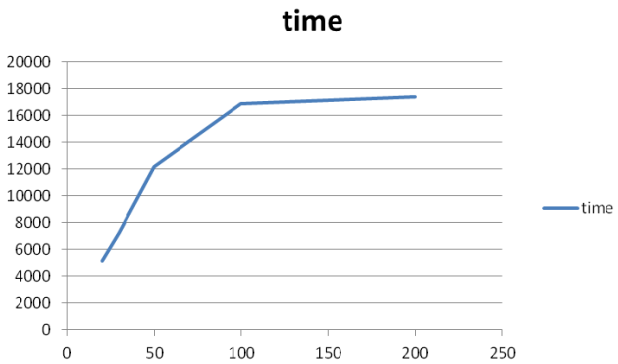
Gambar 8. konfigurasi jaringan pada Director

IV. PENGUJIAN DAN ANALISIS

Sistem *Load balancing* diuji dengan memberikan masukan jumlah client yang berbeda-beada untuk mendapatkan keluaran sesuai dengan parameter-parameter yang telah ditentukan.

A. *Pengujian pada lingkungan simulasi*

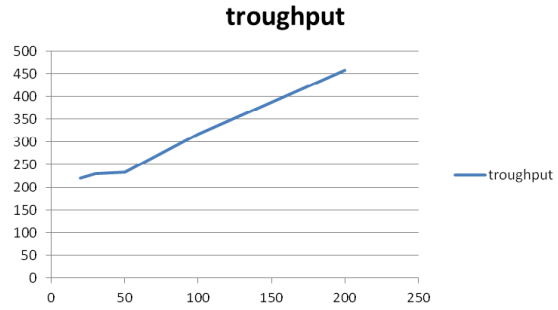
Pengujian dilakukan dengan stress tool dengan menggunakan 100 user.



Gambar 9. Waktu akses pada pengujian sebuah web server

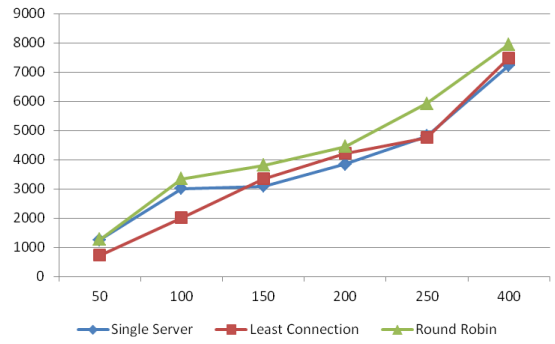
Dalam Gambar 8 terlihat director dengan ip https://siam.ub.ac.id:3636 dan real server dengan alamat ip masing-masing 175.45.184.97, 175.45.184.96 dan 175.45.184.94.

Hasil dari pengujian dengan mengunakan single server untuk jaringan lokal Universitas Brawijaya terlihat pada Gambar 9 dan Gambar 10



Gambar 10. Troughput pada pengujian sebuah web server.

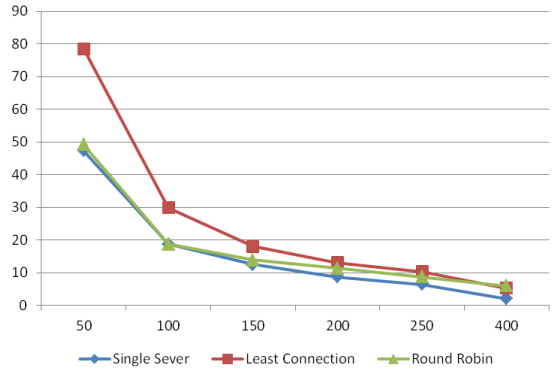
Dari pengujian diperoleh waktu respon dari masing-masing algoritma yang tampak pada grafik 11



Gambar 11. Rata-rata Respon untuk Load Balacing Server di Jaringan Lokal Universitas Brawijaya..

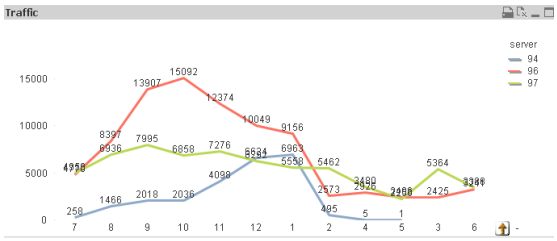
Dari pengujian juga didapatkan Troughput dari masing-masing algoritma yang tampak pada Gambar 12.

B. Hasil Pengujian Pada Lingkungan Produksi



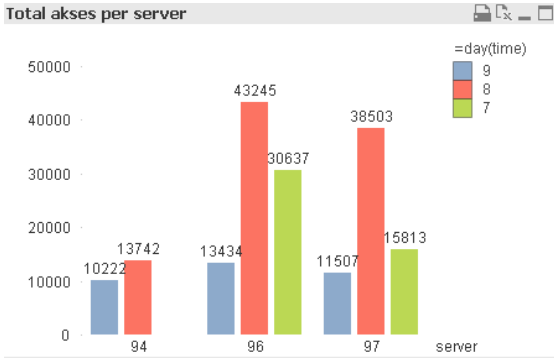
Gambar 12. Troughput untuk Load Balacing Server di Jaringan Lokal Universitas Brawijaya.

Hasil pengujian pada lingkungan simulasi diterapkan pada server SIAM pada level produksi. Metode yang digunakan pada load balancing ini adalah Least Connection yang didasarkan pada hasil simulasi. Server ini secara langsung diakses oleh mahasiswa pada proses daftar ulang dan pemrograman KRS. Pada implementasi di tingkat produksi digunakan 3 buah server dengan nama Server 94, server 96 dan server 97. Nomor pada server menunjukkan IP Real server tersebut. Sedangkan mekanisme balancing dilakukan oleh piranha server yang berada pada url http://siam.ub.ac.id. Pengujian dilakukan pada puncak akses KRS yaitu pada tanggal 7-9 Agustus 2012.

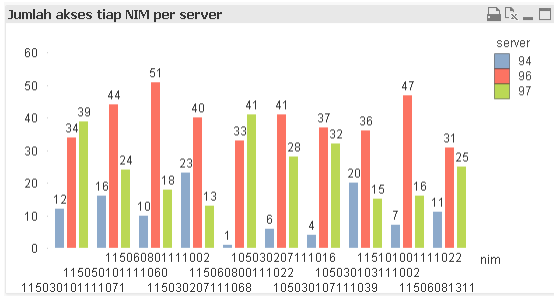


Gambar 13. Jumlah akses ke server SIAM dalam Jam.

Pada Gambar 13 ditunjukkan bahwa puncak akses di terima oleh server 96 pada pukul 10.00 dengan total akses sebanyak 15.093 request.



Gambar 14. Total Akses antara tanggal 7 – 9 Agustus.



Gambar 15. Akses ke server SIAM berdasarkan NIM Mahasiswa.

Gambar 14 menunjukkan bahwa total akses tertinggi adalah pada tanggal 8 Agustus 2012 pada server 96 sebesar 43.245 request.

Gambar 15 Menunjukkan 10 NIM paling sering melakukan akses ke server SIAM. Pada Gambar tersebut ditunjukkan bahwa akses mahasiswa telah terdistribusi ke 3 server. Server yang paling banyak menangani mahasiswa adalah server 96, namun juga di distribusikan ke server SIAM yang lain. Sehingga dengan distribusi beban kerja ketiga server tersebut jumlah akses yang dapat di layani semakin meningkat

V. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang dilakukan terhadap kinerja sistem dapat diambil kesimpulan sebagai berikut:

1. Penggunaan Load balancing mengurangi tingkat error yang terjadi dalam mengakses website. Tingkat error yang terjadi dalam menggunakan single web server adalah 360 sedangkan jika menggunakan Load balancing adalah 152 dengan algoritma Round Robin dan 131 pada Least Connection dengan jumlah user 400 orang.

2. Penggunaan load balancing telah meningkatkan jumlah mahasiswa yang dapat di layani oleh SIAM. Pada puncak akses dalam 1 jam server dapat menangani sebanyak 23.986 request dalam 1 jam. Puncak akses terjadi pada tanggal 8 Agustus 2012 sebanyak 43.245 request dalam 1 hari.

REFERENCES

[1] Hubert Zimmermann, " OSI Reference Model- The ISO Model of Architecture for Open System Interconnection" IEEE transaction on communications, vol.28, issue 4, April 1980.

[2] Vinton G. Cerf, Robert E. Kahn, (May 1974). "A Protocol for Packet Network Intercommunication". IEEE Transactions on Communications 22 (5): 637–648.

[3] Fielding, Roy T.; Gettys, James; Mogul, Jeffrey C.; Nielsen, Henrik Frystyk; Masinter, Larry; Leach, Paul J.; Berners-Lee "Hypertext Transfer Protocol -- HTTP/1.1," The Internet Society, 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2616.txt. [Diakses 18 7 2012].

[4] C. Kopparapu, Load Balancing Server, Firewall, and Caches, Canada: Wiley, 2001.

[5] RedHat, Inc, Linux VirtualServer (LVS) for Red Hat Enterprise Linux, RedHat, 2007.

[6] "Round-Robin Scheduling," 3 February 2006. [Online]. Available: http://kb.linuxvirtualserver.org/wiki/Round-Robin_Scheduling. [Diakses 2012 7 2012].