

Halaman Judul

LAPORAN PENELITIAN
PEMANFAATAN TEKNOLOGI POUND
UNTUK PEMERATAAN BEBAN SERVER WEB



oleh
W A G I T O, S.T., M.T.
NIDN : 0522126901
NPP : 961080

Mendapat Bantuan Biaya Penelitian dari Puslitbang dan PPM
Semester Ganjil 2016/2017

Sekolah Tinggi Manajemen Informatika dan Komputer
AKAKOM YOGYAKARTA
Tahun 2017

HALAMAN PENGESAHAN

1. a. Judul Penelitian : Pemanfaatan Teknologi Pound Untuk Pemerataan Beban server web
b. Bidang Ilmu : Jaringan Komputer
c. Kategori : Implementasi Jaringan Komputer
2. Ketua Peneliti
a. Nama : wagito, S.T., M.T.
b. NIDN : 0522126901
c. NPP : 961080
d. Pangkat/Golongan : Pembina Tk 1 / IV B
e. Jabatan Fungsional : Lektor Kepala
f. Jurusan/Prodi : Teknik Informatika
g. Alamat Institusi : Jalan Raya Janti
Karang Jambe, Yogyakarta
5. Waktu Penelitian : 6 bulan
6. Biaya Penelitian :

Yogyakarta, Juni 2017
Mengetahui

Ketua Prodi


Ir. M. Guntara, M.T.
NPP. 8910919

Ketua Peneliti


wagito, S.T., M.T.
NPP. 961080

Menyetujui

Pjs. Kepala Puslitbang dan PPM
STMIK AKAROM


Edy Prayitno, S.Kom., M.Eng
NPP. 151185

Kata Pengantar

Puji syukur saya panjatkan ke hadirat Allah S.W.T. karena hanya dengan rahmat dan hidayah-Nya. Berkat pertolongan dan tuntunan-Nya serta dengan berbagai usaha akhirnya penelitian ini berhasil diselesaikan dengan baik.

Penelitian yang berjudul Pemanfaatan Teknologi Pound Untuk Pemerataan Beban *server web* dilakukan untuk meneliti distribusi beban akses pada *server web*. Penelitian dilakukan dengan metode pengamatan terhadap jumlah beban akses yang diterima oleh *server web*. Pengamatan akses *web* dilakukan selama enam bulan. Hasil pencatatan beban akses *web* ditampilkan dalam bentuk grafik.

Penulis menyadari bahwa hasil penelitian ini masih banyak kekurangannya, sehingga kritik dan saran yang membangun untuk lebih mengembangkan hasilnya sangat diharapkan. Semoga hasil penelitian ini bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Penulis

Daftar Isi

Halaman Judul.....	
HALAMAN PENGESAHAN.....	ii
Kata Pengantar.....	iii
Daftar Isi.....	iv
Daftar Gambar.....	vii
ABSTRAK.....	viii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
1.6 Target Luaran.....	7
BAB 2 TINJAUAN PUSTAKA.....	8
BAB 3 TEORI.....	10

3.1 Pemerataan Beban (<i>Load Balancing</i>).....	10
3.2 Pound.....	14
3.3 Konfigurasi Pound.....	16
 BAB 4 METODE PENELITIAN.....	 20
4.1 Bahan Penelitian.....	20
4.2 Alat.....	21
4.3 Jalan Penelitian.....	22
4.3.1 Perangkat Keras.....	23
4.3.2 Perangkat Lunak.....	25
4.3.3 Tahapan Penelitian.....	26
 BAB 5 IMPLEMENTASI DAN PEMBAHASAN.....	 28
5.1 Implementasi.....	28
5.1.1 Konfigurasi Server Balancer Pound.....	29
5.1.2 Konfigurasi Server Web.....	32
5.1.3 Konfigurasi Router.....	36
5.2 Pembahasan.....	36
 BAB 6 KESIMPULAN.....	 42
6.1 Kesimpulan.....	42
6.1 Saran.....	42

Daftar Pustaka.....	44
LAMPIRAN.....	L-1
File /etc/init.d/pound.....	L-1
File counter2.php.....	L-3
Count Harian Agustus-Januari.....	L-4
Count Mingguan Agustus-Januari.....	L-7
Count Bulanan Agustus-Januari.....	L-7
Curriculum Vitae.....	L-8
Personalia Penelitian.....	L-9
Biaya Penelitian.....	L-10
Jadwal Penelitian.....	L-11
Surat Keputusan.....	L-12

Daftar Gambar

Gambar 2.1 Isi File Konfigurasi Pound.....	18
Gambar 4.1 Jaringan Pemerata Beban.....	23
Gambar 5.1 Perbedaan Akses Dua Server Web Secara Harian.....	37
Gambar 5.2 Perbedaan Akses Dua Server Web Secara Mingguan.....	38
Gambar 5.3 Perbedaan Akses Dua Server Web Secara Bulanan.....	39

ABSTRAK

Akses *web* merupakan layanan yang banyak disediakan pada jaringan Internet. Beban terhadap akses *web* bisa sangat tinggi apabila layanan *web* tersebut banyak dipakai masyarakat. *Server* harus mampu melayani akses tinggi terhadap *web*. Supaya dapat melayani beban akses yang tinggi, perlu dilakukan distribusi terhadap beban tersebut. Salah satu cara yang dipakai adalah melibatkan beberapa *server web*. Penelitian pemanfaatan teknologi Pound untuk pemerataan beban *server web* bertujuan untuk menerapkan pemerataan beban, menyusun metode untuk mengamati beban *server web* dan menganalisis pembagian beban akses terhadap *server web* selama kurun waktu tertentu.

Beban akses didistribusikan pada masing-masing *server* secara merata. Teknik yang dipakai adalah pemerataan beban. Proses pemerataan beban dilakukan oleh perangkat lunak Pound. Layanan akses *web* dilakukan oleh perangkat lunak Apache. Penelitian dilakukan dengan metode pengamatan terhadap jumlah akses yang diterima oleh *server web*. Pengamatan akses *web* dilakukan selama enam bulan. Hasil pencatatan akses *web* diolah untuk menentukan beda beban akses yang diterima *server web* dan ditampilkan dalam bentuk grafik.

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* dapat terbagi dengan perbedaan jumlah akses harian sebesar 5,5%. Akses mingguan dan akses bulanan menghasilkan angka perbedaan sebesar 3,18 %. Ada sedikit perbedaan hasil persentase jumlah akses harian, mingguan dan bulanan yang disebabkan karena adanya gangguan pada salah satu *server* pada tanggal tertentu.

Kata kunci: pemerataan, beban, akses, *web*

BAB 1 PENDAHULUAN

1.1 Latar Belakang

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, pada saat ini sudah melibatkan Internet seperti melihat acara televisi, mendengarkan musik, melakukan panggilan telepon dan sebagainya. Perkembangan ini menuntun 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 salah satu layanan yang paling banyak dimanfaatkan oleh para pengguna jaringan Internet baik sebagai sarana untuk menyebarkan informasi maupun mencari informasi.

Informasi ditulis dan disebarkan menggunakan program sekrip utama HTML. Dalam perkembangannya, sekrip HTML dikombinasikan dengan sekrip

lain seperti skrip JavaScript yang bekerja pada sisi klien atau pun skrip PHP yang bekerja pada sisi *server*. Kombinasi demikian menghasilkan suatu halaman *web* yang lebih menarik serta informasi data yang dinamis.

Untuk layanan *web* digunakan protokol HTTP atau HTTPS. Protokol HTTP menggunakan *port* virtual 80 sedangkan layanan HTTPS menggunakan *port* virtual 443. Kedua protokol ini secara luas digunakan untuk memberi layanan *web* pada para pengguna Internet. Dua protokol ini hanya berbeda dalam hal bentuk data yang ditransmisikan. Protokol HTTPS menggunakan fitur keamanan untuk transmisi data, sedemikian sehingga transmisi data dilakukan secara enkripsi. Untuk layanan tertentu akan lebih aman menggunakan protokol HTTPS. Namun demikian, untuk layanan *web* umumnya masih menggunakan protokol HTTP.

Layanan HTTP dan HTTPS dilakukan oleh *server web*. Ada beberapa perangkat lunak *server web* yang biasa digunakan pada sistem jaringan komputer, di antaranya Apache, Nginx, Lighttpd, Cherokee, Caddy dan sebagainya. Dari sekian banyak perangkat lunak *server web*, Apache merupakan *server web* yang paling populer dan paling banyak digunakan. Perangkat lunak Apache banyak digunakan karena kemudahan dan kelengkapan fitur yang berkaitan dengan layanan *web*.

Jaringan komputer merupakan sistem yang kompleks, *routing* paket data yang jumlahnya bisa mencapai ratusan, ribuan, atau bahkan jutaan setiap detik. Oleh karena itu, supaya jaringan komputer dapat menangani sejumlah besar data,

penting bahwa data harus disalurkan secara efisien. Misalnya, jika ada sepuluh *router* dalam jaringan namun hanya dua dari seluruh *router* yang melakukan pekerjaan secara penuh, sedang *router* yang lain kurang bekerja penuh. Hal demikian menyebabkan kinerja jaringan komputer tidak berjalan sangat efisien. Jaringan akan berjalan lebih cepat jika setiap *router* menangani lalu-lintas paket data yang relatif berimbang. Demikian juga, jika sebuah situs *web* mendapat ribuan akses setiap detik, akan lebih efisien untuk membagi lalu-lintas akses pada beberapa *server web* daripada hanya mengandalkan satu *server* tunggal untuk menangani beban penuh.

Apache sebagai perangkat lunak penyedia layanan *web* harus mampu menangani beban permintaan 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. Salah satu cara yang dipakai adalah melibatkan beberapa *server web* untuk melayani akses *web* pengguna. Beban akses didistribusikan pada masing-masing *server* secara merata. Teknik yang dipakai adalah pemerataan beban (*load balancing*)

Load balancing membantu membuat jaringan komputer lebih efisien, mendistribusikan pengolahan dan lalu-lintas secara merata di seluruh jaringan dan memastikan tidak ada perangkat tunggal kewalahan. *Server web*, seperti dalam contoh paragraf sebelumnya, sering menggunakan teknologi *load balancing* untuk pemerataan atau membagi beban lalu-lintas akses di antara beberapa *server web*

yang berbeda.

Penerapan *load balancing* pada *server web* jaringan lokal atau jaringan global, membutuhkan perangkat keras atau perangkat lunak yang membagi lalu-lintas masuk antar *server* yang tersedia. 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* seperti ini sering disebut sebagai pemerata beban (*load balancer*).

Salah satu teknologi yang dapat dipakai untuk pemerata beban adalah perangkat lunak Pound. Perangkat lunak Pound berfungsi sebagai *proxy reverse*, pemerata beban dan HTTPS *front-end* untuk *server web*. Pound dikembangkan untuk memungkinkan penyebaran beban antara beberapa *server web* dan memungkinkan SSL *wrapper* bagi seluruh *server web* yang tidak dihadapkan secara langsung. Pound didistribusikan di bawah lisensi GPL, sehingga bebas digunakan dan dikembangkan (Apsis, IT Security, 2016).

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian adalah bagaimana menerapkan perangkat lunak pemerata beban Pound untuk proses pembagian beban akses *server web* serta menganalisis pembagian beban akses *server web* yang dihasilkan oleh perangkat lunak Pound..

1.3 Batasan Masalah

Pada penelitian ini ditetapkan batasan masalah karena dengan cukup luasnya lingkup yang bisa dicakup dalam penelitian. Batasan masalah yang ditetapkan pada penelitian ini adalah sebagai berikut.

- a) Pengamatan terhadap pembagian beban akses *server* dilakukan terhadap *server web* menggunakan protokol HTTP.
- b) Jumlah *server web* yang diamati adalah dua *server* yang dipasang pada jaringan yang sama.
- c) Perangkat lunak *server web* yang digunakan untuk pengamatan pembagian beban akses adalah Apache.
- d) Pengamatan dilakukan pada situs www.akakom.ac.id yang merupakan situs penyedia informasi.
- e) Pengamatan beban akses *server web* dilakukan untuk periode waktu harian, mingguan dan bulanan.
- f) Pengamatan dilakukan selama kurun waktu enam bulan atau satu semester kegiatan akademik.
- g) Perangkat lunak untuk melayani pembagian beban akses *server web* adalah pemerata beban Pound.
- h) Perangkat lunak Pound dipasang pada *server* yang terpisah dengan *server web* yang diamati
- i) Beban akses *web* didistribusikan pada dua *server web* yang identik dalam hal kapasitas sumber daya.

- j) Dilakukan sinkronisasi antara dua *server web* secara berkala untuk menjaga agar dua *server web* selalu identik.
- k) Pencatatan beban hanya dilakukan pada saat pengguna melakukan akses pada halaman depan.
- l) Hasil pencatatan ditampilkan dalam bentuk grafik akses harian, mingguan dan bulanan.

1.4 Tujuan Penelitian

Penelitian analisis pembagian beban *server* menggunakan pemerata beban Pound adalah sebagai berikut.

- a) Menerapkan pemerata beban Pound sebagai perangkat lunak untuk membagi beban akses *server web*.
- b) Menyusun metode untuk mengamati beban akses *server web* berdasarkan catatan akses yang dihasilkan *server web*
- c) Menganalisis pembagian beban akses terhadap *server web* selama kurun waktu tertentu.
- d) Menganalisis beban akses *server web* secara periode waktu harian, mingguan dan bulanan.

1.5 Manfaat Penelitian

Penelitian yang dihasilkan bermanfaat bagi pengelola *server web* untuk menentukan metode pembagian beban akses *server web* khususnya. Bagi

penyedia layanan situs *web* penelitian bermanfaat untuk mengatasi beban tinggi yang sering dialami oleh *server web*, sedemikian sehingga mengurangi risiko kemacetan *server*. Manfaat berikutnya adalah mengurangi risiko kerusakan *server* akibat beban akses yang terlalu tinggi.

1.6 Target Luaran

Hasil penelitian direncanakan dilakukan publikasi dan seminasi pada kegiatan ilmiah. Publikasi yang diinginkan adalah pada jurnal ilmiah nasional yang sudah akreditasi. Jika sulit masuk pada jurnal akreditasi, maka paling tidak bisa masuk jurnal ilmiah nasional.

BAB 2 TINJAUAN PUSTAKA

Beberapa penelitian berkaitan dengan pemerataan beban telah dilakukan dan dipublikasikan dalam bentuk jurnal. Penelitian-penelitian tersebut berbeda-beda pada objek yang diteliti dan metode pelaksanaan penelitian.

Penelitian Ari Budi Noviyanto tahun 2015 yang dipublikasikan pada jurnal JARKOM tentang perancangan dan implementasi *load balancing reverse proxy* menggunakan HAProxy pada aplikasi *web*. Pada penelitian ini digunakan HAProxy sebagai pemerata beban *server web*. Pada penelitian ini dihasilkan hasil pemerataan beban ketika semua *server* dalam keadaan hidup dan hasil pemerataan beban apabila salah satu *server* mengalami kegagalan. Penelitian ini belum dicobakan pada beban akses lalu-lintas data sesungguhnya.

Penelitian I Made Widhi Wirawan tahun 2011 yang dipublikasikan pada Jurnal Ilmu Komputer tentang implementasi *load balance* pada jaringan *multihoming* menggunakan *router* dengan metode *round robin*. Penelitian ini membahas penggunaan *router* untuk proses *load balance* pada jaringan yang punya beberapa jalur sumber Internet. Penelitian juga dilakukan melihat kondisi *backup/failover*. Penelitian ini lebih ditekankan pada *load balance* akses ke jaringan luar.

Penelitian Alimuddin yang dipublikasikan pada jurnal IJCCS tahun 2016

tentang Peningkatan Kinerja Siakad Menggunakan Metode *Load Balancing* dan *Fault Tolerance* Di Jaringan Kampus Universitas Halu Oleo dalam menangani beban tinggi *server web*. Perlu metode dalam pendistribusian beban agar merata pada masing-masing *server* yaitu metode *load balancing* dengan algoritme *round robin*, sehingga skalabilitas siakad tinggi. Sedangkan untuk menangani kegagalan *server* perlu *fault tolerance* agar ketersediaan siakad menjadi tinggi. Hasil penelitian menunjukkan bahwa dengan *load balancing* dapat meminimalkan *response time*, meningkatkan *throughput*, memaksimalkan pemanfaatan sumber daya, dan terhindar dari *overload*.

Penelitian Pemanfaatan Teknologi Pound Untuk Pemerataan Beban *server web* berusaha menerapkan teknologi 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 akses *server web* selama kurun waktu tertentu dan menganalisis beban akses *server web* dengan periode waktu harian, mingguan dan bulanan.

BAB 3 TEORI

3.1 Pemerataan Beban (*Load Balancing*)

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*. (Nginx, 2017)

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. (Nginx, 2017)

Dengan cara demikian, *pemerata* beban melakukan fungsi-fungsi sebagai berikut:

- a) mendistribusikan permintaan klien atau beban jaringan secara efisien di beberapa *server*
- b) Memastikan ketersediaan dan keandalan yang tinggi dengan mengirimkan permintaan hanya untuk *server* yang sedang online
- c) 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: (Nginx, 2017)

- a) *Round Robin* - Permintaan didistribusikan pada seluruh kelompok *server* secara berurutan.
- b) *Least Connections* – Sebuah permintaan baru dikirim ke *server* dengan koneksi sesaat paling sedikit kepada klien. Kapasitas komputasi relatif masing-masing *server* adalah faktor dalam menentukan mana yang memiliki koneksi paling sedikit.
- c) *IP Hash* - Alamat IP dari klien digunakan untuk menentukan *server* mana yang menerima permintaan tersebut.

Pemerata beban biasanya dapat dibagi dalam dua bentuk: berbasis perangkat keras dan berbasis perangkat lunak. Vendor solusi berbasis perangkat keras memuat perangkat lunak *proprietary* pada mesin yang disediakan, yang sering menggunakan prosesor khusus. Untuk mengatasi meningkatnya lalu-lintas pada *website*, harus membeli mesin lebih banyak atau lebih besar dari vendor. Solusi perangkat lunak umumnya berjalan pada perangkat keras komoditas, yang demikian membuat lebih murah dan lebih fleksibel. Perangkat lunak dapat dipasang pada perangkat keras pilihan atau pada lingkungan *cloud* seperti AWS EC2.

Ada beberapa manfaat penggunaan *server* pemerataan beban. Beberapa manfaat diterapkannya pemerata beban adalah sebagai berikut. (Array Networks, 2017)

a) 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

b) Kinerja (*Performance*)

Server pemerataan beban mengurangi waktu respons dan waktu transaksi untuk *website*, aplikasi dan layanan *cloud*. Pengurangan waktu respons dan waktu transaksi dilakukan dengan *caching*, kompresi dan akselerasi

SSL sampai lima kali kinerja aplikasi yang lebih baik

c) 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.

d) 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

e) 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.

3.2 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. (Apsis GmbH, 2017)

Beberapa fitur yang dapat dilakukan perangkat lunak Pound adalah sebagai berikut: (Apsis GmbH, 2017)

- a) *reverse-proxy*: melewati permintaan dari *browser* klien untuk satu atau lebih *server back-end*.
- b) *pemerata* beban: mendistribusikan permintaan dari *browser* klien kepada beberapa *server* yang berlaku sebagai *back-end*, sekaligus menjaga sesi informasi.
- c) pembungkus SSL: dekripsi permintaan protokol HTTPS dari *browser* klien dan melewatkannya sebagai permintaan protokol HTTP biasa pada *server back-end*.
- d) HTTP/HTTPS *sanitizer*: verifikasi permintaan untuk tingkat kebenaran dan hanya menerima dalam bentuk terbaik.
- e) *server failover*: ketika *server back-end* gagal, Pound akan mencatat bukti

dan menghentikan pelewatan lalu-lintas data lewat permintaan sampai *server* tersebut pulih.

- f) *request redirector*: permintaan dapat didistribusikan di antara *server* sesuai dengan URL yang diminta.

Perangkat lunak Pound adalah program yang sangat kecil, mudah diaudit untuk masalah keamanan. Hal ini dapat dijalankan sebagai *setuid/setgid* dan/atau dikungkung *chroot*. Pound tidak mengakses hard-disk sama sekali (kecuali untuk membaca *file* sertifikat pada awal, jika diperlukan) dan dengan demikian seharusnya tidak menimbulkan ancaman keamanan untuk mesin apa pun.

Beberapa hal yang tidak dapat dilakukan oleh perangkat lunak Pound adalah sebagai berikut:

- a) Pound bukan *server web*: dengan sendirinya, Pound tidak melayani isi *web* – Pound mengontak *server back-end* untuk tujuan akses *web*.
- b) Pound bukan *accelerator web*: tidak ada *caching* dilakukan oleh Pound, setiap permintaan dilewatkan sebagaimana adanya ke *server back-end*.

Pada rilis 1,0 Pound dinyatakan sebagai kode produksi berkualitas. Beberapa pengguna telah melaporkan pemanfaatan Pound dan berhasil dalam lingkungan produksi. Volume terbesar dilaporkan sampai saat ini adalah sebuah situs dengan rata-rata sekitar 30M permintaan per hari, puncak pada lebih dari 600 permintaan/sec.

Pound telah berhasil digunakan dalam produksi dengan berbagai *server web*, termasuk Apache, IIS, Zope, WebLogic, Jakarta/Tomcat, iPlanet, dan lain-lain Pound melewati permintaan umum, memberi tanggapan balik dan menjaga tidak berubah. Dengan demikian tidak ada alasan untuk berpikir bahwa ada *server web* yang tidak kompatibel.

Beberapa *browser* klien yang sudah diuji menggunakan perangkat lunak Pound adalah sebagai berikut.

- a) IE 5.0/5.5 and later (Windows) HTTP/HTTPS
- b) Netscape 4.7 (Windows/Linux) HTTP/HTTPS
- c) Mozilla Firefox (Windows/Linux) HTTP/HTTPS
- d) Konqueror (Linux) HTTP/HTTPS
- e) Galleon (Linux) HTTP/HTTPS
- f) Opera (Linux/Windows) HTTP/HTTPS
- g) Lynx (Linux) HTTP

Mengingat bahwa Pound dalam produksi dan tidak ada masalah yang dilaporkan, tidak ada alasan bahwa *browser* lain akan ada masalah. Beberapa masalah yang diamati dengan implementasi SSL, terutama dengan Opera 6, tetapi hal ini harus sudah selesai pada versi ini.

3.3 Konfigurasi Pound

Pengaturan kerja perangkat lunak Pound dilakukan dengan cara mengatur

file konfigurasi */etc/pound.cfg* yang berupa *file* teks. *File* ini dapat diakses menggunakan editor *file* teks yang biasa dipakai untuk konfigurasi jaringan dari jarak jauh seperti *vi*.

Isi *file* konfigurasi Pound yang berkaitan dengan penelitian yang dilakukan adalah sebagai berikut (Segall, R., 2010)

- a) User: user sistem yang menjalankan Pound
- b) Group: group untuk user sistem
- c) ListenHTTP: pengaturan untuk menentukan pengaturan protokol HTTP.
Pada pengaturan ini juga diatur sebagai berikut.
 - Address: alamat IP *server balancer*
 - Port: nomor *port* kerja protokol HTTP, biasanya 80.
- d) ListenHTTPS: pengaturan untuk menentukan pengaturan protokol HTTPS. Pada pengaturan ini juga diatur sebagai berikut.
 - Address: alamat IP *server balancer*
 - Port: nomor *port* kerja protokol HTTP, biasanya 443
- e) Service: layanan *server back-end* yang disediakan. Pada bagian ini terdapat opsi Backend yang mengatur beberapa hal yang berkaitan dengan *server web* yang digunakan untuk menerima beban permintaan layanan *web* sebagai berikut
 - Address: alamat IP *server web back-end*
 - Port: nomor *port* kerja *server web back-end*

Isi *file* konfigurasi yang digunakan untuk mengatur perilaku pemerata beban Pound ditampilkan pada Gambar 2.1.

```
User "<user sistem>"
Group "<group untuk user sistem>"
. . . . .
. . . . .
ListenHTTP
    Address <alamat IP server balancer>
    Port    <nomor port kerja server balancer protokol HTTP>
    . . . . .
End
. . . . .
ListenHTTPS
    Address <alamat IP server balancer>
    Port    <nomor port kerja server balancer protokol HTTPS>
    . . . . .
End
. . . . .
Service
    BackEnd
        Address <alamat IP server web>
        Port    <nomor port kerja server web>
    End
    . . . . .
End
```

Gambar 2.1 Isi File Konfigurasi Pound

Opsi User dan Group hanya bisa berisi satu macam saja. Opsi ListenHTTP dan ListenHTTPS bisa diaktifkan salah satu (salah satu harus aktif) atau keduanya, tergantung jenis protokol apa yang akan dilayani oleh Pound. Opsi Service menentukan jumlah *server web back-end* yang akan terlibat dalam pengaturan *load balancing*. Jumlah service minimal satu (supaya bisa berjalan), sedangkan jumlahnya tergantung pada jumlah *server web* yang dipakai. Namun

jumlah minimal *server web* sebaiknya dua, upaya pengaruh pemerataan beban dapat dilakukan.

Pada penelitian ini, layanan yang akan diuji hanya berupa *web* dengan protokol HTTP yang bekerja pada *port* 80. Dengan demikian opsi yang diaktifkan hanya ListenHTTP, sedangkan opsiListenHTTPS tidak diaktifkan. Dengan kata lain, jika ada permintaan layanan HTTPS pada *server balancer*, maka tidak akan ditanggapi. (Segall, R., 2010)

BAB 4 METODE PENELITIAN

4.1 Bahan Penelitian

Bahan yang digunakan dalam penelitian berupa hasil *log* akses *server web*. *Log server web* berupa *file* teks yang dapat diolah menggunakan perangkat lunak editor teks. Selain itu juga ada bahan penelitian berupa simpanan *database* dalam format SQLite yang dapat diolah menggunakan perangkat lunak *tool* SQLite. Dua *log* ini ditempatkan pada setiap *server web* yang diamati.

Log server web dihasilkan oleh perangkat lunak Apache yang dioperasikan untuk menangani permintaan akses *web*. Catatan *database* SQLite dihasilkan oleh skrip program yang dirancang khusus untuk mencatat berapa jumlah akses terhadap layanan *web* setiap hari. Sebagai data utama adalah catatan *database* SQLite.

Pengamatan akses *web* dilakukan selama kurun waktu enam bulan atau satu semester operasi *server web*. Pencatatan dilakukan setiap hari untuk menghitung berapa jumlah akses layanan *web* pada setiap *server web*. Hasil pencatatan diolah menjadi suatu catatan akses dalam periode harian, mingguan serta bulanan.

4.2 Alat

Alat yang digunakan dalam penelitian ini berupa perangkat keras dan perangkat lunak. Perangkat lunak yang digunakan dalam penelitian berupa sistem operasi dan perangkat lunak pendukung lain sebagai berikut.

- a) Sistem Operasi Linux Mageia 2015.
- b) *server web* Apache.
- c) pemerata beban Pound
- d) Synchronizer Rsync
- e) database SQLite
- f) SQLite3 tools
- g) *server* SSH
- h) SSH client
- i) Mikrotik Router OS
- j) Sshpass
- k) Midnight commander

Perangkat keras yang digunakan dalam penelitian meliputi *server* maupun peralatan pendukung jaringan komputer dengan spesifikasi sebagai berikut.

- *server*
 - a) Prosesor Intel(R) Xeon® X5650 @2.67GHz
 - b) RAM 1 GB.
 - c) Tipe sistem 64 bit.

- d) Harddisk 20 GB.
- e) Gigabit Ethernet
- Peralatan jaringan komputer
 - a) Switch
 - b) Router
 - c) Sistem Pengkabelan

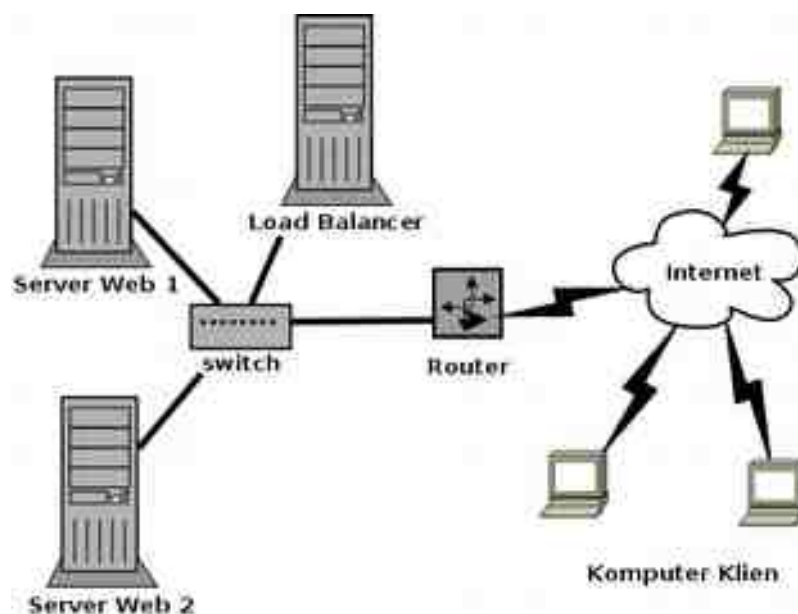
4.3 Jalan Penelitian

Penelitian dilakukan dalam dua tahap yaitu mulai perancangan konfigurasi perangkat keras dan perancangan menggunakan perangkat lunak Virtual Box. Perancangan konfigurasi perangkat keras untuk mengidentifikasi seluruh perangkat keras yang terlibat dalam sistem jaringan serta konfigurasi pada masing-masing mesin. Perancangan pada Virtual Box dimaksudkan untuk memudahkan proses percobaan sebelum diterapkan pada *server* sesungguhnya.

Setelah rancangan pada Virtual Box dipastikan dapat berjalan secara benar, tahap selanjutnya adalah menerapkan hasil rancangan pada *server* sesungguhnya. Rancangan meliputi konfigurasi jaringan *server*, konfigurasi khusus perangkat lunak pemerata beban Pound, sekrip untuk pencatatan jumlah akses yang diterima *server web*, serta sekrip untuk melakukan sinkronisasi antar *server web*. Dalam penelitian ini digunakan objek situs layanan *web* yang menjadi sarana penyebar informasi ke dunia maya Internet.

4.3.1 Perangkat Keras

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* dari luar (*front-end*) dan mendistribusikan beban kepada dua *server web* yang berada di belakangnya. Dua mesin *server web*, masing-masing berfungsi sebagai *server* akhir penerima akses *web* dan menanggapi setiap permintaan akses *web*. Dua mesin *server web* berlaku sebagai *back-end*. 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 menggunakan jaringan kabel maupun jaringan tanpa kabel (*wireless*).



Gambar 4.1 Jaringan Pemerata Beban

Diagram jaringan komputer beserta seluruh peralatan pendukung serta semua mesin *server* yang akan digunakan dalam penelitian ditunjukkan pada Gambar 4.1.

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* pertama dan kedua. *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 *web* yang sesungguhnya dan menanggapi permintaan akses *web* tersebut. Pada penelitian digunakan dua *server web* yang identik baik identik perangkat keras maupun identik perangkat lunak yang dipasang.

Dua *server web* harus selalu melakukan sinkronisasi. Sinkronisasi dilakukan secara periodik pada waktu-waktu tertentu. Sinkronisasi dimaksudkan supaya jika terjadi perubahan pada salah satu *server*, *server* yang lain segera menyesuaikan. Sinkronisasi terutama dilakukan pada aplikasi *web* yang

digunakan pada kedua *server*.

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.

4.3.2 Perangkat Lunak

Router menggunakan sistem operasi Mikrotik RouterOS. Fitur Mikrotik yang digunakan dalam kaitan dengan penelitian adalah IP-Route dan IP-Firewall. Dalam hal ini Router difungsikan sebagai pengatur *route* paket data dan Firewall antara jaringan publik dan jaringan privat. Akses *web* yang berkaitan dengan penelitian diteruskan kepada *server load balancer*.

Mesin *load balancer* menggunakan sistem operasi Linux Mageia 2014. Pada mesin *server* dipasang perangkat lunak pemerata beban Pound. Pound berlaku sebagai penerima akses yang diteruskan dari *router*. Pound akan menyeimbangkan beban akses menuju *server web* yang pada akhirnya diterima oleh aplikasi *web*.

Mesin *server web* juga menggunakan sistem operasi Linux Mageia 2014. Pada mesin *server* dipasang perangkat lunak untuk memberi layanan *web* yaitu untuk aplikasi *web*, *rsync* untuk sinkronisasi antar aplikasi *web* serta perangkat

lunak untuk mendukung komunikasi melalui protokol SSH. Protokol SSH diperlukan sebagai sarana sinkronisasi antar server

Mesin Klien menggunakan sistem operasi apa pun baik Linux, Windows, Android dan sebagainya. Yang penting pada mesin klien dapat digunakan untuk akses layanan *web*. Dengan demikian, yang diperlukan pada mesin klien adalah perangkat lunak *browser*.

4.3.3 Tahapan Penelitian

Penelitian dilakukan dalam beberapa tahapan. Masing-masing tahapan akan dibahas pada bagian selanjutnya. Tahapan-tahapan yang dilakukan pada penelitian ini dapat diuraikan sebagai berikut.

- a) Menyiapkan mesin Router
- b) Menyiapkan mesin Load Balancer
- c) Menyiapkan dua mesin *server web*.
- d) Konfigurasi Mikrotik RouterOS pada mesin Router
- e) Instalasi Mageia 2014 pada mesin Load Balancer
- f) Instalasi Pound pada mesin Load Balancer
- g) Konfigurasi Pound mesin Load Balancer
- h) Instalasi Mageia 2014 pada dua mesin *server web*.
- i) Instalasi Apache, PHP dan aplikasi *web* pada dua mesin *server web*.
- j) Konfigurasi Mageia 2014 pada dua mesin *server web*.
- k) Konfigurasi Apache, PHP dan aplikasi *web* pada dua mesin *server web*.

- l) Menyusun program sekrip sinkronisasi dua mesin *server web*.
- m) Menyusun program sekrip untuk mencuplik data akses *web*
- n) Percobaan akses *web*
- o) Pencatatan akses *web* secara nyata selama enam bulan
- p) Membuat *resume* akses harian, mingguan dan bulanan
- q) Menampilkan grafik akses harian, mingguan dan bulanan
- r) Analisis terhadap hasil percobaan akses *web*.

BAB 5 IMPLEMENTASI DAN PEMBAHASAN

5.1 Implementasi

Konfigurasi jaringan yang diperlukan dalam penelitian ini melibatkan dua *server web*, satu *server load balancer*, satu *router* dan beberapa komputer yang berlaku sebagai klien. Peran serta fungsi masing-masing peralatan, *server* dan perangkat lunak yang dipasang pada *server* diuraikan pada paragraf-paragraf berikut.

Klien digunakan untuk menjalankan aplikasi *browser* yang digunakan untuk melakukan akses layanan *web* pada *server web*. Klien dapat menggunakan sistem operasi apa pun baik Linux, Windows atau sistem operasi yang lain, yang penting dapat menjalankan aplikasi *browser*. Aplikasi *browser* yang dipakai bisa berupa perangkat lunak Firefox, Chrome, Opera, atau aplikasi *browser* lain yang ada pada *smartphone*.

Server load balancer berlaku sebagai *pemerataan* beban akses layanan HTTP serta berlaku juga sebagai *server* yang akan meneruskan akses ke *server web*. Dalam hal ini *server load balancer* berlaku sebagai *server front-end* (penerima akses HTTP dari luar). Perangkat lunak yang dipasang pada *server load balancer* adalah Pound. Perangkat lunak Pound akan meneruskan akses

layanan HTTP kepada *server web* yang ada di belakangnya. Pada penelitian ini, Pound bekerja bekerja di atas sistem operasi Linux Mageia 2014.

Server web berlaku sebagai penerima beban akses layanan HTTP yang sebenarnya. Pada *server web* ditempatkan aplikasi *web* yang akan diuji, dalam hal ini situs layanan *web* www.akakom.ac.id. Pada *server web* terutama dipasang perangkat lunak *server web* Apache. Selain itu, diperlukan juga perangkat lunak pendukung seperti *server* MySQL untuk layanan penyimpanan data serta sekrip sisi *server* PHP.

5.1.1 Konfigurasi Server Balancer Pound

Server balancer digunakan untuk menerima akses layanan HTTP dan meneruskan akses permintaan kepada *server web* yang ada di belakangnya. Dalam hal ini *server balancer* berlaku sebagai *pemerata* beban terhadap layanan beban akses yang ditujukan kepada *server web*.

Pada *server balancer* dipasang perangkat lunak *pemerata* beban akses *web* Pound. Perangkat lunak Pound bertugas menyeimbangkan beban akses *web* dengan cara membagi beban akses layanan *web* kepada semua *server web* yang ada di belakang *server balancer*. Pada penelitian ini digunakan dua *server web* sebagai penerima beban akhir.

Konfigurasi *pemerata* beban Pound terutama dituliskan dalam *file* `/etc/pound/pound.cfg`. Isi konfigurasi Pound yang digunakan dalam penelitian ini adalah sebagai berikut.

```

User "pound"
Group "pound"
Control "/var/run/pound/ctl_socket"

# Main listening ports
ListenHTTP
    Address 172.17.13.14
    Port    80
    xHTTP   1
End

# Catch-all server(s)
Service
    BackEnd
        Address 172.17.13.12
        Port    80
    End
    BackEnd
        Address 172.17.13.13
        Port    80
    End
    Session
        Type    BASIC
        TTL     300
    End
End

```

Penjelasan berikut berkaitan dengan konfigurasi *pemerata* beban Pound yang digunakan dalam penelitian ini.

- a) User dan group yang menjalankan perangkat lunak Pound adalah pound. Dengan demikian dalam *server balancer* harus ada user pound dan group pound. Untuk menjaga keamanan *server*, *user* pound dibuat sedemikian, sehingga tidak dapat melakukan *login*. *User* pound hanya dapat dipakai untuk menjalankan perangkat lunak Pound.
- b) Alamat IP *server* Pound adalah 172.17.13.14. Antar muka tempat melekat alamat IP tersebut sebagai sebagai antarmuka yang mana perangkat lunak Pound bekerja. Pound bekerja pada *port* 80, sama seperti layanan HTTP.

Namun Pound sendiri tidak merespons permintaan layanan HTTP, namun hanya meneruskan permintaan layanan HTTP kepada *server web* lain yang ada di belakangnya.

- c) *Server web* yang menerima beban akses sesungguhnya ada dua yaitu *server web* dengan alamat IP 172.17.13.12 dan *server web* dengan alamat IP 172.17.13.13. Dua *server web* tersebut bekerja pada *port* 80 yang merupakan *port* standar protokol HTTP. Dua *server web* inilah yang sesungguhnya menerima akses permintaan layanan HTTP dan merespon permintaan tersebut. Jika ingin menambah jumlah *server web*, maka bagian ini bisa diperluas sesuai dengan jumlah *server web* yang ada di belakang *server balancer*.

Setelah semua konfigurasi `pound.cfg` ditetapkan, perangkat lunak Pound dapat dijalankan menggunakan perintah

```
service pound start
```

Apabila tidak ada kesalahan konfigurasi, seharusnya perangkat lunak Pound dapat berjalan secara benar dan bekerja pada *port* 80 seperti layanan HTTP.

Apabila dilakukan perubahan pada salah satu bagian konfigurasi `pound.cfg`, maka layanan Pound harus dijalankan ulang menggunakan perintah supaya pengaruh perubahan konfigurasi diterapkan.

```
service pound restart
```

5.1.2 Konfigurasi *Server Web*

Server disiapkan untuk melayani permintaan layanan HTTP yang akan diseimbangkan beban akses terhadapnya. Pada penelitian ini digunakan dua *server web* identik yang dipasang aplikasi *web* www.akakom.ac.id. Pada *server web*, terutama diperlukan perangkat lunak Apache dan PHP. Apache berlaku sebagai perangkat lunak yang melayani akses HTTP, sedangkan PHP digunakan sebagai skrip yang bekerja pada sisi server untuk menjalankan fungsi-fungsi aplikasi web baik tampilan maupun akses pada sistem *database* MySQL. Aplikasi memerlukan simpanan data, namun simpanan data tidak berada pada *server* yang sama dengan aplikasi *web*.

Dua *server web* dibuat sedemikian, sehingga selalu identik dalam hal isi aplikasi *web*. Oleh sebab itu dibuat agar kedua *server* selalu melakukan sinkronisasi dengan *server* lain. Sinkronisasi dilakukan terhadap perubahan aplikasi maupun perubahan data *upload*. Perubahan aplikasi bisa terjadi karena adanya *update* aplikasi oleh pengembang sistem, sedangkan perubahan data *upload* terjadi karena pengguna melakukan *upload file* baru (misalnya untuk *update* berita yang melibatkan *file* gambar).

Program untuk melakukan sinkronisasi aplikasi ditulis dalam bentuk skrip bash. Dalam bentuk versi modifikasi bagian *user* dan *password* untuk keamanan, skrip menjadi berbentuk sebagai berikut.

```
#!/bin/sh  
# design by: wagito
```



```
# $1: server
# $2: user
# $3: pass
# $4: direktory under /home/user/html

function synchronize()
{
    server=$1
    user=$2
    pass=$3
    direktory=$4
    home="/home/$user/html"
    #/bin/chmod o-rwx -R $home/$3
    #/bin/chown $2.apache -R $home/$3
    /bin/sshpass -p $pass rsync -a $home/$direktory $user@$server:$home
}

synchronize "www13.akakom" "userxxxx" "passxxxx" "www.akakom.ac.id"
```

Perangkat lunak pendukung untuk proses sinkronisasi aplikasi antara lain `chmod`, `sshpass` dan `rsync`. `Chmod` digunakan untuk mengubah permisi direktori tempat meletakkan aplikasi, `chown` digunakan untuk mengubah kepemilikan direktori aplikasi *web*, `sshpass` digunakan untuk masuk pada *server* lain secara tidak interaktif, sedangkan `rsync` digunakan untuk proses sinkronisasi direktori aplikasi *web*. Dalam proses sinkronisasi, yang disinkronkan hanya direktori tertentu yang diperlukan saja (terutama direktori tempat meletakkan aplikasi dan *file upload*). Hal demikian perlu dilakukan supaya proses sinkronisasi dua *server* bisa berjalan lebih cepat.

Perangkat lunak `sshpass` diperlukan secara khusus karena untuk sinkronisasi diperlukan login pada *server* secara terprogram. Dengan demikian *username* dan *password* tidak dimasukkan secara interaktif, tetapi harus dituliskan dalam sekrip program. Perangkat lunak `sshpass` menggantikan utilitas `ssh` standar pada utilitas Linux.

Perangkat lunak `rsync` secara khusus diperlukan sebagai utilitas utama

proses sinkronisasi. Perilaku program rsync sama seperti rcp, tetapi memiliki lebih banyak pilihan serta menggunakan protokol rsync *remote-update* yang sangat mempercepat transfer *file* saat *file* tujuan diperbarui. Protokol rsync *remote-update* memungkinkan perangkat lunak rsync untuk mentransfer hanya *file* tertentu yang berbeda antara dua *server web* menggunakan algoritme *checksum-search* yang efisien.

Sinkronisasi dilakukan secara berkala terhadap perubahan yang terjadi pada masing-masing *server*. Sinkronisasi dilakukan terhadap aplikasi maupun data. Sinkronisasi aplikasi dilakukan untuk perubahan aplikasi pada salah satu *server*. Perubahan aplikasi bisa terjadi apabila ada fitur aplikasi *web* yang perlu ditambah atau dikurangi. Sinkronisasi data dilakukan untuk memperbarui data *file upload* situs. Proses *upload file* terjadi karena pada situs ada fitur untuk menambah berita yang dilengkapi dengan *file* gambar yang harus diletakkan pada *server*.

Pada masing-masing *server web* dipasang sekrip untuk melakukan pencacahan jumlah akses yang masuk. Sekrip untuk pencacahan ini disisipkan pada aplikasi *web* situs, sedemikian sehingga setiap kali *server web* mendapatkan akses, sekrip akan dijalankan dan selanjutnya menyimpan informasi waktu dan jumlah akses. Sekrip disisipkan pada halaman muka situs sedemikian, sehingga setiap saat ada pengguna yang membuka halaman situs, sekrip ini akan otomatis dijalankan. Apabila akses oleh pengguna dilakukan beberapa kali dalam satu kunjungan, maka akses akan tercatat beberapa kali juga.

```

<?php
date_default_timezone_set('Asia/Jakarta');
$host="www.akakom.ac.id";
$dir="/var/www/log/";
$day=date('Y/m/d');

if (file_exists($dir.$host))
{
    $db=new SQLite3($dir.$host);
    $db->exec("update counter set count=count+1 where day='".$day."'");
    if ($db->changes()==0) $db->exec("insert into counter values ('".$day."', '1')");
}
else
{
    mkdir($dir);
    $db=new SQLite3($dir.$host);
    $db->exec("create table counter (day date unique, count int)");
    $db->exec("insert into counter values ('".$day."', '1')");
}
$db->close();
?>

```

Sekrip tersebut akan menyimpan data jumlah akses yang diterima *server web* setiap hari. Data disimpan dalam *database* SQLite3. Pencatatan berupa jumlah akses kumulatif tiap hari.

Pada awal sekrip disesuaikan terlebih dahulu zona waktu yang dipakai, supaya pencatatan sesuai dengan waktu setempat (zona waktu di mana *server web* diletakkan). Dalam hal ini, zona waktu *server* diatur di wilayah Asia/Jakarta. Meskipun pengguna mengakses *server web* dari zona waktu yang lain, catatan waktu pada *server* akan mengikuti zona waktu *server*.

Jumlah akses dijumlahkan selama satu hari dari jam 00.00 sampai 24.00. Pencatatan secara otomatis akan menuju pada baris berikutnya apabila terjadi perubahan hari. Apabila sudah ada catatan pada hari tertentu, maka sekrip akan menambahkan cacah hari tersebut. Apabila belum ada catatan waktu, maka sekrip akan mencatat sebagai hari baru.

5.1.3 Konfigurasi Router

Router digunakan sebagai penghubung dari jaringan lokal ke jaringan Internet. Dalam hal ini, *router* difungsikan sebagai *firewall* untuk melakukan translasi alamat jaringan. Fungsi tambahan yang dilakukan *router* adalah sebagai penyaring (filter) terhadap paket data dari jaringan Internet yang masuk pada jaringan lokal.

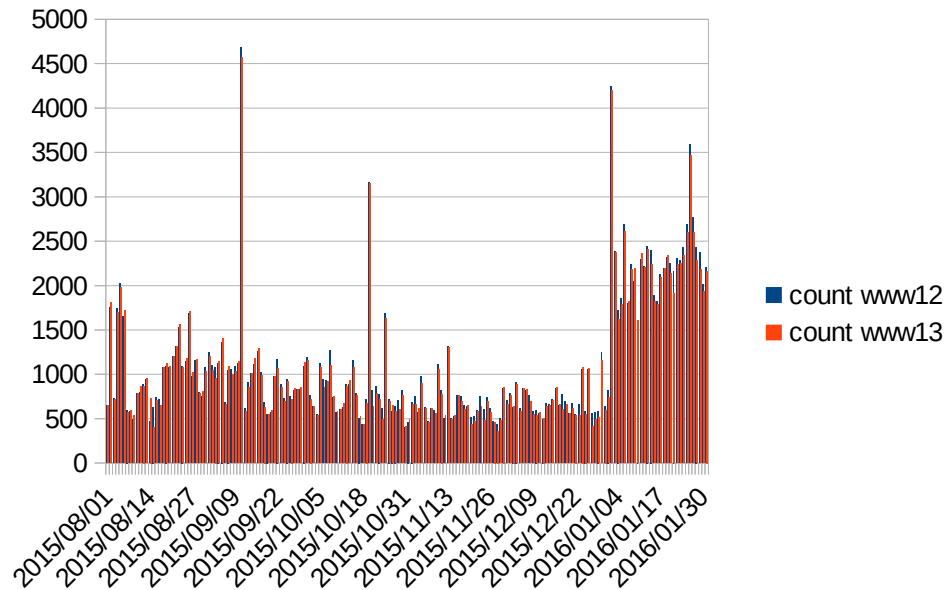
Router yang dipakai dalam sistem jaringan ini adalah Mikrotik. Dalam kaitan dengan penelitian ini, pengaturan dilakukan Router dilakukan pada menu. IP->Addresses untuk menentukan alamat IP publik situs dan menu IP-> Firewall. Bagian menu Firewall paling utama untuk diatur adalah Firewall->NAT yang berkaitan dengan translasi alamat jaringan.

5.2 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 5.1.



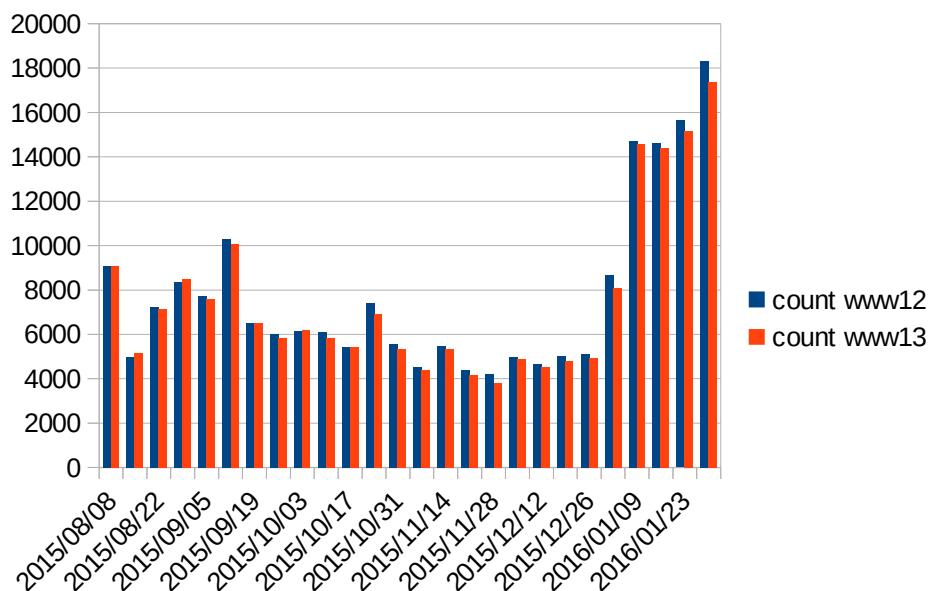
Gambar 5.1 Perbedaan Akses Dua Server Web Secara Harian

Grafiknya kurang bisa membedakan hasil beda cacah beban akses antara dua *server web*. Hal ini 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 ini.

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

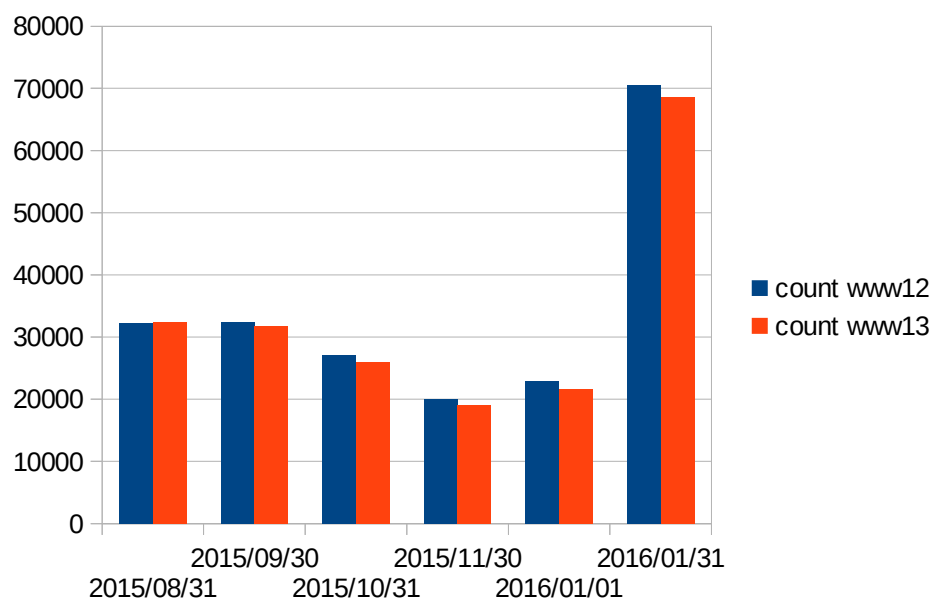


Gambar 5.2 Perbedaan Akses Dua Server Web Secara Mingguan

Gambar 5.2 menunjukkan perbedaan jumlah beban akses pengguna pada

dua *server* yang dijadikan kumulatif secara mingguan.

Secara umum dapat dilihat, tidak ada perbedaan yang signifikan beban akses yang diterima oleh *server web* pertama dan kedua. 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 5.3 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 5.3.

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. Hal ini 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. Hasil *load balancing* akan lebih baik (nilai beda beban akses lebih kecil) apabila tidak ada gangguan pada salah satu *server*.

Penelitian ini bisa juga dikembangkan untuk membandingkan tingkat 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 pengamatan tertentu. Selanjutnya dapat dianalisis penyebab terjadinya gangguan sedemikian, sehingga dapat dilakukan metode pencegahan terhadap gangguan yang terjadi pada *server*. Penyebab terjadinya gangguan mungkin dapat dianalisis dari hasil catatan *log* layanan yang tersedia pada *server*

Pada penelitian ini, dua *server* dapat dikatakan 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 kapasitas prosesor, RAM atau perangkat keras lain. Perbedaan perangkat lunak dapat dilakukan dengan membedakan perangkat lunak *server web* yang digunakan. Ada beberapa pilihan perangkat lunak *server web* yang bisa digunakan. Misalnya *server* pertama menggunakan Apache dan *server* kedua menggunakan Nginx sebagai perangkat lunak *server web*. Apakah terjadi perbedaan jumlah beban akses apabila ada perbedaan perangkat lunak yang digunakan untuk menjalankan aplikasi *web*.

Penelitian juga dapat dikembangkan untuk menambah jumlah *server web* yang dipakai. Pada penelitian ini digunakan dua *server web* yang bisa ditambah jumlahnya. Jika jumlah *server web* ditambah, maka perubahan pengaturan dilakukan pada *server balancing* untuk mengatur konfigurasi Pound. Selain itu juga perlu diatur ulang mekanisme sinkronisasi antar *server web* yang dijadikan *server back-end*. Mekanisme sinkronisasi menjadi lebih kompleks dan mungkin perlu metode tertentu.

BAB 6 KESIMPULAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil pembahasan dan percobaan dalam penelitian ini adalah sebagai berikut.

- Teknologi perangkat lunak Pound dapat diterapkan untuk pemerataan beban akses *server web*. Pada penelitian ini akses *server web* digunakan protokol standar HTTP.
- Metode untuk mengamati beban akses *server web* berdasarkan catatan beban akses yang dihasilkan *server web* disusun dengan kombinasi beberapa utilitas Linux terutama *rsync* dan *sshpas*.
- Beban akses *server web* dapat terbagi dengan perbedaan jumlah beban akses harian sebesar 5,5%. Beban akses mingguan dan beban akses bulanan menghasilkan angka perbedaan sebesar 3,18 %
- Ada sedikit perbedaan hasil persentase jumlah beban akses harian, mingguan dan bulanan yang disebabkan karena adanya gangguan pada salah satu *server* pada tanggal tertentu.

6.1 Saran

Saran yang diajukan untuk pengembangan dan penelitian lebih lanjut dari

penelitian ini adalah sebagai berikut.

- Penelitian dapat dikembangkan untuk menguji kestabilan *server* dibandingkan *server* yang lain ketika menerima beban akses.
- Penelitian bisa dikembangkan untuk meneliti perbedaan hasil *load balancing* beban akses apabila perangkat lunak atau perangkat keras *server back-end* yang berbeda.
- Penelitian dapat dikembangkan untuk mengamati jumlah *server back-end* yang lebih banyak dengan mengubah konfigurasi *server balancer* dan mekanisme sinkronisasi antar *server web*.

Daftar Pustaka

- Alimuddin dan kawan-kawan, 2016, *Peningkatan Kinerja Siakad Menggunakan Metode Load Balancing dan Fault Tolerance Di Jaringan Kampus Universitas Halu Oleo*, jurnal IJCCS, UGM
- Apsis GmbH, 2017, *Pound - Reverse-Proxy And Load-Balancer*,
<http://www.apsis.ch/pound/>
- Apsis, IT Security, 2016, *Pound - Reverse-Proxy and Load-Balancer*,
<http://www.apsis.ch/pound/>
- Ari Budi Noviyanto dan kawan-kawan, 2015, *Perancangan dan implementasi load balancing reverse proxy menggunakan HAProxy pada aplikasi web*, jurnal JARKOM, AKPRIND, Yogyakarta
- Array Networks, 2017, *Server Load Balancing for Application High-Availability*,
<https://www.arraynetworks.com/solutions-server-load-balancing.html>
- 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
- Nginx, 2017, *What Is Load Balancing?*,
<https://www.nginx.com/resources/glossary/load-balancing/>
- Segall, R., 2010, *Pound - HTTP/HTTPS Reverse-Proxy and Load-Balancer*,
<https://linux.die.net/man/8/pound>