

ANALISIS PERBANDINGAN UNJUK KERJA SISTEM PENYEIMBANG BEBAN WEB SERVER DENGAN HAPROXY DAN POUND LINKS

Dite Ardhian¹⁾, Adian Fatchur Rochim²⁾, Eko Didik Widiyanto²⁾

Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro,
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia
email : diteardhian@yahoo.co.id

ABSTRACT

The development of internet technology has many organizations that expanded service website. Initially used single web server that is accessible to everyone through the Internet, but when the number of users that access the web server is very much the traffic load to the web server and the web server anyway. It is necessary for the optimization of web servers to cope with the overload received by the web server when traffic is high.

Methodology of this final project research include the study of literature, system design, and testing of the system. Methods from the literature reference books related as well as from several sources the internet. The design of this thesis uses Haproxy and Pound Links as a load balancing web server. The end of this reaserch is testing the network system, where the system will be tested this stage so as to create a web server system that is reliable and safe.

The result is a web server system that can be accessed by many user simultaneously rapidly as load balancing Haproxy and Pound Links system which is set up as front-end web server performance so as to create a web server that has performance and high availability

Keywords : Performance, Web Server, System, Load Balancing

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi *web* harus didukung pula dengan infrastruktur yang baik dan berkecepatan tinggi agar berbagai kebutuhan tersebut dapat terpenuhi dengan baik. Peningkatan permintaan pada situs, menyebabkan *web server* sibuk menjawab permintaan klien dan terkadang pula *web server* mengalami *server* mati atau kegagalan *server* jika terlalu banyak permintaan sehingga *web server* tidak dapat menanganinya. Hal ini akan merugikan pihak yang mempercayakan situsnya pada suatu *web server*, karena situs-situs tersebut tidak dapat diakses untuk sementara waktu. Parameter seperti waktu tanggap, pemerataan beban dan keberhasilan pelayanan merupakan parameter yang menentukan kinerja suatu *web server*.

Salah satu mekanisme untuk lebih mengoptimalkan penggunaan sumber daya yang ada adalah dengan mekanisme *load balancing* yang akan menyeimbangkan beban dari seluruh *server* yang ada pada *cluster*. Melalui penggunaan mekanisme penyeimbang beban, permintaan koneksi HTTP yang datang pada *web server* akan ditangani oleh server dengan kondisi yang paling baik, dimana pemilihan tersebut dilakukan dengan sistem prioritas *web server*.

Ada beberapa metode penyeimbang beban *web server*. Salah satu metode penyeimbang beban *web* dapat menggunakan *reverse proxy*, yaitu

proxy yang memiliki maksud yang berkebalikan dengan *forward proxy* yaitu menjadi perantara *user-user* di internet terhadap akses ke *web-web server* yang berada pada LAN, sehingga seolah-olah *user* di internet mengakses langsung *web server* yang dimaksud padahal sesungguhnya *user* di internet mengakses *web-web server* yang terdapat di LAN melalui *reverse proxy* tersebut. Tentunya komputer yang bertindak sebagai *reverse proxy* tersebut memiliki IP publik sehingga dapat diakses dari Internet. Salah satu contoh perangkat lunak *opensource* yang menggunakan metode *reverse proxy* adalah Haproxy dan Pound Links, kedua perangkat lunak tersebut dapat digunakan sebagai penyeimbang beban *web server* dan *caching web server* sehingga akses ke *web* menjadi cepat dan beban *web server* menjadi seimbang.

1.2. Tujuan

Menguji dan menganalisis penggunaan program *opensource* Haproxy dan Pound Links, mekanisme kerja sistem penyeimbang beban *web server*, dan perbandingan unjuk kerja sistem penyeimbang beban *web server* dengan Haproxy dan Pound Links.

1.3. Pembatasan Masalah

Penelitian ini dibatasi pada topik sebagai berikut.

1) Mahasiswa Sistem Komputer UNDIP

2) Dosen Sistem Komputer UNDIP

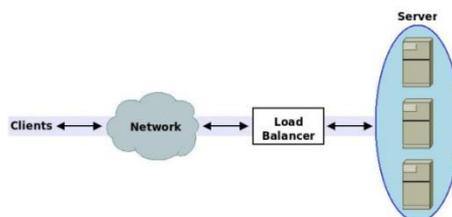
1. Mesin *Server* Haproxy dan Pound Links menggunakan sistem operasi distro Ubuntu server 12.04.
2. Menggunakan replikasi basis data MySQL untuk *update* basis data *web server*.
3. Menggunakan metode sinkronisasi Unison otomatis untuk *update* dokumen WWW *web server*.
4. Menggunakan *web server* Apache2.
5. Sistem penyeimbang beban ini dijalankan di *server* fisik bukan virtual.

2. DASAR TEORI

2.1 Sistem Penyeimbang Beban (*Load balancing*)

Penyeimbang beban (*Load Balancing*) adalah suatu proses dan teknologi yang mendistribusikan trafik situs di antara beberapa *server* dengan menggunakan perangkat berbasis jaringan. Sesuai kriteria dasar penyeimbang beban proses ini mampu mengurangi beban kerja setiap *server* sehingga tidak ada *server* yang kelebihan beban, memungkinkan *server* untuk menggunakan *bandwidth* yang tersedia secara lebih efektif, dan menyediakan akses yang cepat ke situs-situs yang di-*hosting*. Penyeimbang beban dapat diimplementasikan dengan menggunakan perangkat keras, perangkat lunak, atau gabungan keduanya.

Penyeimbang beban merupakan proses fleksibel yang dapat diciptakan dengan berbagai cara dan metode. Proses ini tidak dapat dilakukan oleh sebuah perangkat tertentu atau sebuah perangkat lunak khusus saja. Cukup banyak cara dan pilihan untuk mendapatkan jaringan yang dilengkapi dengan sistem penyeimbang beban. Cara kerja dan prosesnya berdeda-beda satu sama lainnya. Pada dasarnya penyeimbang beban adalah suatu teknik yang digunakan untuk memisahkan antara dua atau banyak *link* jaringan. Dengan mempunyai banyak *link* maka optimalisasi utilisasi sumber daya, *throughput* atau waktu respon akan semakin baik karena mempunyai lebih dari satu *link* yang bisa saling mem-*backup* pada saat jaringan mati dan menjadi cepat pada saat jaringan normal jika memerlukan realibilitas tinggi yang memerlukan 100% koneksi *uptime* dan yang menginginkan koneksi *upstream* yang berbeda dan dibuat saling mem-*backup*.^[6]



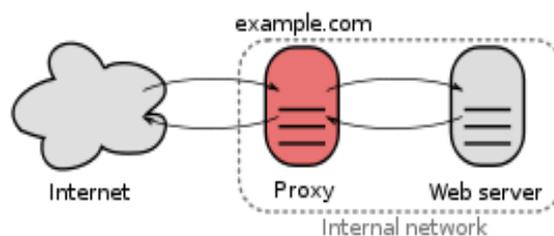
Gambar 1 Skema Sistem penyeimbang beban^[6]

2.2 Reverse proxy

Pengelompokan sistem penyeimbang beban terdapat beberapa jenis terdapat *proxy* yang merupakan bagian dari sistem penyeimbang beban yang bisa bertindak sebagai pembagi beban, *filtering*, dan *caching*. *Proxy* adalah sebuah sistem komputer atau program aplikasi yang melayani permintaan dari klien dengan meminta layanan ke *server* lain. *Proxy server* memiliki 3 fungsi utama yaitu :

1. *Connection sharing* : perantara klien dan *server*.
2. *Filtering* : bekerja pada layer aplikasi yang dapat mem-block paket-paket tertentu.
3. *Caching* : mampu menyimpan informasi yang pernah di akses dari server-server.

Proxy di bagi menjadi 2 yaitu *forward proxy* dan *reverse proxy*. *Proxy forward* adalah *proxy* yang meneruskan data ke *host* tujuan. *Reverse proxy* adalah *proxy* yang menampung permintaan klien dari Internet dan meneruskan ke *server-server* yang berada satu LAN (*local area network*) dengan *proxy* tersebut. Aplikasi yang berfungsi sebagai *reverse proxy* contohnya adalah Haproxy dan Pound Links.



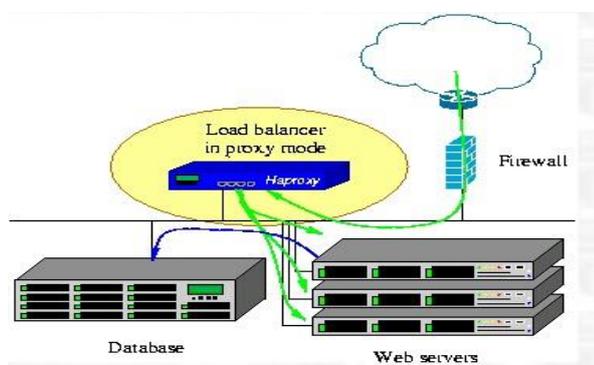
Gambar 2 Skema Reverse Proxy^[7]

Server proxy adalah *server* yang bertindak sebagai perantara untuk melayani permintaan dari klien yang mencari sumber daya dari *server* lain di dalam jaringan komputer. Seorang klien terhubung ke *server proxy* dan meminta beberapa layanan, seperti sambungan *file*, halaman *web*, atau sumber yang lain yang tersedia dari *server* yang berbeda.^[7]

2.2.1 Haproxy

Haproxy adalah produk *opensource* yg mendukung keperluan penyeimbang beban dan *failover web server*, banyak digunakan untuk keperluan *reverse proxy* di *site-site* yang trafik hariannya tinggi. Pada Haproxy memiliki beberapa parameter di dalamnya yaitu sebagai berikut.^[8]

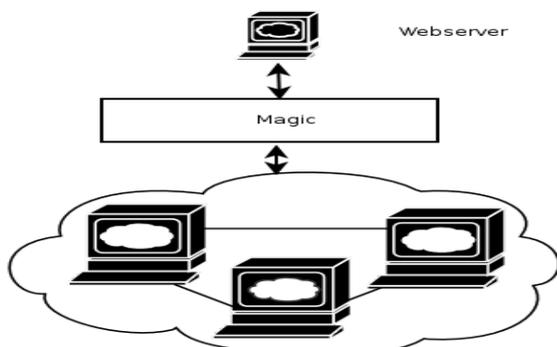
1. Global parameter : berisi parameter manajemen proses dan keamanan, *tuning* kinerja, *debugging*, dan *userlist*
2. *Proxy*
3. Konfigurasi *Server*
4. Manipulasi HTTP
5. *Accesslist*
6. *Logging*
7. Statistik dan monitoring



Gambar 3 Skema Haproxy^[10]

2.2.2 Pound Links

Pound merupakan program *reverse proxy*, penyeimbang beban (*load balancer*), dan *http(s) front end* pada *web server*. Pound dikembangkan untuk memungkinkan mendistribusikan beban antara *web server* dan dapat juga meng-enkripsi dan dekripsi paket data yang akan dilewatkan yaitu SSL. Pound merupakan program *opensource*.^[11]



Gambar 4 Skema Pound Links^[11]

Beberapa fungsi yang dari Pound Links.

1. *Reverse proxy*
2. *Load balancer* (pembagi beban *server*)

3. SSL (enkripsi/dekripsi paket data)
4. Menggunakan protokol HTTP
5. *Fail over system* saat terjadi kegagalan *server*
6. Membagi beban berdasarkan URL

Web Server

Server web dapat merujuk baik pada perangkat keras ataupun perangkat lunak. *Server web* menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS. Penggunaan paling umum *server web* adalah untuk menempatkan situs *web*. Pada prakteknya penggunaannya diperluas sebagai tempat penyimpanan data ataupun untuk menjalankan sejumlah aplikasi kelas bisnis. Fungsi utama sebuah *server web* untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan dan mentransfer seluruh aspek pemberkasan dalam sebuah halaman *web* yang terkait; termasuk di dalamnya teks, gambar, video, atau lainnya. Kriteria dasar *web server* berjalan dengan baik adalah dengan terjalannya komunikasi misalnya protokol HTTP/HTTPS dari *server* ke klien atau sebaliknya tanpa ada data yang hilang.

2.3 Sinkronisasi Otomatis Menggunakan Unison

Unison adalah *tool* untuk *transfer* dan sinkronisasi *file* atau *tree* (struktur direktori dan file) secara dua arah, baik *transfer* lokal (di sistem yang sama) maupun *remote* (jaringan/internet). Unison merupakan *tools* kecil yang powerful karena bisa bertindak sebagai *server* sekaligus sebagai klien.^[15]

2.4 Replikasi MySQL

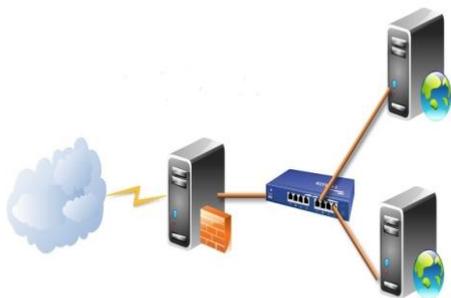
MySQL juga mendukung fungsi replikasi. Replikasi di MySQL merupakan proses dinamis dari sinkronisasi data antara *server (master)* basis data utama dengan satu atau lebih sekunder (*slave server* basis data secara *real-time*). Dengan replikasi, data masih dapat dipulihkan dari salah satu *server* sekunder jika *server* utama gagal, karena adanya salinan dari satu atau lebih basis data tadi.^[12]

3. PERANCANGAN SISTEM

Perancangan sistem ini dibutuhkan tiga *server* yaitu 2 *server* untuk *Web server* dan 1 *server* sebagai *front-end* dari *web server* yang fungsinya sebagai penyeimbang beban *web server*. *Web server* dan *server* penyeimbang

beban berada pada jaringan lokal, alamat IP kedua *web server* adalah IP lokal untuk mengamankan *web server* karena IP nya tidak dapat diakses langsung melalui Internet, sedangkan *server penyeimbang beban* memiliki 2 antar muka yang terhubung dengan kedua *web server* dengan IP lokal dan antar muka satu lagi terhubung dengan internet/ jaringan publik yang dapat diakses dari luar jaringan lokal.

Cara *server penyeimbang beban* memilih *web server* mana yang dipilih untuk melayani permintaan *user/klien* ditentukan oleh konfigurasi aplikasi pada *server penyeimbang beban*. Ada beberapa algoritma dan parameter yang dapat menentukan pembagian beban/pemilihan *web server* tersebut. Skema *physical* topologi dapat dilihat pada gambar 5.

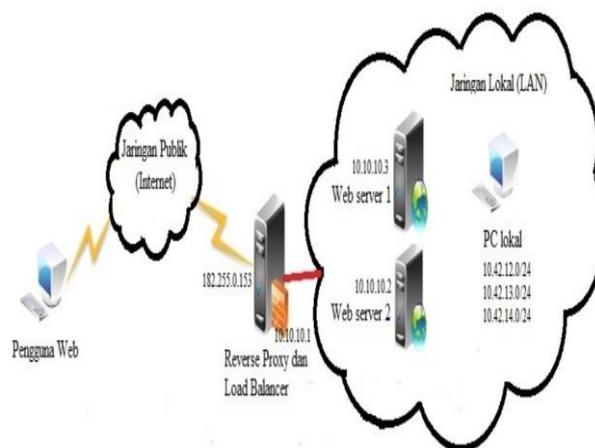


Gambar 5 Skema *Physical* Topologi

Web server menggunakan mekanisme replikasi basis data MySQL yaitu penduplikasian isi basis data yang berada pada *web server* 1 dan *web server* 2, hal ini bertujuan agar database pada *web server* 1 dapat meng-*update* data ke *web server* 2 dan sebaliknya sehingga tidak terdapat informasi yang salah atau berbeda pada kedua *database* di kedua *web server* tersebut sehingga memudahkan dalam pengentrian data.

Web server juga menggunakan mekanisme mensinkronkan data-data *web server* 1 dan *web server* 2 sehingga data akan sama di kedua *web server* jika ada *update* dari *web server* 1 dan sebaliknya.

Jaringan terbagi menjadi dua bagian yaitu jaringan *internal* dan *eksternal*. *Web server* keduanya di beri IP lokal dan penyeimbang beban menggunakan dua buah IP di dua antar muka LAN dan antar muka Internet. Skema *logical* topologi dapat dilihat pada gambar 6.



Gambar 6 Skema *Logical* Topologi

IV. IMPLEMENTASI DAN PENGUJIAN

Implementasi dilakukan dengan membuat layanan-layanan seperti pada perancangan. Dimulai dengan penanaman sistem operasi pada masing-masing *server* sampai dengan implementasi aplikasi-aplikasi yang diperlukan oleh masing-masing *server*. Setelah semua paket telah terpasang, maka hal terpenting yang harus dilakukan adalah melakukan konfigurasi agar semua proses dapat berjalan dengan kinerja yang baik.

4.1 Implementasi dan Konfigurasi Haproxy dan Pound Links

Implementasi dilakukan pada *server front-end* sebagai penyeimbang beban yang akan membagi beban dan meneruskan trafik ke *web server* di *back-end*. Parameter pengaturan penyeimbang beban perlu disesuaikan sesuai dengan kebutuhan seperti menggunakan metode penyeimbang beban round robin, pemberian prioritas pada *server back-end*, dan lain-lain

4.2 Implementasi dan Konfigurasi Web Server

Web server perlu diinstall Apache *web server*, PHP5, php5-dev, dan MySQL *server*. Apache *web server* bertugas menerima dan membalas permintaan situs yang datang dari klien di *web server*. PHP5 berfungsi mendukung bahasa pemrograman PHP pada *web server*. Php5-dev digunakan sebagai syarat implementasi PHP *Accelerator* dengan perangkat lunak eAccelerator. MySQL *server* sebagai basis data berfungsi sebagai syarat mengimplementasi Joomla dan Wordpress.

4.3 Implementasi dan Konfigurasi Unison

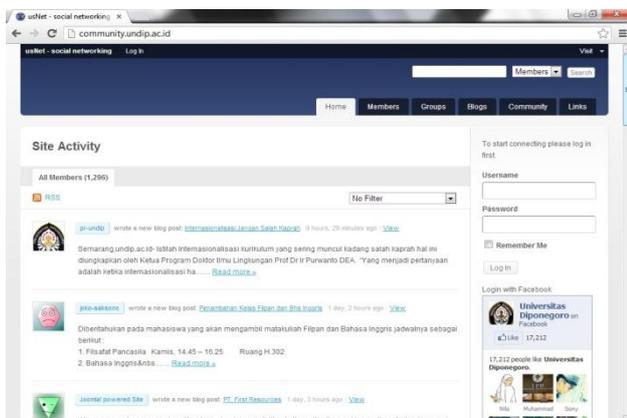
Unison diterapkan pada kedua *web server*. Unison yang digunakan adalah Unison otomatis dimana *web server* yang datanya tidak *update* maka data tersebut di *replace* dengan data yang *update*.

4.4 Implementasi dan Konfigurasi Replikasi MySQL

Replikasi basis data MySQL yaitu penduplikasian isi database yang berada pada *web server* 1 dan *web server* 2, hal ini bertujuan agar basis data pada *web server* 1 dapat meng-*update* data ke *web server* 2 dan sebaliknya sehingga tidak terdapat informasi yang salah atau berbeda pada kedua basis data di kedua *web server* tersebut sehingga memudahkan dalam pengentrian data.

4.5 Hasil pengujian dari Sistem Load Balancing

Pengujian Haproxy, Pound Links, LAMP, dan Unison dilakukan dengan *server* Community UNDIP hostname *community.undip.ac.id* dengan tampilan halaman web pada gambar 7.



Gambar 7 Tampilan *website* community UNDIP

Hasil pengujian di atas, penggunaan lebar jalur data tidak terlalu besar dari lebar jalur data maksimal, ini disebabkan penggunaan Unison untuk sinkronisasi yang akan melakukan *update* jika ada *event update* sehingga tidak menghabiskan lebar jalur data antara *web server*.

Permintaan dari pengguna cukup besar yang diakumulasikan pada besarnya penggunaan lebar jalur data namun, kedua *web server* tersebut masih berjalan stabil dan dapat diakses pengguna dengan lancar.

Penggunaan penyeimbang beban Haproxy sedikit lebih optimal dibandingkan menggunakan

Pound Links dilihat dari *throughput* dan waktu respon.

Tabel 1 Hasil pengujian Httperf

Penyeimbang beban	Rata Koneksi	Rata permintaan	Waktu Respon (ms)	Throughput
Haproxy	1825.5 conn/s	1769.1 req/s	134.2	830.9 KB/s
Pound Links	1497.7 conn/s	1483.4 req/s	234.6	675.2 KB/s
IPTables	173.4 conn/s	100.1 req/s	2600.8	36.1 KB/s

Tabel 2 Hasil pengujian Webbench

Penyeimbang beban	Pages/min	Throughput
Haproxy	193972	654350 bytes/sec
Pound Links	117218	542725 bytes/sec
IPTables	16640	400563 butes/sec

Pada tabel 1 dan tabel 2 menunjukkan Penyeimbang beban Haproxy sedikit lebih unggul dari Pound Links maupun Penyeimbang beban menggunakan IPTables yang telah di teliti oleh peneliti sebelumnya. Sehingga Haproxy dapat digunakan sebagai pilihan pertama penggunaan Penyeimbang beban *web server*.

5. PENUTUP

5.1. Kesimpulan

1. Kinerja sistem penyeimbang beban Haproxy maupun Pound Links pada Ubuntu *server* 12.04 dapat membagi beban CPU, memori dan trafik.
2. Sistem penyeimbang beban ini menggunakan *failover* mode saat salah satu *server* mengalami kegagalan *server* maka layanan *web* tetap dapat diakses klien.
3. Data yang dikirimkan oleh *server* ke klien baik pada sistem penyeimbang beban Haproxy maupun Pound Links tidak rusak maupun hilang.
4. *Web server* menjadi aman karena IP nya menggunakan IP lokal yang tidak dapat di akses maupun di *traceroute* secara langsung dari Internet.

5. Replikasi dua arah basis data MySQL berfungsi untuk *update* basis data *server* secara *realtime*.
6. Sinkronisasi menggunakan Unison memiliki keunggulan yaitu hemat *bandwidth*.
7. Sistem penyeimbang beban menggunakan Haproxy memiliki keunggulan dibanding Pound Links dari parameter waktu respon 70% lebih cepat, *throughput* 20% lebih besar dan akses halaman per menit 60% lebih banyak.

5.2 Saran

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah sebagai berikut.

1. Sistem penyeimbang beban yang telah digunakan dapat diterapkan pula untuk membuat sistem penyeimbang beban *web server* UNDIP khususnya saat diperlukan layanan *web server* dengan jumlah akses ke *web server* sangat tinggi agar layanan dapat teratasi dengan baik.
2. Sistem penyeimbang beban ini dapat dikembangkan lagi dengan jumlah *web server* yang lebih banyak.
3. Sistem penyeimbang beban ini dapat dikembangkan lagi tidak hanya di implementasikan pada *dedicated server* tetapi bisa juga pada *virtual server*.

DAFTAR PUSTAKA

- [1] Abdullah, A, SNMP Simamora, dan HR Andrian. *Implementasi dan Analisa Load Balancing pada suatu Web Server Lokal*. Politeknik Telkom, 2010.
- [2] Bourke, T. *Server Load Balancing*. O'Reilly, 2001
- [3] Lukitasari, Desy dan AF Oklilas. *Analisis Perbandingan Load Balancing Web Server Tunggal Dengan Web Server Cluster Menggunakan Linux Virtual Server*. Universitas Sriwijaya, 2010.
- [4] Nurhatta, Andy Purnama. *Sistem Penyeimbang Beban Web Server Dengan IPTABLES*. Tugas Akhir S-1, Universitas Diponegoro, Semarang, 2012.
- [5] Rijayana, Iwan. *Teknologi Load Balancing Untuk Mengatasi Beban Server*, SNATI, Yogyakarta, 2005.
- [6] Sagala, Muhfi Asbin. *Implementasi Load Balancing Pada Web Server*. Tugas Akhir S-1, Universitas Sumatera Utara, Medan, 2010.
- [7] Stricek, Art. *A Reverse Proxy Is A Proxy By Any Other Name*. Paper Sans Institute, 2002.
- [8] --, Haproxy Documentation, <http://code.google.com/p/haproxy-docs> 25 Juni 2012, 20.00 WIB
- [9] --, HTTP, http://id.wikipedia.org/wiki/Protokol_Transfer_Hiperteks 12 Juni 2012, 21.00 WIB
- [10] --, Httpperf, <http://www.hpl.hp.com/research/linux/httpperf/httpperf-man-0.9.txt> Agustus 2012, 22.30 WIB
- [11] --, Pound, <http://www.apsis.ch/pound/> 25 Juni 2012, 23.00 WIB
- [12] --, Replication Reference Manual 2012, http://google.com/MySQL_5.0_Replication_Reference_Manual 11 Agustus 2012, 09.00 WIB
- [13] --, Webbench http://down.chinaz.com/server/201202/1671_1.htm 2 Februari 2013, 12.00 WIB
- [14] --, Wikipedia Proxy Server, http://en.wikipedia.org/wiki/Proxy_server 22 Agustus 2012, 21.00 WIB
- [15] --, Wikipedia Unison, [http://id.wikipedia.org/wiki/Unison_\(file_sync_hronizer\)](http://id.wikipedia.org/wiki/Unison_(file_sync_hronizer)) 4 Januari 2013, 13.40 WIB

BIODATA



Dite Ardhian, lahir di Semarang 9 Juni 1990. Menempuh pendidikan dasar di SD N Siliwangi Semarang. Melanjutkan ke SMP N 1 Semarang, Dan Pendidikan tingkat atas di SMA N 1 Semarang, lulus tahun 2008. Dari tahun 2008 sampai saat ini tengah menyelesaikan pendidikan Strata Satu di Program Studi Teknik Sistem Komputer, Universitas Diponegoro, Semarang, Indonesia Angkatan Tahun 2008.