

BAB IV

PERANCANGAN DAN IMPLEMENTASI

Dalam bab perancangan ini akan dijelaskan secara rinci rancangan sistem *load balancing* pada *web server* dengan menggunakan algoritma *least connection*. Beberapa hal yang dibahas meliputi analisis kebutuhan, desain dan rancangan sistem.

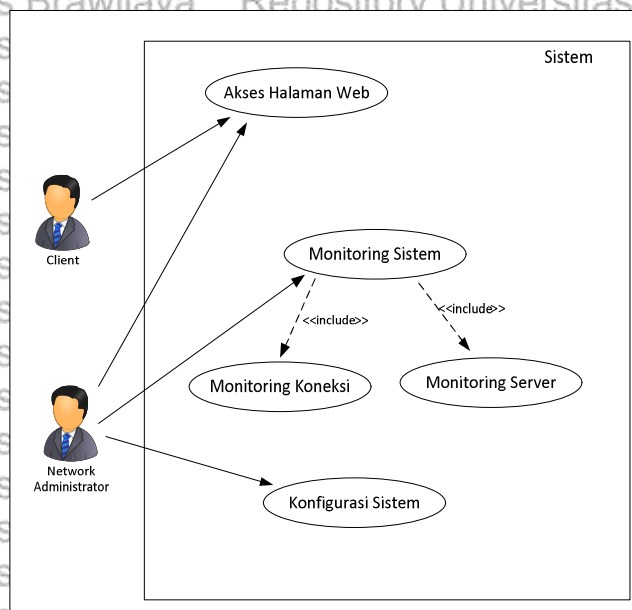
4.1. Analisis Kebutuhan

Pada analisis kebutuhan dibahas komponen-komponen yang digunakan untuk membangun sistem dan mendeskripsikan fungsi-fungsi dari sistem yang dibuat. Komponen-komponen yang dibahas dalam analisis kebutuhan sistem LVS meliputi, *software* dan *hardware* yang digunakan.

Analisis kebutuhan dibedakan menjadi dua yaitu, *user requirements* dan *system requirements* [12].

4.1.1 User Requirement

User dalam sistem *load balancing* pada *web server* adalah *client* atau pengakses *website* dan *network administrator* sistem. Pada Gambar 4.1, merupakan diagram *use case* yang menggambarkan *user* sistem dan hak akses *user*.



Gambar 4.1 : Diagram Use Case User

Sumber : Perancangan



Layanan yang diperoleh *client* adalah dapat mengakses *website* pada *web server*.

Network administrator mendapatkan beberapa layanan sebagai berikut :

1. Memperoleh layanan *monitoring* sistem untuk *monitoring real server* dan koneksi ke *real server*. *Monitoring* pada sistem berguna untuk mengetahui jumlah *real server* yang aktif serta mengetahui jumlah koneksi pada setiap *real server* dalam waktu tertentu.
2. *Network administrator* dapat melakukan beberapa konfigurasi sistem seperti, dapat melakukan konfigurasi IP pada *load balancer*, menentukan *real server* yang terhubung dengan direktor dan menentukan mekanisme dan algoritma penjadwalan *load balancing* yang digunakan.

4.1.2 System Requirements

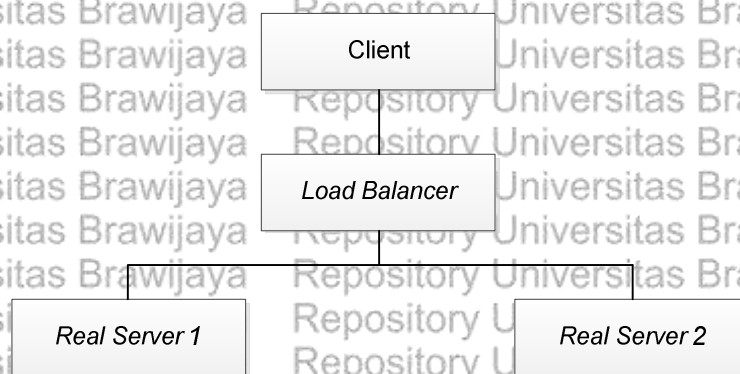
Dalam *system requirements* dibahas komponen-komponen yang diimplementasikan pada sistem *load balancing* pada *web server* serta dijelaskan dengan rinci fungsi dan layanan dari sistem *load balancing* pada *web server*.

4.1.2.1 Sistem Linux Virtual Server (LVS)

Linux *virtual server* atau disingkat LVS merupakan suatu teknologi *clustering* yang dapat digunakan untuk membangun suatu *server* dengan menggunakan kumpulan dari beberapa buah *real server*. *Load balancing* dengan algoritma *least connection* dapat diterapkan pada sistem Linux *virtual server*. Sistem *load balancing* yang dibuat memiliki karakteristik dan fungsi sebagai berikut [2]:

1. Fungsi sistem *load balancing* pada *web server* dengan algoritma *round robin* sehingga pembagian beban ke masing-masing *web server* dapat seimbang.
2. *Website* dapat diakses pengguna melalui IP *load balancer*.
3. Sistem *load balancing* dibuat pada sistem jaringan yang menggunakan IPV4.
4. *Reliability*, sistem *load balancing* pada *web server* akan bekerja lebih baik daripada sebuah *web server* ditinjau dari parameter-parameter seperti, *throughput*, *request loss*, dan waktu respon.

Pada Gambar 4.1, merupakan beberapa *node* dalam sistem LVS. Setiap *node* memiliki fungsi yang berbeda yang akan dijelaskan sebagai berikut :



Gambar 4.2 : Diagram Topologi LVS

Sumber : linuxvirtualserver.org

1. *Client*

Dalam sebuah sistem *web server client* adalah *user* yang mengakses website dalam *web server* tersebut. *Client* dapat berasal dari jaringan dan lokasi yang berbeda.

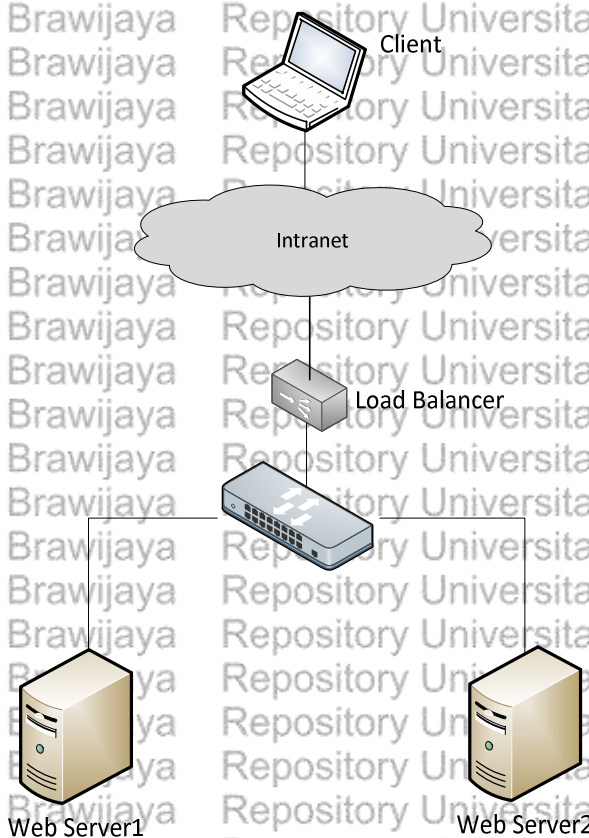
Hal ini menyebabkan *client* sebuah *service web server* sangat banyak. Untuk itu dibutuhkan sistem pembagian beban dalam *cluster* komputer yang dapat melayani *request* yang sangat banyak dari *client*

2. *Load balancer / director.*

Dalam Linux *virtual server director* disebut *virtual Server*. *Client* yang melakukan *request website* akan mengakses IP dari *director*. *Director* dalam LVS akan melanjutkan *request* dari *client* ke *real Server* yang memiliki data *website* yang diminta oleh *client*. *Director* tidak memiliki data *website* dan berfungsi sama seperti *router* yaitu meneruskan paket data ke alamat yang di tuju. Oleh karena itu *director* disebut dengan *virtual server* [2].

3. *Real server / web server*

Real server adalah *server* yang mempunyai data *website* yang sebenarnya. *Real server* terdiri lebih dari satu perangkat *server*. Beban *request* dari *client* akan dibagikan kepada seluruh *real Server* yang aktif. Jumlah *real server* yang lebih dari satu menyebabkan sistem lebih *high availability* karena apabila ada satu *real server* yang mati *request* dari *client* tetap dapat dilayani. Pembagian beban kepada *real server* ditentukan dengan penggunaan algoritma antrian [2].



Gambar 4.3 : Topologi load balancing

Sumber : Kopparapu (2001:21)

Pada sistem *load balancing* dengan LVS, *request* dari *client* akan diteruskan ke *real server*. Tabel 4.1 berikut merupakan tipe *forwarding* pada teknik *load balancing*.

Tabel 4.1 Tipe forwarding

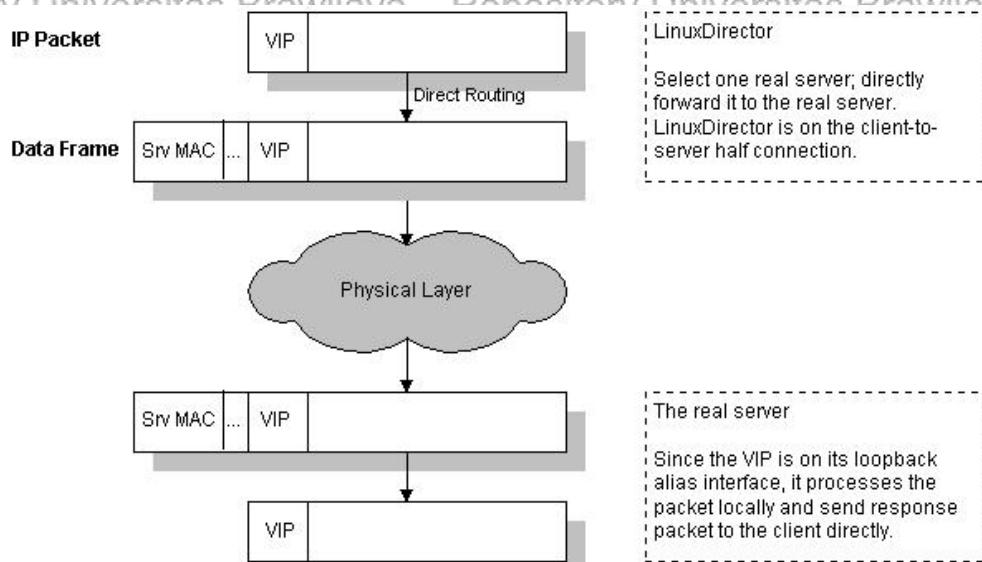
	Layer-2 Forwarding	Layer-3 Forwarding
Two-Ways		Network Address Translation
One-Ways	Direct Routing	IP Tunneling

Sumber : Sierra (2009:14)

Terdapat arsitektur *two-ways* dan *one-way forwarding* pada *load balancing*.

Arsitektur *two-ways* adalah respon dari *server* ke *client* dapat melalui *load balancer* (*director*). Pada arsitektur *one-way* respon dari *server* mencari jalur alternatif langsung menuju *client* tanpa melewati *load balancer*. *Layer-2 forwarding* adalah *forwarding* pada *layer 2 OSI (open systems interconnection)* yaitu pada *data link layer*.

Pada gambar 4.4 merupakan contoh implementasi *layer-2 forwarding*



Gambar 4.4 : Alur kerja *direct routing*

Sumber : <http://www.linuxvirtualserver.org/VS-DRouting.html> [13]

4.1.2.2 Analisis Kebutuhan Perangkat Lunak

Untuk membangun sistem *load balancing* pada *web server* menggunakan sistem operasi dan aplikasi sebagai berikut :

1. Sistem Operasi

Sistem operasi yang digunakan dalam pembuatan sistem *load balancing* pada *web server* ini adalah CentOS 5.7. CentOS 5.7 dengan kernel 2.6.18 digunakan pada *real server* maupun *director*. Sistem operasi linux yang mempunyai kernel dibawah 2.4.28 belum mendukung *virtual server* sehingga tidak dapat digunakan dalam membuat sistem *load balancing*.

CentOS merupakan sistem operasi berbasis Linux, yang mempunyai beberapa fitur untuk melakukan pengamatan penggunaan CPU dan memori, pemetaan alamat memori, yang bermanfaat dalam analisis.

2. Software Load Balancing / Director

Sistem *load balancing* yang dibuat menggunakan perangkat pendukung Linux *virtual server* (EVS) dari Redhat yaitu Piranha. Piranha adalah perangkat *monitoring cluster* dalam LVS dan juga dapat digunakan untuk konfigurasi *director* dan *real server* dalam sistem LVS.

Dalam Piranha terdapat *ipvsadm* (IP *virtual server administration*) yang mengatur kerja dari *director*. Dalam *ipvsadm* dapat diterapkan fungsi-fungsi yang mengatur kerja *director* termasuk juga algoritma penjadwalan. Ketika *director*



mendapat sebuah *request*, *director* akan meneruskan *request* ke *real server* sesuai dengan algoritma yang digunakan. Piranha mendukung algoritma *round robin*, *least-connections*, *weighted round robin*, *weighted least-connections*, *source / destination hash scheduling*.

3. Web Server

Untuk membuat *web server* digunakan Apache. Apache memiliki program pendukung cukup banyak. Berikut ini adalah beberapa program pendukung dari Apache *web server* tersebut, antaranya :

- a. Kontrol akses.
- b. *Common Gateway Interface* (CGI) yang paling terkenal dan sangat sering digunakan adalah Perl (*Practical Extraction and report language*).
- c. Modul PHP (*personal home page*) menggunakan php versi 5.2.
- d. SSI (*Server Side Include*) [14]
- e. Modul MySQL menggunakan versi 5.0.95.

Dalam Apache terdapat konfigurasi yang dapat diatur melalui *httpd.conf* sehingga *web server* dapat bekerja optimal sesuai spesifikasi perangkat keras yang digunakan dalam analisis. Parameter-parameter dasar didalam modul *prefork* yang perlu diubah antara lain:

- *StartServer*, berapa jumlah anak proses yang berjalan saat Apache pertama kali dijalankan.
- *MaxClients*, berupa jumlah anak proses yang boleh berjalan dalam Apache.

4. Database server

Untuk dapat menampilkan halaman yang dinamis *website* membutuhkan sebuah *database*. MySQL adalah sebuah *database management system* (DBMS) yang menggunakan bahasa standar sql (*structured query language*). Dalam pengujian digunakan *mysql-server* dan *mysql-client* versi 5.0.95. Apache membutuhkan sebuah modul tambahan berupa *php-mysql*.

5. Content web

Dalam pengujian dibutuhkan sebuah halaman *website* yang di dalamnya terdapat berbagai macam *file* dasar yang dibutuhkan untuk membentuk sebuah halaman *website* di dalam *web browser* seperti javascript, css, html, dan gambar. Salah satu *content management system* (CMS) yang umum digunakan dan bersifat *opensource*



adalah Joomla berdasarkan 2011 *open source award* oleh Packtpub. Dalam pengujian ini menggunakan CMS Joomla versi 1.5.25 yang dapat berjalan pada php versi 5.1 dan menggunakan *database* MySQL.

6. *Stress Tool*

Dalam pengujian kebutuhan daya listrik pada *web server*, diperlukan suatu keadaan dimana *server* bekerja maksimal, dalam hal ini *server* menerima permintaan dalam jumlah sebesar *maxClients* pada konfigurasi *web server*. Untuk itu dibutuhkan perangkat lunak yang mampu mengirim permintaan HTTP ke *server* dalam jumlah banyak secara serentak. Dalam pengujian digunakan *software stress test* Apache JMeter.

4.1.2.3 Analisis Kebutuhan *Hardware* dan Alat Ukur

Hardware yang digunakan untuk membuat sistem *load balancing* pada *web server* menggunakan *personal computer* (PC) yang digunakan sebagai *director* dan *web server*, ethernet card, kabel UTP, switch dan amperemeter digital.

1. PC untuk *load balancer* dan *web server*

Spesifikasi 2 buah *real server* yang digunakan adalah sama dengan rincian sebagai berikut:

- *Power Supply* : Power Up 380W
- *Motherboard* : Asus P5KPL-AM SE
- *Prosesor* : Intel Core 2 Duo CPU E7400
- *Memori* : 2 x 1GB DDR2
- *Harddisk* : Seagate 7200 500 GB

2. PC untuk *client*

Perangkat keras yang digunakan sebagai *client* menggunakan *build-up desktop PC* Compaq Presario CQ3138L dengan spesifikasi sebagai berikut

- *Motherboard* : Foxconn H-141-uATX
- *Prosesor* : Intel Core 2 Duo E7500
- *Memori* : 1 GB
- *Harddisk* : 500 GB



3. Switch

Switch merupakan perangkat pada *layer datalink*. Switch dapat menangani beberapa sambungan sekaligus pada saat yang sama. Tiap-tiap *port* 100Base-TX pada sebuah switch dapat mengirim dan menerima frame-frame secara bersamaan (*Full-Duplex*).

4. Wattmeter

Untuk mengetahui kebutuhan daya listrik selama *server* melayani permintaan dari *client* dibutuhkan *wattmeter* yang dapat merekam daya listrik atau menampilkan hasil pengukuran kedalam komputer sehingga memudahkan perhitungan. Fluke-43B merupakan sebuah Power Quality Analyzer dengan spesifikasi sebagai berikut :

Tabel 4.2 : Spesifikasi Fluke-43B

	Spesification	
Input	Input impedance	1 MΩ, 20 pF
	Voltage rating	600V rms, CAT III
Power display	Watts, VA, VAR	1-phase and 3-phase, 3 conductor balanced loads
	Ranges:	250 W - 1,56 GW
	Accuracy:	±(4% + 4 counts) Fundamental Power
	Accuracy:	± (2 % + 6 counts) Total Power
	PF Power Factor	
	Range:	0 - 1.0
	Accuracy:	±0.04

Sumber : Fluke-43B *datasheet* [15]

4.2. Desain Topologi dan Konfigurasi Sistem

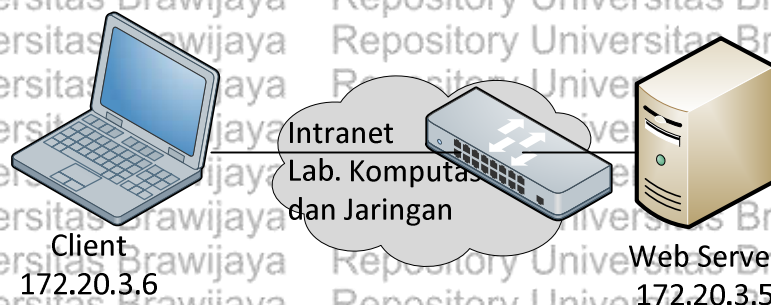
Dalam penelitian ini perancangan desain topologi dan konfigurasi sistem yang dilakukan adalah

- Desain topologi sistem *web server*
- Konfigurasi *web server* dan *load balancer*
- Desain topologi sistem *linux virtual server*
- Konfigurasi *load balancing* menggunakan algoritma *round robin* pada LVS
- Pengkabelan listrik dan pengukuran menggunakan *wattmeter*



4.2.1 Desain Topologi Sistem Web Server

Dalam penelitian ini komunikasi *client-server* dilakukan pada satu segmen jaringan yang sama menggunakan PC pada Laboratorium Komputasi dan Jaringan.



Gambar 4.5 : Topologi Web Server

Sumber : Perancangan

4.2.2 Konfigurasi Sistem Web Server

Beberapa konfigurasi dasar pada sistem operasi yang dilakukan antara lain adalah konfigurasi :

1. Sistem operasi meliputi alamat IP dan *service* yang dijalankan

Tabel 4.3 : Konfigurasi alamat IP pada PC

No	Nama PC	Service yang berjalan	Alamat IP	Interface
1.	Server 1	Load Balancer	172.20.3.2/24	Eth0:1
2.	Server 1	Web Server	172.20.3.4/24	Eth0
3.	Server 2	Web Server	172.20.3.5/24	Eth0

Sumber : Perancangan

2. *Service web* Apache, meliputi konfigurasi *httpd* dan MySQL

Konfigurasi *web server* Apache menggunakan konfigurasi standar dengan perubahan pada variabel *maxClients* (jumlah *client* yang dapat ditangani) sesuai perhitungan *maxClients*. Konfigurasi Apache tersimpan dalam file *httpd.conf* (lampiran 1) dan konfigurasi PHP pada file *php.ini* (lampiran 2). Konfigurasi MySQL-server yang digunakan *web server* menggunakan konfigurasi standar yang tersimpan pada file *my.cnf* (lampiran 3).

3. CMS Joomla berisi halaman contoh standar. CMS Joomla ter-*install* pada folder “/var/www/html/”. Daftar file yang diakses oleh *web browser (client)* ada pada tabel 4.4.



Tabel 4.4 : Daftar file yang diakses client setiap permintaan satu halaman website

No.	Nama File	Ukuran (byte)
1	./index.php	2.049
2	./templates/system/css/system.css	1.385
3	./templates/system/css/general.css	2.777
4	./templates/rhuk_milkyway/css/blue.css	723
5	./templates/rhuk_milkyway/css/template.css	12.924
6	./templates/rhuk_milkyway/css/blue_bg.css	730
7	./media/system/js/mootools.js	74.434
8	./media/system/js/caption.js	1.963
9	./templates/rhuk_milkyway/images/blue/mw_shadow_l.png	224
10	./templates/rhuk_milkyway/images/blue/mw_shadow_r.png	221
11	./templates/rhuk_milkyway/images/mw_joomla_logo.png	6.021
12	./templates/rhuk_milkyway/images/blue/mw_header_t_r.png	856
13	./templates/rhuk_milkyway/images/blue/mw_header_t_l.png	739
14	./templates/rhuk_milkyway/images/blue/mw_header_t.png	363
15	./templates/rhuk_milkyway/images/mw_header_b.png	239
16	./templates/rhuk_milkyway/images/mw_menu_cap_l.png	384
17	./templates/rhuk_milkyway/images/mw_header_r_b.png	243
18	./templates/rhuk_milkyway/images/mw_header_l_b.png	236
19	./templates/rhuk_milkyway/images/mw_menu_normal_bg.png	273
20	./templates/rhuk_milkyway/images/mw_content_t_l.png	329
21	./templates/rhuk_milkyway/images/mw_menu_separator.png	225
22	./templates/rhuk_milkyway/images/mw_content_t.png	203
23	./templates/rhuk_milkyway/images/mw_menu_cap_r.png	709
24	./templates/rhuk_milkyway/images/mw_content_t_r.png	342
25	./templates/rhuk_milkyway/images/blue/mw_box_tr.png	349
26	./templates/rhuk_milkyway/images/blue/mw_box_tl.png	310
27	./templates/rhuk_milkyway/images/blue/mw_box_br.png	1.666
28	./templates/rhuk_milkyway/images/blue/mw_box_bl.png	344
29	./templates/rhuk_milkyway/images/mw_line_grey.png	207
30	./images/M_images/printButton.png	379
31	./images/M_images/pdf_button.png	582
32	./images/M_images/emailButton.png	428
33	./templates/rhuk_milkyway/images/mw_readon.png	307
34	./templates/rhuk_milkyway/images/mw_content_b_l.png	320
35	./images/banners/shop-ad-books.jpg	14.608
36	./templates/rhuk_milkyway/images/mw_content_b.png	216
37	./images/M_images/livemarks.png	725
38	./templates/rhuk_milkyway/images/blue/mw_footer_b_l.png	614
39	./templates/rhuk_milkyway/images/blue/mw_footer_b.png	351
40	./templates/rhuk_milkyway/images/mw_content_b_r.png	326
41	./templates/rhuk_milkyway/images/blue/mw_footer_b_r.png	569

Sumber : Pengujian

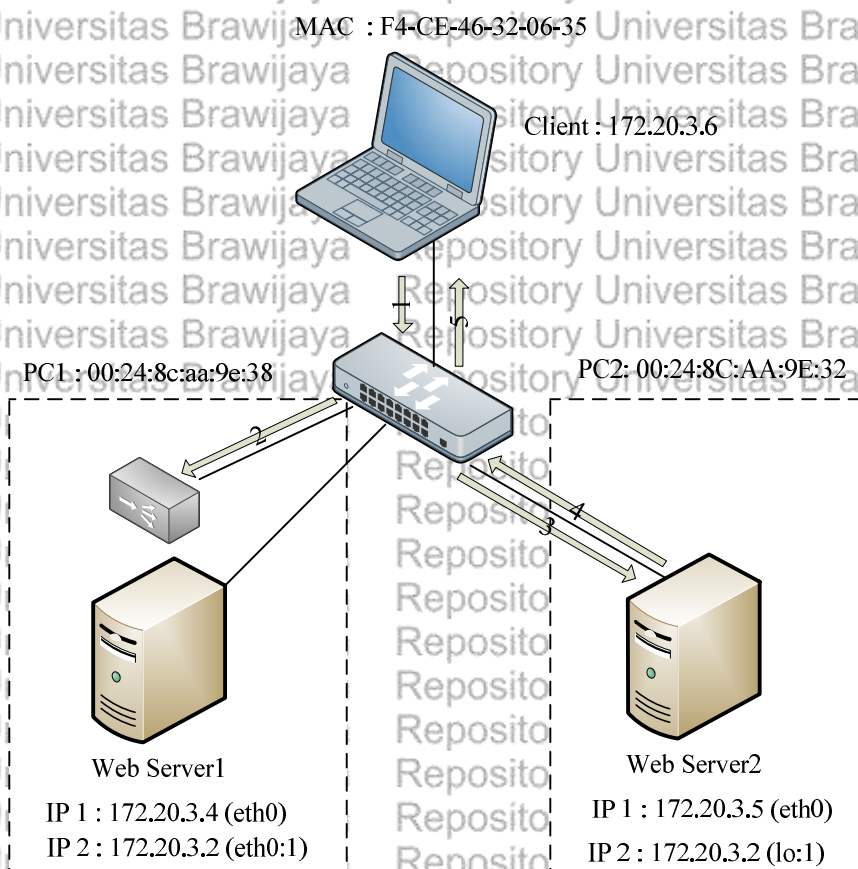


4.2.3 Desain Topologi Linux Virtual Server

Sistem *Load Balancing* pada sistem menggunakan Linux *virtual server* dalam sistem operasi CentOS LVS yang dapat ditemukan di dalam paket Piranha dengan kemudahan konfigurasi melalui halaman *website*. *Director* dapat diletakkan dalam salah satu *web server*. Pada Gambar 4.6, *director* memiliki IP 172.20.3.2.

Topologi yang digunakan untuk membangun sistem *load balancing* pada *Web Server* adalah topologi *star*. Hal ini menyebabkan sistem ini lebih *realible* karena ketika ada salah satu *real server* yang mengalami gangguan, sistem tetap dapat melayani *request* dari *client*.

Load balancing yang digunakan pada pengujian menggunakan *direct routing* dimana alamat IP *load balancer*, alamat IP *real server*, dan alamat *virtual IP* sistem berada pada satu segmen jaringan. *Client* yang akan melakukan permintaan halaman *website* juga berada pada satu segemen jaringan sehingga dapat mengurangi waktu transmisi yang biasa terjadi pada proses *routing*.



Gambar 4.6 : alur paket metode *direct routing* pada satu segmen jaringan

Sumber : Perancangan



Pada Gambar 4.6 dilakukan pertukaran *source* dan *destination* MAC dan IP *addresses* karena *real server* berada pada NAT. Pada gambar 4.4 merupakan aliran paket pada *load balancing* dengan menggunakan *direct routing*.

Dengan konfigurasi seperti pada gambar 4.6 *header* TCP/IP pada aliran paket data yang terjadi akan tampak seperti pada tabel berikut :

Tabel 4.5 : Alur paket pada metode *direct routing*

	IP asal	IP tujuan	Mac asal	Mac tujuan
1	172.20.3.6/24	172.20.3.2/24	F4-CE-46-32-06-35	00:24:8C:AA:9E:38
2	172.20.3.6/24	172.20.3.2/24	F4-CE-46-32-06-35	00:24:8C:AA:9E:38
3	172.20.3.6/24	172.20.3.2/24	F4-CE-46-32-06-35	00:24:8C:AA:9E:32
4	172.20.3.2/24	172.20.3.6/24	00:24:8C:AA:9E:32	F4-CE-46-32-06-35
5	172.20.3.2/24	172.20.3.6/24	00:24:8C:AA:9E:32	F4-CE-46-32-06-35

Sumber : Perancangan

Tabel 4.5 merupakan *header* paket data dari *client* yang melalui *load balancer*.

Terjadi perubahan alamat MAC / alamat fisik pada urutan ke-3 dan ke-3 dengan alamat IP asal dan tujuan sama *load balancer* merubah mac tujuan menjadi *mac* PC lain sehingga permintaan dari *client* ditangani server yang mempunyai *mac* tersebut.

4.2.4 Konfigurasi *load balancing* menggunakan algoritma *round robin* pada LVS

Dalam penelitian ini algoritma yang digunakan adalah *round robin* yang membagi beban ke masing-masing *real server* secara bergantian. Sebuah halaman *website* dinamis memerlukan beberapa permintaan ke *server*. Permintaan pertama berupa "GET / HTTP/1.0" berisi permintaan sebuah halaman *index.php* yang di dalamnya berisi beberapa *script* PHP maupun *query* SQL yang membutuhkan waktu proses lama. Sedangkan permintaan lain hanta berupa *script*, *css*, atau objek lain seperti gambar membutuhkan waktu proses singkat.

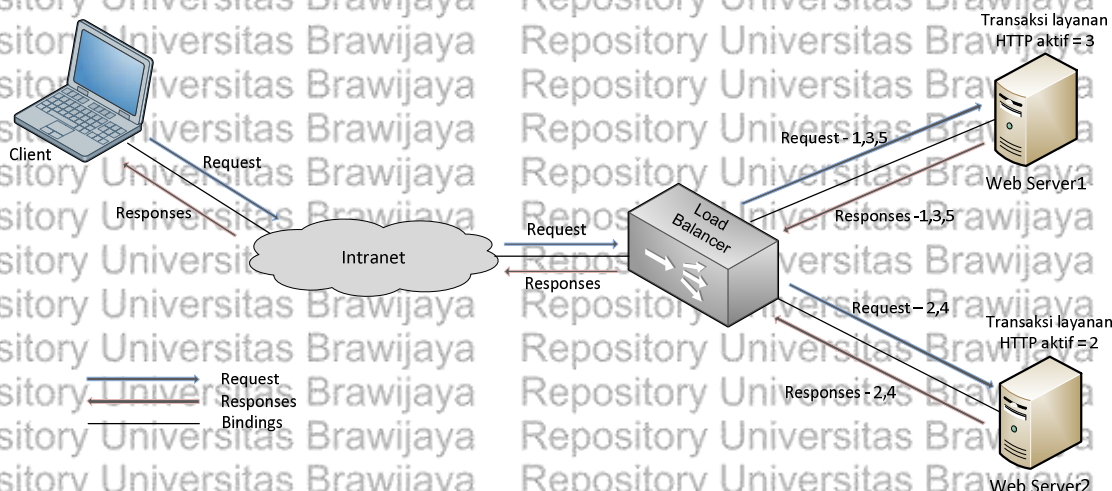
Dengan menggunakan metode *round robin* pembagian jenis permintaan bisa diatur secara merata. Dengan menggunakan metode ini semua *server* akan diperlakukan sama tanpa perlu mengetahui jumlah koneksi yang terbentuk, maupun melihat kesibukan *server*. Dengan kata lain algoritma ini bisa disebut dengan algoritma *stateless* yang menentukan distribusi tanpa melihat koneksi yang aktif. Dalam implementasi dihasilkan sebuah tabel seleksi *server* yang ditunjukkan pada tabel 4.6 [13].



Tabel 4.6 : Pembagian request menggunakan metode *round robin*

Request	Protocol	Src IP	Dst IP	Src Port	Dst Port	Server
1	TCP	172.18.3.18	172.18.3.2	53420	80	Server1
2	TCP	172.18.3.18	172.18.3.2	53422	80	Server2
3	TCP	172.18.3.100	172.18.3.2	45415	80	Server1
4	TCP	172.20.17.7	172.18.3.2	65228	80	Server2
5	TCP	172.20.17.7	172.18.3.2	65229	80	Server1

Sumber : Perancangan



Gambar 4.7 : Konfigurasi algoritma *round robin*.

Sumber : Perancangan

Pada Gambar 4.7 dapat dilihat ada dua layanan *virtual server* memilih layanan untuk setiap koneksi yang masuk secara bergantian sesuai algoritma *round robin* :

- a) *Server-1* yang menangani 3 transaksi aktif.
- b) *Server-2* yang menangani 2 transaksi aktif.

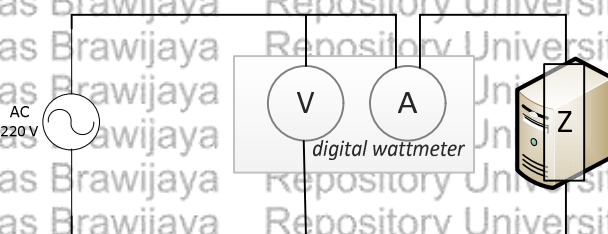
4.2.5 Pengkabelan dan Pengukuran Menggunakan *Wattmeter*

Pengujian penggunaan daya listrik dilakukan pada saat menggunakan sistem satu buah perangkat *server* dan pada sistem *load balancing* yang menggunakan dua buah perangkat *server*. Pengujian kebutuhan daya listrik pada masing-masing sistem dilakukan bersamaan dengan pengujian performansi.



4.2.5.1 Pada Saat Pengujian Sebuah *Web Server*

Pengujian pada sebuah *server* dilakukan seperti pada pengukuran daya listrik pada umumnya, dengan meletakkan ampermeter dengan posisi lebih dekat dengan beban yang diukur, dengan asumsi beban yang diukur melewati arus yang kecil.

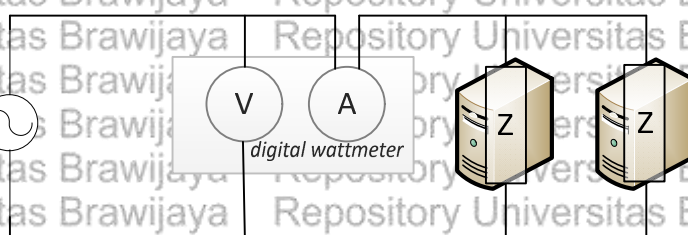


Gambar 4.8 : Pengkabelan pada sebuah sever.

Sumber : Perancangan

4.2.5.2 Pada Saat Pengujian Sistem *Load Balancing* dengan Dua *Server*

Pengujian pada sistem *load balancing* dengan menggunakan dua buah *web server* dilakukan seperti pengujian pada sebuah *server*. Dengan memasang beban secara paralel satu sama lain.



Gambar 4.9 : Pengkabelan pada *load balancing* sever.

Sumber : Perancangan

4.3. Implementasi Sistem

Implementasi sistem sesuai perancangan diterapkan dalam perangkat keras dan konfigurasi perangkat lunak yang telah dipersiapkan hasil konfigurasi tersebut antara lain :

4.3.1. *Web Server* Apache pada Sistem Operasi Linux

Hasil konfigurasi alamat IP pada sistem operasi CentOS, instalasi *piranha*, serta pengujian performansi dan penggunaan energi listrik ditampilkan pada gambar 4.10 sampai gambar 4.15 berikut :



```

root@server:~
[root@server ~]# ip -f inet a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    inet 127.0.0.1/8 scope host lo
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    inet 172.20.3.4/24 brd 172.20.3.255 scope global eth0
    inet 172.20.3.2/24 brd 172.20.3.255 scope global secondary eth0:1
[root@server ~]# service httpd status
httpd (pid 2822) is running...
[root@server ~]# service mysqld status
mysqld (pid 2747) is running...
[root@server ~]# php -v
PHP 5.1.6 (cli) (built: Feb  2 2012 18:24:47)
Copyright (c) 1997-2006 The PHP Group
Zend Engine v2.1.0, Copyright (c) 1998-2006 Zend Technologies
[root@server ~]#
  
```

Gambar 4.10 : Hasil konfigurasi IP dan instalasi web server pada server 1

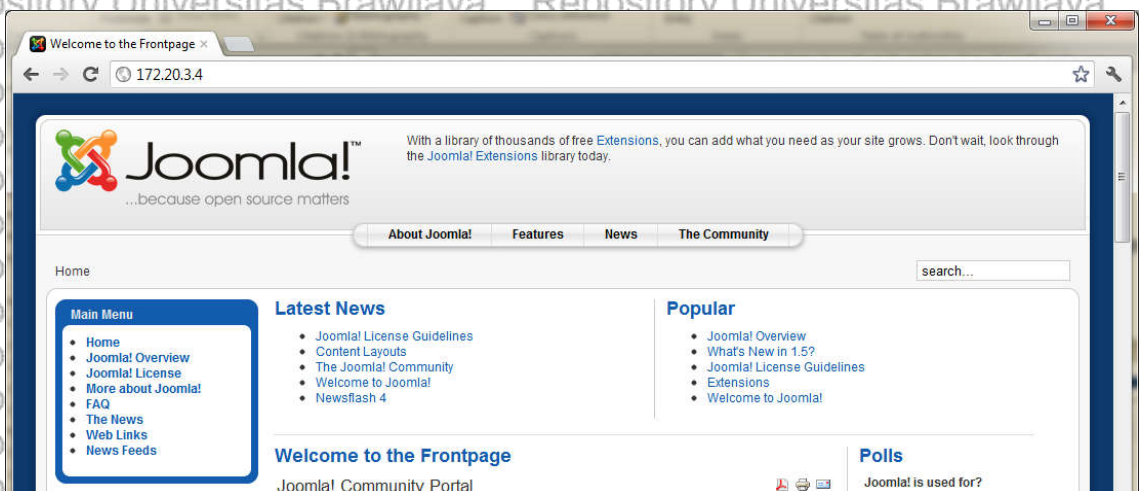
Sumber : Implementasi

```

root@server2:/var/www/html
[root@server2 html]# ip -f inet a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
    inet 172.20.3.2/32 brd 172.20.3.2 scope global lo:1
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    inet 172.20.3.5/24 brd 172.20.3.255 scope global eth0
[root@server2 html]# service httpd status
httpd (pid 2701) is running...
[root@server2 html]# service mysqld status
mysqld (pid 2620) is running...
[root@server2 html]# php -v
PHP 5.1.6 (cli) (built: Jun 27 2012 12:21:16)
Copyright (c) 1997-2006 The PHP Group
Zend Engine v2.1.0, Copyright (c) 1998-2006 Zend Technologies
[root@server2 html]#
  
```

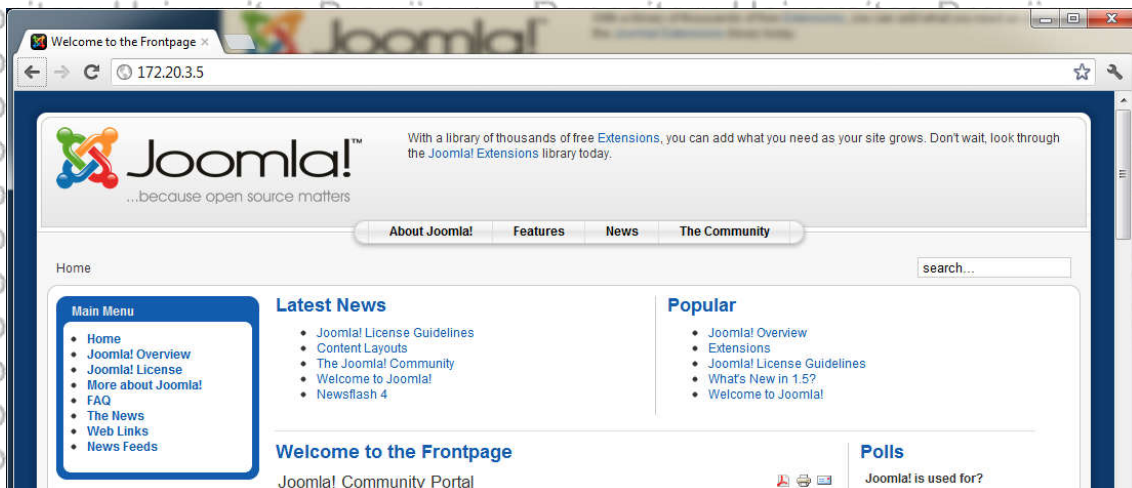
Gambar 4.11 : Hasil konfigurasi IP dan instalasi web server pada Perangkat server 2

Sumber : Implementasi



Gambar 4.12 : Tampilan halaman website di server 2

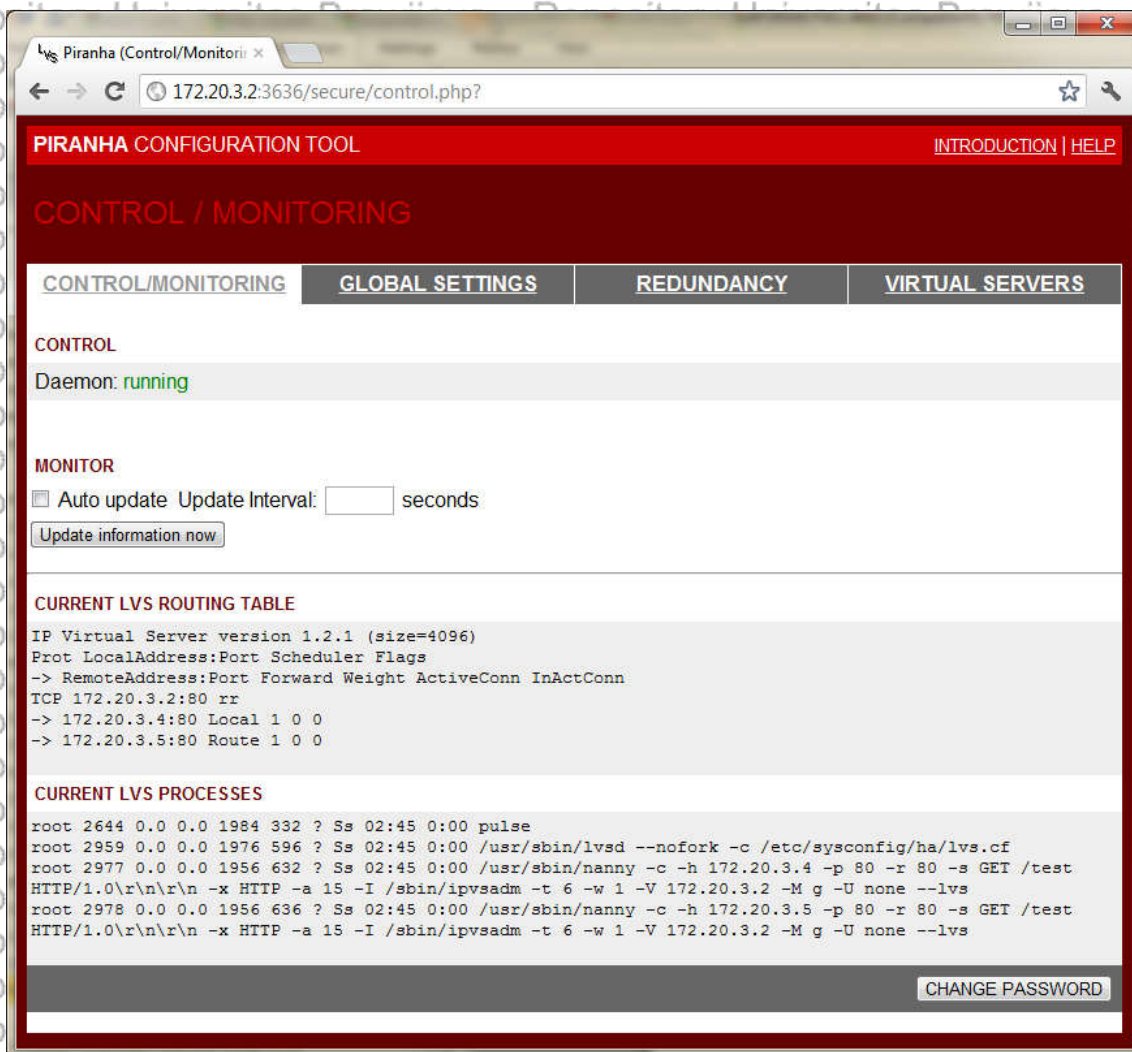
Sumber : Implementasi



Gambar 4.13 : Tampilan halaman website di server 2

Sumber : Implementasi

4.3.2. Load Balancing Menggunakan Piranha



Gambar 4.14 : Hasil instalasi dan konfigurasi LVS pada perangkat server 1

Sumber : Implementasi



```

root@server:~
[root@server ~]# ip -f inet a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    inet 127.0.0.1/8 scope host lo
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    inet 172.20.3.4/24 brd 172.20.3.255 scope global eth0
    inet 172.20.3.2/24 brd 172.20.3.255 scope global secondary eth0:1
[root@server ~]# service httpd status
httpd (pid 2822) is running...
[root@server ~]# service mysqld status
mysqld (pid 2747) is running...
[root@server ~]# service pulse status
pulse (pid 2644) is running...
[root@server ~]# service ipvsadm status
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
-> RemoteAddress:Port      Forward Weight ActiveConn InActConn
TCP  172.20.3.2:80 rr
-> 172.20.3.4:80            Local    1      0      0
-> 172.20.3.5:80            Route   1      0      0

[root@server ~]#

```

Gambar 4.15 : Hasil instalasi dan konfigurasi LVS pada perangkat server 1
Sumber : Implementasi

4.3.3. Pengujian Menggunakan JMeter

Pengujian performansi menggunakan JMeter sebagai user yang mengirimkan banyak permintaan secara bersamaan akan menghasilkan data berupa *throughput*, *error*, dan waktu respon seperti pada gambar 4.16 :

Pengujian JMeter100.jmx (D:\My Documents\Dropbox\pengujian\JMeter\Pengujian JMeter100.jmx) - Apache JMeter (2.6 r1237317)

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Browse... LogDisplay Only: Errors Successes

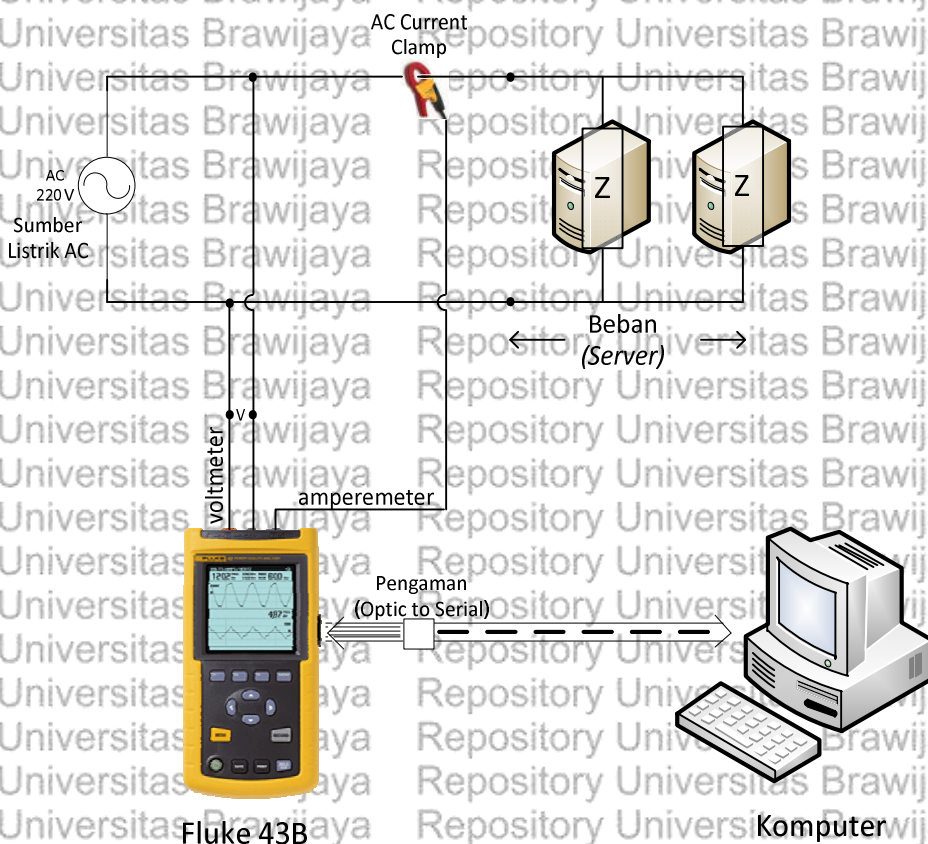
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/ser
/	200	4024	843	5509	1043.68	0.00%	15.2/sec	476
/media/syst...	200	55	1	249	36.49	0.00%	24.6/sec	53
templates/s...	200	51	2	427	46.43	0.00%	24.4/sec	38
templates/h...	200	47	2	318	38.06	0.00%	24.4/sec	72
templates/h...	200	50	2	286	37.62	0.00%	24.4/sec	23
templates/h...	200	78	4	419	53.64	0.00%	24.0/sec	309
templates/s...	200	61	1	214	37.77	0.00%	24.0/sec	22
/images/M_...	200	63	1	160	38.67	0.00%	24.0/sec	22
/media/syst...	200	138	9	265	69.43	0.00%	24.1/sec	175
/images/M_...	200	64	1	155	40.84	0.00%	24.0/sec	19
/images/M_...	200	60	2	157	42.96	0.00%	24.2/sec	14
/images/M_...	200	54	1	156	39.49	0.00%	24.6/sec	16
templates/h...	200	53	1	153	39.17	0.00%	24.6/sec	11
templates/h...	200	49	1	140	36.49	0.00%	24.7/sec	14
templates/h...	200	42	1	138	34.02	0.00%	24.8/sec	23
templates/h...	200	41	1	151	33.15	0.00%	24.9/sec	11
templates/h...	200	40	1	159	30.24	0.00%	25.1/sec	27
templates/h...	200	38	2	160	29.52	0.00%	25.4/sec	15

Include group name in label? Save Table Data Save Table Header

Gambar 4.16 : Contoh hasil Pengujian server 2 menggunakan Apache JMeter

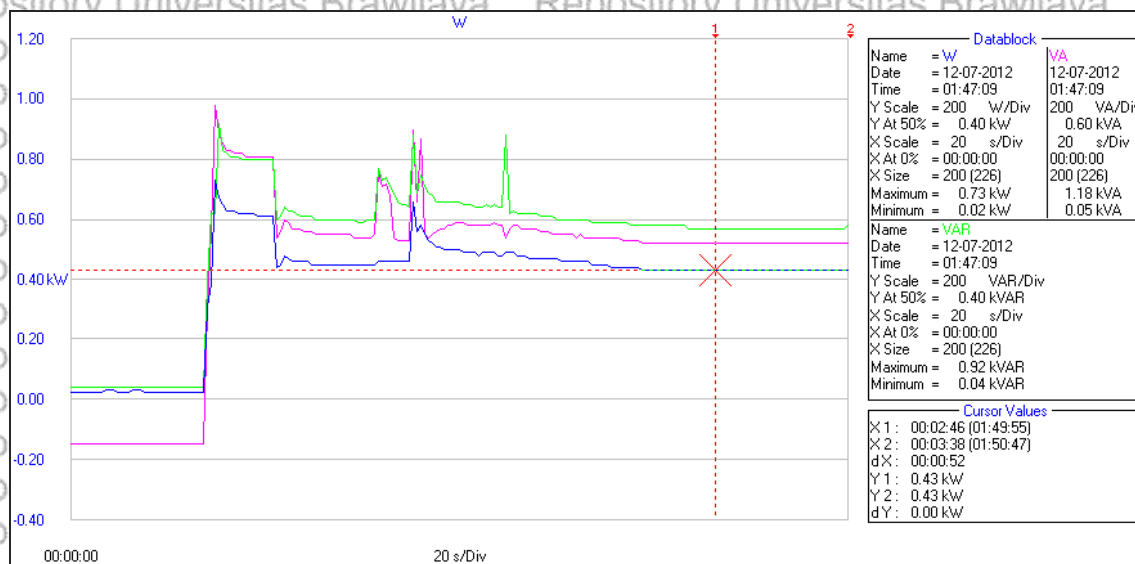


4.3.4. Pengukuran Menggunakan Wattmeter



Gambar 4.17 : Pengkabelan alat ukur untuk menguji penggunaan daya listrik menggunakan Fluke43B sebagai wattmeter

Sumber : Perancangan



Gambar 4.18 : Hasil pengukuran penggunaan daya listrik pada saat perangkat server 2 mulai dinyalakan

Sumber : Pengujian