

---

## IMPLEMENTASI LAOD BALANCING PADA WEB SERVER MENGUNAKAN APACHE DENGAN SERVER MIRROR DATA SECARA REAL TIME

Kresna Adi Pratama<sup>1</sup>, Ridho Taufiq Subagio<sup>2</sup>, Muhammad Hatta<sup>3</sup>, Victor Asih<sup>4</sup>

Universitas Catur Insan Cendekia

Jl. Kesambi 202, Kota Cirebon, Jawa Barat Tlp : (0231) 220250

e-mail : kresna.adhie@gmail.com<sup>1</sup>, ridho.taufiq@cic.ac.id<sup>2</sup>, muhammad.hatta@cic.ac.id<sup>3</sup>,  
victor.asih@cic.ac.id<sup>4</sup>

### ABSTRAK

PT.Trimitra Data Teknologi adalah perusahaan yang bergerak dalam bidang teknologi dan informasi, *website* menjadi salah satu cara jembatan komunikasi antara *client* dan perusahaan. Banyaknya *client* yang mengakses membuat beban sebuah *web server* dalam perusahaan menjadi berat dan menimbulkan masalah yaitu *down nya server* yang membuat *client* sulit untuk mengakses *website* perusahaan. Untuk membantu mengatasi masalah yang terjadi diterapkannya metode *load balancing* dengan algoritma *request counting algorithm* dimana bertujuan untuk membagi beban secara merata dalam *web server* dan memperkecil waktu respon antara *client* dan *server*, beban terbagi dengan anggota *server* yang terdaftar dalam *server load balancing*. Dengan penerapan metode *load balancing* maka kerja *server* akan menjadi lebih maksimal karena adanya sistem *high availability* dimana saat salah satu *server* mati maka kerja *server* akan diambil alih oleh *server* yang lain. Selain metode *load balance* penerapan sistem dengan *server mirror* yang dilakukan dapat membantu memaksimalkan metode *load balance* karena adanya replikasi otomatis antara *web server* yang menjadi anggota *load balance* baik konten *website* ataupun *database*. Hasil yang terjadi adalah *web server* perusahaan akan menjadi sistem yang mampu bekerja secara baik saat melayani *client* dalam hal layanan *web server* karena beban terbagi dengan baik dan kecilnya waktu respon sehingga tidak adanya kesulitan *client* untuk mengakses *website* perusahaan.

**Kata kunci :** *Load Balance, Web Server, Mirror Server.*

### ABSTRACT

*PT.Trimitra Data Technology is a company engaged in the field of technology and information, the website becomes one of the bridges of communication between the client and the company. The number of clients that access makes the burden of a web server in the company becomes heavy and causes problems, namely the down server that makes it difficult for clients to access the company's website. To help overcome the problems that occur the implementation of the load balancing method with the request counting algorithm which aims to evenly distribute the load on the web server and reduce the response time between client and server, the load is shared with server members registered in the load balancing server. With the application of the load balancing method, server work will be maximized because of the high availability system where when one server dies the server work will be taken over by another server. In addition to the load balance method the application of the system with a mirror server can help maximize the load balance method due to automatic replication between web servers that are members of the load balance of website content or databases. The results that occur are the company's web server will be a system that is able to work well when serving clients in terms of web server services because the load is well divided and the response time is small so there is no difficulty for the client to access the company's website.*

**Keywords:** *Load Balance, Web Server, Mirror Server.*

## 1. PENDAHULUAN

### 1.1. Latar Belakang Masalah

Perubahan besar dalam dunia teknologi informasi sudah tidak bisa dipungkiri dimana bisa tercermin dalam meningkatnya juga pengguna internet yang menjadikannya sebagai media penyedia informasi. Dalam jaringan komputer *server* mempunyai peranan penting dalam mengatur para *client* untuk mendapatkan informasi yang mereka butuhkan dan dengan semakin bertambahnya *client* yang mengakses maka beban *server* juga semakin bertambah dikarenakan banyaknya *request* dari *client* terhadap *server*. PT. Trimitra Data Teknologi adalah perusahaan yang bergerak dalam bidang IT dimana penggunaan *web* menjadi salah satu yang sangat dibutuhkan untuk membantu menyediakan layanan jasa dan tentunya *web* juga merupakan salah satu jembatan antara *client* dengan perusahaan karena tidak semua *client* mampu datang ke kantor secara langsung disaat tertentu, serta ada banyak juga *client* yang mengeluhkan tentang sulitnya mengakses *website* perusahaan. Ternyata terdapat kekurangan sistem dalam topologi yang berjalan pada perusahaan dimana masih belum menyediakan *load balance* karena hanya tersedia satu *web server*. *Load balance* merupakan salah satu penanggulangan jika suatu saat *server* menurun kinerjanya karena melayani banyak permintaan dari *client* secara bersamaan, perusahaan PT. Trimitra Data Teknologi menyangkan kejadian tersebut bila terjadi, karena jika *client* yang sedang membutuhkan jasa dalam bidang IT terganggu saat mengakses *website* dalam hal ini tentu bisa sangat merugikan perusahaan. Maka sebab itu diterapkan solusi dengan metode *load balance* yang merupakan teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. Dengan belum diterapkannya teknik *load balancing* dalam *web server* PT. Trimitra Data Teknologi bisa menyebabkan beratnya kinerja *server* dalam sistem yang berjalan dan untuk mengatasinya maka penulis tertarik membuat judul “Implementasi Load Balancing pada Web Server Menggunakan APACHE dengan Server Mirror secara Real Time(Studi Kasus : PT. Trimitra Data Teknologi)”.

### 1.2. Identifikasi Masalah

Berdasarkan latar belakang masalah yang didapat dari penjelasan diatas diuraikan beberapa identifikasi masalahnya :

1. Belum diterapkannya metode *load balancing* sehingga beban *request* masih tertumpu pada satu *web server*.
2. Sistem yang sedang berjalan kurang efisien dan rentan dalam ketersediaan layanan karena hanya tersedia satu *web server*.
3. Bagaimana membuat sebuah sistem yang memberikan ketersediaan layanan *web server* jika *server down*?

### 1.3. Batasan Masalah

Penulis membatasi permasalahan dalam penulisan ini agar pembahasan tidak menyimpang dari tujuan, maka dilakukan pembatasan masalah sebagai berikut :

1. Pembagian *request* dalam *web server* hanya menggunakan *web server* yang sudah diterapkan metode *load balancing*.
2. Layanan dalam *web server* menggunakan dua *web server* untuk memaksimalkan ketersediaan layanan.
3. Pembuatan sistem untuk ketersediaan layanan *web server* ketika *down* hanya menggunakan dua *web server* yang sudah di *load balancing*.

### 1.4. Tujuan Penelitian

Tujuan dari pembuatan *load balance* pada *web server* ini adalah :

1. Konfigurasi *load balance* yang dilakukan dapat membagi beban secara merata dan mampu meringankan kinerja *server*.
2. Membuat jaringan dimana *server* mampu menyediakan layanan yang baik untuk *client* dalam hal ini merupakan ketersediaan layanan yang terdapat dalam *web server*.
3. *Load balancing* pada *web server* juga dapat memberikan solusi *failover* yang terjadi dikarenakan akibat kegagalan sistem dari *web server*.

## 2. KAJIAN PUSTAKA

### 2.1 Tinjauan Pustaka

Pada penelitian yang berjudul “Analisis Load Balancing Pada Web Server Menggunakan Algoritme Weighted Least Connection” menyebutkan bahwa algoritma *weighted least connection* membagi jumlah

koneksi yang dapat ditampung oleh suatu *server* dihitung dengan *weight* semakin besar *weight* maka *server* dapat menampung banyak koneksi dan pengujian sistem yang dilakukan pada penelitian diuji dalam beberapa pengujian dengan mengukur nilai *throughput*, *response time* dan *CPU usage*[1]. Lalu pada penelitian lainnya yang berjudul “Pengujian *Algoritma Load Balancing* pada *Web Server* Menggunakan *NGINX*” menjelaskan Dari hasil analisis pengujian menggunakan uji rata-rata kecepatan akses dari 2 populasi independen, yaitu dari algoritma *round robin* dan *least connection* didapatkan hasil bahwa penggunaan algoritma *least connection* untuk *load balancing* lebih memberikan performa yang baik dibandingkan algoritma *round robin* terhadap kecepatan akses *website* saat terjadi *request* yang sangat banyak dari pengguna dalam waktu yang bersamaan[2]. Dan juga pada penelitian yang berjudul “Implementasi *Load Balancing Web Server* menggunakan *Haproxy* dan *Sinkronisasi File* pada Sistem Informasi” Sistem *load balancing* dapat bekerja dengan baik ketika *request* datang dari *client* telah berhasil didistribusikan *balancer* secara merata kepada setiap *node cluster*. Sehingga *server* tidak mengalami *overload* dan kemampuan *web server* bisa melayani 10.000 *request* dengan tidak mengalami *error request*, penerapan sinkronisasi file bekerja dengan baik dimana file yang diupload pada *node 1* akan disinkron ke setiap *node 2* dan *node 3* pada *cluster*, begitu juga sebaliknya karena sinkronisasi file ini bersifat dua arah dan dengan implementasi *clustering server* dapat meningkatkan respon time *web server* dan meningkatkan jumlah *current connection* yang dapat dilayani oleh *server* [3].

## 2.2 Dasar Teori

### 2.2.1. Jaringan Komputer

Jaringan komputer didefinisikan sebagai sekumpulan komputer (lebih dari satu) yang terhubung satu dengan lainnya menggunakan media tertentu sehingga memungkinkan diantara komputer tersebut berinteraksi, bertukar data, dan berbagi peralatan bersama misalkan *printer*, *scanner* dan sumber informasi lainnya[4].

### 2.2.2. OSI Layer

*OSI layer* adalah standar komunikasi yang dipakai atau diterapkan dalam suatu jaringan[5]. Terdapat 7 lapisan atau layer dalam OSI untuk memproses pertukaran data dalam sistem yang berlainan melalui *hierarki* atau tingkatan protokol komunikasi.

### 2.2.3. TCP / IP

*TCP layer* adalah standar komunikasi data yang sering di gunakan pada internet untuk suatu proses tukar – menukar atau saling mengirim data dari suatu komputer ke komputer lainnya dalam suatu jaringan internet[6]. *TCP/IP* memiliki 4 *layer* kumpulan protokol yang bertingkat.

### 2.2.4. Sistem Operasi Linux

Sistem operasi berbasis *Linux* dapat dirancang dan dilaksanakan melalui menggabungkan *Linux* (sebagai *kernel* dari sistem operasi) dan alat-alat lain seperti kompiler, konektor, perakitan, dan alat-alat tingkat yang lebih tinggi seperti editor teks dan intermediet grafis[7].

### 2.2.5. Database

Secara konsep basis data atau *database* adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan dengan tata cara yang tertentu untuk membentuk data baru atau informasi data (*database*) contoh dari *database* yang biasa digunakan yaitu *MySQL* yang merupakan sebuah perangkat lunak atau *software* sistem manajemen basis data *SQL* atau *DBMS Multithread* dan *multi user*[8].

### 2.2.6. Replikasi

Replikasi adalah suatu teknik untuk melakukan *copy* dan pendistribusian data dan objek-objek *database* dari satu *database* ke *database* lain dan melaksanakan sinkronisasi antara *database* sehingga konsistensi data dapat terjamin[9].

### 2.2.7. Web Server Apache

*Web server* adalah sebuah perangkat lunak *server* yang berfungsi menerima permintaan *HTTP* atau *HTTPS* dari *client* yang di kenal dengan *web browser*, dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen *HTML* beberapa jenis *web server* diantaranya *Apache Web Server / The HTTP Web Server*, *Apache Tomcat*, *Internet Information Services*[10].

### 2.2.8. Load Balancing

*Load Balancing* merupakan teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi[11].

## 3. ANALISIS DAN PERANCANGAN SISTEM

### 3.1 Bahan dan Alat Penelitian

#### 3.1.1 Bahan Penelitian

Bahan penelitian merupakan bahan yang digunakan untuk melakukan penelitian, dalam perancangan dan pembuatan *load balance* pada *web server* akan membutuhkan data primer dan sekunder dari perusahaan terkait mengenai sistem sebelumnya. Pengumpulan data primer bisa dilakukan dengan wawancara terhadap seorang *network administrator* dimana dapat menghasilkan sebuah data berupa topologi yang sedang berjalan, alokasi *IP Address* dalam jaringan, jenis unit yang dipakai, spesifikasi unit meliputi *hardware* dan *software* dalam jaringan yang berjalan, jumlah unit yang sudah dipakai. Lalu untuk pengumpulan data sekunder dilakukan dengan cara membaca dan mempelajari literatur berupa buku, makalah, jurnal, artikel, atau bahan-bahan dari *internet* yang sesuai dengan topik yang sedang dibahas dalam penelitian ini, data yang telah didapat ini nantinya akan diolah dan dimasukkan dalam sistem yang akan diusulkan sehingga mampu dianalisis dan diuji.

#### 3.1.2. Alat Penelitian

Alat yang digunakan dalam penelitian memakai piranti *tools-tools* untuk membantu melakukan pengolahan atau pemberian perlakuan terhadap bahan penelitian.

1. Perangkat Keras (*Hardware*) yang akan dibutuhkan untuk pembuatan server :

- a. PC dengan Processor I5 @2.4 GHz
- b. RAM 8GB
- c. Hardisk 500GB
- d. Kabel UTP dengan konektor rj45

2. Perangkat Lunak (*Software*) yang dibutuhkan untuk mendukung pembuatan *load balancing* pada *web server* :

- a. Sistem Operasi : window 10, CentOS 7
- b. VirtualBox - 6.0.20
- c. Google Chrome - 83.0.4103.61
- d. Microsoft Visio
- e. APACHE Web Server
- f. Lsyncd
- g. MySQL

### 3.2. Prosedur Penelitian

Dalam tahapan ini prosedur ataupun metode yang digunakan adalah PPDIOO dimana model siklus hidup metode pengembangan jaringan dengan konsep PPDIOO ini, memberikan langkah-langkah kunci dalam keberhasilan perencanaan jaringan, baik itu pada tahapan desain, implementasi dan operasional nantinya. Pendekatan dengan model *top-down design*, mengarahkan infrastruktur jaringan untuk beradaptasi pada aplikasi-aplikasi apa saja yang dibutuhkan oleh suatu jaringan.

### 3.3. Teknik Pengumpulan Data

Pengumpulan data merupakan tahapan penting dalam proses penelitian, karena dengan mendapatkan data yang tepat maka riset akan berlangsung sesuai dengan perumusan masalah yang sudah ditentukan. Metode pengumpulan data yang penulis lakukan adalah dengan penelitian lapangan dan studi pustaka. Dalam tahapan ini. Penulis melakukan observasi maupun wawancara serta mempelajari teori-teori yang terkait dengan topik penelitian yang dapat mendukung pemecahan masalah penelitian. Pencarian referensi dilakukan di perpustakaan maupun secara online melalui internet. Selain itu, penulis juga mempelajari jurnal-jurnal dari hasil penelitian yang sudah pernah dikerjakan sebagai bahan perbandingan terhadap penelitian yang akan penulis kerjakan. Pustaka-pustaka yang dijadikan acuan dapat dilihat di Daftar Pustaka.

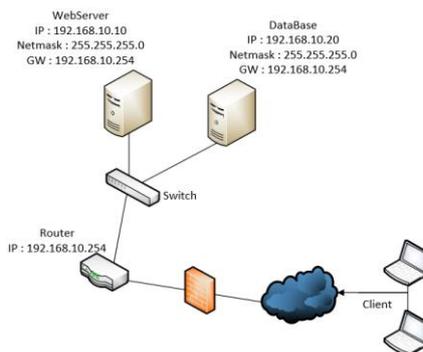
#### 4. IMPLEMENTASI DAN PENGUJIUAN

##### 4.1 Analisis

Analisis merupakan sebuah langkah awal dalam pengembangan sebelum masuk dalam tahap lainnya, analisis dilakukan untuk mengetahui dasar dari suatu masalah yang terjadi dimana hasil dari analisis dapat dijadikan kajian untuk membuat sistem yang lebih efisien. Dalam hal ini penganalisisan dilakukan karena masalah yang timbul dari sistem yang telah berjalan dimana sistem tidak dapat memberikan dukungan terhadap sistem layanan ketika sedang mengalami masalah.

##### 4.1.1. Skema Jaringan

Analisis ini dilakukan pada jaringan yang berjalan dan merupakan tahap awal, tahapan pertama akan dijabarkannya penjelasan tentang topologi yang sedang berjalan, *internet*, *database* dan layanan *server*.



**Gambar 1. Gambar Topologi Berjalan**

Dari yang terlihat dalam gambar hanya ada satu *web server* dengan alamat IP 192.168.10/24 dengan *gateway* 192.168.10.254 dan juga *database* untuk penyimpanan data. *IP Address* berfungsi sebagai tanda pengenalan dalam jaringan, disini terlihat layanan yang diberikan hanya mengandalkan dari satu *web server* dalam operasional normal tentu tidak akan menyebabkan banyak masalah untuk *client* tetapi jika saat *server down* maka akan menjadi suatu kelemahan dalam sistem nya. *Client* yang terhubung dengan *internet* bisa dapat mengakses langsung jaringan lokal *web server*.

##### 4.1.2. Pengalamatan IP Address

Suatu jaringan tentunya tidak akan mungkin tanpa adanya *IP Address* karena merupakan sebuah identitas supaya bisa dikenal dalam jaringan baik lokal maupun tidak. Berikut adalah alamat *IP Address* yang digunakan.

**Tabel 1. Analisis IP Address**

Perangkat	Alamat IP	Subnet Mask	Gateway
Web Server	192.168.10.10	255.255.255.0	192.168.10.254
Database Server	192.168.10.20	255.255.255.0	192.168.10.254
Router	192.168.10.254	255.255.255.0	-

##### 4.1.3. Perangkat Keras Jaringan

Perangkat keras berfungsi sebagai pengatur *software*, berikut adalah perangkat yang digunakan :

**Tabel 2. Analisis Perangkat Jaringan**

No	Nama perangkat	Spesifikasi	unit
1.	Web Server	Intel Core i5 @ 2.40GHZ, RAM 8GB, Storage 500GB SSD, Inter HD Grafis	1

2.	Database Server	Intel Core i5 @ 2.40GHZ, RAM 8GB, Storage 500GB SSD, Inter HD Grafis	1
3.	Router	Mikrotik RB750,MIPS-BE,CPU ARZ7241 400MHz Main Stroge 64MB, RAM 32 MB, LAN Ports 5, RouterOS, Level 4	1
4.	Modem	LAN Ports 5, Wi-Fi 2.4 / 5 GHz, Wired,WPS	1
5.	Switch	5 port	1
6.	Kabel UTP 3M	Balden	2

**4.1.4. Perangkat Lunak Jaringan**

Supaya dapat mengatur *input output* yang masuk ke komputer dari perangkat keras, berikut adalah tabel perangkat lunak yang digunakan :

**Tabel 3. Analisis Perangkat Lunak Jaringan**

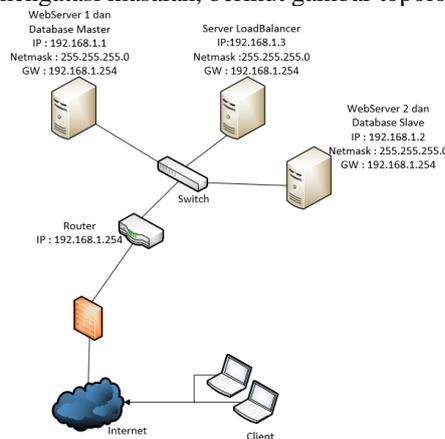
No	Perangkat	Distro	Jenis OS	Service
1.	Web server	Linux	CentOS 7	HTTP
2.	Database server	MySQL	MySQL	MariaDB
3.	Router	Mikrotik	RouterOS	MISP-BE

**4.2. Perancangan**

Perancangan merupakan sebuah langkah awal dalam pengembangan sebelum masuk dalam tahap lainnya, perancangan dilakukan untuk mengetahui kebutuhan apa saja yang akan di terapkan dan yang dibutuhkan. Dalam hal ini perancangan dilakukan untuk membuat sistem yang mampu memaksimalkan sistem yang berjalan.

**4.2.1. Topologi Usulan**

Setelah melihat adanya hal yang kurang dalam sistem yang sedang berjalan maka dari itu diajukannya sistem usulan untuk mengatasi masalah, berikut gambar topologi usulannya.



**Gambar 2. Gambar Topologi Usulan**

Dalam topologi yang diusulkan pada Gambar 2 terdapat beberapa tambahan berupa *web server* kedua dengan IP 192.168.1.2/24 dengan *gateway* 192.168.1.254 dan *web server load balance* dengan IP 192.168.1.3/24 dengan *gateway* 192.168.1.254 dimana semua terhubung dengan *switch* . *Web server load balance* merupakan tempat request *client* tertuju karena sebagai *front end* dari jaringan yang diusulkan. *Database* yang di usulkan pun bertambah yaitu *database slave* 192.168.1.2/24 karena letak *database* didalam *web server* kedua jadi IP yang digunakan mengikuti IP *web server* kedua, *database* diterapkan dengan sistem *master* dan *slave* untuk menjadi suatu solusi jika nanti *database master* mengalami kerusakan. Sedangkan kedua *web server* akan terkonfigurasi secara *mirroring* untuk memudahkan *maintenance* maupun pengubahan konten didalam layanan *web server*, hasil perubahan yang terjadi akan sama persis dengan apa yang dilakukan terhadap *web server* utama yaitu *web server* satu. Dengan adanya sistem seperti ini maka jaringan yang ada akan lebih maksimal dan para *client* yang mengakses akan

mendapatkan layanan yang lebih baik karena sistem yang diusulkan sudah mendukung ketersediaan yang tinggi dalam hal pelayanan *web*.

**4.2.2. Perancangan Infrastruktur Usulan**

Perancangan terhadap infrastruktur menjadi suatu awal dalam langkah dimana dapat mengetahui dan menetapkan apa yang dibutuhkan dalam pembuatan jaringan.

**Tabel 4. Perangkat Keras Usulan**

No	Nama perangkat	Spesifikasi	unit
1.	Web Server 1	Intel Core i5 @ 2.40GHZ, RAM 8GB, Storage 500GB SSD, Inter HD Grafis	1
2.	Web Server 2	Intel Core i5 @ 2.40GHZ, RAM 8GB, Storage 500GB SSD, Inter HD Grafis	1
3.	Server Load Balance	Intel Core i5 @ 2.40GHZ, RAM 8GB, Storage 500GB SSD, Inter HD Grafis	1
4.	Router	Mikrotik RB750,MIPS-BE,CPU ARZ7241 400MHz Main Stroge 64MB, RAM 32 MB, LAN Ports 5, RouterOS, Level 4	1
5.	Modem	LAN Ports 5, Wi-Fi 2.4 / 5 GHz, Wired,WPS	1
6.	Switch	5 port	1
7.	Kabel UTP 3M	Balden	2

**4.2.3. Alokasi IP Address**

Dibawah ini menunjukan alamat IP yang digunakan dalam perancangan infrastruktur usulan, IP merupakan tanda pengenalan dalam jaringan yang digunakan.

**Tabel 5. Alokasi IP Address**

Perangkat	Alamat IP	Subnet Mask	Gateway
Web Server 1 dan DB Master	192.168.1.1	255.255.255.0	192.168.1.254
Web Server 2 dan DB Slave	192.168.1.2	255.255.255.0	192.168.1.254
Server Load Balance	192.168.1.3	255.255.255.0	192.168.1.254
Router	192.168.1.254	255.255.255.0	-

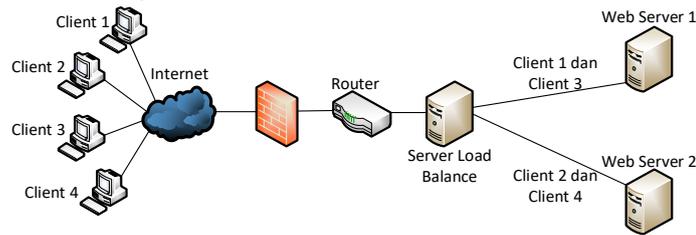
**4.2.4. Perangkat Lunak**

Komponen pendukung yang dimaksudkan adalah merupakan sebuah komponen yang bersifat perangkat lunak yang diperlukan dalam membantu pembuatan jaringan yang akan diusulkan.

**Tabel 6. Komponen Pendukung**

No	Nama Paket	Sumber	Fungsi
1.	Linux CentOS 7	https://centos.org	Sistem Operasi dalam server
2.	Apache	Repositories CentOS 7	Aplikasi pembuatan web server
3.	Lsyncd	Repositories CentOS 7	Aplikasi mirror dalam web server
4.	MySQL	https://dev.mysql.com	Database untuk server

#### 4.2.5. *Arsitektur Load Balancing*



**Gambar 3. Pembagian Kuota Load Balance**

Dalam pembagian *request client* yang menuju ke *server load balance* maka akan terbagi secara merata karena adanya konfigurasi *sticky session* karena jika tanpa *sticky session* maka tidak adanya pembagian merata karena *client* akan dilayani oleh 2 *web server* secara bergantian ketika akan melakukan *request* ulang, *sticky session* merupakan cara supaya *user* diberi penanda biasanya berupa *cookie*. Setelah *client* diberikan tanda maka *server load balance* akan membaca *cookie* tersebut dan mengarahkan ke *server* yang sudah pernah mengurus *client* tersebut. Dalam gambar bisa terlihat bahwa *client 1* dan *client 3* akan dilayani oleh *web server 1* sedangkan untuk *client 2* dan *client 4* akan dilayani oleh *web server 2*. *Sticky session* dalam konfigurasi di *server load balance* yang dilakukan menggunakan *header cookie* dimana *client* diberi penanda sebelum masuk lalu untuk *web server* akan terbagi menjadi 2 yaitu *web server 1* berada dalam *route* ke 1 dan *web server 2* akan berada dalam *route 2* setelah dilakukan konfigurasi tersebut maka *server load balancer* dapat membaca perubahan *route* untuk layanan pada *web server* yang menjadi member dalam *server load balance*.

#### 4.2.6. *Mirror Data*

Penerapan *server* dalam usulan juga menggunakan konfigurasi *mirror* yaitu untuk memudahkan *server* mempunyai replikasi dari konten *web server* maupun isi *database* tanpa harus mengkonfigurasi *web server* dan *database server* yang kedua, *mirror* ini dapat menghemat waktu dalam perubahan yang terjadi dalam konten *web server* maupun isi dalam *database* karena saat selesainya pekerjaan dalam *web server* pertama atau melakukan perubahan dalam *database* pertama maka akan terreplikasi secara identik terhadap *web server* kedua dan *database* kedua. Setelah *server master* membuat *database* hal pertama yang dilakukan adalah menyamakan nilai *variabel* dari *log database* yang terbuat di *server master* untuk menandakan *database* mana yang akan di *mirror*, setelah itu selanjutnya setiap perubahan dalam *database server master* yang tercatat di *binary log server master* akan dibaca oleh *I/O thread server slave* yang kemudian akan dicatat dalam *relay log server slave*. Karena perubahan data sudah berada dalam *relay log server slave* maka selanjutnya *SQL thread* akan bekerja untuk membaca data dalam *relay log* yang kemudian akan dimasukkan menjadi data dalam *database server slave*.

### 4.3 Implementasi Dan Pengujian

#### 4.3.1 Implementasi

Merupakan sebuah tindakan yang dilakukan setelah melakukan analisis dan perancangan, dalam hal ini implementasi dilakukan dalam langkah-langkah berupa instalasi dan konfigurasi.

##### 4.3.1.1 *Instalasi OS*

Sistem operasi merupakan sebuah sistem perangkat lunak yang berfungsi sebagai pengatur sumber daya dari *hardware* dan *software*, berikut langkah-langkah untuk melakukan *install OS*.

##### 4.3.1.2 *Konfigurasi IP statik*

Konfigurasi *IP static* dilakukan supaya tidak berubah saat *server reboot* dan untuk menetapkan alamat *address server* itu sendiri serta memudahkan konfigurasi antar *server* sedangkan jika memakai *IP DHCP* maka *IP server* akan bisa terus berubah jika dilakukan *reboot*, hal pertama yaitu masuk dalam script *enp0s3*.

##### 4.3.1.3 *Install Apache*

*HTTP* merupakan protokol yang digunakan halaman *browser* untuk menampilkan halaman *website* dari *web server*. *Web server* merupakan *server* yang mengirimkan *HTTP response* ke *browser* dan memprosesnya menjadi halaman *web*.

##### 4.3.1.4 *Konfigurasi Apache*

Setelah *install* selesai maka jalankan *service* dari *HTTP* dengan cara *systemctl start httpd.service* dan *systemctl enable httpd.service*, dua fungsi *command* tersebut adalah pertama menjalankan *service* dan kedua jika *server* dilakukan *reboot* maka *service* akan berjalan secara otomatis.

#### 4.3.1.5 Install Lsyncd

*Lsyncd* merupakan *tool* yang digunakan untuk melakukan replikasi terhadap data-data yang berada dalam direktori tertentu dalam *server*, dalam hal ini target untuk replikasi adalah direktori `/var/www/html`. Dalam proses replikasi antar *server* harus bisa dilakukan tanpa perlu menginput *password* lagi maka dari itu pertama dilakukan pembuatan *keygen* untuk memudahkan komunikasi antara *server*.

#### 4.3.1.6 Konfigurasi Lsyncd

Pertama jalankan *daemon* dengan *command* `lsyncd -rsync /var/www/html/ /192.168.1.2:/var/www/html/`, untuk `/var/www/html/` itu adalah *source* dari direktori yang ingin dilakukan *mirror* sedangkan `/192.168.1.2:/var/www/html/` adalah *target* dan direktori mana akan diletakkannya hasil *mirror*.

#### 4.3.1.8 Install MySQL

*MySQL* merupakan sebuah *software management database* yang digunakan untuk menyimpan data-data dengan terstruktur, berikut langkah yang dilakukan untuk menginstall *MySQL* dengan menerapkan sistem *master slave* yang bertujuan untuk melakukan otomatisasi replikasi *database* tertentu sehingga dapat memaksimalkan *high availability* dalam *database*.

#### 4.3.1.9 Konfigurasi MySQL

Setelah install paket *mariaDB* lalu untuk menjalankannya masukan *command* `systemctl start mariadb` dan `systemctl enable mariadb`. Untuk mengamankan *database* setelah melakukan instalasi maka harus terlebih dahulu melakukan konfigurasi untuk mengganti *password* dengan *command* `mysql_secure_installation`.

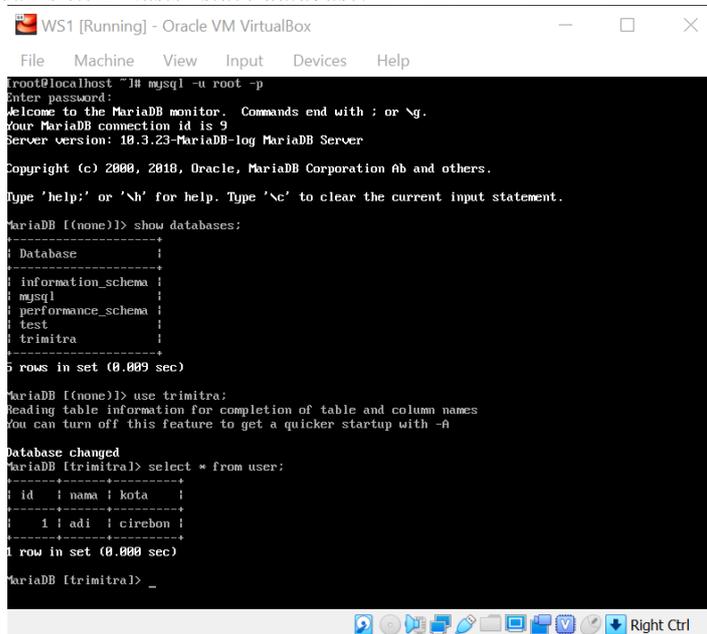
#### 4.3.1.10 Konfigurasi Load Balancing

*Load balance* dilakukan untuk membagi beban secara merata di antara *web server* yang terdaftar sebagai anggota *load balance*, konfigurasi ini dilakukan oleh *server load balance* dimana bertindak sebagai *front end* dan *web server* menjadi *back end*.

### 4.3.2. Pengujian

#### 4.3.2.1 Database Master dan Slave

Hasil dari konfigurasi yang pertama adalah dengan menampilkan sistem *master server* yang berjalan, *database* yang akan di *mirror* dinamakan dengan *trimitra* dengan tabel *user* didalamnya sebagai hasil dari sistem *master slave database*.



```
WS1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@localhost ~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.3.23-MariaDB-log MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
| trimitra |
+-----+
5 rows in set (0.009 sec)

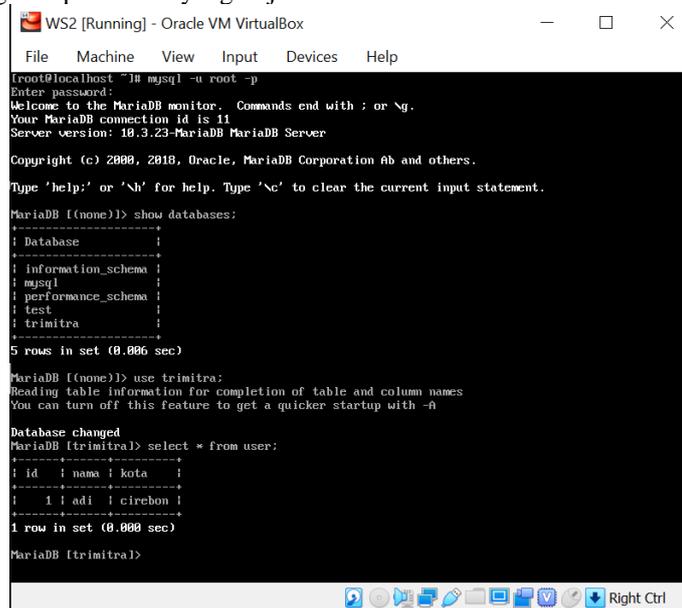
MariaDB [(none)]> use trimitra;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [trimitra]> select * from user;
+----+-----+-----+
| id | nama | kota |
+----+-----+-----+
| 1 | adi | cirebon |
+----+-----+-----+
1 row in set (0.000 sec)

MariaDB [trimitra]> _
```

Gambar 4. Database Web Server 1

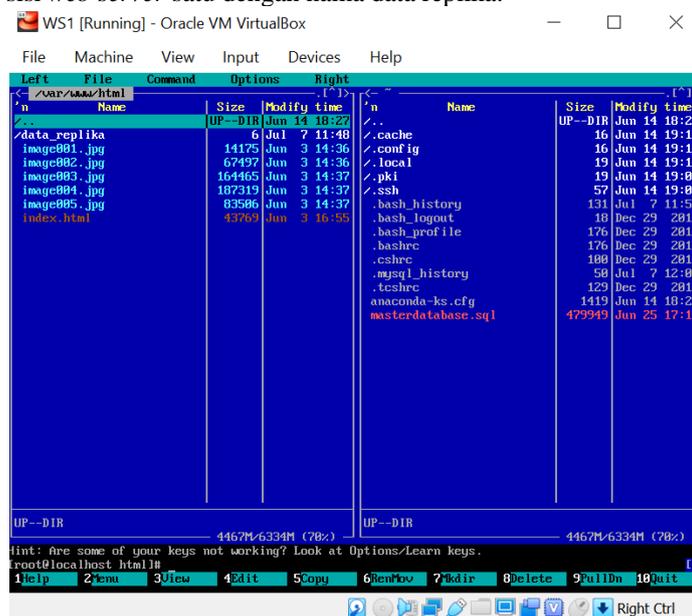
Dalam *database web server* kedua terdapat juga *database* trimitra dengan tabel *user* sebagai contoh dari hasil konfigurasi sistem *master slave database*. Isi dari tabel *user* akan otomatis berubah mengikuti perubahan yang terjadi dalam *database server master*.



Gambar 5. Database Web Server 2

4.3.2.2 Mirror Server

Lalu untuk hasil *mirror* konten *web server* dalam */var/www/html* dilakukan dengan pembuatan direktori dari sisi *web server* satu dengan nama data replika.



Gambar 6. Direktori Web Server 1

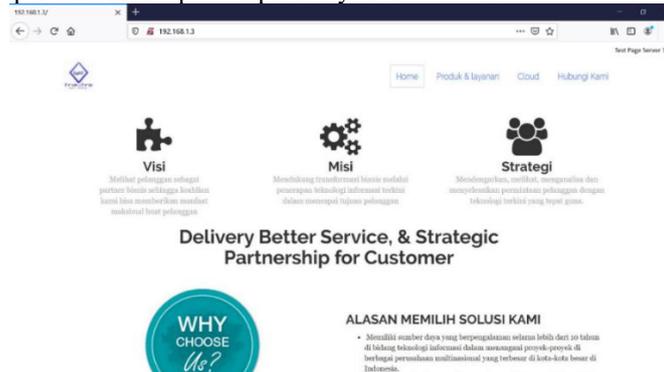
Direktori yang telah dibuat dalam */var/www/html web server* satu akan terbuat juga dalam *web server* dua dikarenakan adanya proses sinkronisasi yang terjadi dengan *interval* waktu tertentu dan *real time*. Perubahan dari pembuatan baru maupun penghapusan direktori hingga perubahan isi file dari *web server* satu akan selalu di cek oleh *web server* dua.



Gambar 7. Direktori Web Server 2

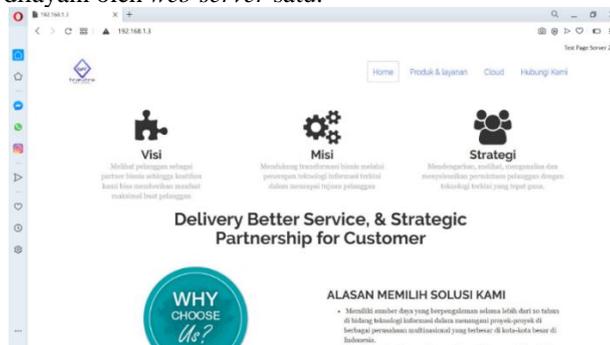
#### 4.3.2.3 Load Balance

Pengujian berikut sudah diterapkannya sistem *load balance*. *Load balance* dapat berjalan dengan baik dan terjadinya pembagian beban secara merata oleh anggota nya dalam hal ini merupakan *web server* yang menyediakan layanan *web*. Dapat terlihat pengaksesan layanan *web* melalui dua *browser*, *client* pertama mendapat tampilan layanan *web* dari *web server* satu.



Gambar 8. Layanan Web Server 1

Lalu *browser* kedua akan mendapat tampilan *web* dari *web server* dua karena sistem nya adalah membagi beban secara merata dan berurutan, maka saat *client* baru akan mengakses lagi selanjutnya akan dilayani oleh *web server* satu.



Gambar 9. Layanan Web Server 2

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari penelitian yang dilakukan dapat diuraikan beberapa kesimpulan :

1. Beban *server* dapat terbagi karena masing-masing *web server* mampu mengidentifikasi *client* yang membutuhkan layanan dengan memberi tanda dalam *header session* sehingga tiap *client* baru yang akan mengakses akan dilayanin sesuai dengan *route web server*.
2. *Load balance* dapat membantu memaksimalkan ketersediaan dalam layanan *web server* karena *load balance* mempunyai beberapa anggota *web server*.
3. Terbentuknya sistem *failover* karena konfigurasi sistem *load balance* yang dapat mengalihkan layanan *web server* jika salah satu dari anggota *load balance* mengalami masalah.

### 5.2. Saran

Adapun saran yang perlu menjadi pertimbangan untuk perkembangan penelitian selanjutnya yaitu Pengembangan kedepannya bisa dapat membandingkan performa jenis *load balance* yang ada pada penelitian ini dengan jenis *load balance* yang lain agar dapat mengetahui konfigurasi performa *server* yang lebih baik.

## DAFTAR PUSTAKA

- [1] D. Rahmana, R. Primananda, and W. Yahya, "Analisis Load Balancing Pada Web Server Menggunakan Algoritme Weighted Least Connection," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 3, pp. 915–920, 2017, doi: 10.1016/0028-3932(72)90008-5.
- [2] D. K. Hakim, D. Y. Yulianto, and A. Fauzan, "Pengujiian Algoritma Load Balancing pada Web Server Menggunakan NGINX," *JRST (Jurnal Ris. Sains dan Teknol.*, vol. 3, no. 2, p. 85, 2019, doi: 10.30595/jrst.v3i2.5165.
- [3] A. Rahmatulloh and F. MSN, "Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi," *J. Nas. Teknol. dan Sist. Inf.*, vol. 3, no. 2, pp. 241–248, 2017, doi: 10.25077/teknosi.v3i2.2017.241-248.
- [4] H. Yuliandoko, Ed., "Jaringan Komputer Wire dan Wireless," in *Jaringan Komputer*, 1st ed., DEEPUBLISH (Grup Penerbitan CV BUDI UTAMA), 2018, p. 4.
- [5] S. S. Mulyanta, "Pengenalan Protokol Jaringan Wireless Komputer," *Google Book. CV ANDI OFFSET*, p. 172, 2017.
- [6] E. Haryanto, "Jaringan Komputer TCP\_IP.pdf." CV ANDI OFFSET, p. 167, 2012.
- [7] E. B. Harjono, "Analisa Dan Implementasi Dalam Membangun Sistem Operasi Linux Menggunakan Metode LSF Dan REMASTER," *Informatika*, vol. I, no. 1, pp. 30–35, 2016.
- [8] Y. Swara, G. Y. Kom. M., & Pebriadi, "Rekayasa Perangkat Lunak Pemesanan Tiket Bioskop Berbasis Web," *J. TEKNOIF*, vol. 4, no. 2, pp. 27–39, 2016.
- [9] H. Maulana, "Analisis Dan Perancangan Sistem Replikasi Database Mysql Dengan Menggunakan Vmware Pada Sistem Operasi Open Source," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 1, no. 1, pp. 32–37, 2016, doi: 10.30743/infotekjar.v1i1.37.
- [10] Sumarna, "Perancangan Clustering Ujian Online Studi Kasus Bina Sarana Informatika," *J. Techno Nusa Mandiri*, vol. XII, no. 1, pp. 35–40, 2015.
- [11] S. Riskiono, "Implementasi Metode Load Balancing Dalam Mendukung Sistem Kluster Server," *Semnas Ristek*, pp. 455–460, 2018.