

# Analisa Dan Perancangan *Load Balancing Web Server* Menggunakan HAProxy

*Analysis And Design Of Web Server Load Balancing Using Haproxy*

Riska<sup>1</sup>, Hendri Alamsyah<sup>2</sup>

<sup>1,2</sup>Rekayasa Sistem Komputer, Universitas Dehasen Bengkulu

E-mail: <sup>1</sup>riska.iskandar@unived.ac.id, <sup>2</sup>hendri.alamsyah@unived.ac.id

## Abstrak

Penelitian ini bertujuan untuk merancang sistem *server load balancing web server* dan selanjutnya melakukan analisis masalah-masalah yang ada sebagai dasar perancangan *server* untuk mengatasi masalah yang ada pada *server* yang akan dibuat. Penelitian ini dilakukan dengan menggunakan pendekatan berlandaskan pada metode perancangan atau eksperimental, selanjutnya dilakukan analisa terhadap kondisi *server* sebelum dan sesudah dirancangnya *load balancing web server* serta aplikasi *website* yang akan diterapkan dengan mengimplementasikan *load balancing web server* menggunakan HAProxy. Hasil dari penelitian ini menunjukkan bahwa *load balancing web server* menggunakan HAProxy dapat meningkatkan performansi *server website* berdasarkan tingkat ketersediaan *server (uptime)* sebesar 99,49% dan rata – rata waktu klik sebesar 7,291ms per user serta konsistensi data pada *server website* Universitas Dehasen Bengkulu. Untuk menjaga konsistensi data, pada pengembangan *server* ini juga memanfaatkan fasilitas replikasi data. Dengan solusi ini tingkat ketersediaan *server* akan terjaga dan konsistensi data yang terjamin.

Kata kunci: *Load Balancing; Web Server; HAProxy*

## Abstract

*The research aims to design an intricate server system and then to do an analysis of existing problems as a basis for the design of the server to address any existing problems on the server that will be created. This research was conducted by using an approach based on the design or experimental methods, further analysis of the condition of the server before and after he designed a web server load balancing and web applications that will be applied to implement a web server load balancing using HAProxy. The results of this study indicate that web server load balancing using HAProxy can improve website server performance based on the server availability level (uptime) of 99.49% and an average click time of 7,291ms per user and data consistency on the Universitas Dehasen Bengkulu website server. To maintain data consistency, this server development also utilizes data replication facilities. With this solution, the level of server availability will be maintained and data consistency is guaranteed.*

Keywords: *Load balancing; Web Server; HAProxy*

## 1. PENDAHULUAN

Dalam pemanfaatan jaringan komputer, tidak terlepas dari adanya sebuah *server*. Terdapat banyak jenis *server*, salah satu diantaranya adalah *server* untuk *website* yang biasa disebut dengan *web server*. *Server* adalah komputer yang bertugas melayani semua permintaan yang dilakukan oleh komputer *client* selanjutnya hasil dari permintaan akan dikirimkan kembali kepada komputer *client*, sedangkan *Web server* adalah perangkat lunak yang menjadi tulang belakang dari *world wide web (www)*[1]. *Web server* melayani *request* data dari *client* yang yang mengakses alamat atau URL dari web tersebut menggunakan web *browser*. Dalam proses layanan dengan

komunikasi data menggunakan protokol HTTP / HTTPS antara web *server* dan *client* ini perlu diperhatikan dalam pemberian layanan untuk ketersediaan akses dan data, dimana *server* harus dapat diakses kapan saja [2]. Keadaan yang sering terjadi dalam mengakses website yang diterapkan pada *web server* adalah sering terjadinya *collaps/down* yang disebabkan banyaknya *client* yang mengaksesnya, apalagi saat *website* tersebut sering diakses oleh pengunjung yang tidak sedikit, contoh saat mahasiswa mencari pengumuman pada *website* suatu Universitas, ini menyebabkan akses terhadap *server* menjadi lambat dan dapat mengganggu proses pelayanan menggunakan media *website* tersebut, bahkan ada kalanya *server* tidak bisa diakses dan tentu saja hal ini sangat tidak diinginkan.

Web *server* Universitas Dehasen Bengkulu saat ini digunakan sebagai media berbagi informasi ataupun pengumuman baik untuk Mahasiswa ataupun Masyarakat secara umum, dimana berdasarkan hasil pengamatan yang dilakukan saat kondisi web Universitas Dehasen Bengkulu dalam kondisi full akses, web *server* hanya mampu melayani permintaan *request* dari *client* 150 sampai 200 request secara bersamaan. Keadaan ini dapat dilihat dari komunikasi yang mengalami time out yang disebabkan oleh padatnya lalu lintas data, ketika beban tersebut menghabiskan seluruh *resource* yang ada, maka aplikasi akan mengalami kegagalan atau *down*[3,4].

Dari permasalahan ini, dapat ditawarkan solusi dengan menerapkan metode *load balancing* terhadap web *server* yang ada di Universitas Dehasen Bengkulu, sehingga *server* yang dibangun dapat melakukan pembagian *request* ke beberapa *server* sekaligus untuk meringankan beban yang ditanggung oleh *server*, dengan kata lain membagi-bagi beban yang datang ke beberapa *server*, jadi yang melayani pengguna tidak hanya terpusat pada satu perangkat saja. Load balancing adalah 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[2,6]. Dalam perancangan *load balancing* ini, banyak *tools* yang dapat digunakan salah satunya adalah HAProxy. HAProxy merupakan singkatan dari *High Availability Proxy*, adalah sebuah *open source software* TCP / HTTP *load balancer* dan solusi proxy yang populer yang mendukung keperluan penyeimbang beban dan *failover* web *server*[6,7]. Selain itu, HAproxy juga memungkinkan adanya sistem *failover*, dimana, *failover* adalah proses perpindahan koneksi secara otomatis suatu *server* ketika *server* yang lainnya mengalami masalah atau kerusakan atau dapat juga dikatakan sebagai duplikasi dan *backup* data[7].

Pada penelitian [7] performansi MySQL *Cluster* dapat sedikit ditingkatkan dengan menggunakan metode *load balancing*, selain itu, penelitian yang dilakukan [2] menyarankan untuk menambahkan *clustering database* ataupun replikasi *database* untuk mengimbangi *performance* dari web *server* serta perlunya proses sinkronisasi yang *realtime*. Oleh sebab itu dalam penelitian ini, selain penerapan *load balancing*, untuk menjaga konsistensi data, pada pengembangan *server* ini juga memanfaatkan fasilitas replikasi data. Replikasi data 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, keuntungan dari penggunaan replikasi ini tergantung dari jenis replikasi tetapi pada umumnya replikasi mendukung ketersediaan data setiap waktu dan dimanapun diperlukan[9]. Untuk mengatasi kegagalan sistem database dapat dilakukan replikasi data pada database yang digunakan, teknik yang dapat digunakan adalah MySQL *cluster* atau disebut dengan *Network DataBase* (NDB), MySQL *cluster* memiliki tiga jenis node yaitu node penyimpanan, node pengelolaan, dan node MySQL *server*[11,12]. Dengan solusi ini tingkat ketersediaan *server* akan terjaga dan konsistensi data yang terjamin.

## 2. METODE PENELITIAN

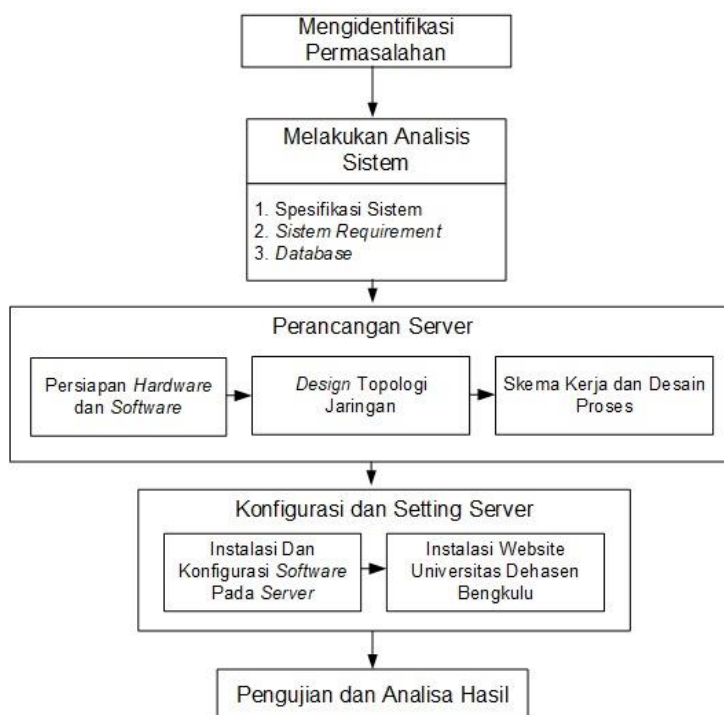
### 2.1 Pengumpulan Data

Teknik pengumpulan data merupakan cara-cara yang akan dilakukan untuk memperoleh data dan keterangan-keterangan yang diperlukan dalam penelitian[11]. Teknik pengumpulan data

pada penelitian ini dilakukan dengan dua cara, yaitu pertama melakukan observasi guna melihat kondisi *server* dan data yang berkaitan dengan penelitian. Kedua menanyakan langsung terkait dengan kondisi *server* dengan pegawai yang bertanggung jawab terhadap objek yang diteliti.

## 2.2 Rancangan Penelitian

Rancangan penelitian merupakan rencana penelitian yang disusun sedemikian rupa sehingga peneliti dapat memperoleh jawaban terhadap rumusan masalah[12]. Rancangan penelitian ini merupakan tahapan-tahapan yang dilakukan untuk menghasilkan *output* yang diinginkan. Adapun tahapan-tahapannya dalam penelitian ini seperti pada gambar 1.



Gambar 1 Rancangan Penelitian

Gambar 1 merupakan tahapan-tahapan yang dilakukan pada penelitian. Adapun penjelasan tahapan-tahapan pada gambar diatas meliputi:

1. Mengidentifikasi Permasalahan  
Identifikasi permasalahan dilakukan untuk menentukan masalah-masalah yang terjadi pada tempat penelitian, kemudian dirumuskan dan diberikan batasan-batasan permasalahan yang akan diteliti.
2. Melakukan Analisis Sistem  
Analisis dilakukan pada *server* yang sedang berjalan untuk mengetahui spesifikasi *server*, topologi jaringan yang berjalan, kelemahan sistem *server*. Selain itu juga dilakukan pada website Universitas Dehasen Bengkulu untuk mengetahui spesifikasi sistem, system requirement dan database yang digunakan. Hasil analisis akan digunakan untuk perancangan *server* yang akan dilakukan.
3. Perancangan *Server*  
Dalam perancangan *server* dibagi menjadi beberapa tahapan yaitu, persiapan hardware dan software, desain topologi jaringan, dan merancang skema kerja dan desain proses.
4. Konfigurasi dan Setting *Server*  
Konfigurasi dan setting *server* dilakukan setelah proses perancangan selesai, pada tahap ini website Universitas Dehasen Bengkulu diakses melalui internet. Website yang berada pada sistem *server* yang menerapkan metode load balancing web *server* menggunakan HAProxy

harus berfungsi dengan baik sesuai dengan rencana perancangan. Adapun pada tahap ini dilakukan beberapa proses yaitu, instalasi dan konfigurasi software pada *server*, Instalasi website Universitas Dehasen Bengkulu pada *server*, dan Konfigurasi *load balancing* web *server* dengan HAproxy dan mysql database *replication*

#### 5. Pengujian Hasil

Pengujian dilakukan untuk memastikan bahwa optimasi yang dilakukan melalui perancangan *server* ini berjalan dengan baik sesuai dengan yang diharapkan. Parameter pengujian hasil perancangan *server* ini adalah tingkat performansi, proses failover dari *server* utama ke *server* cadangan atau sebaliknya dan konsistensi data pada masing-masing *server*. Adapun rencana pengujian dilakukan dengan cara sesuai dengan parameter yang diuji yaitu:

##### a. *Performansi*

Tingkat performansi diuji menggunakan web *server* stress tool dan HAProxy Statistik. Pengujian ini dilakukan dengan cara yaitu dengan mendaftarkan alamat url atau alamat domain website Universitas Dehasen Bengkulu ke <http://StatusCake.com> dan software web *server* stress tool yang merupakan aplikasi untuk menguji ketahanan *server*

##### b. Proses *Failover*

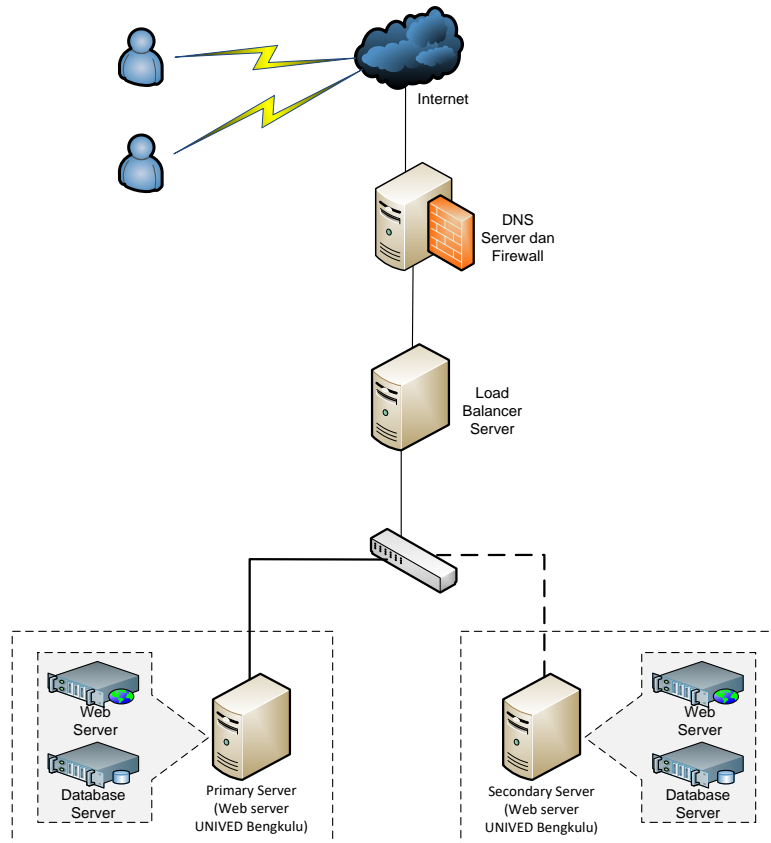
Proses *failover* diuji dengan membuat skenario yaitu dengan cara membuat *server* utama down, dan mengamati waktu perubahan pada tiap *server* yang berada dalam jaringan dengan *load balancing* web *server*.

##### c. Konsistensi Data

Konsistensi data diuji dengan mencoba melakukan *query* (penambahan data, perubahan data, penghapusan data) pada *server* utama kemudian mengamati data pada *backup server* dan sebaliknya.

#### 2.3 Perancangan Topologi

Perancangan topologi ini dibuat sebagai desain atau gambar perancangan interkoneksi pada *server* website Universitas Dehasen Bengkulu yang akan dirancang. Topologi sistem terdiri dari 4 buah *server* yaitu *DNS Server*, *load balancer server*, *Primary Server*, dan *Scondary Server*. Rancangan topologi terlihat pada Gambar 2.



Gambar 2 Rancangan Toplogi

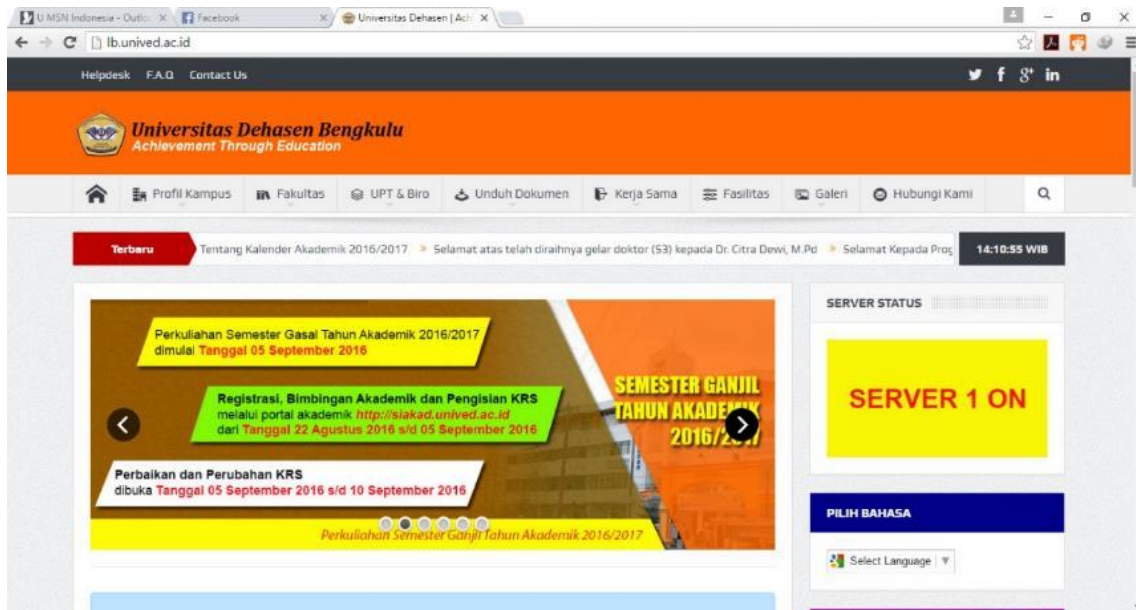
Saat keadaan sistem berjalan normal, *server load balancer* yang melayani *service* untuk *user* akan membagi beban pada *primary server* dan *secondary server*, sehingga beban menjadi merata dan tidak ditujukan hanya pada satu *server* saja. *User* akan mengakses *web server* ini melalui jaringan dengan *DNS server* dan kemudian dapat diakses melalui internet. *Primary server* akan terus melakukan *redudansi* ke *secondary server* dan sebaliknya setiap ada perubahan pada *server* yang sedang berjalan untuk menjamin konsistensi data pada kedua *server*.

### 3. HASIL DAN PEMBAHASAN

Berdasarkan hasil penelitian yang dilakukan dengan tiga parameter pengujian yaitu Performansi, waktu failover, dan konsistensi data pada kedua server. Pengujian penelitian ini dilakukan dengan menggunakan beberapa tool yaitu Pktstat, Web Stress tools, StatusCake.com, Axence Net Tools dan Putty.

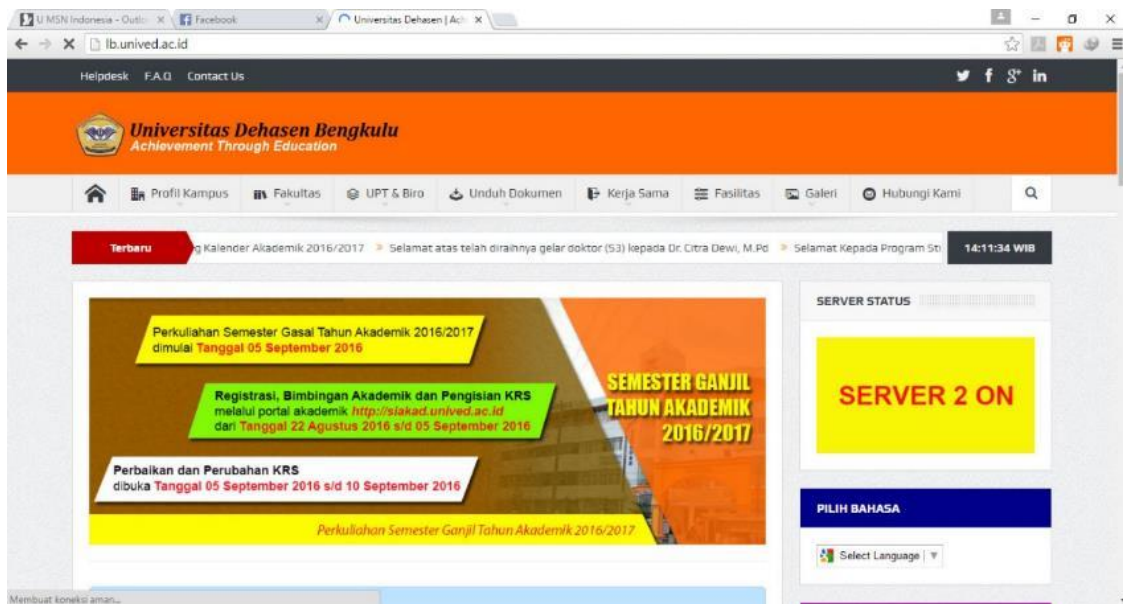
#### 3.1 Performansi Web Server

Pada tahap ini pengujian dilakukan dengan membuat skenario dengan mengakses server dalam kondisi normal. Tampilan utama website Universitas Dehasen Bengkulu terlihat pada Gambar 3.



Gambar 3 Halaman Utama Website UNIVED (Server 1)

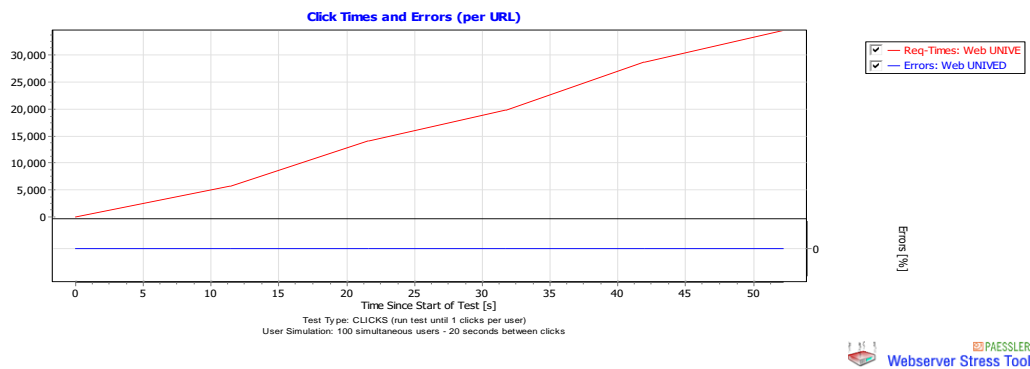
Dalam kondisi normal, url <http://lb.unived.ac.id> dilayani oleh kedua *server* yang berada dalam jaringan *load balancing web server*. Layanan akses *server* tersebut di kontrol atau diatur oleh *load balancer web server (server-lb)*, untuk akses tersebut akan dibagi secara merata pada kedua *server load balancing*, sehingga tampilan yang diperoleh saat mengakses website Universitas Dehasen Bengkulu akan tampil secara random. Saat *primary server (server 1)* sedang melayani koneksi dari user, saat user lain mengakses website, maka akses website akan dilayani oleh *secondary server (server 2)*, seperti terlihat pada Gambar 4.



Gambar 4 Halaman Utama Website (Server 2)

### 3.1.1 Pengujian Performansi Web Server dengan Web Stress Tool

Untuk mengukur tingkat performansi web *server*, dapat diuji dengan mendaftarkan url website Universitas Dehasen Bengkulu (sistem lama) dan setelah perancangan *load balancing web server* pada aplikasi web stress tools. Tingkat performansi *server* website Universitas Dehasen Bengkulu (sistem lama) dapat dilihat pada Gambar 5.



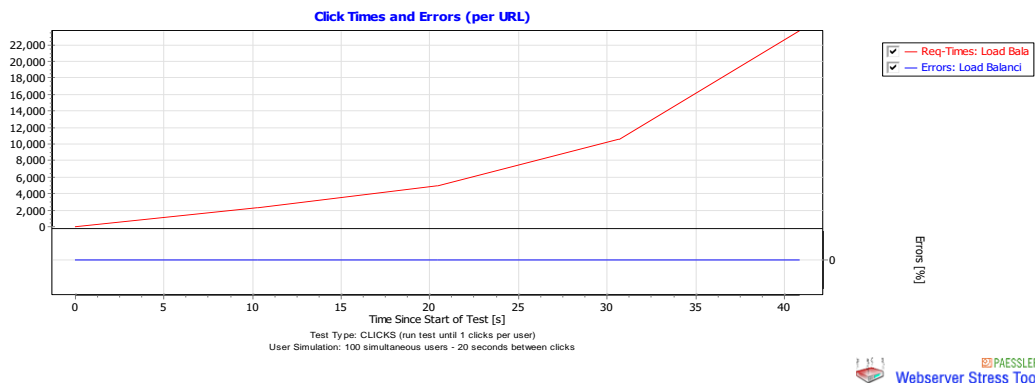
Gambar 5 Pengujian Performansi Server (Sistem Lama)

Untuk lebih jelas lagi pada pengujian dengan *web stress tool*, dapat dilihat pada tabel 1.

Tabel 1 Report Per URL (Sistem Lama)

URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1.	Web UNIVED	64	0	0.00	1,441,936	22,530

Dari tabel 1 terlihat nilai *time spent* sebesar 1,441,936 ms dan *Avarage click time* sebesar 22,530 ms. Nilai tersebut didapatkan dari hasil pengujian performansi menggunakan *web stress tool*. Sedangkan Tingkat performansi *server* website Universitas Dehasen Bengkulu setelah perancangan *load balancing web server* dapat dilihat pada gambar 6.



Gambar 2 Pengujian Performansi *Server Load Balancing Web Server*

Untuk lebih jelas lagi pada pengujian dengan *web stress tool*, dapat dilihat pada tabel 2.

Tabel 2 Report Per URL (Dengan Load Balancing)

URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1.	Load Balancing Web UNIVED	98	0	0.00	714,521	7,291

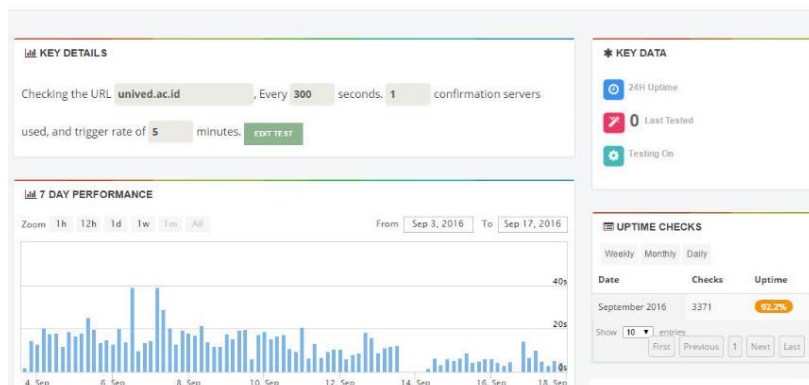
Dari tabel 2 terlihat nilai *time spent* sebesar 714,521 ms dan *Avarage click time* sebesar 7,291 ms. Nilai tersebut didapatkan dari hasil pengujian performansi menggunakan *web stress tool*.

### 3.1.2 Pengujian Performansi Web Server dengan StatusCake.com

Selain menggunakan *web stress tool*, pengujian untuk mengukur tingkat performansi web server juga diuji dengan mendaftarkan url website Universitas Dehasen Bengkulu (sistem lama) dan setelah perancangan *load balancing web server* pada website <http://StatusCake.com>. Yang

mana tingkat performansi dari web server ini di ukur berdasarkan tingkat uptime dari server tersebut yang dinyatakan dalam bentuk persentase. Adapun Tingkat performansi server website Universitas Dehasen Bengkulu (sistem lama) dapat dilihat pada gambar 7.

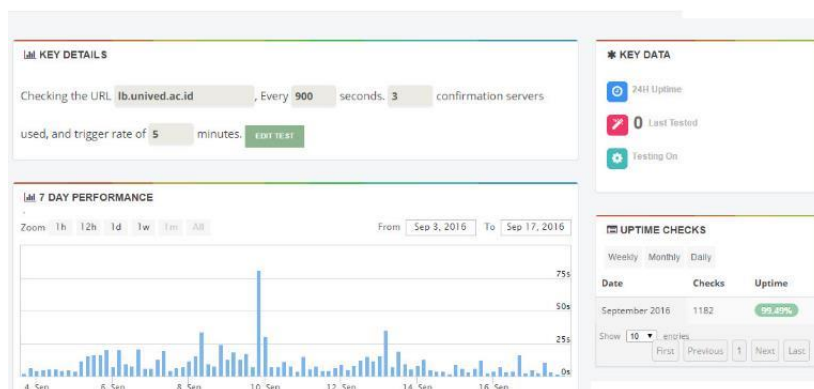
Status Listing For Web Server UNIVED Tanpa Load Balancing



Gambar 3 Pengujian Uptime Server (Sistem Lama)

Pada gambar 7 diatas, terlihat nilai *uptime* sebesar 92,2%. Nilai tersebut didapatkan dari total *check* sebanyak 3371 kali dalam 15 hari monitoring. Sedangkan Tingkat *performansi server* website Universitas Dehasen Bengkulu setelah perancangan *load balancing web server* dapat dilihat pada gambar 8.

Status Listing For Web Server UNIVED dengan load Balancing



Gambar 4 Pengujian Uptime Server Load Balancing Web Server

Dari gambar 8 diatas, terlihat nilai *uptime* sebesar 99,49%. Nilai tersebut didapatkan dari total *check* sebanyak 1181 kali dalam 15 hari monitoring.

### 3.2 Pengujian Load Balancing

Untuk pengujian *load balancing* ini dilakukan dengan me-monitoring *server load balancer (server-lb)*. Pengujian ini dilakukan dengan mengetikkan perintah `#pktstat -i eth0 -nt` pada *server load balancer*. Perintah ini digunakan untuk menampilkan daftar *real-time* koneksi aktif yang terlihat pada antarmuka jaringan, dan berapa banyak *bandwidth* yang sedang digunakan. Dengan menggunakan perintah ini juga akan terlihat proses *load balancing* yang sedang terjadi. Dalam pengujian ini akan dilakukan skenario dengan dua *user* yang sedang mengakses website Universitas Dehasen Bengkulu. Adapun pengujian yang dilakukan dapat dilihat pada gambar 9 dan gambar 10.



```

root@server-lb: ~
interface: eth0
bps

```

bps	%	desc
2.0k	0%	tcp 192.168.1.200:22 <-> 223.255.231.146:27432
171.4	0%	tcp 192.168.1.200:37142 <-> 192.168.1.201:80
171.4	0%	tcp 192.168.1.200:40818 <-> 192.168.1.202:80
171.4	0%	tcp 192.168.1.200:37144 <-> 192.168.1.201:80
171.4	0%	tcp 192.168.1.200:40820 <-> 192.168.1.202:80
171.4	0%	tcp 192.168.1.200:37146 <-> 192.168.1.201:80
171.4	0%	tcp 192.168.1.200:40822 <-> 192.168.1.202:80
		tcp 192.168.1.200:40808 <-> 192.168.1.202:80
		- 200 GET /wp-content/themes/limolas/images/nav-shadow.png
		tcp 114.125.44.233:56834 <-> 192.168.1.200:80
		- 200 GET /wp-content/themes/limolas/images/nav-shadow.png
		tcp 192.168.1.200:37132 <-> 192.168.1.201:80
		- 200 GET /wp-content/themes/limolas/images/nb-dots.png
		tcp 114.125.44.233:56832 <-> 192.168.1.200:80
		- 200 GET /wp-content/themes/limolas/images/nb-dots.png
		tcp 192.168.1.200:40812 <-> 192.168.1.202:80
		- 304 GET /wp-content/plugins/revslider/rs-plugin/assets/large_righ
		tcp 192.168.1.200:40810 <-> 192.168.1.202:80
		- 304 GET /wp-content/plugins/revslider/rs-plugin/assets/bullet.png

Gambar 5 Pengujian Load Balancing User 1

Pada gambar 9 diatas, dapat dilihat IP Address yang yang digunakan oleh user 1 yaitu 114.125.44.233 yang mengakses IP address dari server load balancer yaitu 192.168.1.200:80. Kemudian load balancer akan meneruskan (redirect) permintaan koneksi tersebut ke primary server dengan IP address 192.168.1.201:80 dan secondary server dengan IP address 192.168.1.202. Proses tersebut akan terus berulang hingga user tersebut berhenti mengakses website Universitas Dehasen Bengkulu.

```

root@server-lb: ~
interface: eth0
bps

```

bps	%	desc
12.3k	0%	tcp 192.168.1.200:80 <-> 223.255.231.146:22409
3.2k	0%	tcp 192.168.1.200:22 <-> 223.255.231.146:27432
208.7	0%	tcp 192.168.1.200:37339 <-> 192.168.1.201:80
208.7	0%	tcp 192.168.1.200:41015 <-> 192.168.1.202:80
208.7	0%	tcp 192.168.1.200:37341 <-> 192.168.1.201:80
208.7	0%	tcp 192.168.1.200:41017 <-> 192.168.1.202:80
196.8	0%	tcp 192.168.1.200:37336 <-> 192.168.1.201:80
		- 200 GET /wp-content/uploads/2014/11/logo_unived.png
		tcp 192.168.1.200:80 <-> 223.255.231.146:22400
		- 200 GET /wp-includes/js/wp-emoji-release.min.js?ver=4.5.3
		tcp 192.168.1.200:80 <-> 223.255.231.146:22399
		- 200 GET /wp-content/plugins/js_composer/assets/js/js_composer_fro
		tcp 192.168.1.200:41010 <-> 192.168.1.202:80
		- 304 GET /wp-content/plugins/revslider/rs-plugin/assets/large_left
		tcp 192.168.1.200:41008 <-> 192.168.1.202:80
		- 304 GET /wp-content/plugins/revslider/rs-plugin/assets/bullet.png
		tcp 192.168.1.200:37334 <-> 192.168.1.201:80
		- 304 GET /wp-content/plugins/revslider/rs-plugin/assets/large_righ

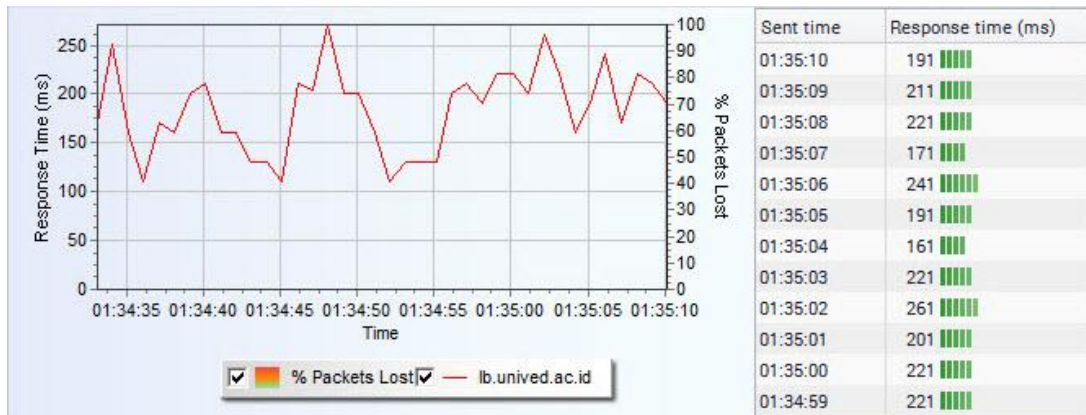
Gambar 6 Pengujian Load Balancing User 2

Pada gambar 10 diatas, dapat dilihat IP address yang yang digunakan oleh user 2 yaitu 223.255.231.146 yang mengakses IP address dari server load balancer yaitu 192.168.1.200:80. Kemudian load balancer akan meneruskan (redirect) permintaan koneksi tersebut ke secondary server dengan IP address 192.168.1.202:80 dan primary server dengan IP address 192.168.1.201. Proses tersebut akan terus berulang hingga user tersebut berhenti mengakses website Universitas Dehasen Bengkulu.

### 3.3 Waktu Failover

Pengujian untuk mengukur waktu failover dari server load balancing web server menggunakan metode Round Robin dengan HAProxy adalah dengan mengamati trafik service HTTP website Universitas Dehasen Bengkulu. Monitoring trafik dilakukan menggunakan tool

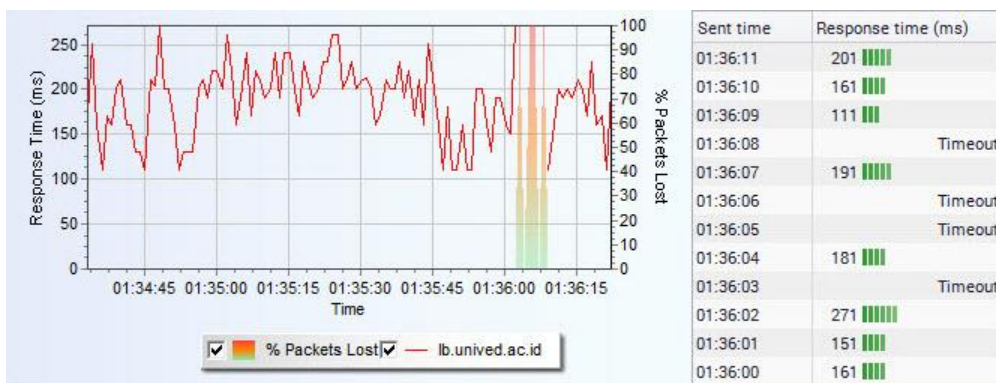
Axence Net Tool. Adapun hasil pengujian dengan melakukan ping ke *url* website Universitas Dehasen Bengkulu dalam kondisi normal terlihat pada gambar 11.



Gambar 7 Trafik service HTTP (Normal)

Dari gambar di atas, terlihat respon time yang stabil dari URL <http://lb.unived.ac.id>.

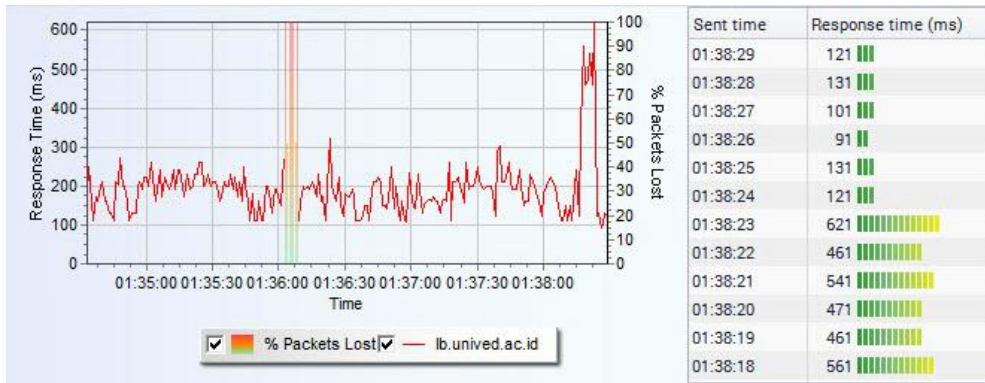
Selanjutnya pengujian dengan melakukan *ping* ke *url* website Universitas Dehasen Bengkulu dalam kondisi *primary server down*. Untuk lebih jelasnya dapat dilihat pada gambar 12.



Gambar 8 Trafik service HTTP (Primary Server Down)

Dari pengujian proses *failover* diatas, proses yang terjadi sangat cepat, di mana saat proses *failover* ataupun *failback* hanya terjadi *timeout* selama 4 detik yang berselang. Lamanya proses *failover* juga dipengaruhi oleh kualitas koneksi dari kedua *server*.

Setelah melakukan pengujian pada kondisi *primary server down*, dilanjutkan dengan mengamati trafik *service HTTP* website Universitas Dehasen Bengkulu saat *primary server* kembali normal. Adapun proses yang terjadi dapat dilihat pada gambar 13.



Gambar 9 Trafik service HTTP (Kembali Normal)

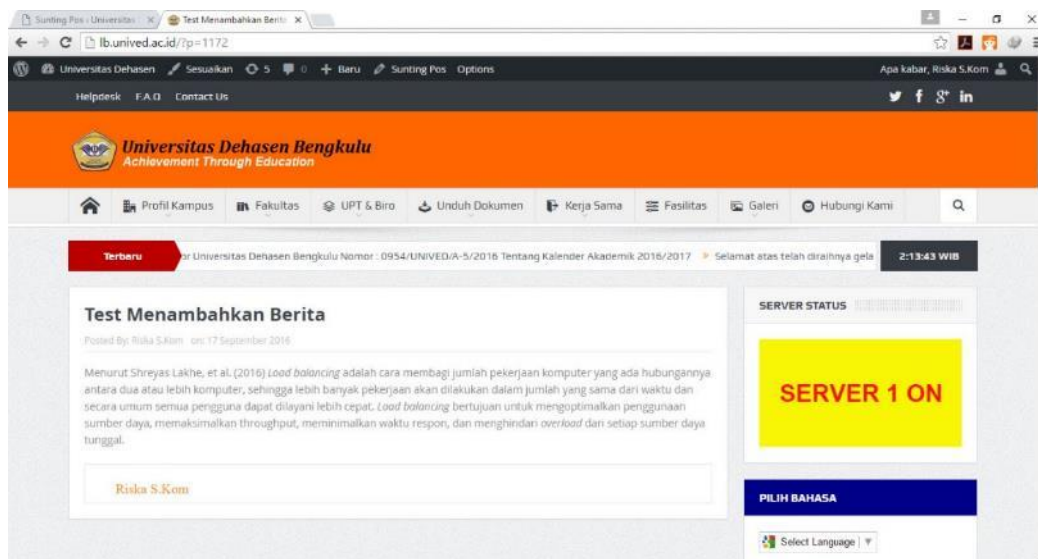
Pada gambar 13 diatas, tidak terlihat adanya timeout dari proses *monitoring* trafik *service website* Universitas Dehasen Bengkulu saat *primary server* kembali normal.

### 3.4 Konsistensi Data

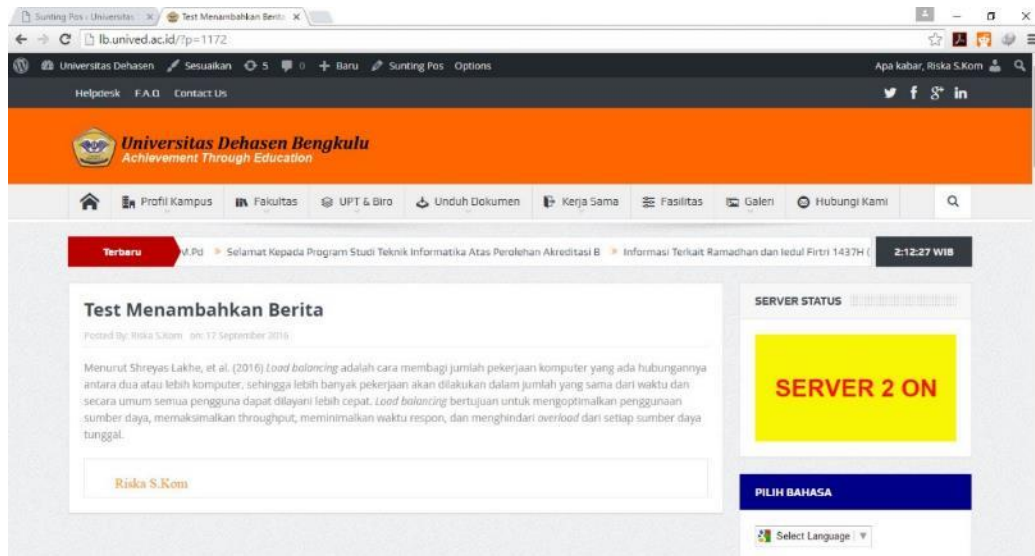
Pengujian konsistensi data pada *load balancing web server* dilakukan dengan menambahkan konten pada kondisi *server* berjalan normal dan menambahkan konten pada kondisi *primary server crash* atau *down*. Adapun pengujian konsistensi data pada penelitian ini dibagi menjadi beberapa skenario pengujian berikut ini.

#### 3.4.1 Skenario 1

Melakukan penambahan konten pada website Universitas Dehasen Bengkulu pada kondisi normal. Dari hasil pengujian pada skenario I, terlihat konten yang ada pada *secondary server* sama dengan konten yang ada pada *primary server*. Untuk lebih jelasnya dapat dilihat pada gambar 5.21 dan gambar 14.



Gambar 10 Konten pada Primary Server



Gambar 11 Konten pada Secondary Server

Dari gambar 14 dan gambar 15 menunjukkan bahwa konten yang ada pada *primary server* dan *secondary server* adalah sama.

### 3.4.2 Skenario II

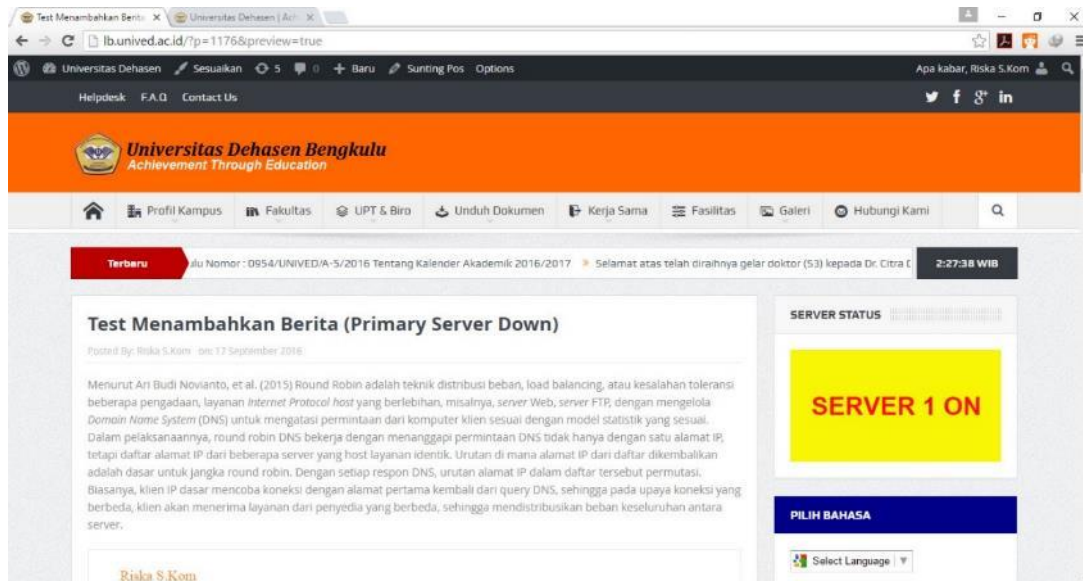
Skenario ini dilakukan pada saat kondisi *primary server down*, dan *request* aplikasi hanya dilayani oleh *secondary server backup*. Pada saat *primary server down* dilakukan penambahan konten pada website. Adapun konten yang ditambahkan dapat dilihat pada Gambar 16.



Gambar 12 Konten Saat Primary Server Down

Kemudian dalam waktu 15 menit *primary server* diaktifkan sehingga *request* aplikasi dilayani kembali oleh *primary server* dan *secondary server*. Selanjutnya dilakukan pengamatan pada *website* Universitas Dehasen Bengkulu, hasil pengamatan penulis menunjukkan konten di *primary server* juga telah berubah seperti pada *secondary server*. Adapun konten setelah *primary server* aktif kembali terlihat pada gambar 17.





Gambar 13 Konten Setelah Primary Server Aktif

#### 4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang dilakukan pada perancangan *load balancing web server* menggunakan HAProxy dengan, maka ditarik kesimpulan *Load balancing web server* menjadi salah satu alternatif untuk mengatasi masalah beban trafik terhadap *website* Universitas Dehasen Bengkulu, sebab dengan menggunakan teknik ini dapat membagi beban secara merata pada masing–masing *server* yang terhubung dalam jaringan *server load balancing*. Penerapan *load balancing web server* dapat meningkatkan ketersediaan pada *server (uptime)* pada *website* Universitas Dehasen Bengkulu hingga 99,49% dan rata – rata waktu klik sebesar 7,291 ms. Sistem dapat menjalankan fungsinya dengan baik berdasarkan waktu *failover* yang terjadi saat *primary server* mengalami *crash/down* terjadi cukup cepat sehingga *user* tidak merasa jika terjadi *downtime*. Serta penerapan *mysql database replication* dapat mempertahankan konsistensi data pada kedua *server* sehingga data pada *primary server* sama dengan data pada *secondary server*.

Untuk penelitian selanjutnya dapat menerapkan *load balancing web server* ini menggunakan mesin di lokasi yang berbeda untuk menjaga kestabilan dari *server load balancing*, misalnya menggunakan *virtual private server (VPS)* untuk beberapa *server* yang akan digunakan.

#### DAFTAR PUSTAKA

- [1] E. Fernando, “Arsitektur teknologi webserver berbasis mini pc dengan raspberry pi,” *Akademika*, vol. 9, no. August, pp. 281–285, 2016.
- [2] 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.
- [3] D. B. Prasetyo, R. I. Miftah, and R. I. Perwira, “Implementasi Network Notification System Dengan Menggunakan Teknologi Firebase Cloud Messaging (Fcm) Berbasis Android,” *Telematika*, vol. 16, no. 2, p. 62, 2019.
- [4] M. A. Nugroho and R. Kartadie, “Analisis Kinerja Penerapan Container untuk Load Balancing Web Server,” *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 1, no. 02, pp. 7–15, 2016.
- [5] A. Noviyanto, Ari Budi; Kumalasari N, Erna; Hamzah, “Perancangan Dan Implementasi Load Balancing Reverse Proxy Menggunakan Haproxy Pada Aplikasi Web,” *J. JARKOM*, vol. 2, no. 2, pp. 32–43, 2015.
- [6] C. R. Shreyas Lakhe, Ashutosh Shinde, Nandan Sukthankar, “Server Load Balancing

- Using Haproxy,” *J. information, Knowl. Res. Comput. Eng.*, pp. 810–817, 2016.
- [7] T. A. Gani, A. Arafat, and . Melinda, “Analisis Kinerja MySQL Cluster Menggunakan Metode Load Balancing,” *J. Rekayasa Elektr.*, vol. 11, no. 4, p. 129, 2015.
- [8] Suryanto, “Implementasi Clustering Database Server Menggunakan Pgcluster Untuk Optimalisasi Kinerja Sistem Basis Data,” *J. Tek. Komput. AMIK BSI*, vol. 1, no. 1, pp. 134–143, 2015.
- [9] R. P. F. Isra, Wanda Ichsanul; Abidin, Taufik Fuadi; Afidh, “Analisa Keefektifan Mysql Cluster Dan Non-Cluster Dalam Memproses Data Akademik,” in *Seminar Nasional Sistem Informasi Indonesia*, 2016, no. November, pp. 3–8.
- [10] I. Putri Andhikha, “Implementasi Dan Analisis Kinerja Mysql Cluster Menggunakan Metode Load Balancing,” *J. Manaj. Inform.*, vol. 8, no. 1, 2017.
- [11] A. Fransisca and H. Wijoyo, “Implementasi Mettā Sutta Terhadap Metode Pembelajaran Di Kelas Virya Sekolah Minggu Sariputta Buddhies,” *J. Ilmu Agama dan Pendidik. Agama Buddha*, vol. 2, no. 1, pp. 1–12, 2020.
- [12] S. N. Siti Fatimah, “Gambaran Pengetahuan Ibu Hamil Tentang Risiko 4T Desa Jahiang Kecamatan Salawu Kabupaten Tasikmalaya,” *J. Kesehat. Bakti Tunas Husada J. Ilmu Ilmu Keperawatan, Anal. Kesehat. dan Farm.*, vol. 20, no. 1, pp. 107–111, 2020.