

## IMPLEMENTASI FAILOVER DATABASE MENGUNAKAN HEARTBEATS

**Antika Sari<sup>1\*</sup>, Toibah Umi Kalsum<sup>2</sup>, Riska<sup>3</sup>**

Program Studi Rekayasa Sistem Komputer, Fakultas Ilmu Komputer,  
Universitas Dehasen Bengkulu<sup>1,2,3</sup>

*antikasari0599@gmail.com<sup>1\*</sup>, cicik.umie@gmail.com<sup>2</sup>*

### ABSTRACT

*For implement this high availability server, you can use special application software for clustering, such as heartbeat. This heartbeat is part of linux high availability project. Using heartbeat, you can do failover by promoting an active server to serve requests from clients and replace the function of the main server. This is done to ensure that the availability of existing data on the database server is always available when there is a request by the client. The purpose of this research is to build a database server that is always available by implementing a failover server and implementing data replication system on the database server. Database failover technology using heartbeats and using master to master replication system, data availability and system performance increase because the data has a backup. Failover database using heartbeats using master to master replication system has advantages (can do data backup automatically) and disadvantages (requires additional devices), so using this technology needs to be adjusted to the needs of system development.*

**Keywords:** *Failover Data Base, Heartbeats,*

### ABSTRAK

Untuk menerapkan *high availability* server ini dapat menggunakan software aplikasi khusus untuk clustering seperti heartbeat. Heartbeat ini Merupakan bagian dari proyek linux high availability. Dengan menggunakan heartbeat maka dapat melakukan failover dengan mempromosikan server aktif untuk melayani request dari client dan menggantikan fungsi dari server utama. Hal ini dilakukan untuk menjaga agar ketersediaan data yang ada pada server data base selalu tersedia saat ada request oleh client. Tujuan penelitian ini untuk membangun Server database yang selalu tersedia dengan menerapkan failover server dan menerapkan sistem replikasi data pada server database. Dengan teknologi failover database menggunakan heartbeats dengan menggunakan sistem replikasi master to master maka ketersediaan data dan kinerja sistem meningkat karena data telah memiliki cadangan. Failover database menggunakan heartbeats dengan menggunakan sistem replikasi master to master memiliki kelebihan (dapat melakukan backup data secara otomatis) serta kekurangan (membutuhkan perangkat tambahan), sehingga untuk menggunakan teknologi ini perlu disesuaikan dengan kebutuhan pengembangan sistem.

**Kata kunci :** *Failover Database, Heartbeats,*

### 1. Pendahuluan

Pada era perkembangan teknologi informasi yang sangat pesat, ketersediaan akses data dan informasi merupakan bagian terpenting saat ini. Untuk menunjang ketersediaan akses data dan informasi dibutuhkan koneksi internet yang selalu aktif dan juga server dari pengelola data dan informasi yang selalu online setiap saat. Namun banyak faktor yang dapat menyebabkan server ini tidak tersedia setiap saat, seperti kegagalan fungsi Hardware dan software

klaster dengan server utama. Tujuan failover adalah agar pengguna tetap dapat mengakses ke server walaupun server sedang mengalami kegagalan software, hardware atau lainnya yang mengakibatkan server tersebut tidak dapat diakses.

Untuk menerapkan *high availability* server ini dapat menggunakan software aplikasi khusus untuk clustering seperti heartbeat. Heartbeat ini Merupakan bagian dari proyek linux high availability. Dengan menggunakan heartbeat maka dapat melakukan failover dengan

mempromosikan server aktif untuk melayani request dari client dan menggantikan fungsi dari server utama. Hal ini dilakukan untuk menjaga agar ketersediaan data yang ada pada server data base selalu tersedia saat ada request oleh client. Secara sederhana, cara kerja failover adalah jika server master mengalami kegagalan maka secara otomatis server backup akan mengambil alih tugas server master. Dengan kata lain implementasi failover ini dapat menghindari single point of failure, dimana single point of failure adalah sebuah komponen penting dalam sistem yang dimana jika komponen ini berhenti maka keseluruhan sistem akan berhenti pula.

Dari uraian latar belakang di atas, penulis tertarik untuk mengangkat judul penelitian “Implementasi **failover Data Base Menggunakan Heartbeats**”

## 2. Tinjauan Pustaka dan Pengembangan Hipotesis

### A. High availability

Menurut Pratama (2021:5573), *High availability* dalam teknologi informasi dapat diartikan sebagai sebuah sistem atau komponen yang dapat beroperasi secara terus menerus dalam jangka waktu yang lama tanpa mengalami gangguan. Pada dasarnya, high availability mengimplementasikan satu atau lebih server backup dalam mode standby, yang kemudian dapat menjadi siaga dalam beberapa saat ketika ditemukan kegagalan dalam server master.

Menurut Purnomo dan Edi (2017:2), High Availability merupakan server berbeda berkerja bersama-sama untuk memastikan downtime pada resource dikurangi seminimal mungkin. Tujuan dari High Availability Cluster adalah untuk memastikan bahwa sumber daya mencapai Availability semaksimal mungkin, Jika server down atau jika sumber daya yang berhenti, HA cluster akan memonitor dan memastikan resource atau sumber daya dihidupkan ulang pada tempat lain dalam sistem cluster, sehingga dapat digunakan lagi setelah mengalami gangguan. Konsep tersebut berkaitan dengan kemampuan sistem untuk mengatasi terjadinya gangguan, kerusakan hardware, crash/down, kesalahan jaringan bahkan kegagalan server yang di sebabkan software yang gagal melakukan tugas semestinya. Solusi yang ditawarkan berupa backup data atau failover data yang dilakukan secara real time. Saat server utama berhenti berjalan, maka server slave akan mengambil alih peran server utama dengan kualitas penanganan input dan output yang sama dengan server utama. Sistem akan selalu melakukan sinkronisasi data diantara keduanya atau mungkin lebih untuk mendapatkan redundancy data.

Pengujian *availability* atau ketersediaan tinggi terhadap *server* dapat dilakukan dengan menggunakan rumus berikut Almightly Ryandreut. Berikut.

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \times \frac{X}{100} \dots\dots\dots(1)$$

Dimana,

- a. *Mean Time Between Faults* (MTBF) merupakan waktu rata-rata *uptime*.
- b. *Mean Time to Repair* (MTTR) merupakan rata-rata waktu yang diperlukan untuk mengembalikan layanan setelah mengalami *serverdown* sampai *server backup* menangani dan memberikan layanan kembali.

### B. Failover

Menurut Pratama (2021:5573), *Failover* merupakan sebuah teknik untuk menghindari kegagalan pada sistem. Dengan implementasi *failover* pada sistem, sistem akan terus berjalan untuk melayani pengguna karena *failover* memindahkan *traffic* dari komponen yang mengalami kegagalan ke komponen *backup*. Proses *failover* dapat dirancang secepat mungkin setelah saat terjadi kegagalan. Didalam *failover* terdapat proses *failback*, *failback* adalah proses kembalinya komponen yang sebelumnya mengalami kegagalan. Terdapat dua model dari *failover*, yaitu *failover active-active* dan *failover active-standby*.

Menurut Purnomo dan Edi (2017:2), *Failover* merupakan sistem komunikasi dua atau lebih *server* ditempat yang berbeda yang dapat saling *backup* data dan mampu menggantikan pelayanan bila *server* lain down. *Failover* bertujuan untuk membantu menjaga akses client ke sumber daya di *server*, ketika *server* mengalami kegagalan

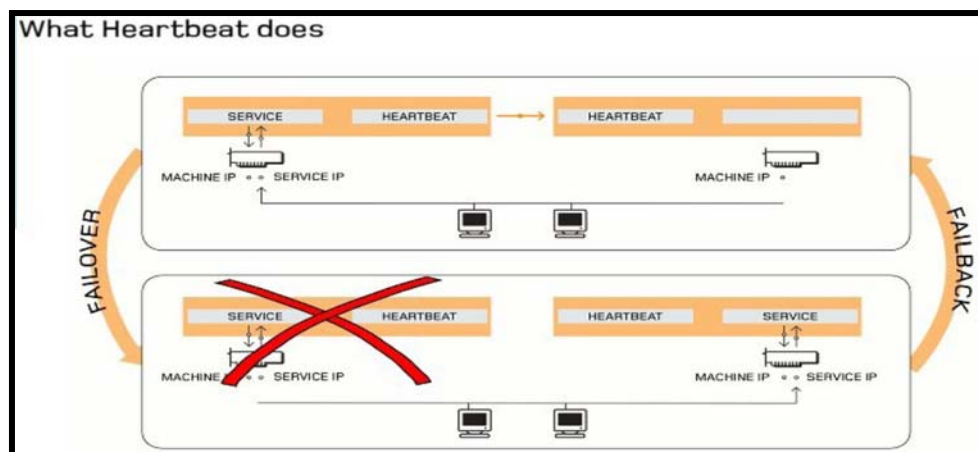
fungsi *software*, atau kegagalan akses *server*. *Failover cluster* merupakan sekumpulan *server* yang saling bekerjasama untuk memberikan pelayanan meskipun berada ditempat yang berbeda, dan memiliki kualitas data atau sumberdaya yang sama antara *server* yang satu dengan *server* lainnya. Sistem *Failover* akan bekerja untuk menghubungkan *server-server* yang menjadi anggota *cluster*-nya untuk mengambil alih tugas *server* utama saat terjadi kegagalan fungsi dalam waktu tertentu

### C. Heartbeat

Menurut Pratama (2021:5574), Heartbeat merupakan sebuah perangkat lunak *failover* berbasis klaster, yang memastikan bahwa *resources* akan selalu tersedia untuk diakses. Dalam jangka waktu tertentu, masing-masing *node* dari Heartbeat akan mengirimkan packet melewati jaringan kepada *node* Heartbeat lain sebagai sinyal tetap hidup. Jika tidak ada pesan Heartbeat dari *node master*, maka diasumsikan *node master* telah mati kemudian *node backup* menjalankan teknik *failover* serta mengambil alih peran dari *node master*.

Menurut Afif dan Suryono (2013:66), Heartbeat merupakan aplikasi dasar untuk Linux-HA (*Linux High Availability*). Heartbeat akan menjalankan script inisialisasi untuk HA dan saat *node* atau *server* mati dan hidup. *Heartbeating* heartbeat juga melakukan perpindahan IP dari satu *node* ke *node* yang lain (*IP floating*).

Komunikasi heartbeat bisa dilakukan melalui serial ports, UDP/IP *broadcast* (*ethernet*, etc), UDP/IP *multicast* (*ethernet*). Selain itu heartbeat juga menangani *service-service* apa saja yang akan dijalankan pada saat *node* atau *server* menjadi aktif.



Gambar 2.1 Konsep Heartbeat

### D. Database Server

Menurut Rahmatulloh dan MSN (2017:243), Database server adalah aplikasi komputer yang menyediakan layanan data ke komputer atau program komputer dan memiliki fungsi sebagai tempat penyimpanan data, seperti yang ditetapkan oleh model klien-server. Istilah ini juga merujuk kepada sebuah komputer yang didedikasikan untuk menjalankan program server database. Salah satu contoh aplikasi database adalah MySQL, Oracle, MariaDB dan lain sebagainya.

Menurut Darmawan, dkk (2018:94), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (Structured Query Language) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL

### E. Web Server

Menurut Rahmatulloh dan MSN (2017:242), Web server adalah sebuah software yang memberikan layanan berbasis data dengan menggunakan protokol HTTP atau HTTPS dari

client menggunakan aplikasi web browser untuk request data dan server akan mengirim data dalam bentuk halaman web dan pada umumnya berbentuk dokumen HTML. Halaman web yang diminta bisa terdiri dari berkas teks, video, gambar, file dan banyak lagi. Salah satu program dari Web Server adalah Apache. Apache merupakan web server yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX, untuk saat ini telah tersedia apache yang di desai untuk sistem operasi lainnya. Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunaanya.

Menurut Masykur dan Prasetyowati (2016:95), Apache adalah sebuah nama web server yang bertanggung jawab pada permintaan dan jawaban HTTP dan logging informasi secara lebih detail. Selain itu, Apache juga merupakan suatu web server yang modular, mengikuti standar protokol HTTP dan sangat disukai.

#### F. HAProxy

Menurut Noviyanto, dkk (2015:24), HAProxy adalah produk open source yang mendukung keperluan penyeimbang beban dan failover web server. HAProxy ini banyak digunakan untuk keperluan reverse proxy di site- site yang trafik hariannya tinggi. HAProxy sangat diperlukan apabila aplikasi menuntut SLA yang cukup ketat dan tidak mentolerir adanya downtime. Selain itu HAProxy memungkinkan adanya backup dari load balancer yang digunakan sehingga apabila terjadi down pada load balancer utama maka load balancer yang menjadi backup dalam waktu singkat secara otomatis akan menggantikan load balancer yang mati tersebut. Pada HAProxy ini memiliki beberapa parameter di dalamnya yaitu Global parameter, Proxy, Konfigurasi Server, Manipulasi HTTP, Accesslist, Logging, Statistik dan monitoring.

#### G. Jaringan Komputer

Menurut Madcoms dalam Rahadjeng dan Ritapuspitasari (2018:53), Jaringan komputer merupakan kumpulan dari beberapa komputer dan peralatan penunjang lainnya yang terhubung dalam satu kesatuan dan saling terkoneksi.

Menurut Foruzen dalam Pratama (2014:21), Jaringan komputer merupakan hasil dari koneksi (hubungan) dari sejumlah perangkat atau komputer yang dapat saling berkomunikasi satu sama lain. Perangkat yang dimaksud pada definisi ini mencakup semua jenis perangkat komputer (komputer desktop, komputer jinjing, smartphone, PC tablet) dan perangkat penghubung (*router, switch, modem, wireless access point*).

#### H. Ubuntu Server

Menurut Husen dan Surbakti (2020:23), Linux Ubuntu Server adalah sistem operasi turunan dari Linux Ubuntu yang di desain khusus dengan kernel yang sesuai untuk beroperasi sebagai sistem operasi server. Kernel Linux Ubuntu Server di desain khusus untuk bisa bekerja dengan lebih dari satu proses (multiprocessor) dengan dukungan NUMA pada 100Hz in-ternal timer frequency dan menggunakan penjadwalan deadline I/O.

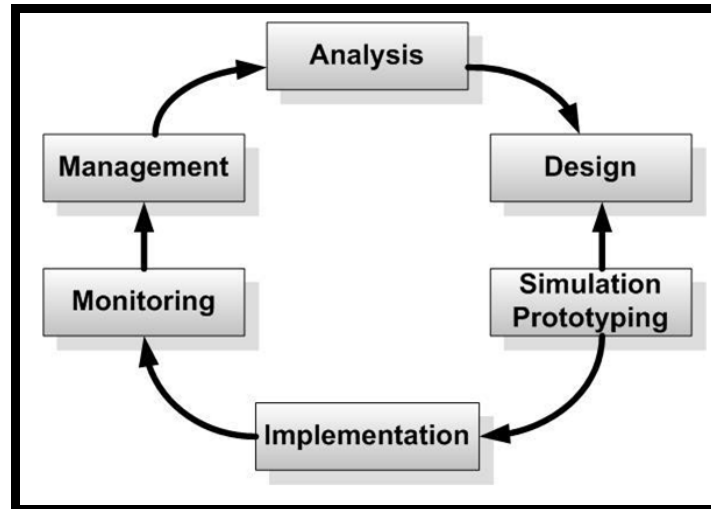
Menurut febriantoro, dkk (21:2013) Ubuntu server adalah istilah yang dikenal dengan terminal. Terminal sering disebut command prompt atau shell. Dimasa lalu, hal ini adalah cara pengguna untuk berinteraksi dengan komputer, dan para pengguna berpendapat bahwa penggunaan perintah melalui shell. Ubuntu suatu Operating System (OS) dari distro Linux. Sifatnya Open Source yang artinya semua orang bisa mengembangkan tanpa terpusat oleh suatu pengembang Ubuntu server memiliki kelebihan akan lebih cepat dibanding melalui aplikasi berbasis grafik dan hal ini masih berlaku sampai sekarang. Salah satu keunggulan lagi dari Ubuntu ini adalah adanya repository. Repository adalah tempat penyimpanan kumpulan software/aplikasi yang bias didownload untuk digunakan karena pada dasarnya setiap menginstall software di Linux dibutuhkan

repository, maka Linux akan mencari software tersebut di repository. Jika telah ditemukan maka proses instalasi bisa dilanjutkan

### 3. Metode Penelitian

Metode penelitian yang digunakan adalah metode *Network Development Life Cycle* (NDLC). Dengan menggunakan metode penelitian NDLC ini diharapkan penulis dapat

mendefinisikan siklus proses perancangan atau pengembangan suatu sistem jaringan komputer terutama dalam proses implementasi Pi Hole DNS *server* sebagai ads-blocker dan sistem *filtering* website pada jaringan hotspot. Berikut merupakan gambar dan tahapan dari metode penelitian NDLC, yaitu:



Gambar 3.1 NDLC Model

Dari Gambar 3.1 NDLC Model, dapat dijelaskan tahapan yang akan dilakukan seperti berikut ini:

1. Tahap *Analysis* (Analisis)

Tahapan analisis merupakan tahapan awal yang dilakukan dalam menganalisis kebutuhan, analisis permasalahan yang ada, analisis keinginan *user*, dan analisa topologi jaringan yang sudah ada, bisa dibidang tahapan ini adalah tahapan pengumpulan data yang dibutuhkan untuk perumusan masalah dalam menyelesaikan kendala yang ada. Dengan mengidentifikasi sistem yang sedang berjalan lalu mencoba untuk menganalisis suatu pengembangan sistem seperti apa yang akan diterapkan pada *database server* untuk menerapkan ketersediaan tinggi atau *high availability* menggunakan teknik *failover* dengan heartbeat.

2. Tahap *Design*

Pada tahapan ini, penulis akan membuat desain gambar topologi jaringan yang akan dibangun, desain akses data dan sebagainya dalam upaya implementasi *failover database server* menggunakan heartbeat.

3. Tahap *Simulation Prototype*

Pada tahapan ini, dilakukan simulasi pengembangan *database server* dengan menerapkan *failover database server* menggunakan heartbeat pada jaringan komputer. Hal ini dimaksudkan untuk melihat kinerja dari jaringan yang akan dibangun dan menjadi bahan presentasi dan *sharing* dengan pengembangan sistem jaringan.

4. Tahap *Implementation*

Pada tahapan ini, merupakan tahapan dimana implementasi *failover database server* menggunakan heartbeat. Pada tahapan ini juga akan terlihat bagaimana pengembangan yang akan dibangun serta apakah jaringan tersebut akan memberikan pengaruh terhadap sistem yang ada.

5. Tahap *Monitoring*

Tahapan *monitoring* merupakan tahapan penting agar jaringan dan komunikasi dapat berjalan sesuai dengan keinginan dan tujuan penulis pada tahap awal analisis.

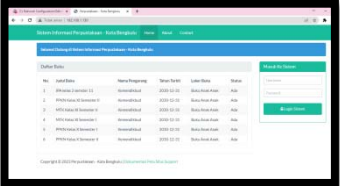
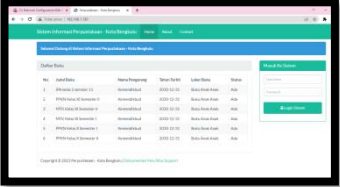
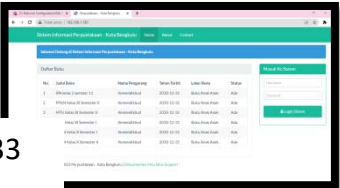
6. Tahap *Management*

Pada tahap manajemen ini akan dilakukan beberapa langkah pengelolaan agar sistem yang telah dibangun dapat berjalan sesuai dengan yang diharapkan..

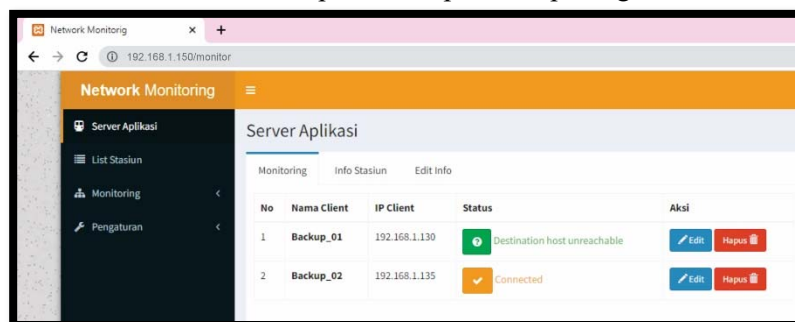
#### 4. Hasil dan Pembahasan

Pengujian Failover ini dilakukan dengan cara mematikan server aplikasi 1 (Utama) secara langsung, dengan kondisi seperti ini maka semua resources dari server aplikasi 1 (Utama) akan segera dipindahkan ke server aplikasi 2 (Cadangan) dan menjalankan service pada server cadangan untuk melakukan tugasnya seperti server aplikasi 1 (Utama). Akses utama yaitu server aplikasi 1 (utama), dimana jika server utama dan cadangan kedua-duanya dalam kondisi hidup, maka akses akan ke server utama. Dan ketika server utama mati akses akan secara otomatis pindah ke server cadangan. Adapun skenario pengujian yang dilakukan seperti pada table dibawah ini:

Tabel 4.1 Skenario Pengujian

No	Skenario	Hasil
1	Server Aplikasi 1 = Hidup Server Aplikasi 2 = Hidup	Apliksi web dapat di buka 
2	Server Aplikasi 1 = Mati Server Aplikasi 2 = Hidup	Apliksi web dapat di buka 
3	Server Aplikasi 1 = Hidup Server Aplikasi 2 = Mati	Apliksi web dapat di buka 

Status salah satu server mati dapat dilihat pada tampilan gambar dibawah ini:



Gambar 4.1 Tampilan Status Server

Dari tampilan gambar diatas dapat dilihat server aplikasi 1 dengan alamat IP Address 192.168.1.130 dalam kondisi mati (disconnect), yang dibuktikan dengan melakukan ping ke server aplikasi, seperti dapat dilihat pada tampilan gambar dibawah ini:



```

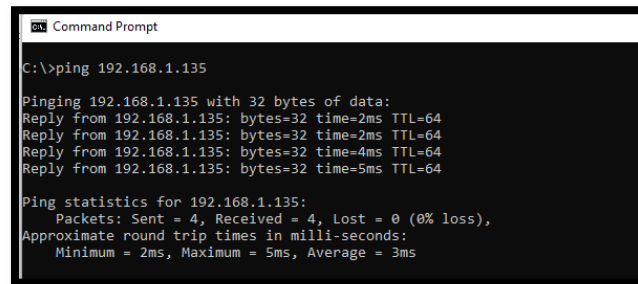
C:\>ping 192.168.1.130

Pinging 192.168.1.130 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.1.4: Destination host unreachable.
Request timed out.

Ping statistics for 192.168.1.130:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
  
```

**Gambar 4.2 Tampilan Ping ke Server Aplikasi 1**

Dari tampilan gambar 4.2 diatas dapat dilihat server aplikasi 1 dengan alamat IP Address 192.168.1.130 dalam kondisi mati (disconnect) ditandai dengan *Request Time Out*. Sedangkan server aplikasi 2 dalam keadaan hidup, seperti hasil ping dibawah ini:



```

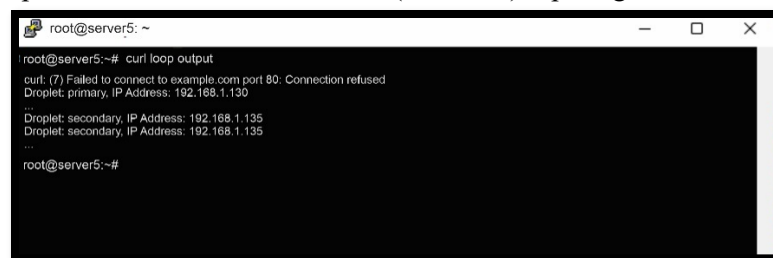
C:\>ping 192.168.1.135

Pinging 192.168.1.135 with 32 bytes of data:
Reply from 192.168.1.135: bytes=32 time=2ms TTL=64
Reply from 192.168.1.135: bytes=32 time=2ms TTL=64
Reply from 192.168.1.135: bytes=32 time=4ms TTL=64
Reply from 192.168.1.135: bytes=32 time=5ms TTL=64

Ping statistics for 192.168.1.135:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 5ms, Average = 3ms
  
```

**Gambar 4.3 Tampilan Ping ke Server Aplikasi 2**

Dari tampilan gambar 4.3 diatas dapat dilihat server aplikasi 2 dengan alamat IP Address 192.168.1.135 dalam kondisi Hidup (*connect*) ditandai dengan *Ping Reply* Sedang tampilan dari terminal server linux (heartbeat) seperti gambar dibawah ini:



```

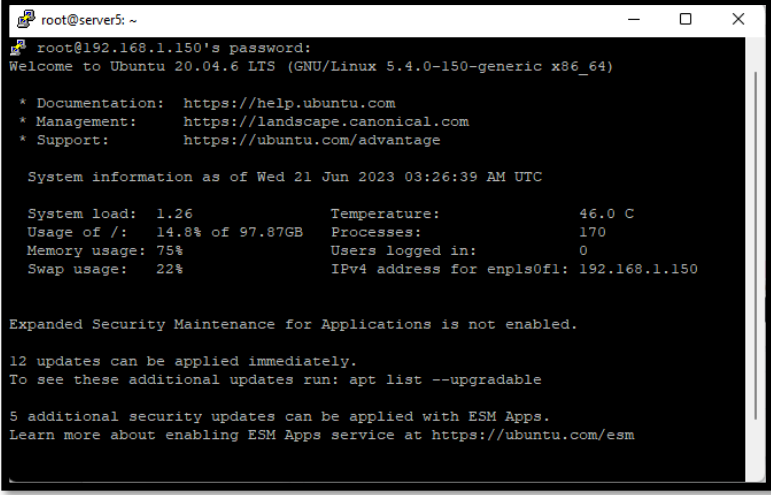
root@server5: ~
root@server5:~# curl loop output
curl: (7) Failed to connect to example.com port 80: Connection refused
Droplet: primary, IP Address: 192.168.1.130
...
Droplet: secondary, IP Address: 192.168.1.135
Droplet: secondary, IP Address: 192.168.1.135
...
root@server5:~#
  
```

**Gambar 4.4 Tampilan Server A Mati Dari Terminal Linux**

Dari serangkaian pengujian yang dilakukan system implementasi failover data base menggunakan *heartbeats* menggunakan linux ubuntu server dan MySQL berjalan dengan baik, sesuai dengan konfigurasi-konfigurasi yang diterapkan. Pengujian *Failover* terjadi jika pada salah satu *server failure*. *Heartbeat* akan perpindahan antara server node1 dan node2, yang akan diakses oleh *client*. *Heartbeat* selalu mengecek antara dua server bila salah satu down.

## Hasil Pengujian

Pengujian dilakukan terhadap beberapa indicator yang tidak terlepas dari rencana pengujian. Adapun IP Address yang digunakan pada masing-masing server dapat dilihat dari tampilan gambar dibawah ini:



```

root@server5: ~
root@192.168.1.150's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed 21 Jun 2023 03:26:39 AM UTC

System load:  1.26          Temperature:   46.0 C
Usage of /:   14.8% of 97.87GB  Processes:    170
Memory usage: 75%          Users logged in: 0
Swap usage:  22%           IPv4 address for enp1s0f1: 192.168.1.150

Expanded Security Maintenance for Applications is not enabled.

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

5 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

```

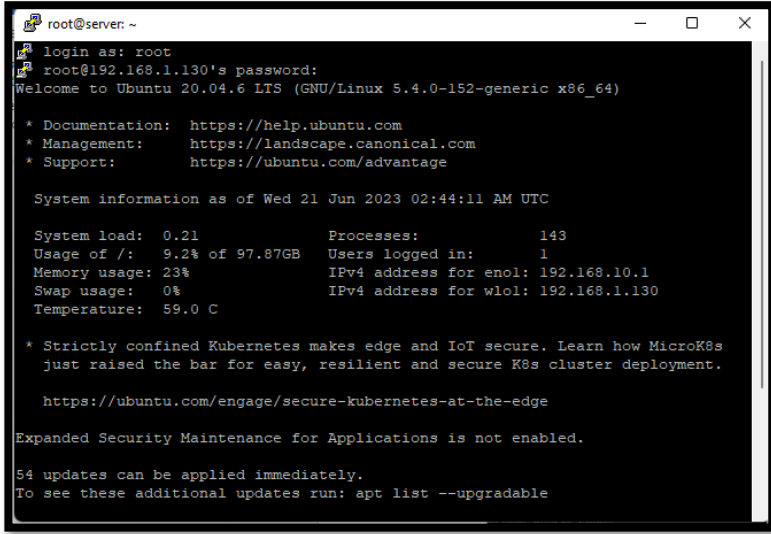
**Gambar 4.33 Tampilan Server Heartbeat**

Tampilan gambar diatas merupakan tampilan pada server heartbeats, dapat di lihat:

Sistem Operasi : Linux Ubuntu Server 20.04

IP Address : 192.168.1.150

IP Address yang digunakan pada server aplikasi 1 dapat dilihat dari tampilan gambar dibawah ini:



```

root@server: ~
login as: root
root@192.168.1.130's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-152-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed 21 Jun 2023 02:44:11 AM UTC

System load:  0.21          Processes:    143
Usage of /:   9.2% of 97.87GB  Users logged in: 1
Memory usage: 23%          IPv4 address for eno1: 192.168.10.1
Swap usage:  0%            IPv4 address for wlo1: 192.168.1.130
Temperature:  59.0 C

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

```

**Gambar 4.34 Tampilan Server 1**

Tampilan gambar diatas merupakan tampilan pada server heartbeats, dapat dilihat:

Sistem Operasi : Linux Ubuntu Server 20.04

IP Address : 192.168.1.130

IP Address yang digunakan pada server aplikasi 2 dapat dilihat dari tampilan gambar dibawah ini:



```

root@server: ~
root@192.168.1.135's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-152-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed 21 Jun 2023 03:07:58 AM UTC

System load:  0.29          Processes:    180
Usage of /:   9.4% of 97.87GB   Users logged in:  0
Memory usage: 58%          IPv4 address for docker0: 172.17.0.1
Swap usage:   1%            IPv4 address for wlp3s0b1: 192.168.1.135
Temperature: 26.8 C

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

8 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

```

Gambar 4.35 Tampilan Server 2

Tampilan gambar diatas merupakan tampilan pada server heartbeats, dapat di lihat:

Sistem Operasi : Linux Ubuntu Server 20.04  
 IP Address : 192.168.1.135

Masing-masing server tersebut saling terhubung pada Jaringan Local Area Network (LAN), selanjutnya dilakukan pengujian sebagai berikut:

#### 1. Pengujian Ketersediaan (Availability)

Pengujian dilakukan dalam rentang waktu 28 April – 3 Mei 2023 dengan mematikan server secara sengaja dan didapatkan waktu uptime atau waktu saat server aktif dan waktu response time atau waktu yang dibutuhkan server heartbeat untuk proses failover berlangsung seperti pada tabel berikut :

Tabel 4.4 Perhitungan Availability

No	Tanggal	Up (Menit)	Down (Menit)	Jumlah Up	Jumlah Down
1	28 – 4 – 2023	1438.25	2.50	2	0
2	29 – 4 – 2023	1438.10	2.70	2	0
3	30 – 4 – 2023	1428.20	2.50	1	1
4	31 – 4 – 2023	1437.25	3.50	2	0
5	3 – 5 – 2023	1425.15	2.50	1	1
6	2 – 5 – 2023	1439.40	3.70	1	1
7	3 – 5 – 2023	1432.05	2.50	2	0

Setelah mendapatkan uptime dan downtime, selanjutnya melakukan perhitungan untuk mendapatkan MTBF (*Mean Time Between Failure*), MTTR (*Mean Time To Repair*) dan persentase availability dengan rumus:

$$\text{Availability} = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

A = Availability (Ketersedian)

MTBF = uptime / jumlah uptime

MTRR = responsetime / jumlah downtime

Hasil dari perhitungan MTBF dan MTTR :

Tabel 4.5 Hasil Perhitungan Availability

No	Tanggal	Up (Menit)	Down (Menit)	Availability (%)
1	28 – 4 – 2023	1438.25	2.50	99.83
2	29 – 4 – 2023	1438.10	2.70	99.81
3	30 – 4 – 2023	1428.20	2.50	99.83
4	31 – 4 – 2023	1437.25	3.50	99.76
5	3 – 5 – 2023	1425.15	2.50	99.82
6	2 – 5 – 2023	1439.40	3.70	99.74
7	3 – 5 – 2023	1432.05	2.50	99.83
<b>Total</b>				<b>698.82</b>

Dari hasil pengujian availability didapatkan rata-rata persentasenya adalah:  
Average Availability =  $698.82 / 7$

$$= 99.80\%$$

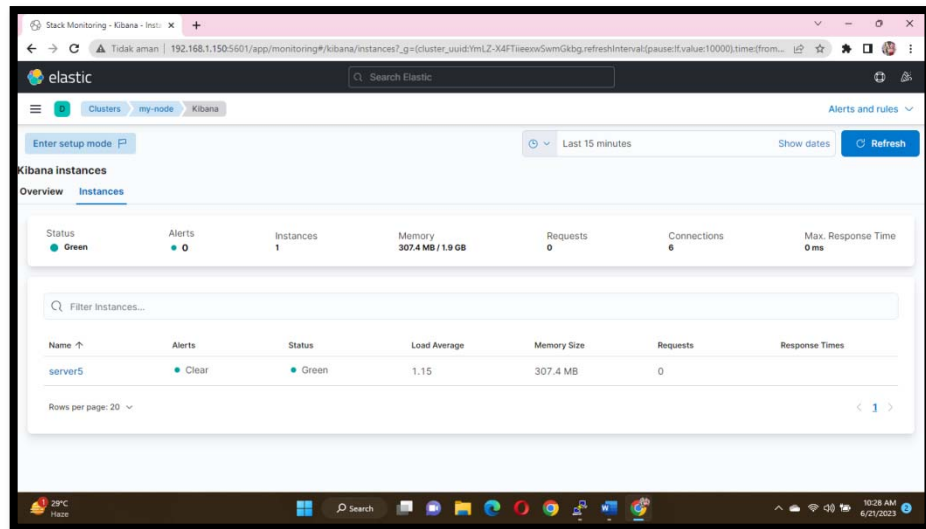
## 2. Pengujian *Downtime* dan *Response Time*

Heartbeat memudahkan proses *failback* agar data selalu sinkron satu sama lain. Proses *failback* berjalan dengan memindahkan pelayanan *service* dari *node2* ke server *node1*. Proses ini sesuai konfigurasi Heartbeat yaitu 5 detik. Pada peristiwa *failback* system akan bekerja seperti semula. Dan *node* yang dijadikan *node primary* yaitu *node1* akan kembali pulih menjadi *node primary* dan mensinkronkan data dengan *node secondary* yaitu *node2*

Tabel 4.6 Waktu *Delay* Pepindahan *Node*

Pengujian	Waktu (Detik)
1	4
2	3
3	2
4	3
5	4
6	3
7	2
	21

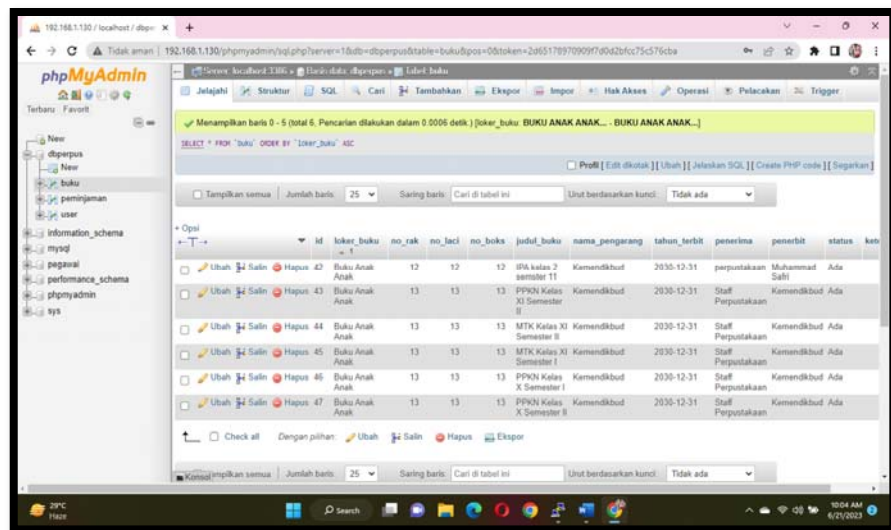
Untuk melakukan failover dan monitor server digunakan heartbeat dan ELK elastic. Akses utama yaitu server 1 dan jika jalur utama mengalami gangguan (*down*) maka secara otomatis akan pindah ke server 2, seperti tampilan gambar di bawah ini:



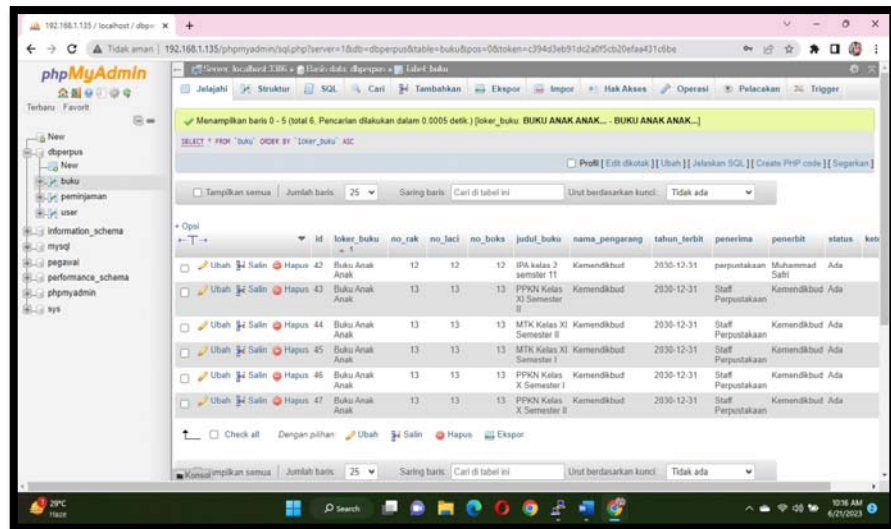
Gambar 4.36 Tampilan Status Server

### 3. Pengujian Sinkronisasi Data

Waktu yang dibutuhkan dalam melakukan replikasi database dipengaruhi oleh besar transfer data (database), kondisi jaringan (sibuk atau tidak sibuk) serta kualitas perangkat yang digunakan. Adapun untuk akses ke database (MySQL) melalui jaringan lokal, server A dapat diakses melalui alamat <http://192.168.1.130/phpmyadmin> dan untuk server B <http://192.168.1.135/phpmyadmin>, seperti dapat dilihat pada tampilan gambar dibawah ini :



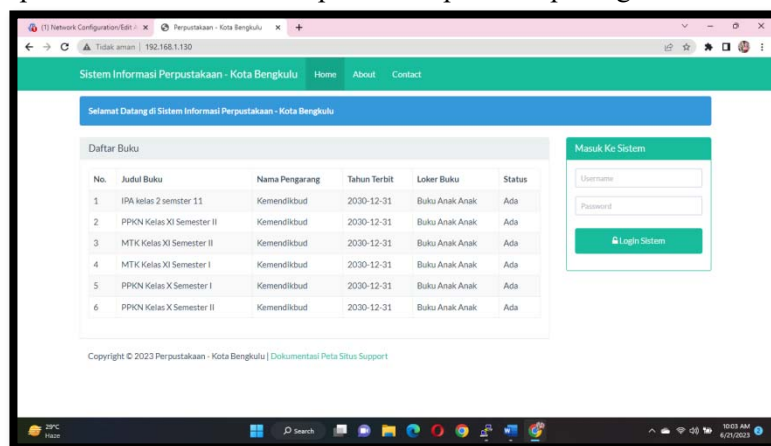
Gambar 4.37 Tampilan Informasi Server 1



**Gambar 4.38 Tampilan Informasi Server 2**

Dari serangkaian ujicoba yang dilakukan 95% proses sinkronisasi database (MySQL) berhasil dilakukan dengan kecepatan replikasi data +/- 2 sampai dengan 10 detik.

Hasil pengujian didapatkan perpindahan server yang dilakukan oleh heartbeat memakan waktu rata-rata selama 3 detik, seperti dapat dilihat pada table 4.1 dibawah. Adapun aplikasi pada node1 dan node2 dapat dilihat pada tampilan gambar dibawah ini:



**Gambar 4.39 Tampilan Aplikasi Yang di Gunakan**

Pada pengujian failback atau proses pengembalian fungsi server primary (node1) dari server secondary (node2) meliputi perubahan yang terjadi pada server node2 selama menggantikan fungsi dari server node1. Layanan yang diberikan aplikasi Heartbeat memudahkan proses failback agar data selalu sinkron satu sama lain. Proses failback berjalan dengan memindahkan pelayanan service dari node2 ke server node1. Proses ini sesuai konfigurasi Heartbeat yaitu 5 detik. Pada peristiwa failback system akan bekerja seperti semula. Dan node yang dijadikan node primary yaitu node1 akan kembali pulih menjadi node primary dan mensinkronkan data dengan node secondary yaitu node2.

Dari serangkaian pengujian yang dilakukan didapatkan hasil sesuai table dibawah ini:

Tabel 4.7 Hasil Pengujian

No	Jenis Pengujian	Kriteria	Hasil	Keterangan
1.	Pengujian Ketersediaan (Availability)	Pengujian dilakukan dengan menggunakan rumus availability.	99.80%	Baik
2.	Pengujian Downtime dan Resptime	Pengujian ini bertujuan untuk mengamati proses Failover dari kedua server	Berjalan sesuai rancangan dimana ketika satu server mati akan otomatis pindah keserver lainnya dan kedua server dapat melakukan sinkronisasi data melalui server heratbeat	Baik
3.	Pengujian Waktu Sinkronisasi data base	Pengujian ini dilakukan untuk mengetahui Waktu yang dibutuhkan MySQL server untuk sinkronisasi data.	Waktu yang dibutuhkan untuk sinkronisasi data kecil dari 1 menit	Baik

## 5. Kesimpulan

Kesimpulan yang dapat diambil setelah mengimplementasikan sistem failover database menggunakan heartbeats dengan menggunakan sistem replikasi master to master adalah sebagai berikut :

1. Dengan teknologi failover database menggunakan heartbeats dengan menggunakan sistem replikasi database maka ketersediaan data dan kinerja sistem meningkat karena data telah memiliki cadangan.
2. Akses utama yaitu server aplikasi 1 (utama), dimana jika server utama dan cadangan keduanya dalam kondisi hidup, maka akses akan ke server utama. Dan ketika server utama mati akses akan secara otomatis pindah ke server cadangan
3. failover database menggunakan heartbeats dengan menggunakan sistem replikasi database memiliki kelebihan (dapat melakukan backup data secara otomatis) serta kekurangan (membutuhkan perangkat tambahan), sehingga untuk menggunakan teknologi ini perlu disesuaikan dengan kebutuhan pengembangan sistem.

## REFERENSI

1. Afif, Mohammad Faruq;, and Tito Suryono. 2013. "Implementasi Disaster Recovery Plan Dengan Sistem Fail Over Menggunakan DRBD Dan Heartbeat Pada Data Center FKIP UNS." *Indonesian Jurnal on Networking and Security (IJNS)* 2(2): 64–69. <http://ejournal.unsa.ac.id/index.php/seruni/article/view/582>.
2. Amin, Zaid. 2017. *Desain Dan Implementasi Tunneling Ipv4 Berbasis Unix Dengan Esp (Encapsulating Security Payload)*. STMIK PalComTech. Palembang.
3. Husen, Zakaria, and M.Syukri Surbakti. 2020. *Membangun Server Dan Jaringan Komputer Dengan Linux Ubuntu*. Banda Aceh: Syiah Kuala University Press.
4. Masykur, Fauzan dan Prasetyowati, Fiqiana. 2016. "Aplikasi Rumah Pintar (Smart Home) Pengendali Peralatan Elektronik Rumah Tangga Berbasis Web". *Jurnal Sains, Teknologi dan Industri*, Vol. 14, No. 1. Hal 93-100.
5. Noviyanto, Ari Budi; Kumalasari N, Erna; Hamzah, Amir. 2015. "Perancangan Dan Implementasi Load Balancing Reverse Proxy Menggunakan Haproxy Pada Aplikasi Web." *Jurnal JARKOM* 2(2): 32–43.
6. Pratama, Dian. 2021. "Perbandingan Kinerja Teknologi Failover Berbasis Klaster ( Heartbeat ) Dengan Teknologi Failover Berbasis Jaringan ( Keepalived )." 5(12): 5572–81.
7. Pratama, I Putu Agus Eka. 2014. *Smart City Beserta Cloud Computing dan Teknologi -*

- Teknologi Pendukung Lainnya. Informatika, Bandung, 582 Halaman.
8. Purnomo, J, and S M W Edi. 2017. "Implementasi Dan Analisis High Availability Server Dengan Teknik Failover Clustering Menggunakan Heartbeat." *Artikel Ilmiah. Universitas Kristen Satya Wacana*: 1–26.
  9. Rahadjeng, Indra Riyana dan Ritapuspitari. 2018. Analisis jaringan local area network (LAN) pada PT. Mustika ratu tbk Jakarta Timur. *Jurnal PROSISKO*, Vol. 5 No. 1, 53-60.
  10. Rahmatulloh, Alam, and Firmansyah MSN. 2017. "Implementasi Load Balancing Web Server Menggunakan Haproxy Dan Sinkronisasi File Pada Sistem Informasi Akademik Universitas Siliwangi." *Jurnal Nasional Teknologi dan Sistem Informasi* 3(2): 241–48.
  11. Setiawan, Aris, Asep. 2015, *Perancangan Sistem Informasi Pengadaan Barang Menggunakan Metode Berorientasi Objek*. UNIKOM. Bandung.
  12. Sofana, Iwan. 2012. *CISCO CCNA dan Jaringan Komputer Edisi Revisi*. Informatika. Bandung. 614 hal.
  13. Syafrizal, Melwin. 2005. *Pengantar Jaringan Komputer*. Andi. Yogyakarta. 274 hal.