

PERANCANGAN INFRASTRUKTUR WEB SERVER DAN DATABASE MENGUNAKAN METODE REPLICATION MIRROR DAN FAILOVER CLUSTERING

Zaenal Mutaqin Subekti, S.Kom., M.Kom.¹, Subandri, S.Kom., M.Kom.², Galih Rakasiwi³

¹Jurusan Teknik Komputer, STMIK Bani Saleh

²Jurusan Teknik Informatika, STMIK Bani Saleh

³Jurusan Teknik Informatika, STMIK Bani Saleh

Jl. M Hasibuan No. 68 Bekasi 17113

E-mail: zms.stmikbanisaleh@gmail.com 1), andrisubandri@ymail.com 2), galihrakasiwi2309@gmail.com.com 3)

ABSTRAKS

Data dan informasi merupakan suatu hal yang sangat penting. Suatu sistem yang digunakan oleh setiap instansi bergantung pada informasi dan aplikasi yang memproses data informasi tersebut. Salah satu kebutuhan manusia yang paling dasar adalah Informasi. Teknologi informasi dalam konteks teknis dapat diartikan sebagai sekumpulan infrastruktur untuk mendukung pengelolaan informasi yang meliputi proses mengumpulkan, menyimpan, mengambil, menyebarkan dan menggunakan kembali informasi. High Availability Server merupakan salah satu metode failover clustering teknologi ini digunakan untuk mengantisipasi kegagalan atau kerusakan hardware pada komputer server yang dapat mengganggu request dari client. Masalah yang muncul bisa disebabkan karena server utama mati dan tidak ada server backup yang menggantikan kerja dari server utama yang mati sehingga sehingga proses request terganggu. Salah satu solusi untuk mengatasi masalah tersebut dengan merancang failover clustering dan replication mirror.

Kata Kunci: failover, clustering, server backup, replication, mirror

1. PENDAHULUAN

1.1 Latar Belakang

Pesatnya pertumbuhan era digital saat ini, mendorong setiap organisasi untuk menyimpan data pada sebuah media penyimpanan yang kemudian diolah menjadi data informasi, dalam memudahkan organisasi tersebut mengetahui informasi apa yang dibutuhkan. Kemajuan jaringan komputer saat ini dapat memberikan layanan komunikasi yang semakin efektif dan efisien.

Data dan informasi merupakan suatu hal yang sangat penting. Suatu sistem yang digunakan oleh setiap instansi bergantung pada informasi dan aplikasi yang memproses data informasi tersebut. Salah satu kebutuhan manusia yang paling dasar adalah Informasi. Teknologi informasi dalam konteks teknis dapat diartikan sebagai sekumpulan infrastruktur untuk mendukung pengelolaan informasi yang meliputi proses mengumpulkan, menyimpan, mengambil, menyebarkan dan menggunakan kembali informasi.

Teknologi Informasi (TI), atau dalam bahasa Inggris dikenal dengan istilah *Information technology (IT)* adalah istilah umum untuk teknologi apa pun yang membantu manusia dalam membuat, mengubah, menyimpan, mengomunikasikan dan/atau menyebarkan informasi. TI menyatukan komputasi dan komunikasi berkecepatan tinggi untuk data, suara, dan video. Contoh dari Teknologi Informasi bukan hanya berupa komputer pribadi, tetapi juga telepon, TV, peralatan rumah tangga

elektronik, dan peranti genggam modern (misalnya ponsel)

Instansi pendidikan merupakan salah satu organisasi yang menyimpan informasi pentingnya pada peralatan teknologi informasi. Sekolah merupakan instansi yang bergerak di bidang pendidikan. Semua proses pengolahan data dan pengolahan informasi terkait dalam bidangnya pasti akan bergantung pada kemajuan teknologi, khususnya dalam penggunaan aplikasi pengolahan data yang berbasis *website* yang ada di *server* local. Ketergantungan tersebut menjadi pemicu utama kerusakan dan kehilangan data pada server dan belum dapat ditangani oleh pihak sekolah. Kehilangan data yang ada di *server* atau kerusakan *hardware* bisa saja terjadi jika pengelolaan *server* yang kurang baik. Berdasarkan keadaan tersebut maka pengambilan keputusan sangatlah penting untuk menjaga data informasi instansi,

High Availability Server merupakan metode *failover clustering* teknologi ini digunakan untuk mengantisipasi kegagalan atau kerusakan *hardware* pada komputer *server* yang dapat mengganggu *request* dari *client*. Masalah yang muncul bisa disebabkan karena *server* utama mati dan tidak ada *server backup* yang menggantikan kerja dari *server* utama yang mati sehingga sehingga proses *request* terganggu. Salah satu solusi untuk mengatasi masalah tersebut dengan merancang *failover clustering* dan *replication mirror*.

Dipilihnya *failover clustering* sebagai solusi dikarenakan metode ini menyediakan sebuah teknik

yang jika *server* utama mati, maka *server* yang lain mengambil alih peran dari *server* yang mati tersebut untuk melayani request dari *client*. Dan ketika *server* utama yang sebelumnya tadi mati telah aktifkan kembali maka tugas untuk melayani request *client* diambil alih kembali oleh *server* utama tersebut dan *server backup* kembali bersifat pasif.

Untuk membangun metode *failover clustering* dibutuhkan sebuah *replication mirror* yang dapat mereplika *server* utama secara otomatis. Metode *failover clustering* dan *Replication mirror* ini dapat menjawab kebutuhan akan adanya High Availability *server*.

1.2 Referensi

1.2.1 Replication

(Halim Maulana, 2016:1) 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. Replikasi database dapat digunakan apabila sebuah organisasi atau perusahaan didukung oleh hardware dan aplikasi software dalam sebuah sistem yang terdistribusi melalui koneksi jaringan lokal maupun internet.

1.2.2 Clustering

Clustering server merupakan yang menggabungkan beberapa sumber daya yang bekerja bersama - sama sehingga tampak seolah - olah merupakan suatu sistem tunggal (Irfani, 2015:1). Dalam dunia TI sendiri *Clustering server* merupakan sebuah infrastruktur yang menggunakan lebih dari satu *server* yang menyediakan redundant interconnection, sehingga pengguna hanya mengetahui hanya ada satu sistem yang berjalan dan pengguna tidak akan menyadari jika terjadi kerusakan fisik *server* maupun kegagalan dari sistem *server* sendiri itu sehingga *server* mati karena tersedia *server* lain berguna sebagai *backup*.

1.2.2.1 Failover Clustering

failover Cluster adalah sekelompok komputer independen yang bekerja bersama untuk meningkatkan ketersediaan dan skalabilitas peran *cluster* (sebelumnya disebut aplikasi dan layanan *cluster*). *Server* ditempatkan bersamaan (disebut *node*) dihubungkan oleh kabel fisik dan oleh perangkat lunak. Jika satu atau lebih dari *node cluster* gagal, *node* lain mulai menyediakan layanan (proses yang dikenal sebagai *failover*).

1.2.3 Heartbeat

(Maskur Purwiadi 2018:5) *Heartbeat* merupakan sebuah aplikasi yang dapat mendeteksi apabila *server* utama *down* maka *Heartbeat* akan secara otomatis mengarahkan peran *server utama* kepada *server backup*. *Heartbeat* menjalankan *script*

inisialisasi untuk menjalankan *service* lain saat *Heartbeat* dijalankan atau bisa juga mematikan *service* lain saat *Heartbeat* dimatikan.

1.2.4 Web Server

Menurut Askari Azikin (2011:175), Untuk Menampilkan halaman dari situs yang kita miliki pada halaman *web browser* dan dapat diakses oleh orang banyak dibutuhkan sebuah *web server*.

1.2.5 Domain Name System

(Athaulah,2012) DNS merupakan singkatan dari Domain Name System yang merupakan sebuah sistem untuk menyimpan informasi tentang nama *host* atau nama *domain* dalam sebuah basis data tersebar (*distributed database*) di dalam jaringan komputer, misalnya internet. DNS menyediakan alamat IP untuk setiap nama *host* dan mendata setiap *mail exchange server* yang menerima *email* untuk setiap domain.

1.2.6 SSH (Secure Shell)

SSH yaitu program yang memungkinkan anda untuk login ke sistem remote dan memiliki koneksi yang terenkripsi. SSH merupakan paket program yang digunakan sebagai pengganti yang aman untuk *login*, *rsh* dan *rcp*. Ia menggunakan *public-key cryptography* untuk mengenkripsi komunikasi antara dua host, demikian pula untuk autentikasi pemakai (Ika Dwi Cahyani 2010:2).

1.2.7 Rsync

Menurut Andrew Tridgell dalam (Wikipedia:2019) Rsync adalah utilitas untuk mentransfer dan menyinkronkan file secara efisien antara komputer dan hard drive eksternal dan di seluruh komputer jaringan dengan membandingkan waktu modifikasi dan ukuran file.

1.2.8 Crontab

I Putu Eka Suparwita (2012:31) *Crontab* adalah sebuah perintah yang sangat berguna untuk menjalankan tugas-tugas yang terjadwal, sehingga akan mengurangi waktu administrasi.

1.2.9 Linux

Menurut *id.wikipedia.org* Ubuntu *server* merupakan sistem operasi yang di install sebagai sistem operasi *server*, yang dibangun diatas kernel linux. Nama Ubuntu berasal dari filosofi dari Afrika Selatan yang berarti "kemanusiaan kepada sesama" Proyek Ubuntu resmi disponsori oleh Canonical Ltd. Ubuntu dilengkapi dengan banyak pilihan lingkungan desktop, di antaranya yang paling terkenal adalah GNOME, KDE, Xfce, dan LXDE.

1.3 Tinjauan Pustaka

- a. Penelitian Dari jurnal Irfani (2015) yang juga membahas *Failover* juga dengan judul Implementasi *High Availability Server* Dengan Teknik *Failover Virtual Computer Cluster*
- b. Penelitian *Failover* juga pernah dibahas oleh Abdul Hakim, Meirina Suci Ridha, Sujiliani Heristian, Arina Selawati, Pradnya Paramita (2018) dalam jurnalnya Implementasi *Failover Clustering Server* Untuk Mengurangi Resiko *Downtime* Pada *Web Server*
- c. Penelitian tentang *server cluster high availability* juga pernah di bahas oleh Komariyah Fitri dan Argyawati Harum (2016) Implementasi *Server Cluster High Availability* Pada *Web Server* dengan Sistem Operasi Turnkey Linux Menggunakan *Heartbeat*

2. METODE PENELITIAN

2.1 Metode Pengumpulan Data

a. Studi Literatur

Menggunakan Berbagai macam literatur mengenai teori Failover Clustering. Untuk melengkapi data yang diinginkan dan pengetahuan akan masalah yang diambil maka dilakukan peninjauan melalui membaca Dalam melakukan penelitian ilmiah harus dilakukan teknik penyusunan yang sistematis untuk memudahkan langkah-langkah yang akan diambil. Begitu pula yang dilakukan penulis dalam penelitian ini, langkah pertama yaitu dengan melakukan studi literatur pada buku-buku yang membahas tentang High Availability, jurnal, dan penelitian yang telah dilakukan yang berkaitan dengan *Failover*, *Mirror*, dan *Replication*.

b. Referensi Internet

Mencari, Mendownload, dan mengumpulkan data yang berhubungan dengan teori failover clustering dan referensi yang berkaitan dan mencari tutorial yang berkaitan.

2.2 Metode Pengembangan Sistem

Ada enam tahap yang dilakukan ketika menerapkan metode *PPDIOO* yaitu *prepare*, *plan*, *design*, *implementation*, *operate* dan, *optimize*.

1. Prepare

Pada tahap ini dimana dilakukan analisa kebutuhan yang mencakup identifikasi. Pada tahap ini didapatkannya hasil analisa berupa data – data kebutuhan pihak instansi dalam meningkatkan segi teknis. Data - data tersebut akan menjadi bahan pertimbangan dalam perancangan sistem yang akan diterapkan.

Dalam tahap ini dibutuhkan perangkat lunak dan perangkat keras dan persiapan sebelum memulai membuat *failover* dan *mirroring server* yaitu:

a. Kebutuhan Perangkat Keras

Beberapa perangkat keras yang diperlukan dalam penelitian ini adalah sebagai berikut :

- 2 unit PC Server.

PC server di *install* sistem operasi dan *software* – *software* pendukung lainnya yang telah disebutkan di kebutuhan perangkat lunak diatas. *Server* tersebut memiliki spesifikasi sebagai berikut :

Tabel 2.1 Spesifikasi Server Utama

Hardware	Tipe	Ukuran
Motherboard	Acer	
Processor	Intel Core i5-8300H	
Memory	DDR 3	4 Gb
Hardisk	Seagate	320 Gb
CD / DVD	DVD	
NIC	On Board	
Software	Linux Ubuntu Server 16.04 Xerial Xerus	

Hardware	Tipe	Ukuran
Motherboard	Acer	
Processor	Intel Core i5-8300H	
Memory	DDR 3	4 Gb
Hardisk	Seagate	320 Gb
CD / DVD	DVD	
NIC	On Board	
Software	Linux Ubuntu Server 16.04 Xerial Xerus	

Tabel 2.2 Spesifikasi Server Backup

b. Kebutuhan Perangkat Lunak

Beberapa perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut :

- Sistem Operasi Linux Ubuntu Server 16.04 Xerial Xerus. Sistem operasi ini digunakan sebagai *server system* pada *server* SMK Negeri 15 Kota Bekasi.
- Sistem Operasi Window 10 Sistem operasi ini digunakan pada komputer client. Komputer client akan digunakan untuk melakukan percobaan serta evaluasi terhadap sistem jaringan yang baru menggunakan *failover* dan *mirroring server*.

- Aplikasi untuk membuat *Web server*
Aplikasi yang digunakan untuk membuat web *server* pada kedua sisi *server* adalah Apache. Aplikasi ini bisa diperoleh secara bebas (open source).
- Aplikasi untuk menampilkan file PHP
Untuk menampilkan file php yang ada pada web server maka digunakan satu aplikasi yaitu PHP. Aplikasi ini bisa diperoleh secara bebas (open source)
- Aplikasi untuk membuat *Database*
Untuk membuat *database* pada sisi *server* di gunakan satu aplikasi yang sama yaitu MySQL *Server*. Aplikasi ini bisa diperoleh secara bebas (open source).
- Aplikasi untuk membuat *Failover*
Untuk membuat *failover* pada kedua sisi *server* di gunakan satu aplikasi yang sama yaitu *Heartbeat*. Aplikasi ini bisa diperoleh secara bebas (open source).
- Aplikasi untuk membuat *Mirroring server*
Untuk membuat *mirroring* pada sisi *server backup* di gunakan satu aplikasi yang sama yaitu *Rsync*. Aplikasi ini bisa diperoleh secara bebas (open source).

2. Plan

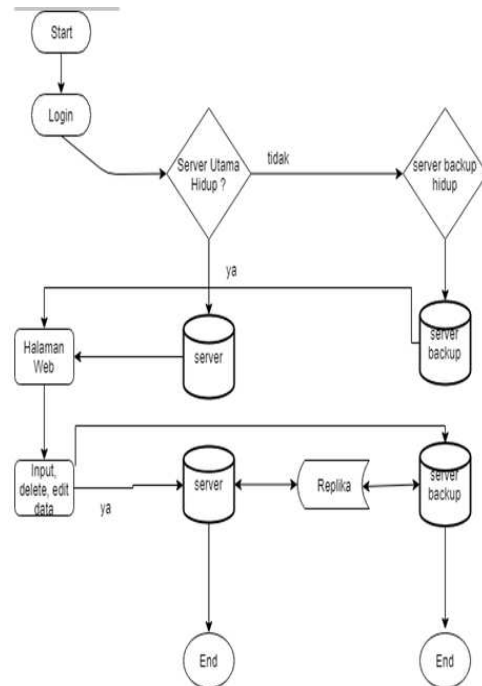
Pada tahap plan, telah didapatkan informasi / data dari tahap sebelumnya, maka dibuatnya sebuah perencanaan dalam membuat rancangan infrastruktur yang berdasarkan tujuan, fasilitas kebutuhan rancangan yang dibuat.

- Penyelesaian rancana dan desain
- Membuat Rancangan *Failover*
- Membuat Rancangan *Replication mirror*
- Membuat Rancangan *Replication Database*

3. Design

Ada tiga desain yang peneliti buat yaitu *Flow Chart*, Topologi Jaringan dan Cara Kerja *failover dan Replication mirror*.

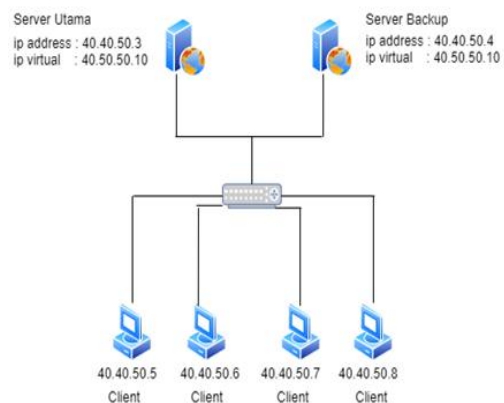
- Flow Chart



Gambar 2. 1 Flow Chart *Failover dan Replication*

b. Topologi Jaringan

Setelah melakukan pembahasan, dilakukan penyesuaian desain Topologi dengan menggunakan topologi *Star* sebagai berikut :



Gambar 2. 2 Topologi Desain

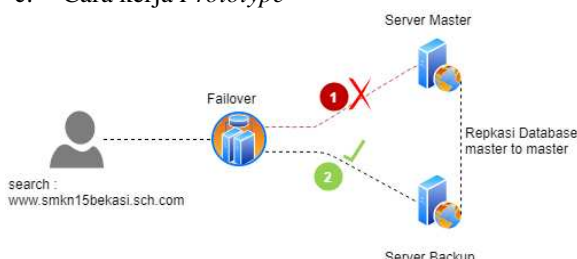
TABEL 2. 1 Tabel Informasi Server Utama

Hostname	Server15
Operating System	Linux Ubuntu Server 16.04 Xenial Xerus
Aplikasi	Apache, PHP, MySql, SSH, Heartbeat, dan Rsync.
Ip address	40.40.50.3
Subnet Mask	255.255.255.0
Ip Virtual Cluster	40.40.50.10
DNS	smkn15bekasi.sch.com

TABEL 2. 4 Tabel Informasi Server Utama

Hostname	Backup15
Operating System	Linux Ubuntu Server 16.04 Xenial Xerus
Aplikasi	Apache, PHP, MySql, SSH, Heartbeat, dan Rsync.
Ip address	40.40.50.3
Subnet Mask	255.255.255.0
Ip Virtual Cluster	40.40.50.10
DNS	smkn15bekasi.sch.com

c. Cara kerja *Prototype*



Gambar 3. 3 Cara Kerja Failover dan Replication Mirror

Client Melakukan Pemanggilan pada www.smkn15bekasi.com dan input data pada web server jika server utama mati maka proses input pada server akan berpindah ke server backup.

4. Implement

Semua yang dirancang dalam tahapan desain akan penulis terapkan di tahap *implement* ini. Penerapan yang dilakukan penulis yaitu melakukan instalasi pada *server* yang akan digunakan. Setelah itu lakukan konfigurasi pada *server* tersebut agar bisa melakukan proses *failover* dan mengimplementasikan proses replikasi *mirror* dan replikasi *database* yang digunakan saling mereplika data.

5. Operate

Pada tahap operate ini, semua perangkat dijalankan dan di integrasikan serta diawasi. Dalam tahapan ini performa kedua *server* diawasi serta kesalahan – kesalahan

atau kegagalan – kegagalan yang mungkin terjadi terus di awasi untuk menjadi bahan pertimbangan di tahap *optimize*.

6. Optimize

Pada tahapan optimasi ini dilakukan manajemen *server* dan dilakukan modifikasi sistem yang telah penulis buat jika terjadi suatu ketidaksesuaian terhadap kebutuhan, akan tetapi lebih kepada sistem failover dan replikasi *datasenya*, seperti penggunaan alamat ip pada *server* dan *client*. Jika sistem yang baru dirancang ternyata tidak sesuai dan menimbulkan banyak permasalahan pada pihak sekolah dapat meminta peneliti untuk merancang kembali desain sistem yang diterapkan. Syarat – syarat yang dimodifikasi ulang mengarah kepada awal siklus metode PPDIOO.

3. HASIL DAN PEMBAHASAN KONFIGURASI *FAILOVER CLUSTERING*

```

GNU nano 2.5.3 File: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto em0s3
iface em0s3 inet static
address 40.40.50.3
netmask 255.255.255.0
gateway 40.40.50.1
broadcast 40.40.50.254
    
```

Gambar 3.1 Konfigurasi *network interface* Pada *Server Utama*

Konfigurasi yang sama dilakukan pada *network interface* server backup dengan ip 40.40.50.4. Langkah berikutnya adalah konfigurasi *hostname* kedua *server*. *Server Utama* `node1.smkn15bekasi.sch.com`

```

root@server15:/home/server15# hostnamectl set-hostname node1.smkn15bekasi.sch.com
    
```

Gambar 3.2 Konfigurasi *hostname* Pada *Server Utama*

Lalu untuk server backup konfigurasi *hostname* `node2.smkn15bekasi.sch.com`

```

root@node2:/home/Backup15# hostnamectl set-hostname node2.smkn15bekasi.sch.com
    
```

Gambar 3.3 Konfigurasi *hostname* Pada *Server Backup*

Langkah berikutnya melakukan konfigurasi *heartbeat* di kedua *server* dengan konfigurasi yang sama seperti di *server* utama yaitu *nano*

/etc/ha.d/ha.cf/. Yang membedakan hanya konfigurasi pada *script ucast*, makasud dari *ucast* yaitu ip lawan atau ip *server backup*.

```
GNU nano 2.5.3 File: /etc/ha.d/ha.cf
#keepalive 2
warntime 5
deadtime 15
initdead 90
udpport 694
auto_failback off
ucast enp0s3 40.40.50.4
logfile /var/log/ha-log
node node1.smkn15bekasi.sch.com node2.smkn15bekasi.sch.com
```

Gambar 3.4 Konfigurasi /etc/ha.d/ha.cf/ Pada Server Utama

```
GNU nano 2.5.3 File: /etc/ha.d/ha.cf
#keepalive 2
warntime 5
deadtime 15
initdead 90
udpport 694
auto_failback off
ucast enp0s3 40.40.50.3
logfile /var/log/ha-log
node node1.smkn15bekasi.sch.com node2.smkn15bekasi.sch.com
```

Gambar 3.5 Konfigurasi /etc/ha.d/ha.cf/ Pada Server Backup

Langkah berikutnya membuat file kunci /etc/ha.d/authkeys di kedua *server* dan berikan hak chmod 600 pada authkeys

```
GNU nano 2.5.3 File: /etc/ha.d/authkeys
auth 2
2 crc
```

Gambar 3.6 Konfigurasi /etc/ha.d/authkeys Pada Server utama dan backup

```
root@server15:/home/server15# chmod 600 /etc/ha.d/authkeys
root@server15:/home/server15#
```

Gambar 3.7 Konfigurasi chmod 600 pada file authkeys

Langkah berikutnya adalah konfigurasi file /etc/haresources pada *server* utama dan *server backup*

```
GNU nano 2.5.3 File: /etc/ha.d/haresources Modified
node1.smkn15bekasi.sch.com IPaddr::40.40.50.10/24/enp0s3:0 apache2
```

Gambar 3.8 Konfigurasi /etc/haresources

Langkah berikutnya adalah edit file /etc/hosts pada kedua *server*

```
GNU nano 2.5.3 File: /etc/hosts Modified
127.0.0.1 localhost
127.0.1.1 server15
40.40.50.3 node1.smkn15bekasi.sch.com node1
40.40.50.4 node2.smkn15bekasi.sch.com node2
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Gambar 3.9 Edit file /etc/hosts

Setelah selesai semua di konfigurasi maka lakukan start dan enable heartbeat di kedua sisi *server*.

Konfigurasi DNS pada server utama dan backup

Pertama yaitu *edit file named.conf.local* pada direktori /etc/bind menggunakan teks *editor*. Pada *file* ini ditentukan untuk zona *forward* agar dns dapat terbaca oleh *client* yang terhubung dengan *server*.

```
GNU nano 2.5.3 File: /etc/bind/named.conf.local
// Do any local configuration here
//
zone "smkn15bekasi.sch.com" {
    type master;
    file "/etc/bind/db.smkn15bekasi.sch.com";
};
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```

Gambar 3.10 Edit File named.conf.local

Selanjutnya, salin *file* db.local menjadi db.smkn15bekasi.sch.com pada direktori /etc/bind, dan *edit file* tersebut menggunakan teks.

```
GNU nano 2.5.3 File: /etc/bind/db.smkn15bekasi.sch.com
; BIND data file for local loopback interface
$TTL 604800
@ IN SOA smkn15bekasi.sch.com. root.smkn15bekasi.sch.com. (
    2 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS ns.smkn15bekasi.sch.com.
@ IN A 40.40.50.10
@ IN A www.smkn15bekasi.sch.com.
@ IN AAAA ::1
ns IN A 40.40.50.10
www IN A 40.40.50.10
```

Gambar 3.11 Edit File db.smkn15bekasi.sch.com

```
GNU nano 2.5.3 File: /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
domain smkn15bekasi.sch.com
search smkn15bekasi.sch.com
nameserver 40.40.50.10
nameserver 8.8.8.8
```

Gambar 3.12 Edit File Resolv.conf

Selanjutnya, *edit file* hosts pada direktori /etc/hosts.

```
GNU nano 2.5.3 File: /etc/hosts
127.0.0.1 localhost
127.0.1.1 server15
40.40.50.3 node1.smkn15bekasi.sch.com node1
40.40.50.4 node2.smkn15bekasi.sch.com node2
40.40.50.10 smkn15bekasi.sch.com www.smkn15bekasi.sch.com ftp.smkn15bekasi.sch.com ns.smkn15bekasi.sch.com
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Gambar 3.13 Edit File /etc/hosts

Terakhir, *restart* dari layanan bind9 dengan menggunakan perintah “/etc/init.d/bind9 restart”.

Kemudian, lakukan konfigurasi yang sama seperti konfigurasi di atas pada *server backup*. Untuk nama domain dan alamat yang di konfigurasi pada *server backup* harus sama dengan domain dan alamat yang terdapat pada *server* utama, agar tidak terjadi kesalahan dalam

pemanggilan domain apabila *server* sedang melakukan *failover clustering*.

Konfigurasi SSH Pada Server Utama

Pertama ketik perintah `ssh-keygen -t rsa` pada *server* utama, untuk membuat id khusus.

```
root@server15:/var/www/siakad# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:blSk6gSF2Hr1OmOvb/OnMk1Y2ebpRAA4diUhaaw root@node1.smkn15bekasi.sch.com
The key's randomart image is:
+----[RSA 2048]-----+
|
|o=+o
|o=..
|+ . . o
|E . . .
|o oo S
|o oooo..
|_+ = +_
|++ B++=
|oo+#+
+----[SHA256]-----+
root@server15:/var/www/siakad#
```

Gambar 3.14 Pembuatan SSH Tanpa password(1)

copy id selanjutnya kirimkan id tersebut ke *server backup* dengan `ssh-copy-id`. Tahap ini adalah tahap dimana *server* utama mengirimkan id khusus nya ke *server backup*. Agar *server backup* memberikan.

```
root@server15:/var/www/siakad# ssh-copy-id -i ~/.ssh/id_rsa.pub backup15@40.40.50.4
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '40.40.50.4 (40.40.50.4)' can't be established.
ECDSA key fingerprint is SHA256:rSycbsp6ZIVILn04d9hc9;rPW0n0ulcVaPcvfjBEUNQ.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
backup15@40.40.50.4's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'backup15@40.40.50.4'"
and check to make sure that only the key(s) you wanted were added.
root@server15:/var/www/siakad#
```

Gambar 3.15 Pembuatan SSH Tanpa password(2)

Konfigurasi Replikasi Database Mysql Secara dua arah

Pada tahap ini dilakukan proses konfigurasi pada masing - masing *database server*, agar bisa melakukan replikasi dua arah. Proses pertama dilakukan konfigurasi pada *file mysqld.cnf* pada direktori `/etc/mysql/mysql.conf.d/` di *server* utama. Dan merubah data *bind-address*, *server-id* dan *log-bin*.

```
GNU nano 2.5.3 File: /etc/mysql/mysql.conf.d/mysqld.cnf Modified
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 40.40.50.3
```

Gambar 3.16 Konfigurasi Replikasi Database mysqld.cnf server utama (1)

```
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
# other settings you may need to change.
server-id               = 1
log_bin                = /var/log/mysql/mysql-bin.log
expire_logs_days       = 10
max_binlog_size        = 100M
#binlog_do_db          = include_database_name
#binlog_ignore_db      = include_database_name
```

Gambar 3.17 Konfigurasi Replikasi Database mysqld.cnf server utama (2)

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 40.40.50.4
```

Gambar 3.18 Konfigurasi Replikasi Database mysqld.cnf server backup (1)

```
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
# other settings you may need to change.
server-id               = 2
log_bin                = /var/log/mysql/mysql-bin.log
expire_logs_days       = 10
max_binlog_size        = 100M
#binlog_do_db          = include_database_name
#binlog_ignore_db      = include_database_name
```

Gambar 3.19 Konfigurasi Replikasi Database mysqld.cnf server backup (2)

Berikutnya, setelah konfigurasi *file mysqld.cnf* di `/etc/mysql/mysql.conf.d/` pada *server* utama dan *server backup* maka lakukan *restart service mysql*. Selanjutnya, buat *user* pada *mysql server* utama dan *server backup* berikan hak untuk mereplikasi *database* menggunakan *user* yang telah dibuat.

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> create user 'repl'@'%' identified by 'mastertomaster';
Query OK, 0 rows affected (0.02 sec)

mysql> grant replication slave on *.* to 'repl'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> show master status;
+-----+-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000001 | 747 | | | |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 3.20 Konfigurasi Replication User Database server Utama

```
affiliates. Other names may be trademarks of their respective owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> create user 'replica'@'%' identified by 'mastertomaster';
Query OK, 0 rows affected (0.01 sec)

mysql> grant replication slave on *.* to 'replica'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> show master status;
+-----+-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000001 | 753 | | | |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 3.21 Konfigurasi Replication User Database server Backup

Setelah membuat *user* di *server* utama dan *server backup*, maka selanjutnya lakukan konfigurasi untuk merubah data utama pada *database server* utama dan data utama pada *database server backup* dan lakukan pengecekan *slave* pada setiap *server*.

```

root@backup15:/home/backup15# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.27-0ubuntu0.16.04.1-log (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CHANGE MASTER TO
-> MASTER_HOST='40.40.50.3',
-> MASTER_USER='repl',
-> MASTER_PASSWORD='mastertomaster';
Query OK, 0 rows affected, 2 warnings (0.08 sec)

mysql> start slave;
Query OK, 0 rows affected (0.00 sec)

mysql>
    
```

Gambar 3.22 Rubah data utama di server utama

```

root@server15:/var/www/siakad# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.27-0ubuntu0.16.04.1-log (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CHANGE MASTER TO
-> MASTER_HOST='40.40.50.4',
-> MASTER_USER='replica',
-> MASTER_PASSWORD='mastertomaster';
Query OK, 0 rows affected, 2 warnings (0.14 sec)

mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql>
    
```

Gambar 3.23 Rubah data utama di server backup

Berikutnya, cek *slave status* pada *mysql server* utama dan *server backup*.

```

-> MASTER_PASSWORD='mastertomaster';
Query OK, 0 rows affected, 2 warnings (0.14 sec)

mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql> show slave status \G;
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 40.40.50.4
Master_User: replica
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000001
Read_Master_Log_Pos: 753
Relay_Log_File: node1-relay-bin.000002
Relay_Log_Pos: 966
Relay_Master_Log_File: mysql-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
    
```

Gambar 3.24 Status slave server utama

```

-> MASTER_PASSWORD='mastertomaster';
Query OK, 0 rows affected, 2 warnings (0.08 sec)

mysql> start slave;
Query OK, 0 rows affected (0.00 sec)

mysql> show slave status \G;
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 40.40.50.3
Master_User: repl
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000001
Read_Master_Log_Pos: 747
Relay_Log_File: node2-relay-bin.000002
Relay_Log_Pos: 960
Relay_Master_Log_File: mysql-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
    
```

Gambar 3.25 Status slave server backup

Konfigurasi Rsync

Di Tahap konfigurasi ini adalah dengan mengirim data *website* dari *server* utama ke *server backup*.

```

root@server15:/var/www/siakad# rsync -avzh /var/www/siakad/ backup15@40.40.50.4:/var/www/siakad/
    
```

Gambar 3.26 Konfigurasi Web Server Pada Server Backup (1)

Berikutnya, *backup* aplikasi *website* secara berkala menggunakan *service* *crontab* pada *server* utama ke *server backup*.

```

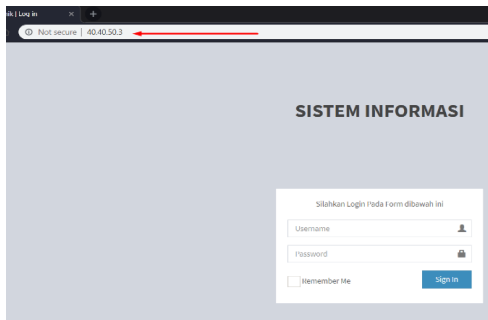
GNU nano 2.5.3 File: /tmp/crontab.6921in/crontab Modified
# daemon's notion of time and timezones.
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
# For example, you can run a backup of all your user accounts
# at 5 a.m. every week with:
# 0 5 * * 1 tar -czf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# h dom mon dow   command
* * * * * rsync -avzh /var/www/siakad/ backup15@40.40.50.4:/var/www/siakad
    
```

Gambar 3.27 Konfigurasi Web Server Pada Server Backup (2)

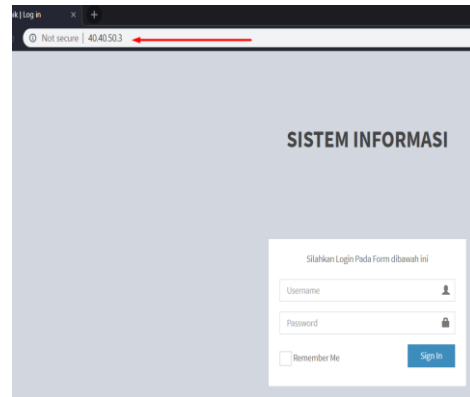
Uji Coba Failover Clustering

Pada bagian ini, uji coba dilakukan dengan cara menjalankan pemanggilan *web server* pada *browser* di sisi *client* menggunakan komputer *client*. Pengujian ini dilakukan terhadap alamat 40.40.50.10 (*virtual IP* pada *cluster* yang dibuat *heartbeat*). Pengujian mengenai *web server* dari *client* menuju alamat 40.40.50.10 berjalan baik pada setiap kondisi berikut :

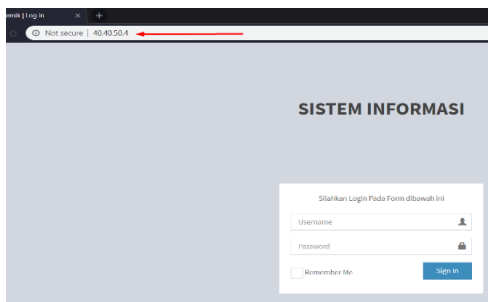
1. *Server* utama dan *server backup* dalam kondisi hidup.



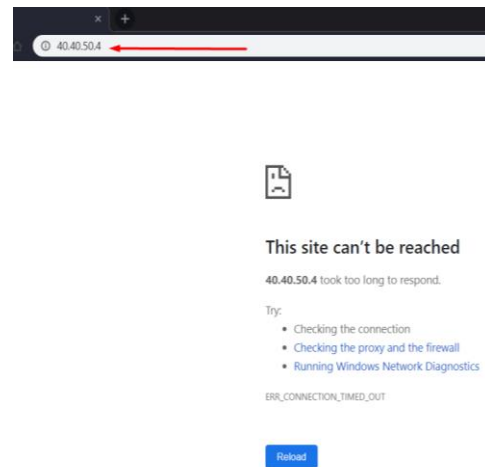
Gambar 3.28 Uji Coba Pemanggilan Ip Server Utama di browser Client (1)



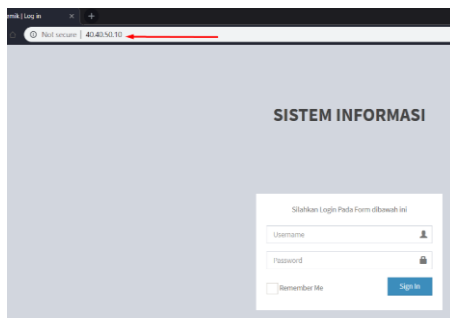
Gambar 3.31 Uji Coba Pemanggilan Ip Server Utama di browser Client (2)



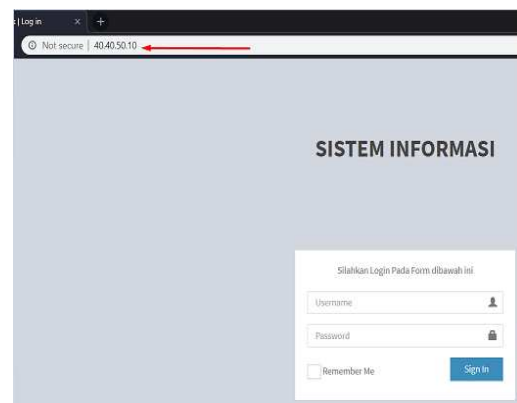
Gambar 3.29 Uji Coba Pemanggilan Ip Server Backup di browser Client (1)



Gambar 3.32 Uji Coba Pemanggilan Ip Server Backup di browser Client (2)



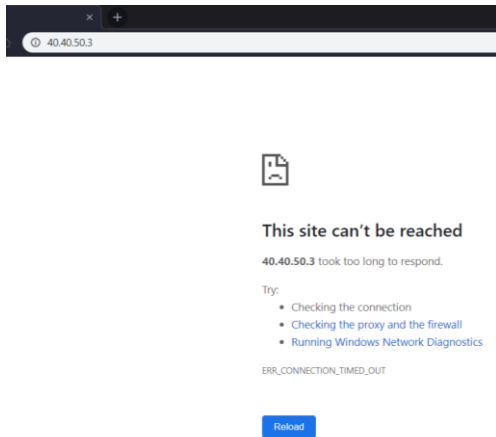
Gambar 3.30 Uji Coba Pemanggilan Ip Virtual di browser Client (1)



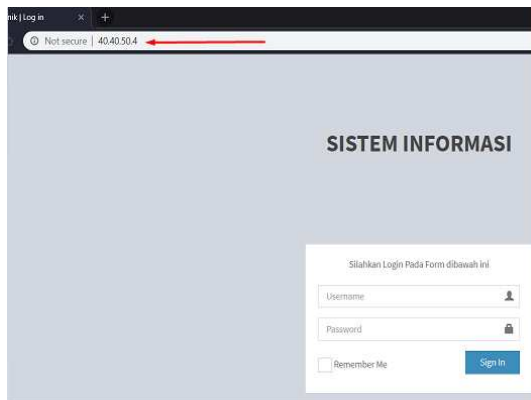
Gambar 3.33 Uji Coba Pemanggilan Ip Virtual di browser Client (2)

2. *Server* Utama dalam kondisi hidup dan *server backup* dalam kondisi mati atau down

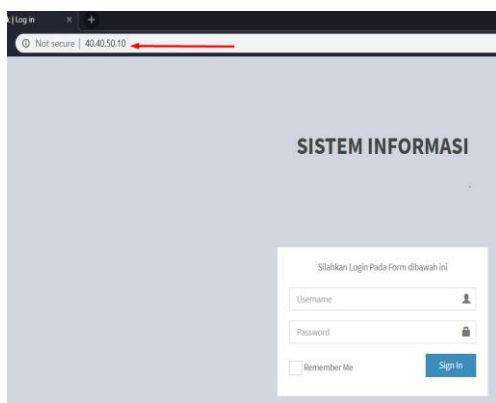
3. *Server* Utama dalam kondisi mati atau down dan *server backup* dalam kondisi hidup.



Gambar 3.34 Uji Coba Pemanggilan Ip Server Utama di browser Client (3)



Gambar 3.35 Uji Coba Pemanggilan Ip Server Backup di browser Client (3)



Gambar 3.36 Uji Coba Pemanggilan Ip Virtual di browser Client (3)

Uji Coba Akses Website Menggunakan DNS Melalui Client

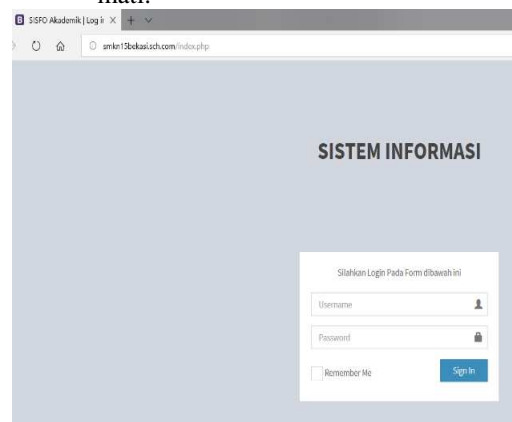
Pada bagian ini, uji coba dilakukan dengan cara mengakses nama domain pada *server* melalui *browser* dari *client* menuju alamat *smk15bekasi.sch.com*. Pengujian mengenai koneksi dari *client* menuju alamat *smkn15bekasi.sch.com* berjalan baik pada setiap kondisi berikut

1. *Server* utama dan *server backup* dalam kondisi hidup.



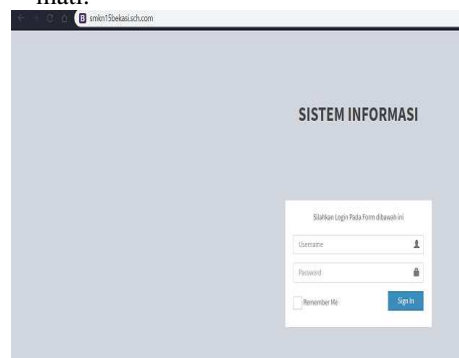
Gambar 3.37 Uji Coba Akses Website Menggunakan DNS (1)

2. *Server* utama dalam kondisi hidup, sementara *server backup* dalam kondisi mati.



Gambar 3.38 Uji Coba Akses Website Menggunakan DNS (2)

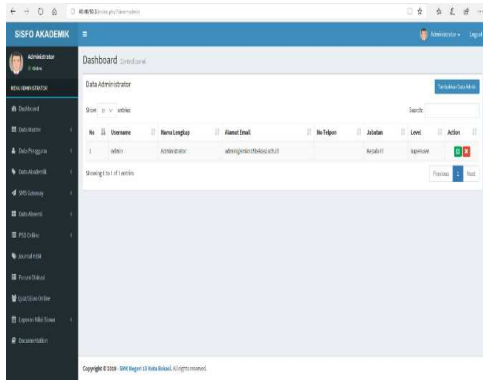
3. *Server backup* dalam kondisi hidup, sementara *server* utama dalam kondisi mati.



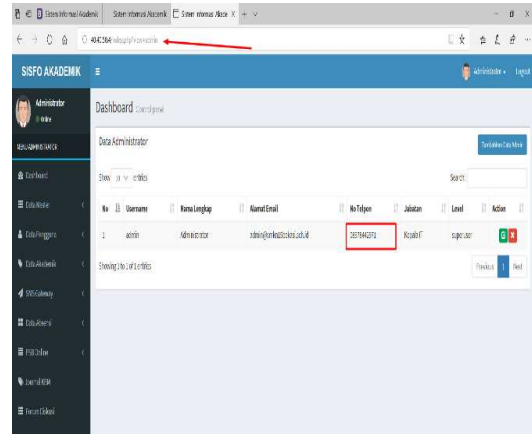
Gambar 3.39 Uji Coba Akses Website Menggunakan DNS (3)

Uji Hasil Replikasi Database

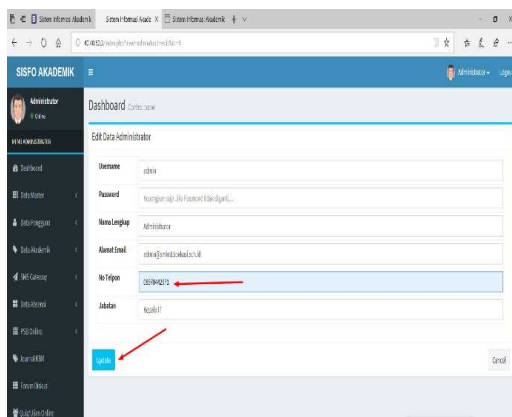
Pada bagian uji replikasi dilakukan pengujian dengan cara memanggil *web server* dengan melalui ip *server* utama dan melakukan proses edit data admin dengan menambahkan nomor telepon.



Gambar 3.40 Uji Coba Update Data Admin (1)

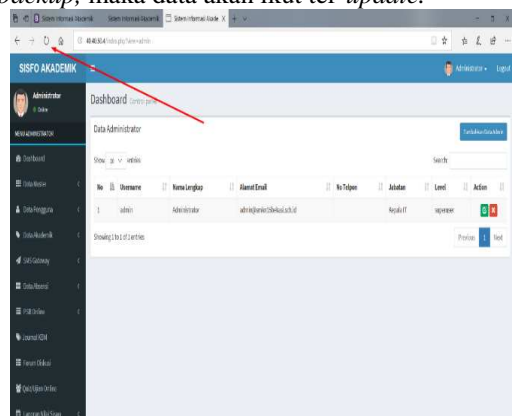


Gambar 3.43 Uji Coba Update Data Admin (4)



Gambar 3.41 Uji Coba Update Data Admin (2)

selanjutnya lakukan proses pengecekan data pada server backup dengan memanggil alamat ip server backup lalu cek data admin pada server backup, maka data akan ikut ter-update.



Gambar 3.42 Uji Coba Update Data Admin (3)

UJI HASIL KIRIM DATA OTOMATIS

Pada tahap ini, melakukan uji coba dari hasil *rsync* dan layanan *crontab* yang telah di konfigurasi pada directory */var/www/siakad* dengan membuat sebuah file baru dengan perintah *touch*.

Cek direktory */var/www/siakad* pada server backup

```
root@node2:/var/www/siakad# ls
40.40.50.4      files          main-header.php  print-psb2.php
application    footer.php     masterdump.sql   print-psb3.php
backup15@40.40.50.4  foto_pegawai menu-admin.php   print_raport
bootstrap      foto_siswa    menu-guru.php    print-raport_uts.php
build          index.php     menu-kepek.php   print-siswa.php
config         info user.txt menu-siswa.php   psb
dist           login.php     pages            siakad
dokumentasi   logo.png     plugins          siakad.sql
download.php  logout.php   print-psbl.php
root@node2:/var/www/siakad#
```

Gambar 3.44 Uji Coba Hasil Layanan Rsync dan Crontab (1)

Cek dan Buat file pada directory */var/www/siakad* pada server utama dengan perintah *touch* file_baru.txt dan edit file_baru.txt isi file dengan "Rsync sukses"

```
root@node1:/home/server15# cd /var/www/siakad
root@node1:/var/www/siakad# ls
application  footer.php     main-header.php  print-psb2.php
bootstrap    foto_pegawai  menu-admin.php   print-psb3.php
build        foto_siswa    menu-guru.php    print_raport
config       index.php     menu-kepek.php   print-raport_uts.php
dist         info user.txt menu-siswa.php   print-siswa.php
dokumentasi login.php     pages            psb
download.php logo.png     plugins          siakad
files        logout.php   print-psbl.php  siakad.sql
root@node1:/var/www/siakad#
root@node1:/var/www/siakad# touch file_baru.txt
root@node1:/var/www/siakad#
```

Gambar 3.45 Uji Coba Hasil Layanan Rsync dan Crontab (2)

```
GNU nano 2.5.3      File: file_baru.txt
Rsync Sukses
```

Gambar 3.46 Uji Coba Hasil Layanan Rsync dan Crontab (3)

Cek directory /var/www/siakad pada server backup dengan perintah ls, cek apakah file_baru.txt sudah masuk.

```
root@node2:/var/www/siakad# ls
40.40.50.4      file_baru.txt  logout.php    print-psbl.php
application    files         main-header.php print-psb2.php
backups15840.40.50.4 footer.php    masterdump.sql print-psb3.php
bootstrap      foto_pegawai menu-admin.php print_report
build          foto_siswa   menu-guru.php print-report_uts.php
config         index.php    menu-kepek.php print-siswa.php
dist          info_user.txt menu-siswa.php psb
dokumentasi   login.php    pages        siakad
download.php  logo.png    plugins      siakad.sql
root@node2:/var/www/siakad#
```

Gambar 3.47 Uji Coba Hasil Layanan Rsync dan Crontab (4)

```
GNU nano 2.5.3      File: file_baru.txt
Rsync Sukses
```

Gambar 3.48 Uji Coba Hasil Layanan Rsync dan Crontab (5)

TABEL 4. 3 Hasil Pengujian

Uji Hasil	Hasil	Rate
Failover Clustering	Sukses	100%
Replication Database	Sukses	100%
Replication Mirror	Sukses	100%

4. KESIMPULAN

Dari hasil percobaan, maka dapat disimpulkan hal – hal sebagai berikut :

1. *Replication mirror* dan dengan bantuan *failover clustering* yang telah dirancang, dapat menjadi pilihan solusi yang terbaik, karena dapat menggantikan *server* yang mati ke *server* yang lain dan *client* tetap bisa mengakses *web server*.
2. Meningkatkan nilai ketersediaan yang tinggi atau *high availability* melalui metode *failover clustering* dan *replication mirror* menjadi solusi bagi SMK Negeri 15 Kota Bekasi yang memiliki infrastruktur *client – server* sebagai instansi pendidikan yang mulai meningkat saat ini..
3. Proses *replication database* yang dirancang ini berfungsi mendukung *high availability*, karena replikasi *database* ini hasilnya sama persis dengan *server* utama sehingga terjadinya kerusakan dan kehilangan data dapat ditangani.

Saran – saran yang dapat dilakukan untuk pengembangan lebih lanjut dari skripsi ini adalah :

1. Agar keamanan data lebih terjaga dari bencana seperti kebakaran dan lain-lain diharapkan untuk menggunakan layanan pihak ketiga sebagai data *backup* penyimpanan data.

2. Agar sistem replikasi *database* dua arah antara *server* aman sebaiknya dibuat pengamanan dengan mekanisme pengamanan SSL.

PUSTAKA

Athailah. 2012. *Panduan Membuat Mail Server dengan Zimbra*. Jakarta: Jasakom

Azikin, Askari. 2011. *Debian GNU/Linux*. Informatika Bandung. Bandung.

Hakim, Abdul et. all. 2018. *Implementasi Failover Clustering Server Untuk Mengurangi Resiko Downtime Pada Web Server*.

Irfani. 2015. *Implementasi High Availability Server Dengan Teknik Failover Virtual Computer Cluster*.

Komariyah, Fitri dan Argyawati, Harum. 2016. *Implementasi Server Cluster High Availability Pada Web Server dengan Sistem Operasi Turnkey Linux Menggunakan Heartbeat*. Jurnal Penelitian Ilmu Komputer, System Embedded & Logic. Universitas Islam “45”

Lumingkewas, Albert D. et. all. 2016. *RSYNC dalam jurnalnya “Perancangan Dan Implementasi Gateway Redundancy Untuk Peningkatan Reliabilitas Jaringan Menggunakan Protokol CARP”*

Maulana, Halim. 2016. *Analisis Dan Perancangan Sistem Replikasi Database Mysql Dengan Menggunakan Vmware Pada Sistem Operasi Open Source*.

Palit, Randi V. et. all. 2015. *Rancangan Sistem Informasi Keuangan Gereja Berbasis Web Di Jemaat GMIM Bukit Moria Malalayang*. E-Journal Teknik Elektro dan Komputer vol. 4 no. 7. Unsrat

Prismana, I Gusti Lanang Putra Eka dan Supramana. 2016. *IMPLEMENTASI LOAD BALANCING PADA WEB SERVER DENGAN MENGGUNAKAN APACHE*. Universitas Negeri Surabaya

Purwadi, Maskur, et. all. 2018. *High Availability Controller Software Defined Network Menggunakan Heartbeat dan DRBD*

Ratomil, Husnul dan Nadhori, Isbat Uzzin. 2013. *MANAJEMEN BACKUP DATA OTOMATIS PADA JARINGAN MENGGUNAKAN RSYNC*. Politeknik Elektronika Negeri Surabaya. Institut Teknologi Sepuluh Nopember

Saputra, Adhe. 2016. *PENGEMBANGAN JARINGAN WIRELESS LOCAL AREA*

*NETWORK (WLAN) MENGGUNAKAN
METODE PPDIOO.* Fakultas Ilmu Komputer
Universitas Binadarma

Suparwita, I Putu Eka. 2012. *IMPLEMENTASI
SISTEM BACKUP OTOMATIS VIRTUAL
PRIVATE SERVER DENGAN CRONTAB.*
Jurnal Elektronik Ilmu Komputer - Universitas
Udayana

Subekti, Zaenal Mutaqin Kurniawan, Rizky. 2019.
*PERANCANGAN JARINGAN VOIP
BERBASIS OPEN SOURCE DENGAN DNS
PADA MIKROTIK.* Jurnal Cendekia – AMIK
Dian Cipta Cendekia

Wikipedia. 2019. *Sistem operasi.*
https://id.wikipedia.org/wiki/Sistem_operasi

Wikipedia. 2019. *Rsync.*
<https://en.wikipedia.org/wiki/Rsync>