



RANCANGAN REDUDANSI *SESSION BORDER CONTROLLER(SBC)*

FREESWITCH

MENGGUNAKAN *PACEMAKER – COROSYNC*

PADA PT. BATAM BINTAN TELEKOMUNIKASI (OFON)

TUGAS AKHIR

**UNIVERSITAS
MERCU BUANA**

Wisnu Nur Hidayat

41517310013

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MERCU BUANA

JAKARTA

2021



**RANCANGAN REDUDANSI *SESSION BORDER CONTROLLER(SBC)*
FREESWITCH
MENGUNAKAN *PACEMAKER – COROSYNC*
PADA PT. BATAM BINTAN TELEKOMUNIKASI (OFON)**

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:

Wisnu Nur Hidayat

41517310013

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA**

2021

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM : 41517310013

Nama : Wisnu Nur H

Judul Tugas Akhir : Rancangan Redudansi *Session Border Controller (SBC) FreeSwitch*
Menggunakan *Pacemaker – Corosync* Pada PT. Batam Bintang
Telekomunikasi (OFON)

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 13 Agustus 2021



Wisnu Nur Hidayat



UNIVERSITAS
MERCU BUANA

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Wisnu Nur Hidayat
NIM : 41517310013
Judul Tugas Akhir : Rancangan Redudansi *Session Border Control (SBC) FreeSwitch* Menggunakan *Pacemaker-Corosync* Pada PT, Batam Bintang Telekomunikasi (OFON)

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Non eksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Non eksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 13 Agustus 2021


Wisnu Nur Hidayat

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Wisnu Nur Hidayat
 NIM : 41517310013
 Judul Tugas Akhir : Rancangan Redudansi *Session Border Controller (SBC) FreeSwitch* Menggunakan *Pacemaker – Corosync* Pada PT. Batam Bintang Telekomunikasi (OFON)

Menyatakan bahwa :

1. Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis		Status	
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi		Diajukan	✓
		Jurnal Nasional Terakreditasi	✓		
		Jurnal International Tidak Bereputasi		Diterima	
		Jurnal International Bereputasi			
Disubmit/dipublikasikan di :	Nama Jurnal	: JTIK			
	ISSN	: 2355-7699, e-ISSN: 2528-6579.			
	Link Jurnal	: https://jtiik.ub.ac.id/			
	Link File				
	Jurnal Jika	:			
	Sudah di Publish				

2. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.
3. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan

Demikian pernyataan ini saya buat dengan sebenarnya.

Mengetahui
Dosen Pembimbing TA

Jakarta, 13 Agustus 2021



Sri Dianing Asri, ST, M.Kom

Wisnu Nur Hidayat



LEMBAR PERSETUJUAN

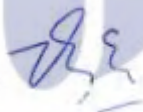
<info: hanya untuk syarat sidang, **tidak dilampirkan** di laporan akhir/yudisium>

Nama Mahasiswa : Wisnu Nur Hidayat
NIM : 41517310013
Judul Tugas Akhir : Rancangan Redudansi *Session Border Controller*
(SBC) *FreeSwitch* Menggunakan *Pacemaker* –
Corosync Pada PT. Batam Bintang Telekomunikasi
(OFON)

Tugas Akhir ini telah diperiksa dan disetujui

Jakarta, 13 Agustus 2021

Menyetujui,



(Sri Dianing Asri, ST, M.Kom)

Dosen Pembimbing

UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI

NIM : 41517310013

Nama : Wisnu Nur Hidayat

Judul Tugas Akhir : Rancangan Redudansi *Session Border Controller (SBC) FreeSwitch* Menggunakan *Pacemaker – Corosync* Pada PT. Batam Bintang Telekomunikasi (OFON)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 13 Agustus 2021



UNIVERSITAS
MERCU BUANA
(Giri Purnama, S.Pd., M.Kom)

LEMBAR PERSETUJUAN PENGUJI

NIM : 41517310013


Nama : Wisnu Nur Hidayat

Judul Tugas Akhir : Rancangan Redudansi *Session Border Controller (SBC) FreeSwitch* Menggunakan *Pacemaker – Corosync* Pada PT. Batam Bintang Telekomunikasi (OFON)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 13 Agustus 2021

UNIVERSITAS
MERCUBUANA


(Sukma Wardhana, S.Kom, M.Kom.)

LEMBAR PERSETUJUAN PENGUJI

NIM : 41517310013

Nama : Wisnu Nur Hidayat

Judul Tugas Akhir : Rancangan Redudansi *Session Border Controller (SBC) FreeSwitch* Menggunakan *Pacemaker – Corosync* Pada PT. Batam Bintang Telekomunikasi (OFON)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 13 Agustus 2021



UNIVERSITAS
MERCUBUANA
(Wawan Gunawan, S. Kom, MT)

LEMBAR PENGESAHAN

NIM : 41517310013
Nama : Wisnu Nur Hidayat
Judul Tugas Akhir : Rancangan Redudansi *Session Border Controller (SBC)*
FreeSwitch Menggunakan *Pacemaker – Corosync* Pada
PT. Batam Bintang Telekomunikasi (OFON)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 13 Agustus 2021

Menyetujui,

(Sri Dianing Asri, ST, M.Kom)

Dosen Pembimbing

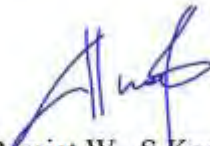
UNIVERSITAS
MERCU BUANA

Mengetahui,



(Wawan Gunawan, S.Kom, MT)

Koord. Tugas Akhir Teknik Informatika



(Herry Derajat W., S.Kom, MM)

Ka. Prodi Teknik Informatika

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Allah SWT atas segala rahmat dan nikmat-Nya yang telah diberikan, sehingga penulis dapat menyelesaikan Tugas Akhir sebagai persyaratan guna memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Herry Derajat W., S.Kom, MM selaku kepala program studi Teknik Informatika.
2. Ibu Sri Dianing Asri, ST, M.Kom. selaku Dosen Pembimbing.
3. Team Technology OFON yang telah membantu dan mengarahkan dalam proses penelitian.
4. Teristimewa kepada keluarga besar penulis, Ayahanda tercinta Maryono, Almarhumah Ibunda tersayang Sukarti, semua Saudara dan Saudari penulis yang telah memberikan kasih sayang dan juga dukungan secara moril maupun materil serta doa tulus yang tiada hentinya kepada penulis.
5. Fauzia Allamatul Fithri yang telah membantu dan memberi dukungan kepada penulis dalam penyelesaian Tugas Akhir ini.
6. Semua rekan dan teman yang tidak dapat penulis sebutkan satu persatu yang mana banyak memberi doa dan dukungan kepada penulis.

Akhir kata, penulis berharap Tugas Akhir ini dapat bermanfaat bagi rekan – rekan mahasiswa dan juga para pembaca sekalian untuk menjadi pembelajaran ataupun refrensi. Semoga Allah SWT selalu memberikan rahmat dan nikmat-Nya kepada kita semua.

Jakarta, 15 Juli 2021

Wisnu Nur Hidayat

DAFTAR ISI

<u>HALAMAN SAMPUL</u>	i
<u>HALAMAN JUDUL</u>	i
<u>LEMBAR PERNYATAAN ORISINALITAS</u>	ii
<u>SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR</u>	iii
<u>SURAT PERNYATAAN LUARAN TUGAS AKHIR</u>	iv
<u>LEMBAR PERSETUJUAN</u>	vi
<u>LEMBAR PERSETUJUAN PENGUJI</u>	vii
<u>LEMBAR PENGESAHAN</u>	x
<u>ABSTRAK</u>	xi
<u>ABSTRACT</u>	xii
<u>KATA PENGANTAR</u>	xiii
<u>DAFTAR ISI</u>	xiv
<u>NASKAH JURNAL</u>	2
<u>KERTAS KERJA</u>	7
<u>BAB 1. LITERATUR REVIEW</u>	17
<u>BAB 2. ANALISIS DAN PERANCANGAN</u>	20
<u>BAB 3. SOURCE CODE</u>	25
<u>BAB 4. DATASET</u>	35
<u>BAB 5. TAHAPAN EKSPERIMEN</u>	38
	xiv

<u>BAB 6. HASIL SEMUA EKSPERIMEN</u>	42
<u>DAFTAR PUSTAKA</u>	55
<u>LAMPIRAN DOKUMEN HAKI</u>	57
<u>LAMPIRAN KORESPONDENSI</u>	60



NASKAH JURNAL

RANCANGAN REDUDANSI SESSION BORDER CONTROLLER (SBC) FREESWITCH MENGGUNAKAN PACEMAKER – COROSYNC PADA PT. BATAM BINTAN TELEKOMUNIKASI (OFON)

Wisnu Nur H^{*1}, Sri Dianing Asri, ST, M.Kom^{*2}

¹Wisnu Nur Hidayat

²Sri Dianing Asri, ST, M.Kom

Email: ¹41517310013@mercubuana.ac.id, ²dianing.asri@gmail.com

*Penulis Korespondensi

(Naskah masuk: dd mmm yyyy, diterima untuk diterbitkan: dd mmm yyyy)

Abstrak

Perkembangan teknologi komunikasi sangat cepat berkembang. Dari yang kita tahu bahwa pada awalnya kita hanya mengenal telepon analog yang mana masih menggunakan sambungan kabel secara langsung agar dapat melakukan panggilan. Semakin canggih perkembangan telepon, hingga saat ini banyak yang sudah mengenal *Voice Over Internet Protocol* (VoIP) sebagai solusi telepon yang lebih fleksibel dimana kita dapat menggunakan panggilan selama kita terhubung dengan internet dimanapun kita berada. OFON sebagai salah satu penyedia layanan telepon dengan menggunakan teknologi VoIP tersebut tentu ingin menyediakan layanan terbaik bagi masyarakat, dimana OFON menjanjikan *Service Level Agreement* (SLA) sebesar 95% layanan akan selalu berjalan. Penelitian ini bertujuan untuk mengembangkan server VoIP dimana dengan kebutuhan pada perusahaan yang menargetkan SLA 95% tersebut dapat tercapai. Untuk itu diperlukan persiapan yang matang dalam penyiapan server yang digunakan untuk melayani pelanggan. Kami merancang redudansi server VoIP yang akan digunakan sebagai *Session Border Controller* (SBC) yang mana akan digunakan sebagai pengatur lalulintas jalur panggilan bagi pelanggan.

Kata kunci: VoIP, Session Border Controller (SBC), Redudansi, High Availability

FREESWITCH SESSION BORDER CONTROLLER (SBC) REDUNDANCY DESIGN USING PACEMAKER – COROSYNC AT PT. BATAM BINTAN TELECOMMUNICATIONS (OFON)

Abstract

The development of communication technology is growing very fast. At first we only knew analog telephones which still used a direct cable connection to make calls. Until now many people are familiar with Voice Over Internet Protocol (VoIP) as a more flexible telephone solution, we can use calls as long as we are connected to the internet wherever we are. OFON as a telephone service provider using VoIP technology certainly wants to provide the best service for the customer. OFON promises a Service Level Agreement (SLA) of 95% the service will always run properly. This study goals to develop a VoIP server where OFON 95% SLA can be achieved. For this reason, careful preparation is needed in preparing the server used to serve customers. We designed a redundant VoIP server that will be used as a Session Border Controller (SBC) which will be used as a traffic controller for customers calls..

Keywords: VoIP, Session Border Controller (SBC), Redundancy, High Availability

PENDAHULUAN

Komunikasi adalah kebutuhan dasar bagi manusia, seperti yang kita tahu bahwa manusia adalah makhluk sosial yang mana perlu saling berinteraksi satu sama lain. Dengan kita berkomunikasi dengan orang lain banyak hal yang bisa kita dapatkan seperti informasi terbaru, kabar mengenai sanak keluarga, dan lainnya.

Seperti yang dijelaskan sebelumnya bahwa komunikasi menjadi kebutuhan mendasar. Berdasarkan hal tersebut banyak pengembang teknologi yang mengembangkan alat telekomunikasi, agar dapat terjadinya komunikasi jarak jauh melalui teknologi saat ini.

Perkembangan teknologi komunikasi tidak dapat dipungkiri bahwa hal tersebut berkembang dengan cepat. Pada awalnya kita hanya mengenal telepon analog, kemudian berkembang menjadi *handphone*, dan untuk saat ini sudah banyak beredar menggunakan *Voice Over IP* (VoIP).

OFON sebagai salah satu penyedia layanan telepon berbasis VoIP. Ingin memberikan layanan terbaik bagi pelanggannya. Dimana pelanggan merasa layanan telepon yang digunakan jauh lebih fleksibel namun juga memiliki kualitas yang baik dari segi suara dan juga dalam segi ketersediaan layanan tersebut. Untuk itu perlunya perancangan yang matang agar dapat terciptanya kondisi *high availability* agar dapat meminimalisir gangguan yang dirasakan oleh pelanggan.

TEORI DAN PERANCANGAN

2.1 LANDASAN TEORI

Voice Over Internet Protocol (VoIP)

Menurut [Fanani, 2019] VoIP (*Voice over Internet Protocol*) adalah teknologi yang mampu melewati trafik suara, video dan data yang berbentuk paket secara *real-time* dengan jaringan IP (*Internet Protocol*). VoIP ini dapat memanfaatkan infrastruktur internet yang sudah ada untuk berkomunikasi seperti layaknya menggunakan telepon biasa dan tidak dikenakan biaya telepon biasa untuk berkomunikasi dengan pengguna VoIP lainnya. Teknologi ini memungkinkan komunikasi suara menggunakan jaringan berbasis IP (*internet protocol*) untuk dijalankan diatas infrastruktur jaringan *packet network*. Teknologi ini bekerja

dengan jalan mengubah suara menjadi format digital tertentu yang dapat dikirimkan melalui jaringan IP.

Session Initiation Protocol (SIP)

Session Initiation Protocol (SIP) adalah protocol yang berjalan pada *layer* ke 3 dalam *osi layer* yaitu pada *layer application*. Dimana SIP bertugas sebagai pengatur berjalannya sesi panggilan dalam VoIP dari proses awal panggilan hingga berakhirnya panggilan. Dalam setiap sesi SIP dibantu oleh protocol lainnya, yaitu *Real Time Protocol* (RTP), *Real Time Control Protocol* (RTCP), dan juga *Session Description Protocol* (SDP). RTP digunakan sebagai transmisi media panggilan yang mana suara kita dirubah menjadi sinyal digital dan dikirimkan melalui jaringan IP. Dan juga RTCP digunakan sebagai *monitor* yang akan memberikan laporan yang menjadi parameter kualitas suara dalam sesi panggilan tersebut. Sedangkan SDP sendiri digunakan sebagai protocol yang mendeskripsikan alamat IP yang akan digunakan sebagai media dalam sesi panggilan tersebut.

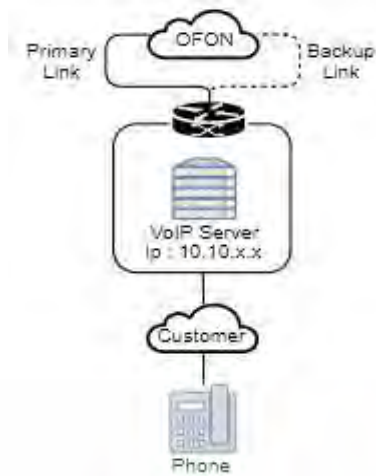
Session Border Controller (SBC)

Session Border Controller (SBC) adalah sebuah *server* yang diterapkan pada jaringan VoIP, yang bertugas sebagai pelindung, pengontrol setiap sesi panggilan VoIP yang terjadi. Dalam sebuah sesi panggilan VoIP ada 2 *end-point* yang disebut *user agent* (ua). SBC berfungsi sebagai pengontrol sesi panggilan dimana *user agent* pertama yang melakukan panggilan dan *user agent* kedua yang menerima panggilan tersebut hanya mengetahui alamat IP dari SBC. Ini berfungsi sebagai salah satu bentuk perlindungan agar masing – masing *end point* tidak mengetahui alamat IP dari jaringan lawan bicaranya.

High Availability (HA)

Menurut [Umam, 2018] *High availability* (HA) adalah suatu kemampuan dari suatu sistem untuk melakukan fungsinya secara berkesinambungan (tanpa adanya interupsi) untuk jangka waktu lebih lama dari pada ketahanan yang di berikan oleh masing-masing komponennya. *Availability* biasanya di ukur berdasarkan istilah *nine* semakin banyak *nine* maka semakin tinggi sebuah sistem *availability*. Mengapa *nine* karena tidak mungkin mendapatkan ketersediaan sebesar seratus persen

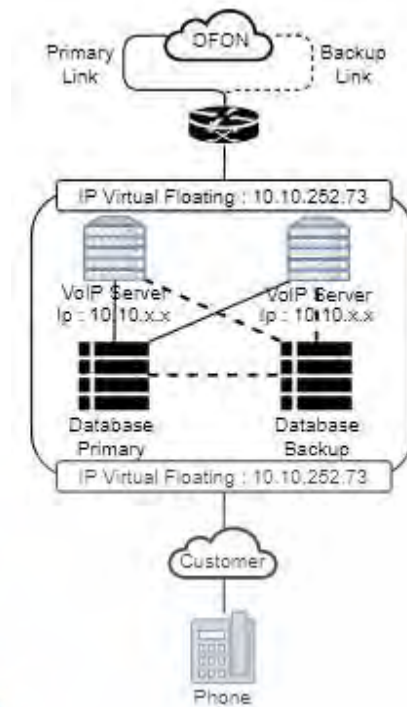
2.2 TOPOLOGI AWAL



Gambar 1. Topologi awal jaringan VoIP OFON.

Dapat dilihat pada gambar 1 bahwa saat ini jaringan VoIP yang saat ini berjalan hanya memiliki redundansi dalam segi jaringan. Namun hal itu menjadi salah satu kondisi *single point of failure*, yang berarti bila kendala yang terjadi bukan pada jaringannya namun pada *server* VoIP yang mengatur layanan panggilan, sehingga menyebabkan terjadinya kendala yang akan dirasakan oleh pelanggan. Untuk itu perlunya *upgrade* pada jaringan VoIP yang saat ini berjalan sehingga dapat memberikan pelayanan telepon yang lebih baik pada pelanggan.

2.3 TOPOLOGI RANCANGAN



Gambar 2. Rancangan topologi redundansi.

Seperti yang dapat dilihat pada gambar 2 diatas, dari topologi tersebut maka adanya *backup server* yang akan memberikan layanan ketika *primary server* sedang mengalami kendala. Metode yang digunakan adalah *active standby* dimana *primary server* yang akan memberikan layanan telepon pada pelanggan, namun jika pada *primary server* mengalami kendala maka *backup server* yang akan menjalankan tugas tersebut.

Dengan adanya redundansi pada *server* dan juga pada *database* maka diharapkan dapat meminimalisir kendala yang akan dirasakan oleh pelanggan. Dengan begitu OFON dapat mempertahankan kestabilan layanan telepon yang diberikan. Dengan rancangan tersebut maka secara fungsi atau *logical* kedua *server* tersebut memiliki satu ip yang sama yaitu 10.10.252.73 , namun secara fisik memiliki dua *server* dengan dua alamat ip yang berbeda.

Dapat dilihat pada tabel 1 sebagai perbandingan pada topologi awal dan juga topologi rancangan yang telah dibuat. Dengan topologi rancangan ini diharapkan agar dapat memberi pelayanan telepon yang lebih maksimal dan dapat mencegah masalah yang akan terjadi.

Tabel 1. Perbandingan topologi awal dengan rancangan.

	Redudansi Jaringan	Redudansi Server	Redudansi Database
Topologi awal	Aktif	Tidak aktif	Tidak aktif
Topologi rancangan	Aktif	Aktif	Aktif

TAHAPAN UJICoba

Pada tahap ini maka akan dilakukan instalasi server sesuai dengan yang sudah dirancang dan akan dilakukan ujicoba agar menjadi bukti bahwa rancangan tersebut dapat berjalan sebagaimana mestinya.

3.1. TOOLS

Dalam penelitian ini ada beberapa perlengkapan yang perlu disiapkan penulis, yaitu sebagai berikut :

- 2 Virtual Machine dengan spesifikasi : 2 core cpu, RAM 2GB dan penyimpanan 20GB.
- Sistem operasi Centos 7.
- *Service* pada sistem operasi linux seperti : *wget*, *nano*, *pacemaker – corosync*, *fusionpbx*, *freeswitch*, *postgresql*.

3.2. UJI COBA RANCANGAN REDUDANSI

Membangun sistem redudansi dengan metode *high availability* tentu memerlukan standarisasi target yang perlu dicapai, berikut untuk target *Quality Of Service* (QOS) :

Tabel 2. Target QOS yang perlu dicapai.

#	Target QOS	Status	Keterangan
1	<i>Floating IP</i>	Aktif	Minimal 2 server dengan 1 <i>virtual floating IP</i> .
2	<i>Backup Database</i>	Aktif	Mode <i>master and slave</i> , dimana <i>database slave</i> akan meng-copy berkas pada <i>database master</i> dan dapat mem-backup ketika <i>database master</i> sedang <i>down</i> .

Berikut hasil uji coba redudansi pada server VoIP SBC untuk memenuhi QOS Floating IP :

- *Ping* dari *client* ke *primary server*, *backup server* dan *floating ip*.

Melakukan ping dari *client* ke ip 10.10.252.71 sebagai *primary server*, ping dari *client* ke ip 10.10.252.72 sebagai *backup server* dan ke ip 10.10.252.73 sebagai *floating virtual ip* dengan hasil yang dapat dilihat pada tabel 3 berikut :

Tabel 3. Hasil ping dari client ke server.

Server	Bytes	Time	TTL	Packet loss
<i>Primary</i>	32 ms	0,8 ms	60 ms	0%
<i>Backup</i>	32 ms	0,9 ms	60 ms	0%
<i>Floating ip</i>	32 ms	4 ms	60 ms	0%

- Uji coba ping pada *floating service*.

Setelah melakukan *test ping* dari *client* ke masing – masing *server* dalam kondisi normal. Maka akan dilakukan uji coba dengan skema sebagai berikut :

- *Ping* dari *client* ke semua *server* dan *floating ip* dengan jumlah 20 kali ping.
- Matikan *primary server* ketika ping mencapai 10 kali.
- Apakah *floating ip* tetap berjalan?
- Apakah *backup server* telah aktif?
- Apakah mengalami RTO? Berapa %?
- Apakah *latency ping floating ip* naik?

Hasil uji coba dapat dilihat pada tabel 4 berikut :

Tabel 4. Hasil uji coba floating ip.

Server	Bytes	Time	TTL	Packet loss
<i>Primary</i>	32 ms	0,8 ms	60 ms	50%
<i>Backup</i>	32 ms	0,9 ms	60 ms	0%
<i>Floating ip</i>	32 ms	24 ms	60 ms	0%

Dapat dilihat dari tabel 4 diatas, bahwa pengetesan dari segi ping tidak banyak mempengaruhi ping kearah *floating ip* yang mana akan dipublikasikan ke pelanggan sebagai alamat *ip server SBC OFON*.

- Uji coba *database*.

Setelah dilakukan uji coba ping pada *floating ip*. Melihat dari tabel 4, redudansi melalui *service floating ip* sudah berhasil dan tentu saja dengan *service database* yang berada pada *localhost* masing – masing *server*.

Sehingga uji coba *database* dilakukan dengan cara menguji *backup*

server untuk menyalin data secara otomatis dari *primary server*.

Hasil uji coba dapat dilihat pada tabel 5 berikut :

Tabel 5. Hasil uji coba penambahan data pada *database*.

<i>Server</i>	Data awal	Penambahan data manual	Data bertambah otomatis	Total data
<i>Primary</i>	5	2	0	7
<i>Backup</i>	5	0	2	7

Dari tabel 5 tersebut ketika ada penambahan data pada *primary server* maka secara otomatis *backup server* akan menyalin data yang telah ditambahkan ke *primary server*. Dengan begitu ketika *primary server* terjadi kendala maka tidak akan terjadi *miss* pada data *backup server*.

KESIMPULAN

Dari hasil uji coba yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut :

- Redudansi berjalan dengan baik seperti yang diharapkan. Dimana dengan menggunakan *floating ip* sangat membantu dalam rancangan redudansi.
- Service pacemaker – corosync* sebagai *service* yang menjalankan fungsi *floating ip* dapat berjalan dengan baik.
- Database postgresql* dapat menjalankan fungsi redudansi dengan mode *master – slave*.
- Penggunaan *floating ip* akan memberikan keamanan ekstra pada infrastruktur VoIP kita, dikarenakan pelanggan ataupun mitra bisnis tidak akan mengetahui alamat ip dari *server* yang sebenarnya.

DAFTAR PUSTAKA

- C. Umam, L. B. Handoko, and G. M. Rizqi, "Implementation And Analysis High Availability Network File System Based Server Cluster," *J. Transform.*, vol. 16, no. 1, p. 31, 2018, doi: 10.26623/transformatika.v16i1.841.
- D. Setiawan, A. F. Rochim, and R. Isnanto, "Voice over Internet Protocol (VoIP) Menggunakan Asterisk Sebagai Session Initiation Protocol (SIP) Server," 2011, [Online]. Available: <http://eprints.undip.ac.id/25422/>.
- E. Saputra and I. Lestari, "Analisa dan Perancangan Voice over Internet Protokol (VoIP) Menggunakan Teknologi Open Source pada Pusat Teknologi Informasi dan Pangkalan Data UIN SUSKA Riau," *J. Sains, Teknol. dan Ind.*, vol. 12, no. 1, pp. 106–111, 2014.
- M. R. Fanani, M. D. Atmadja, P. Studi, J. Telekomunikasi, T. Elektro, and P. N. Malang, "Implementasi Sistem VoIP IP-PBX Menggunakan Session Border Controller (SBC) Berbasis Single Board Computer di Politeknik Negeri Malang," *J. JARTEL ISSN*, pp. 109–114, 2019.
- P. P. Gentayu and M. Alaydrus, "Analisa Layanan Keamanan, Performansi Pensinyalan dan Kualitas Panggilan Interkoneksi SIP International Direct Dialing Menggunakan Softswitch Class 4 dan Session Border Controller," *J. Telekomun. dan Komput.*, vol. 3, no. 2, p. 201, 2017, doi: 10.22441/incomtech.v3i2.1120.
- S. Jalendry and S. Verma, "A Detail Review on Voice over Internet Protocol (VoIP)," *Int. J. Eng. Trends Technol.*, vol. 23, no. 4, pp. 161–166, 2015, doi: 10.14445/22315381/ijett-v23p232.
- S. Shastri, A. Hamid, and V. Mansotra, "Voip: Conceptual Model Implementation," *Int. J. Adv. Comput. Eng. Netw.*, no. 5, pp. 2320–2106, 2017, [Online]. Available: <http://iraj.in>.

UNIVERSITAS
MERCU BUANA

KERTAS KERJA

Ringkasan

Dalam naskah jurnal penulis telah menuangkan semua point penting dalam penelitian ini, kecuali sebagai berikut:

SOURCE CODE

1. TOOLS

Dalam penelitian ini ada beberapa perlengkapan yang perlu disiapkan penulis, yaitu sebagai berikut :

- 2 Virtual Machine dengan spesifikasi : 2 core cpu, RAM 2GB dan penyimpanan 20GB.
- Sistem operasi Centos 7.
- *Service* pada sistem operasi linux seperti : *wget*, *nano*, *pacemaker* – *corosync*, *fusionpbx*, *freeswitch*, *postgresql*.

2. TAHAPAN INSTALASI

Lakukan instalasi sistem operasi linux centos 7 pada masing-masing VM. Untuk mempermudah proses instalasi dan juga *maintenance* kedepannya maka password *root* pada *server* dapat dibuat sama. Setelah melakukan instalasi sistem operasi linux centos 7 dan juga memastikan agar mendapat akses internet dan dapat saling terhubung dalam jaringan pada masing-masing VM, maka langkah selanjutnya sebagai berikut :

- Konfigurasi alamat IP pada masing-masing VM dengan detail berikut:
 VM 1 : 10.10.252.71
 VM 1 : 10.10.252.72
- Konfigurasi hostname pada masing-masing VM.

```
[root@node1sbc ~]#hostnamectl set-hostname node1@SBC
```

```
[root@node2sbc ~]#hostnamectl set-hostname node2@SBC
```
- Instalasi nano yang akan digunakan sebagai *text editor* linux.

```
yum install nano
```

 Kemudian tunggu hingga proses instalasi selesai.

Universitas Mercu Buana

- Instalasi *wget* yang akan digunakan untuk mengunduh *file* dari internet.
yum install wget
Kemudian tunggu hingga proses instalasi selesai.
- Setting *file hosts* untuk penambahan DNS lokal.

Pada *node1@SBC* :

```
[root@node1sbc ~]# nano /etc/hosts
10.10.252.72 node2sbc
```

Pada *node2@SBC* :

```
[root@node2sbc ~]# nano /etc/hosts
10.10.252.71 node1sbc
```

Setelah melakukan instalasi *service nano* dan *wget* yang akan digunakan untuk membantu instalasi *server SBC*, yang mana *SBC* nanti akan menggunakan *freeswitch* sebagai *core* utama *server telepon*, *fusionpbx* sebagai *web user interface* untuk pengoperasiannya dan *postgresql* sebagai *database*.

Langkah selanjutnya adalah instalasi *fusionpbx*, *postgresql* dan juga *freeswitch* yang sudah dalam satu paket, dengan cara sebagai berikut :

- *Disable selinux* pada centos 7.
sudo setenforce 0
- *Disable firewall* pada centos 7.
systemctl stop firewalld
- Unduh paket *file fusionpbx* dan juga *freeswitch*.
wget -O - https://raw.githubusercontent.com/fusionpbx/fusionpbx-install.sh/master/centos/pre-install.sh | sh;
Setelah itu tunggu hingga paket *file* sudah selesai diunduh.
- Lakukan instalasi pada paket yang selesai diunduh.
cd /usr/src/fusionpbx-install.sh/centos && ./install.sh

- Tunggu hingga proses instalasi selesai dilakukan, setelah proses instalasi selesai maka akan muncul tampilan seperti berikut :

Installation has completed.

Use a web browser to login.

domain name: https://10.10.252.71

username: admin

password: oINarGBZWWyR7VY

The domain name in the browser is used by default as part of the authentication.

If you need to login to a different domain then use username@domain.

username: admin@10.10.252.71

Official FusionPBX Training

*Fastest way to learn FusionPBX. For more information
<https://www.fusionpbx.com>.*

Available online and in person. Includes documentation and recording.

Location: Online

Admin Training: TBA

Advanced Training: TBA

Continuing Education: 17th December 2020 (1 Day)

Timezone: <https://www.timeanddate.com/weather/usa/idaho>

Additional information.

<https://fusionpbx.com/members.php>

<https://fusionpbx.com/training.php>

<https://fusionpbx.com/support.php>

<https://www.fusionpbx.com>

<http://docs.fusionpbx.com>

- Lakukan hal yang sama pada *server node2@SBC*.
- Setelah selesai melakukan instalasi *fusionpbx* dan *freeswitch*, *server web* dapat diakses sesuai dengan informasi yang muncul setelah instalasi selesai dilakukan. Dan dapat diganti dengan menggunakan password yang sama agar dapat memudahkan proses *maintenance*.

Universitas Mercu Buana

Setelah proses instalasi *server SBC* yang menggunakan *fusionpbx* dan juga *freeswitch* selesai dilakukan maka langkah selanjutnya adalah melakukan instalasi *pacemaker – corosync* sebagai *service* yang berjalan untuk melakukan redundansi pada kedua *server SBC*.

Berikut langkah-langkah instalasi dan *set up* pada *pacemaker – corosync* :

- Lakukan instalasi pada *service pacemaker-corosync*.

```
[root@node1sbc ~]# yum install -y pacemaker pcs fence-agents-all psmisc
policycoreutils-python
```

- *Allow port* untuk *service corosync* pada *firewall*.

```
firewall-cmd --permanent --add-service=high-availability
firewall-cmd --reload
```

- Ganti password untuk *hacluster pacemaker – corosync*.

```
echo "SbcCluster" | passwd --stdin hacluster
```

- Jalankan *service pacemaker – corosync*.

```
systemctl start pcsd && systemctl enable pcsd
```

- Lakukan hal yang sama pada *node2@SBC*.

- Setelah kedua *server* sudah selesai dilakukan instalasi *service pacemaker – corosync*. Maka setelah itu melakukan *setting service pacemaker – corosync* agar kedua *server* dapat terhubung.

```
pcs cluster auth node1sbc node2sbc -u hacluster
```

```
Password: (masukkan password)
```

```
node1sbc: Authorized
```

```
node2sbc: Authorized
```

- Konfigurasi *cluster name* dan *cluster node* pada *server node1@SBC*.

```
pcs cluster setup --name sbc node1sbc node2sbc
```

- Kemudian akan muncul notif seperti berikut :

```
Shutting down pacemaker/corosync services...
```

```
Redirecting to /bin/systemctl stop pacemaker.service
```

```
Redirecting to /bin/systemctl stop corosync.service
```

```
Killing any remaining services...
```

```
Removing all cluster configuration files...
```

```
node1sbc: Succeeded
```

```
node2sbc: Succeeded
```


Synchronizing pcsd certificates on nodes node1, node2...

node1sbc: Success

node2sbc: Success

Restarting pcsd on the nodes in order to reload the certificates...

node1sbc: Success

node2sbc: Success

- Sbc ini adalah nama *cluster* yang kita buat yang mana berisi dengan 2 *server* node1sbc dan node2sbc.

- Pada node1sbc jalankan *service cluster* sbc yang telah dibuat.

```
[root@node1sbc ~]#pcs cluster start -all
```

```
[root@node1sbc ~]#pcs cluster enable -all
```

- Konfigurasi *property* pada kedua *server*.

```
[root@node1sbc ~]#pcs property set stonith-enabled=false
```

```
[root@node1sbc ~]#pcs property set no-quorum-policy=ignore
```

```
[root@node1sbc ~]#pcs property set default-resource stickiness="INFINITY"
```

Setelah selesai melakukan instalasi dan konfigurasi *cluster* pada kedua *server*. Maka untuk saat ini perlu dilakukan konfigurasi *floating* IP yang mana akan memberikan alamat IP pada *cluster* sbc yang sudah dibuat.

Berikut langkah-langkah konfigurasi *floating* IP pada *service corosync*:

- Buat *corosync resource* dengan alamat IP *floating* 10.10.252.73 pada node1sbc.

```
[root@node1sbc ~]#pcs resource create SbcFailover
ocf:heartbeat:IPaddr2 ip=10.10.252.73 cidr_netmask=32 op monitor
interval=30ms
```

- Pada node1sbc jalankan *service cluster* sbc agar *corosync resource* dapat berjalan.

```
[root@node1sbc ~]#pcs cluster start -all
```

```
[root@node1sbc ~]#pcs cluster enable -all
```

3. REDUDANSI DATABASE

Setelah instalasi *server* dan pemasangan redundansi menggunakan *service corosync*, maka selanjutnya adalah pemasangan redundansi database dengan skema 1 *primary* dan 1 *backup*. Dimana *backup database* akan menggunakan *database localhost* masing – masing *server*.

Pada penelitian ini masing – masing *server* akan tersambung pada *database primary* kemudian akan menyalin data ke *database localhost*. Agar ketika *database primary* sedang bermasalah, maka masing – masing *server* masih dapat mengakses data dari *database localhost*.

Langkah – langkah *setting* pada *database primary* agar dapat mengirim data ke *database backup* secara otomatis sebagai berikut :

- Masuk ke *database primary* agar dapat *setting password user* baru dengan *password Sbc@Cluster21*.

```
su – postgres
-bash-4.2$ psql
psql (9.6.22)
Type "help" for help.
postgres=# \password postgres
Enter new password:
Enter it again:
```

- Pada *database primary* masuk ke *directory data* pada *database postgresql*. Kemudian merubah file *postgresql.conf* dengan cara :

```
cd /var/lib/pgsql/9.6/data
nano postgresql.conf
```

Kemudian merubah pada bagian *#listen_addresses = 'localhost'* menjadi *listen_addresses = '10.10.252.71'* atau dapat juga dirubah menjadi *listen_addresses = '*'*. Konfigurasi ini bertujuan agar *database primary* membaca *traffic* pada semua *interface* atau *ip* tertentu secara spesifik.

- Merubah pada bagian *#wal_level = minimal* menjadi *wal_level = hot_standby*.
- Merubah pada bagian *#synchronous_commit = on* menjadi *synchronous_commit = local*. Hal ini digunakan agar *level synchronization* menjadi *local sync*.

- Aktifkan *archiving mode* dengan merubah pada bagian `#archive_mode = off` dan `archive_command = ''` menjadi seperti berikut :

```
archive_mode = on
```

```
archive_command = 'cp %p /var/lib/pgsql/9.6/archive/%f'
```

- Merubah pada bagian `#max_wal_senders = 0` dan `#wal_keep_segments = 0` menjadi seperti berikut :

```
max_wal_senders = 2
```

```
wal_keep_segments = 10
```

Hal ini dikarenakan *server database* yang digunakan 2 dengan tiap *segment* sebesar 10MB.

- Merubah pada bagian `#synchronous_standby_names = ''` menjadi `synchronous_standby_names = 'dbbackup01'`. Akan digunakan untuk menyambungkan *database backup* nantinya.

- Setelah selesai maka dapat keluar dari *editor nano* dengan menekan `ctrl+x` kemudian *save*.

- Setelah selesai melakukan konfigurasi *file postgresql.conf* maka perlu membuat *file* tersebut dapat dijalankan oleh *server* dengan cara sebagai berikut :

```
mkdir -p /var/lib/pgsql/9.6/archive/
```

```
chmod 700 /var/lib/pgsql/9.6/archive/
```

```
chown -R postgres:postgres /var/lib/pgsql/9.6/archive/
```

- Merubah *file pg_hba.conf* untuk menambahkan alamat ip kedua *server database*.

```
# PostgreSQL Primary IP address
```

```
host replication replica 10.10.252.71/32 md5
```

```
# PostgreSQL SLave IP address
```

```
host replication replica 10.10.252.72/32 md5
```

- Kemudian lakukan *restart service database*.

```
systemctl restart postgresql-9.6
```

- Setelah selesai melakukan konfigurasi redundansi pada *database primary*, kemudian membuat akun *user* baru untuk *database backup* dengan *username replica* dan *password Sbc@Cluster21*.

```
su - postgres
```

```
createuser --replication -P replica
```

Enter New Password:

Enter it again:

Langkah – langkah setting redundansi pada *database backup* agar dapat menerima data dari *database primary* secara otomatis sebagai berikut :

- Matikan *service database postgresql* pada *database backup* terlebih dahulu.

```
systemctl stop postgresql-9.6
```

- Kemudian *backup directory* data pada *database postgresql* menjadi *directory data-backup*.

```
cd /var/lib/pgsql/9.6/
```

```
mv data data-backup
```

- Setelah *directory* data berubah menjadi *data-backup* maka membuat *directory* data yang baru. Kemudian merubah *ownership permission* agar dapat diakses oleh *database*.

```
mkdir -p data/
```

```
chmod 700 data/
```

```
chown -R postgres:postgres data/
```

- Kembali ke *database primary* kemudian mengirimkan berkas pada *directory* data ke *user database backup*.

```
su - postgres
```

```
pg_basebackup -h 10.10.252.71 -U replica -D /var/lib/pgsql/9.6/data -P --xlog
```

Password:

- Pada *database backup* masuk ke *directory* data pada *database postgresql*. Kemudian merubah *file postgresql.conf* dengan cara :

```
cd /var/lib/pgsql/9.6/data
```

nano postgresql.conf

Kemudian merubah pada bagian *#listen_addresses = 'localhost'* menjadi *listen_addresses = '10.10.252.72'* atau dapat juga dirubah menjadi *listen_addresses = '*'*. Konfigurasi ini bertujuan agar *database primary* membaca *traffic* pada semua *interface* atau *ip* tertentu secara spesifik.

- Mengaktifkan *mode hot_standby* dengan merubah *#hot_standby = off* pada *script file postgresql.conf* menjadi seperti berikut :

hot_standby = on

- Setelah selesai maka dapat keluar dari *editor nano* dengan menekan *ctrl+x* kemudian *save*.

- Membuat *file* baru dengan nama *recovery.conf*.

nano recovery.conf

- Setelah itu masukan *script* seperti berikut :

tandby_mode = 'on'

*primary_conninfo = 'host=10.10.252.71 port=5432 user=replica
password= Sbc@Cluster21 application_name= dbbackup01'*

trigger_file = '/tmp/postgresql.trigger.5432'

- Setelah selesai maka dapat keluar dari *editor nano* dengan menekan *ctrl+x* kemudian *save*.

- Merubah *ownership permission file recovery.conf* agar dapat diakses oleh *database*.

chmod 600 recovery.conf

chown postgres:postgres recovery.conf

- Setelah itu jalankan kembali *service database*.

systemctl start postgresql-9.6

Setelah melakukan *setting* pada *database primari* dan *backup* agar *database backup* dapat menyalin *file* secara otomatis. Maka selanjutnya melakukan *setting* agar *server VoIP backup* dapat mengakses *database primary*, dengan langkah – langkah sebagai berikut :

- Masuk pada *directory fusionpbx* pada *server backup*.

cd /etc/fusionpbx

- Kemudian *edit file config.php* pada bagian *//pgsql: database connection information* menjadi seperti berikut :

```
//pgsql: database connection
```

```
$db_host = '10.10.252.71';
```

```
$db_port = '5432';
```

```
$db_name = 'fusionpbx';
```

```
$db_username = 'fusionpbx';
```

```
$db_password = 'UoFWDRRNJUjhNo7cGhAAlhU6mE';
```

Untuk *username* dan *password* dapat dilihat pada *server primary* pada *file* yang sama.

- Kemudian *edit file config.lua* pada bagian *--database information* menjadi seperti berikut :

```
--database information
```

```
database = {}
```

```
database.type = "pgsql";
```

```
database.name = "fusionpbx";
```

```
database.path = [[]];
```

```
database.system = "pgsql://hostaddr=127.0.0.1 port=5432
```

```
dbname=fusionpbx
```

```
user=fusionpbx
```

```
password=UoFWDRRNJUjhNo7cGhAAlhU6mE options="";
```

```
database.switch = "pgsql://hostaddr=127.0.0.1 port=5432
```

```
dbname=freeswitch
```

```
user=fusionpbx
```

```
password=UoFWDRRNJUjhNo7cGhAAlhU6mE options="";
```

```
database.backend = {}
```

```
database.backend.base64 = 'luasql'
```

- Dengan begitu konfigurasi *redundansi database* telah selesai dilakukan.