



**IMPLEMENTASI HIGH AVAILABILITY CALL CENTER BERBASIS
OPEN SOURCE DI UNIVERSITAS MERCUBUANA**

TUGAS AKHIR

Gilang Hamdan Muarif
41516110023

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2020**



**IMPLEMENTASI HIGH AVAILABILITY CALL CENTER BERBASIS
OPEN SOURCE DI UNIVERSITAS MERCUBUANA**

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:
Gilang Hamdan Muarif
41516110023

UNIVERSITAS
MERCU BUANA
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2020

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM : 41516110023

Nama : Gilang Hamdan Muarif

Judul Tugas Akhir : Implementasi High Availability Call Center Berbasis Open Source Di Universitas Mercubuana

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, 31 Juli 2020



Gilang Hamdan Muarif



UNIVERSITAS
MERCU BUANA

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

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

Nama Mahasiswa : Gilang Hamdan Muarif
NIM : 41516110023
Judul Tugas Akhir : Implementasi High Availability Call Center Berbasis Open Source Di Universitas Mercubuana

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

Dengan Hak Bebas Royalti/Noneksklusif 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.

UNIVERSITAS
MERCU BUANA

Jakarta, 31 Juli 2020



SURAT PERNYATAAN LUARAN TUGAS AKHIR

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

Nama Mahasiswa : Gilang Hamdan Muarif
 NIM : 41516110023
 Judul Tugas Akhir : Implementasi High Availability Call Center
 Berbasis Open Source Di Universitas Mercubuana

Menyatakan bahwa 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 : Jurnal RESTI | |
| | | ISSN : 2580-0760 | |
| 2 | Kertas Kerja, Merupakan material hasil penelitian sebagai kelengkapan Artikel Jurnal. Terdiri dari (minimal 4) | Literatur Review | [✓] |
| | | Hasil analisa & perancangan aplikasi | [✓] |
| | | Source code | [✓] |
| | | Data set | [✓] |
| | | Tahapan eksperimen | [✓] |
| | | Hasil eksperimen seluruhnya | [✓] |
| 3 | HAKI Disubmit / Terdaftar | HKI | Diajukan |
| | | Paten | Tercatat |
| | | No & Tanggal Permohonan : | |
| | | No & Tanggal Pencatatan : | |

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 31 Juli 2020



Gilang Hamdan Muarif

LEMBAR PERSETUJUAN PENGUJI


NIM : 41516110023.....
.....

Nama : Gilang Hamdan Muarif
.....
.....

Judul Tugas Akhir : Implementasi High Availability Call Center Berbasis Open
Source Di Universitas Mercubuana
.....
.....
.....
.....

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 2020



UNIVERSITAS
MERCU BUANA
(Dr. Ida Nurhaida, MT.)

LEMBAR PERSETUJUAN PENGUJI

| | | |
|-------------------|---|--|
| NIM | : | 41516110023..... |
| Nama | : | <u>Gilang Hamdan Muarif</u> |
| Judul Tugas Akhir | : | <u>Implementasi High Availability Call Center Berbasis Open Source Di Universitas Mercubuana</u> |

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 2020



(Diky Firdaus, S.Kom, MM)



LEMBAR PERSETUJUAN PENGUJI

NIM : 41516110023
Nama : Gilang Hamdan Muarif
Judul Tugas Akhir : Implementasi High Availability Call Center Berbasis Open Source Di Universitas Mercubuana

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 2020



LEMBAR PENGESAHAN

NIM : 41516110023
Nama : Gilang Hamdan Muarif
Judul Tugas Akhir : Implementasi High Availability Call Center Berbasis Open Source Di Universitas Mercubuana

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 2020

Menyetujui,



(Desi Ramayanti, S.Kom, MT)
Dosen Pembimbing

Mengetahui,



UNIVERSITAS
MERCU BUANA

(Diky Firdaus, S.Kom, MM)
Koord. Tugas Akhir Teknik Informatika

(Desi Ramayanti, S.Kom, MT)
Ka. Prodi Teknik Informatika

ABSTRAK

Nama : Gilang Hamdan Muarif
NIM : 41516110023
Pembimbing TA : Desi Ramayanti, S.Kom, MT
Judul : Implementasi High Availability Call Center
Berbasis Open Source Di Universitas Mercubuana

Media komunikasi menjadi salah satu penunjang operasional kampus agar bisa memberi pelayanan terbaik, dimana akan memudahkan komunikasi antara *stakeholder* dengan pihak pelayanan kampus. Masalah yang akan diteliti adalah keterbatasan fitur dan pelayanan pada sistem PABX eksisting yang menyebabkan menurunnya kualitas layanan yang diberikan. Tujuan dari penelitian ini adalah membuat sebuah sistem IP PBX berbasis *Voip* yang dapat menyediakan fitur *High Availability*, *Call Detail Record*, *Agent Monitoring*, adanya integrasi antar cabang yang berbasis *open source* sehingga dapat diimplementasikan tanpa memerlukan adanya biaya tambahan. Metode perancangan sistem menggunakan metode *Cisco Lifestyle Service PPDIOO* dengan metode pengujian menggunakan metode *Black Box*. Proses pengujian menggunakan aplikasi *Startrinity* untuk dapat menguji beban pada server dan jaringan *voip*, pada proses pengujian semua fungsi dapat berfungsi dengan baik pada aplikasi *call center*. Pemanfaatan IP PBX *Asterisk* untuk menyediakan fitur dasar telepon dan alur panggilan untuk menu IVR *call center*, aplikasi *CDR Viewer* untuk menyediakan data keluar masuk nya panggilan, aplikasi *Queue Monitoring* untuk memonitor operasional *call center* dan konfigurasi IP PBX server dengan konsep HA sudah cukup dalam memenuhi kebutuhan fungsionalitas bagi seluruh pengguna sistem komunikasi yang ada di Kampus UMB

Kata kunci:

High Availability, IP PBX Asterisk, CDR Viewer, Queue Monitoring, IVR.

ABSTRACT

Name : Gilang Hamdan Muarif
Student Number : 41516110023
Counsellor : Desi Ramayanti, S.Kom, MT
Title : Implementasi High Availability Call Center
Berbasis Open Source Di Universitas Mercubuana

Communication media is one of campus operational support in order to be able to provide best service, which will facilitate communication between stakeholders with customer service in campus. The problem to be investigated is the limitations of features and services in the existing PABX system which causes a decrease in the quality of services provided. The purpose of this research is to create a Voip-based IP PBX system that can provide High Availability, Call Detail Record, Agent Monitoring, inter-branch integration with open source-based integration so that it can be implemented without requiring additional costs. System design method uses the Cisco Lifestyle Service PPDIOO method with testing method using the Black Box method. The testing process uses Startrinity application to be able to test load on the server and voip network, in the testing process all functions can work properly in the call center application. Utilization of Asterisk IP PBX to provide basic telephone features and call flow for IVR call center menus, CDR Viewer to provide data in/out of calls, Queue Monitoring to monitor call center operations and IP PBX server configuration with HA concept is sufficient to meet the functionality requirements for all system communication users on UMB Campus.

Key words: *High Availability, IP PBX Asterisk, CDR Viewer, Queue Monitoring, IVR.*

KATA PENGANTAR

Puji syukur kita panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan Tugas Akhir yang berjudul Perancangan dan Implementasi Sistem Call Center Berbasis Open Source Menggunakan Pabx Asterisk Di Universitas Mercubuana ini tepat pada waktunya. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari ibu Desi Ramayanti, S.Kom, MT, selaku pembimbing Tugas Akhir ini penulis tidak dapat menyelesaikan Tugas Akhir ini sebaik mungkin. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Trival Apriyadi selaku Kepala Biro Manajemen Gedung dan Sarana
2. Bapak Aji Taufik selaku narasumber dari BMGS
3. Bapak Dede selaku narasumber dari BMGS
4. Muhammad Muttaqin selaku rekan proyek Tugas Akhir
5. Ahmad Ramadan selaku tim R&D PT. Selindo Alpha
- 6.

Akhir kata, penulis berharap Tugas Akhir ini dapat bermanfaat khususnya bagi penulis dan umumnya bagi semua pihak yang berkepentingan. Dan semoga Allah SWT membalas semua amal dan kebaikan kepada pihak-pihak yang telah membantu menyelesaikan Tugas Akhir ini. Aamiin.

Jakarta, 31 Juli 2020

Penulis

DAFTAR ISI

| | |
|---|------|
| HALAMAN SAMPUL..... | i |
| HALAMAN JUDUL | i |
| LEMBAR PERNYATAAN ORISINALITAS | ii |
| SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR... iii | |
| SURAT PERNYATAAN LUARAN TUGAS AKHIR..... | iv |
| LEMBAR PERSETUJUAN PENGUJI | v |
| LEMBAR PENGESAHAN | viii |
| ABSTRAK | ix |
| ABSTRACT | x |
| KATA PENGANTAR..... | xi |
| DAFTAR ISI..... | xii |
| NASKAH JURNAL | 1 |
| KERTAS KERJA..... | A |
| BAGIAN 1. LITERATUR REVIEW | B |
| BAGIAN 2 ANALISIS DAN PERANCANGAN..... | L |
| BAGIAN 3 SOURCE CODE | Y |
| BAGIAN 4 DATASET..... | BB |
| BAGIAN 5 TAHAPAN EKSPERIMEN | DD |
| BAGIAN 6 HASIL SEMUA EKSPERIMEN | GG |

NASKAH JURNAL

1. Pendahuluan

Teknologi informasi dan komunikasi yang terus berkembang menjadi pilar utama dalam setiap inovasi teknologi, maka dapat dikatakan bahwa teknologi informasi menjadi teknologi yang sangat penting dalam kehidupan sehari-hari. Termasuk dalam ruang lingkup kampus media komunikasi menjadi salah satu penunjang operasional kampus agar bisa memberi pelayanan terbaik. Oleh karena itu menjadi sebuah kebutuhan dengan adanya sebuah media komunikasi yang terintegrasi dan bisa menyediakan fasilitas *call center* sehingga bisa memberikan pelayanan terbaik.

Berbagai macam media komunikasi yang banyak digunakan saat ini memiliki kelebihan dan kekurangan masing-masing. Tabel 1 berikut memperlihatkan perbandingan beberapa media komunikasi. Dari Tabel 1 tersebut dapat dilihat media komunikasi telepon meskipun merupakan teknologi yang sudah ada sejak lama namun masih memiliki kelengkapan fitur yang tidak kalah dengan teknologi komunikasi lain seperti *email* dan *chat*, sehingga pengguna nya pun masih tetap banyak hingga sekarang.

Tabel 1. Tabel Perbandingan Media Komunikasi[1]

| Fitur | Email | Pos | Telepon | Chat |
|-----------------------|----------------|---------|------------------------------------|--------|
| Sinkronisme | Ya | Ya | Tidak ^{a)} | Ya |
| Instantaneity | Tinggi | Tidak | Teratas | Ya |
| Interaktivitas | Tinggi | Rendah | Teratas | Tinggi |
| Surat massal | Ya | Ya | Lemah (<i>Teleconference</i>) | Tidak |
| Pengarsipan | Elektro nik | Manual | Mesin penjawab | Lemah |
| Editing/ suntingan | Alami | Mungkin | Tidak / Sangat lemah | Lemah |
| Tekstual | Ya | Ya | Ya | Ya |
| Lampiran | Ya | Ya | Tidak | Ya |

^{a)}Dengan pengecualian pesan suara dan mesin penjawab

Dalam dunia pendidikan tinggi seperti pada Universitas Mercu Buana (UMB) layanan komunikasi merupakan sarana yang sangat penting, dimana akan memudahkan komunikasi antara *stakeholder* (contohnya mahasiswa) dengan pihak penanggung jawab pelayanan. Media komunikasi ini digunakan pada semua unit yang ada di UMB seperti unit keuangan, unit penerimaan mahasiswa baru unit operasional perkuliahan, fakultas, program studi dan lain-lain. Seiring dengan peningkatan jumlah mahasiswa maka sudah menjadi kewajiban penyedia layanan untuk selalu meningkatkan kualitas pelayanan demi kepuasan para pelanggan. Kualitas layanan memerlukan koordinasi dan bagian dari kualitas koordinasi ditentukan oleh komunikasi yang baik, oleh karena itu keterbatasan fitur dan tidak adanya integrasi

sangat berpengaruh bagi kampus dalam memberikan pelayanan terbaik.

Universitas Mercu Buana (UMB) adalah perguruan tinggi swasta yang didirikan di Jakarta pada tahun 1985 di bawah pembinaan Yayasan Menara Bhakti[2]. UMB saat ini memiliki 4 kampus utama yaitu kampus Meruya, Menteng, Warung Buncit dan Jatisampurna. Dalam kegiatan operasional kampus saat ini perangkat PABX Analog menggunakan produk Panasonic tipe TDA 600 berada di bawah infrastruktur departemen Biro Manajemen Gedung dan Sarana (BMGS) dan *support* dari Dian Husada sebagai *vendor*.

Sistem PABX saat ini hanya menghubungkan unit-unit pada satu area kampus saja dan belum terintegrasi antar area kampus. Sehingga untuk melakukan komunikasi antar area kampus masih menggunakan layanan PSTN yang membutuhkan biaya tambahan untuk melakukan panggilan antar cabang. Sistem PABX saat ini juga belum mendukung *high availability* sehingga jika terjadi masalah pada sistem PABX maka semua telepon yang terhubung pada area kampus tersebut tidak dapat digunakan dan pada penggunaan skala besar akan timbul permasalahan lain seperti sulitnya proses pengecekan jika terjadi permasalahan sistem PABX yang harus dilakukan langsung secara *on site*. Selain hal tersebut, sistem PABX yang ada saat ini pun masih belum terdapat fitur rekaman panggilan dan daftar lengkap siapa saja yang melakukan panggilan ke *call center* UMB. Hal ini menyebabkan jika ada panggilan yang tidak terjawab maka tidak ada data yang jadi acuan siapa yang sedang membutuhkan layanan. Sehingga memungkinkan menurunnya tingkat kepuasan pelanggan dan tidak maksimalnya pelayanan yang diberikan oleh pihak UMB.

Untuk permasalahan pertama, terkait dengan sistem komunikasi antar area kampus, UMB saat ini sudah memiliki sistem jaringan komputer dan internet pada semua area kampus. Sehingga dengan kondisi sistem jaringan yang terintegrasi, maka solusi dari masalah ini adalah dengan membangun sebuah komunikasi suara berbasis sistem jaringan komputer dan internet. Teknologi komunikasi yang sesuai adalah dengan mengimplementasikan *Voice over Internet Protocol* (VoIP), dimana implementasi VoIP sangat mudah dilakukan tanpa penambahan biaya untuk membangun infrastruktur baru.

Voice over Internet Protocol (VoIP) atau *IP Telephony* adalah suatu teknologi yang memungkinkan pengiriman informasi berupa suara melalui perantara internet atau jaringan IP. Teknologi ini memanfaatkan infrastruktur jaringan komputer dan sebuah server untuk melakukan sebuah komunikasi suara antar pengguna. Perbedaan VoIP dengan telepon tradisional adalah masalah

infrastukturnya, jika VoIP menggunakan jaringan komputer sedangkan telepon tradisional menggunakan infrastruktur telepon yang telah dibangun oleh perusahaan telepon konvensional[3].

Berdasarkan penelitian Arief Agus Sukmandhani dan Febryo Ponco Sulisty, mengenai integrasi sistem telekomunikasi multibrand pada lembaga pendidikan, dimana pada penelitian tersebut dilakukan pengujian langsung terhadap sistem PABX yang terdapat pada semua cabang agar bisa saling berkomunikasi dengan memanfaatkan jaringan internet. Hasil dari penelitian tersebut menyebutkan bahwa integrasi antar cabang terutama *multibrand* PABX harus menyesuaikan dengan kualitas jaringan dan *brand* PABX itu sendiri sehingga komunikasi dapat berjalan normal dan optimal[4].

Untuk permasalahan kedua terkait dengan belum mendukungnya sistem *High Availability* (HA) pada sistem IP PBX, sudah banyak produk seperti *FreePBX* dan *Haast* yang menyediakan HA namun bersifat komersial. *Pacemaker* merupakan HA *resource manager* yang dapat digunakan untuk *server cluster* berskala besar ataupun kecil yang bersifat *open source*. *Pacemaker* mendukung banyak skenario, dari yang paling sederhana *2-node cluster standby* sampai dengan *16-node* aktif. *Pacemaker* juga dapat secara signifikan mengurangi biaya *hardware* dengan memungkinkan beberapa aktif maupun pasif *cluster* untuk digabungkan dan berbagi node cadangan[5].

Untuk permasalahan ketiga terkait dengan sistem PABX yang ada saat ini masih belum terdapat fitur rekaman panggilan dan daftar lengkap siapa saja yang melakukan panggilan ke *call center* UMB, maka berdasarkan penelitian Akri Meidi dan Ida Nurhaida, mengenai perancangan *call center inbound* menggunakan IP PBX *Asterisk* di Universitas Mercu Buana, dimana pada penelitian tersebut dibuat sebuah rancangan sistem *call center* yang berfokus pada pelayanan *inbound call*. Pada penelitian tersebut juga dilakukan simulasi pengujian sistem *call center* dengan melakukan stress tes panggilan terhadap sistem PABX dan didapatkan hasil standarisasi pengukuran QoS versi TIPHON dengan rata-rata delay 18-20 ms, jitter 0.03-1 ms, packet loss 0.00%, throughput 20-22 kbps menunjukkan bahwa kualitas IP PBX yang dirancang sudah bagus pada sebuah *call center* dan layak untuk diimplementasikan[6].

Namun dari penelitian tersebut terdapat kekurangan yang masih harus ditingkatkan dan dilengkapi seperti sistem yang dibangun menggunakan produk *Elastix* yang memiliki keterbatasan lisensi, belum terdapat fitur *Call Detail Record* yang dapat digunakan untuk *reporting*, pembuatan *dashboard monitoring* yang tidak dijelaskan secara detail, sistem yang masih berfokus pada pelayanan *inbound call* dan tidak adanya integrasi antar cabang yang pada penelitian ini fitur-fitur tersebut akan dilengkapi.

Sistem *Internet Protocol-Private Branch Exchange* (IP PBX) *Asterisk* yang akan dibangun adalah versi gratis dimana menyediakan fitur bawaan yang cukup lengkap seperti sistem panggilan *single number*, sistem penjawab otomatis (*Interactive Voice Response*), distribusi panggilan, *transfer* panggilan, *hold* panggilan, pesan suara, *music on hold*, *call record/monitor* dan *speed dial*[7]. Sistem IP PBX dalam penelitian ini juga akan dibangun dengan konsep *High Availability Cluster* berbasis *open source*. Lalu untuk mengoptimalkan layanan *call center* maka akan ditambahkan fitur aplikasi *Call Detail Record* (CDR) *Viewer* (melihat detail panggilan setiap *extension* dan mendengarkan rekaman panggilan) dengan bahasa pemrograman PHP dan aplikasi *Queue monitoring* (memonitor panggilan secara *real time* dan *historical*) dengan bahasa pemrograman *Python* dan basis data menggunakan *MySQL*.

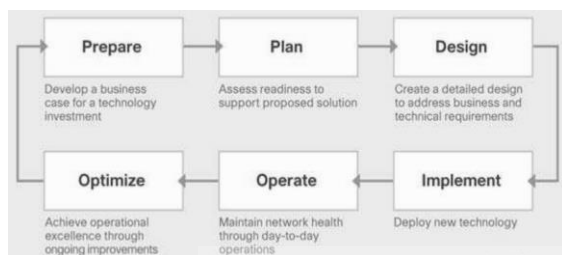
Asterisk adalah sebuah IP-PBX server yang bersifat *open-source*, ini berarti *Asterisk* dapat digunakan dengan gratis dan dapat dimodifikasi sesuai keperluan. *Asterisk* berlisensi GPL dan nonGPL dan ditulis dengan bahasa C[7]. *Asterisk* memiliki sebuah *set driver* yang mendukung koneksi ke basis data, yaitu *Asterisk Realtime Architecture*. Arsitektur ini memungkinkan *developer* menyimpan semua konfigurasi *Asterisk* dalam basis data dan kembali memuatnya kapanpun. *Developer* dapat memaksa *reload* melalui AMI (*Asterisk Manager Interface*) atau dengan memanggil *Asterisk* dari *shell script*[8].

Distributed Replicated Block Device (DRBD) bisa dianalogikan sebagai mekanisme RAID-1 (*mirroring*, bisa juga tipe RAID lain yang menggunakan prinsip *mirroring*), yang melakukan duplikasi data melalui *network*. Duplikasi data ini dilakukan dalam mekanisme *block devices*, bukan dalam bentuk data mentah. Jika RAID-1 melakukan duplikasi isi dan data suatu partisi ke partisi lain, DRBD melakukan hal yang sama, hanya saja melalui *network*. DRBD memiliki suatu keunggulan dibandingkan hardisk RAID, yaitu *backup server* berada terpisah dengan sumber *backup*. Pemisahan ini membawa keuntungan *preventif*, jika ada masalah pada salah satu server, server lainnya akan bertindak sebagai server pengganti. Jika server utama sudah kembali pulih, kendali akan dikembalikan ke server utama[5].

Interactive Voice Response (IVR) adalah sebuah sistem telepon yang memungkinkan interaksi antara sistem dengan pemanggil, mengumpulkan informasi dan menyalurkan panggilan tersebut ke penerima yang sesuai. Sebuah sistem IVR menerima kombinasi masukan suara telepon dan pilihan tombol-tombol pesawat telepon (sinyal DTMF) dan memberikan respon dalam bentuk suara, *fax*, *callback*, *e-mail* dan media lainnya. Sebuah sistem IVR terdiri atas peralatan telepon, aplikasi, sebuah basis data dan infrastruktur pendukung lainnya[9].

2. Metode Penelitian

Dalam penelitian ini, penulis menggunakan metode perancangan *Cisco Lifestyle Service PPDIIO (Prepare, Plan, Design, Implement, Operate, and Optimize)*. Metode PPDIIO merupakan metode perancangan jaringan dari Cisco yang dirancang untuk mendukung berkembangnya suatu sistem jaringan[10]. Gambar 1 menunjukkan diagram alur metode perancangan PPDIIO.



Gambar 1. Metode perancangan PPDIIO[10]

Tahapan metode PPDIIO adalah sebagai berikut:

2.1. Prepare (Persiapan)

Pada tahapan ini dilakukan identifikasi permasalahan terkait dengan penerapan sistem PABX pada kampus UMB. Untuk melakukan proses identifikasi pada tahapan ini, maka penulis melakukan tahapan observasi, wawancara dan studi literatur untuk proses pengumpulan data.

1) Observasi

Observasi dilakukan terkait dengan sistem PABX kampus UMB khususnya *Call Center* Layanan Kampus Meruya dengan nomor 021-5840815, Kampus Menteng dengan nomor 021-31935454 dan Kampus Jatisampurna dengan nomor 021-8449635. Observasi yang dilakukan pada tanggal 20 sampai dengan 22 Juli 2020 dengan cara melakukan tes panggilan, pengamatan dan pencatatan. Tabel 2 merupakan data lengkap hasil observasi, dari proses observasi didapatkan hasil bahwa untuk menghubungi *call center* UMB melalui layanan telepon sangat sulit, dari total 18 panggilan yang dilakukan ke 3 layanan kampus tersebut, hanya 3 panggilan yang berhasil tersambung dengan pihak kampus dan hanya kampus Menteng yang panggilan nya dapat dilayani oleh petugas kampus. Hal tersebut tentunya sangat menyulitkan bagi pelanggan untuk mendapatkan informasi yang dibutuhkan dari pihak kampus. Dari hasil observasi juga didapatkan data bahwa tidak adanya filter jam operasional di sistem PABX, hal tersebut akan berdampak pada reputasi pelayanan kampus yang seakan tidak melayani padahal sudah berada diluar jam operasional layanan.

Tabel 2. Tabel Hasil Observasi

| Kampus | Time | Connect | IVR | Pickup |
|--------|---------------|---------|-----|--------|
| Meruya | 20 juli 14:05 | No | No | No |
| | 20 juli 15:58 | No | No | No |
| | 21 juli 10:16 | Yes | Yes | No |
| | 21 juli 13:23 | Yes | Yes | No |

| | | | | |
|--------------|---------------|-----|-----|-----|
| Menteng | 22 juli 15:50 | Yes | Yes | No |
| | 22 juli 19:54 | Yes | Yes | No |
| | 20 juli 14:07 | Yes | No | Yes |
| | 20 juli 16:00 | Yes | No | Yes |
| | 21 juli 10:20 | Yes | No | No |
| | 21 juli 13:26 | Yes | No | No |
| Jatisampurna | 22 juli 15:49 | Yes | No | Yes |
| | 22 juli 19:56 | Yes | No | No |
| | 20 juli 14:10 | Yes | Yes | No |
| | 20 juli 16:03 | Yes | Yes | No |
| | 21 juli 10:18 | Yes | Yes | No |
| | 21 juli 13:25 | Yes | Yes | No |
| | 22 juli 15:47 | Yes | No | No |
| | 22 juli 19:58 | Yes | No | No |

2) Wawancara

Wawancara dilakukan terkait dengan Sistem PABX UMB yang sudah ada. Wawancara dilakukan dua kali pada tanggal 11 November 2019 dan pada tanggal 24 juni 2020 dengan pihak Biro Manajemen Gadung dan Sarana (BMGS) selaku pengelola jaringan telepon seluruh kampus UMB. Pada wawancara ini penulis mengajukan pertanyaan mengenai spesifikasi sistem yang sudah berjalan saat ini, kekurangan dan kelebihan serta fitur yang dibutuhkan namun belum bisa disediakan oleh sistem yang ada. Hasil yang didapatkan dari wawancara yaitu jumlah *extension* sebanyak 204, jumlah line PSTN terdapat 28 *channel* dan jumlah operator 4 orang yang ada pada kampus Meruya. Data tersebut bisa menjadi gambaran kapasitas sistem PABX yang ada sedang berjalan saat ini di kampus UMB.

Dari hasil wawancara juga didapatkan data bahwa sistem PABX yang digunakan di UMB saat ini adalah tipe TDA600 Panasonic, sistem PABX tersebut masih merupakan sistem PABX analog yang memiliki keterbatasan hak akses untuk melakukan penambahan atau perubahan pada sistem, keterbatasan fitur telepon, keterbatasan integrasi dengan sistem atau jaringan lain, tidak adanya fitur *recording extension* dan tidak adanya fitur *reporting* yang dibutuhkan untuk membangun sebuah *call center*.

3) Studi Pustaka

Proses studi pustaka dilakukan dengan melakukan pencarian dan pengumpulan data dengan cara membaca buku, jurnal-jurnal yang berkaitan dengan objek penelitian dan dapat dijadikan sebagai dasar teori serta dapat dijadikan bahan perbandingan.

2.2. Plan (Perencanaan)

Pada tahapan ini dilakukan analisa perencanaan hasil yang akan dicapai dan perencanaan tindakan dalam melakukan uji penerapan. Tahapan ini meliputi hal-hal sebagai berikut :

1) Analisa Kebutuhan Fungsionalitas

Proses analisa kebutuhan fungsionalitas dilakukan untuk melakukan pemetaan kebutuhan bagi masing-masing pengguna, hal tersebut untuk memudahkan kita

dalam melihat fitur-fitur apa saja yang dibutuhkan masing-masing pengguna. Daftar kebutuhan fungsionalitas pengguna seperti pada Tabel 3 berikut :

Tabel 3. Tabel fungsionalitas setiap pengguna

| Pengguna | Fungsionalitas |
|----------------------|--|
| Penelpon | Mendapatkan informasi layanan apa saja yang terdapat pada kampus. Mendapatkan informasi daftar program studi yang ada pada kampus. Mendapatkan informasi fasilitas pada kampus. Mendapatkan informasi/pengumuman penting yang dibutuhkan oleh pihak mahasiswa ataupun dosen. |
| Staff Call center | Menyimpan data penelepon dan sejarah percakapan untuk keperluan <i>follow up</i> penelepon, <i>reporting</i> , analisa data serta pemrosesan dan pengambilan kebijakan bisnis. Jam operasional <i>call center</i> adalah senin sampai sabtu jam 9 sampai dengan jam 4 sore. Staff <i>call center</i> berperan sebagai operator di setiap kampus yang berperan sebagai staff yang membantu penelepon jika tidak mengetahui <i>extension</i> yang akan dituju, tidak mengetahui departemen yang dituju atau ingin menanyakan informasi seputar kampus. |
| Staff Kampus | Dapat melakukan panggilan antar <i>extension</i> dalam satu kampus atau antar kampus. Dapat melakukan panggilan keluar dan menerima panggilan dari luar. Dapat menjalankan fitur dasar telepon seperti transfer panggilan, <i>hold</i> panggilan dan <i>voice mail</i> . |
| Sistem administrator | Mengubah, menambahkan, dan menghapus akun pengguna. Memodifikasi dialplan. Menambahkan dan menghapus berkas suara dan menu yang ada pada IVR. |

2) Analisa Kebutuhan *Hardware* dan *Software*

Proses analisa kebutuhan server mengambil data dari proses wawancara yang sudah dilakukan sebelumnya, maka sistem yang akan dibangun untuk menunjang kebutuhan tersebut dijabarkan pada Tabel 4 Kebutuhan *Hardware* dan Tabel 5 Kebutuhan *Software* berikut :

Tabel 4. Tabel Kebutuhan *Hardware*

| Specification | Server for GNS3 | Asterisk | Client |
|---------------|-----------------|----------|--------|
| CPU | 2.7 GHz | 2 GHz | 1 GHz |
| Memory | 24 GB | 2 GB | 1 GB |
| HDD | 1 TB | 16 GB | 20 GB |

Tabel 5. Tabel Kebutuhan *Software*

| Software | Version | Function |
|------------------------|-----------------------|-------------------|
| GNS 3 | 2.2.10 | Network Simulator |
| VMWARE Workstation Pro | 15.5.2 build-15785246 | VM Host |

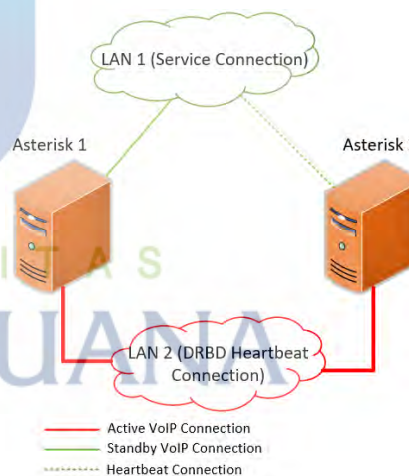
| | | |
|------------------------|------------|------------------|
| CentOS | 7 | Server OS |
| Windows | 7 ultimate | Client OS |
| Asterisk | 16.9.0 | IP PBX Software |
| Zoiper | 5.3.8 | Softphone |
| Startrinity SIP Tester | 2020-07-31 | Stress Test Tool |

2.3. Design (Desain)

Pada tahapan ini dibuat sebuah gambaran atau skema sistem IP PBX dan *call center* yang akan berjalan.

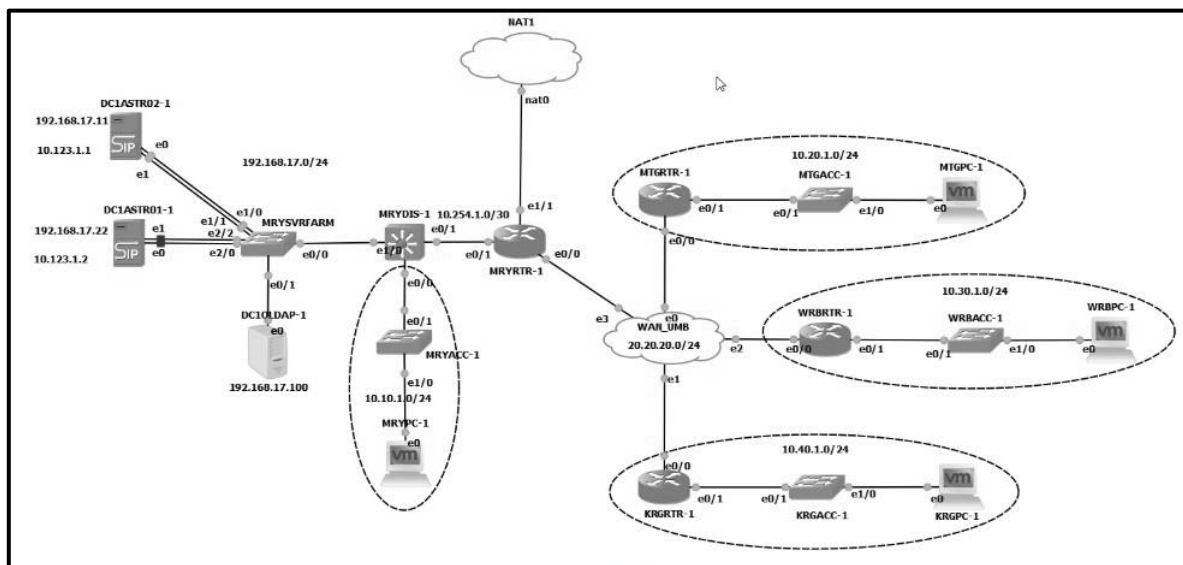
1) Desain Sistem IP PBX

Rancangan sistem ini disimulasikan menghubungkan empat area kampus yaitu Meruya, Menteng, Warung Buncit dan Jatisampurna. Gambar 2 merupakan gambaran sistem IP PBX Asterisk dirancang dengan konsep *High-availability Clusters* dengan konfigurasi *Active/Passive Failover* sehingga memiliki 2 koneksi, satu koneksi digunakan untuk komunikasi dari user kearah kluster *asterisk* dan koneksi lainnya digunakan untuk komunikasi antar server, dimana *node* aktif bertugas untuk melakukan eksekusi terhadap aplikasi atau tugas tertentu, sedangkan *node* pasif berstatus *standby* dengan tidak melakukan tugas apapun sampai mendeteksi bahwa terdapat masalah pada *node* utama/aktif.



Gambar 2. Design Sistem IP PBX Asterisk

Gambar 3 merupakan topologi jaringan yang akan digunakan untuk mensimulasikan sistem IP PBX dan jaringan kampus UMB. Server IP PBX Asterisk akan diposisikan di kampus UMB Meruya, dan untuk kampus Menteng, Warung Buncit dan Jatisampurna mengakses sistem IP PBX menggunakan jaringan internet yang sudah terintegrasi antar kampus.

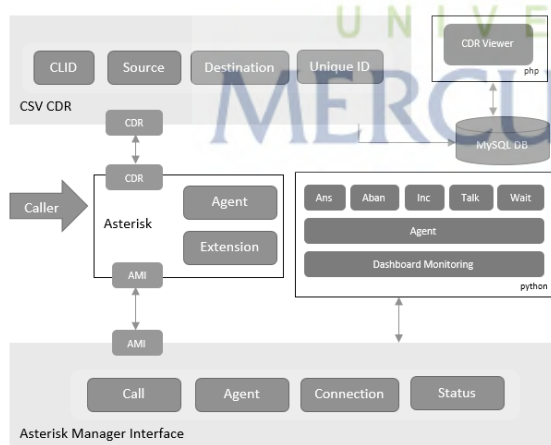


Gambar 3. Topologi jaringan kampus Mercu Buana

Untuk menunjang kebutuhan *call center*, maka akan diintegrasikan aplikasi *CDR Viewer* dan *Queue Monitoring* dengan IP PBX *Asterisk*. Gambar 4 merupakan skenario sistem aplikasi *call center* yang akan berjalan pada sistem IP PBX *Asterisk*. Aplikasi *Queue Monitoring* akan diintegrasikan dengan IP PBX *Asterisk* menggunakan fitur *Asterisk Manager Interface (AMI)*. Sedangkan aplikasi *CDR Viewer* akan diintegrasikan dengan sistem IP PBX *Asterisk* dengan menggunakan fitur CSV CDR yang dibuat oleh IP PBX *Asterisk*, data akan diunggah secara otomatis kedalam basis data MySQL untuk dijadikan data oleh *CDR Viewer* dalam menampilkan fitur-fitur *reporting* yang diperlukan untuk *call center*.

CDR Viewer juga memiliki beberapa fitur lain yaitu mengunduh panggilan yang direkam, mengunduh hasil pencarian panggilan kedalam berkas dengan format csv, laporan panggilan dalam bentuk grafik, laporan panggilan yang dilakukan bersamaan dalam rentan waktu tertentu, laporan jumlah menit panggilan yang dilakukan tiap *extension* dan laporan panggilan yang masuk ke *agent call center*.

Gambaran basis data yang akan digunakan untuk pengolahan data pada aplikasi *CDR Viewer* dapat dilihat pada Gambar 5. Terdapat dua tabel yang akan digunakan pada aplikasi *CDR Viewer*, yaitu tabel CDR dimana pada tabel ini menyimpan data histori panggilan yang ada pada IP PBX *Asterisk* dan tabel *User* dimana pada tabel ini menyimpan daftar user yang terdaftar untuk mengakses aplikasi *CDR Viewer*.



Gambar 4. Skenario Sistem Call Center

| CDR | | User | |
|-------------|-------------|------------|--------------|
| id (PK) | int(11) | id (PK) | int |
| calldate | datetime | user | varchar(50) |
| accountcode | varchar(20) | password | varchar(250) |
| src | varchar(80) | created_at | datetime |
| dst | varchar(80) | | |
| dcontext | varchar(80) | | |
| clid | varchar(80) | | |
| channel | varchar(80) | | |
| dstchannel | varchar(80) | | |
| lastapp | varchar(80) | | |
| lastdata | varchar(80) | | |
| start | datetime | | |
| answer | datetime | | |
| end | datetime | | |
| duration | int(11) | | |
| billsec | int(11) | | |
| disposition | varchar(45) | | |
| amaflags | int(11) | | |
| uniqueid | varchar(32) | | |
| userfield | varchar(32) | | |

Gambar 5. Konsep Basis Data CDR Viewer

2) Desain Aplikasi CDR Viewer

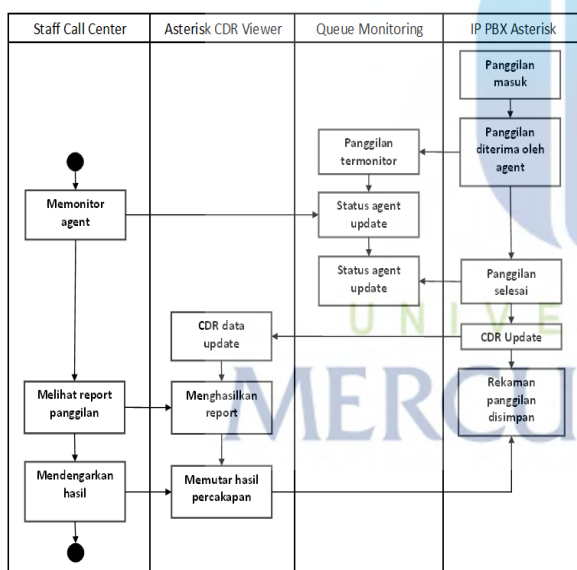
Aplikasi *CDR Viewer* memiliki fitur utama sebagai aplikasi yang menyediakan *report* untuk detail aktifitas panggilan yang dilakukan di setiap *extension* yang terdaftar di IP PBX *Asterisk*. Detail yang ditampilkan yaitu diantaranya detail waktu panggilan, durasi panggilan, sumber dan destinasi panggilan.

3) Desain Aplikasi Queue Monitoring

Aplikasi *Queue Monitoring* memiliki fitur utama untuk memonitor aktifitas *agent* dalam *call center* yang terdaftar di IP PBX Asterisk secara *real time*. Data yang ditampilkan yaitu nama layanan, jumlah panggilan masuk ke semua layanan, jumlah panggilan tidak terjawab ke semua layanan, jumlah panggilan masuk yang sedang berlangsung di semua layanan, rata-rata waktu panggilan di semua layanan dan rata-rata waktu antrian yang dibutuhkan penelpon untuk diterima oleh *agent* di semua layanan. Selain itu terdapat juga fitur detail *agent* yang terdaftar dalam layanan, status *agent*, jumlah panggilan per *agent*, tombol untuk *spy* dan *whisper* dan detail penelpon yang sedang dalam antrian.

4) Proses Bisnis *Call center*

Gambaran *activity diagram* proses bisnis dalam *call center* yang dibangun pada IP PBX Asterisk yang digambarkan pada Gambar 6. Dalam proses bisnis dijelaskan bahwa IP PBX Asterisk bertindak sebagai sistem utama dalam memberikan data pada aplikasi *Queue Monitoring* dan *CDR Viewer*. Staff *call center* bisa memonitor status *agent* secara *realtime* melalui aplikasi *Queue monitoring* dan melihat *report* dan hasil percakapan *agent* di aplikasi *CDR Viewer*.



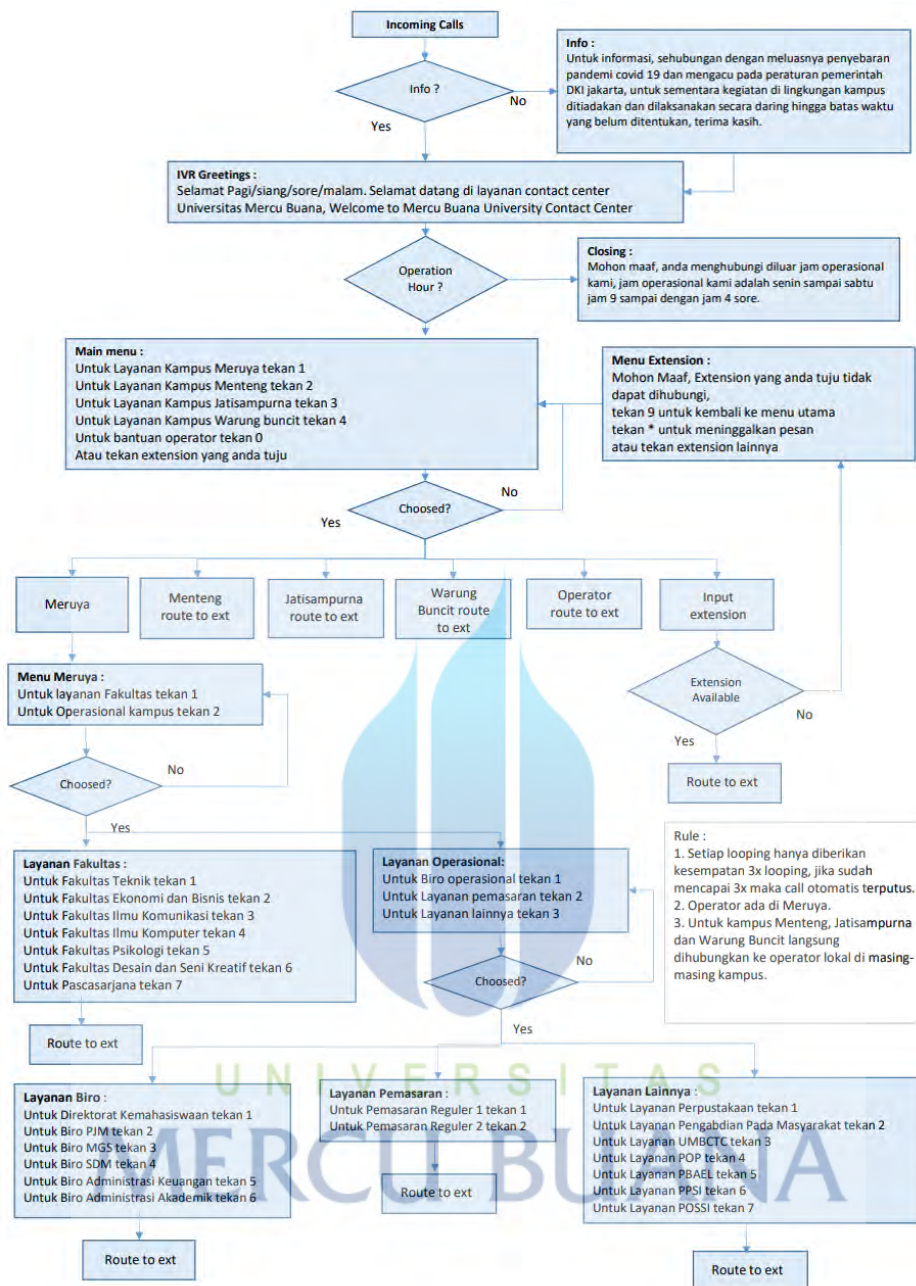
Gambar 6. Proses Bisnis *Call Center*

5) Desain IVR *Call Center*

IP PBX *Asterisk* memiliki fitur utama sebagai pengatur rute panggilan baik itu panggilan masuk ataupun panggilan keluar. Layanan *call center* yang merupakan layanan telepon masuk memiliki kebutuhan untuk konfigurasi IVR dimana pada IVR tersebut akan diatur jam operasional layanan, informasi penting yang perlu diketahui oleh penelpon dan kemana panggilan akan diarahkan.

Gambaran desain alur panggilan masuk *call center* yang akan dibangun didalam sistem IP PBX *Asterisk* digambarkan pada Gambar 7. Disimulasikan Mercubuana menggunakan sistem *single number* sehingga cukup menghubungi satu nomor utama untuk bisa terhubung ke semua cabang kampus, setelah penelepon melakukan panggilan ke nomor akses, penelepon akan mendengarkan *greeting* awal lalu sistem akan melakukan pengecekan jam operasional, jika panggilan dilakukan diluar jam operasional maka akan diarahkan ke informasi penting kampus (jika ada) dan pemberitahuan diluar jam operasional.

Namun jika panggilan dilakukan didalam jam operasional maka penelepon akan mendengarkan *greeting* main menu untuk memilih akan diarahkan ke Kampus Meruya, Menteng, Warung Buncit, Jatisampurna atau tekan langsung *extension* yang akan dituju atau jika penelepon tidak mengetahui *extension* yang akan dituju bisa menekan tombol 0 untuk masuk ke Menu Operator dan akan dihubungkan ke staff *call center*. Jika sudah memilih kampus yang akan dituju maka penelepon tinggal mengikuti arahan untuk menuju departemen yang dituju atau untuk mendengarkan informasi penting kampus.



Gambar 7. Topologi Call Flow Inbound

2.4. Implementation (Implementasi)

Pada tahapan ini dibuat implementasi sistem dimulai dengan konfigurasi DRBD dan pacemaker untuk *high availability system*, instalasi *CDR Viewer* dan *Queue monitoring* untuk fitur *call center* lalu mengintegrasikan seluruh aplikasi dengan IP PBX *Asterisk* dan melakukan pengujian seluruh aplikasi *call center* menggunakan metode *Black box* agar dapat memastikan semua fitur berfungsi normal dan sistem sudah dibangun sesuai kebutuhan.

Seluruh sistem dibangun dan disimulasikan dalam sebuah aplikasi *Graphical Network Simulator3 (GNS3)*

yang merupakan sebuah aplikasi yang dapat mensimulasikan topologi jaringan berbasis GUI seperti *router cisco, juniper, mikrotik, firewall, virtual switch, hub, cloud* dan masih banyak lagi perangkat yang bisa disimulasikan[11].

Gambar 8 merupakan *service – service* yang berjalan pada kluster IP PBX *Asterisk*, ada beberapa komponen dalam kluster IP PBX *Asterisk* yang dibangun seperti *MariaDB, file system MySQL, DRBD, drbdlinks, dan Virtual IP (VIP)*.


```
[root@DRCASTRO2 ~]# pcs status
Cluster name: UMBASTERISKCLUSTER
Stack: corosync
Current DC: DC1 (version 1.1.21-4.el7-f14e36fd43) - partition with quorum
Last updated: Sat Aug 1 14:52:45 2020
Last change: Sat Aug 1 14:52:37 2020 by root via cibadmin on DRC

2 nodes configured
8 resources configured

Online: [ DC1 DRC ]

Full list of resources:

Master/Slave Set: databasebothnode [Database_mysql]
Masters: [ DC1 ]
Slaves: [ DRC ]
fsDRBD_mysql (ocf::heartbeat:Filesystem): Started DC1
MariaDB_Service (systemd:mariadb): Started DC1
MariaDB_VIP (ocf::heartbeat:IPAddr2): Started DC1
Asterisk_VIP (ocf::heartbeat:IPAddr2): Started DC1
drbdlinks (ocf::heartbeat:drbdlinks): Started DC1
asterisk (ocf::heartbeat:asterisk): Started DC1

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
[root@DRCASTRO2 ~]# █
```

Gambar 8. Service Pada Kluster IP PBX Asterisk

Setiap komponen (*resource*) pada kluster ini dikelola oleh *pacemaker* sebagai *high availability resource manager*. Setiap komponen tersebut dikelola agar berjalan bersamaan pada 1 node yang sama untuk menghindari inkonsisten data atau *call service* pada kluster *asterisk*. Selain itu *pacemaker* juga memonitor status node IP PBX *Asterisk* yang *standby* apakah *online* atau tidak. Gambar 9 merupakan konfigurasi *Pacemaker* yang diterapkan pada IP PBX *Asterisk*.

```
[root@DC1ASTR01 corosync]# cat corosync.conf
totem {
    version: 2
    cluster_name: UMBASTERISKCLUSTER
    secauth: off
    transport: udpu
}

nodelist {
    node {
        ring0_addr: DC1
        nodeid: 1
    }
    node {
        ring0_addr: DRC
        nodeid: 2
    }
}

quorum {
    provider: corosync_votequorum
    expected_votes: 2
    two_node: 1
    wait_for_all: 0
}

logging {
    to_logfile: yes
    logfile: /var/log/cluster/corosync.log
    to_syslog: yes
}
[root@DC1ASTR01 corosync]# █
```

Gambar 9. Konfigurasi Pacemaker IP PBX Asterisk

Gambar 10 merupakan konfigurasi *Distributed Replicated Block Device* (DRBD) yang diterapkan pada kluster IP PBX *Asterisk*. *Block Device* yang digunakan pada sistem *DRBD* ini menggunakan *logical volume*. Hal ini dimaksudkan untuk memudahkan administrator jika ada kebutuhan penambahan kapasitas di masa mendatang.

```
[root@DC1ASTR01 drbd.d]# cat global_common.conf
global {
    usage-count yes;
}
resource Asterisk_drbd {
    protocol C;
    handlers {
        pri-on-incon-degr "/usr/lib/drbd/notify-pri-on-incon-degr.sh;
/usr/lib/drbd/notify-emergency-reboot.sh; echo b > /proc/sysrq-trigger
; reboot -f";
        pri-lost-after-sb "/usr/lib/drbd/notify-pri-lost-after-sb.sh;
/usr/lib/drbd/notify-emergency-reboot.sh; echo b > /proc/sysrq-trigger
; reboot -f";
        local-io-error "/usr/lib/drbd/notify-io-error.sh; /usr/lib/drbd
/notify-emergency-shutdown.sh; echo o > /proc/sysrq-trigger; halt -F"
;
        split-brain "/usr/lib/drbd/notify-split-brain.sh root";
    }
    startup {
        wfc-timeout 60;
        degr-wfc-timeout 120;
    }
    options {
    }
    disk {
        on-io-error detach;
    }
    net {
        allow-two-primaries no;
        after-sb-0pri discard-zero-changes;
        after-sb-1pri discard-secondary;
        after-sb-2pri disconnect;
        rr-conflict disconnect;
    }
    on DC1ASTR01.mercubuana.ac.id {
        device /dev/drbd1;
        disk /dev/drbd_vg/drbd_lv;
        address 192.168.17.11:7789;
        meta-disk internal;
    }
    on DRCASTRO2.mercubuana.ac.id {
        device /dev/drbd1;
        disk /dev/drbd_vg/drbd_lv;
        address 192.168.17.22:7789;
        meta-disk internal;
    }
}
[root@DC1ASTR01 drbd.d]#
```

Gambar 10. Konfigurasi DRBD pada IP PBX Asterisk

IP PBX *Asterisk* dibangun dalam sistem operasi *Linux CentOS* dimana seluruh berkas konfigurasi utama terdapat pada direktori */etc/asterisk*. Berkas utama yang dikonfigurasi untuk membangun sistem IVR yaitu terdapat pada *extensions.conf* sebagai *dialplan* yang mengatur alur panggilan.

```
extensions.conf
[IVR Mercuru]
exten = 15000,1,NoOp(IVR Mercuru)
GotoIfTime(00:00-11:00,* ,*,*?pagi)
GotoIfTime(11:01-15:00,* ,*,*?siang)
GotoIfTime(15:01-18:00,* ,*,*?sore)
GotoIfTime(18:01-23:59,* ,*,*?malam)

GotoIfTime(* ,*,26-31,jul?info)
GotoIfTime(09:00-16:00,mon-
sat,* ,*?main_menu)
;Main Menu
Read(Digits,Main_menu,4)
GotoIf("${Ddigits}" = ""?noresponse)
GotoIf("${Ddigits}" = "1"?meruya)
GotoIf("${Ddigits}" = "2"?menteng)
GotoIf("${Ddigits}" = "3"?jatisampurna)
GotoIf("${Ddigits}" = "4"?warungbuncit)
GotoIf("${Ddigits}" = "0"?operator)
GotoIf("${Ddigits}" > "4"?invalid)
GotoIf("${LEN(${Ddigits})}" > "1"?menu_ext)

;menu_ext
GotoIf("${LEN(${Ddigits})}" < "4"?invalid)
Dial(SIP/${Ddigits},60)
Goto($DIALSTATUS)

;NOANSWER
Playback(no_answer)

;BUSY
Playback(busy)

;CHANUNAVAIL
Playback(unavaial)

Read(Digits,Layanan_ext,4)
GotoIf("${Ddigits}" = ""?noresponse)
GotoIf("${Ddigits}" = "9"?main_menu)
GotoIf("${Ddigits}" = "*"?voicemail)
GotoIf("${LEN(${Ddigits})}" > "1"?menu_ext)
```

Aplikasi *CDR Viewer* dibuat dengan menggunakan bahasa pemrograman PHP, basis data yang digunakan adalah MySQL, data yang terdapat dalam basis data diambil dari CDR CSV yang terdapat dalam direktori `/var/log/asterisk/cdr-csv/Master.csv`. File CSV akan diupload secara berkala kedalam basis data MySQL menggunakan fitur *cron job* yang ada pada CentOS. Berkas yang harus dikonfigurasi agar *CDR Viewer* dapat terhubung dengan basis data yaitu `..installdir../asterisk-cdr-viewer/include/config.inc.php`.

config.inc.php

```
$db_type = 'mysql';
$db_host = 'localhost';
$db_port = '3306';
$db_user = 'dbuser';
$db_pass = 'dbpass';
$db_name = 'dbname';
$db_table_name = 'tablename';
$db_options = array();
```

Tampilan layar utama aplikasi *CDR Viewer* seperti pada Gambar 11 terdapat filter untuk *report* yang akan ditampilkan dan jenis *report* yang akan ditampilkan. Filter bisa disesuaikan dengan *Call Date* (tanggal dan waktu panggilan), *Source channel*, *Source*, *Caller ID*, *Extension*, *DID*, *Destination Channel*, *Userfield*, *Account Code*, *Duration*, *Bill Second* dan *Disposition*.

Sedangkan jenis report yang akan ditampilkan bisa dalam bentuk *CDR Search* (tampilan tabel CDR yang ditampilkan langsung dalam aplikasi *CDR Viewer* dimana terdapat fitur untuk mengunduh rekaman panggilan), *CSV file* (menampilkan tombol untuk mengunduh berkas hasil pencarian dalam bentuk .csv), *Call Graph* (grafik yang menampilkan statistik panggilan), *Concurrent Calls* (jumlah panggilan secara bersamaan dalam rentan waktu tertentu), *Minutes Report* (jumlah waktu panggilan dalam rentan waktu tertentu), *Answer-Seizure Ratio/Average Call Duration* (ASR/ACD) *Report* (Jumlah waktu panggilan sukses dan rata-rata waktu panggilan).



Gambar 11. Tampilan *CDR Viewer*

Aplikasi *Queue Monitoring* dibuat dengan menggunakan bahasa pemrograman Python dimana

aplikasi akan diintegrasikan dengan IP PBX Asterisk menggunakan fitur *Asterisk Manager Interface* (AMI), berkas konfigurasi yang harus dikonfigurasi yaitu `manager.conf` yang ada pada direktori `/etc/asterisk/` dan berkas `config.ini` yang ada pada direktori `..installdir../qpanel/`.

manager.conf

```
[general]
enabled = yes
port = 5038
bindaddr = 0.0.0.0
webenabled = no

[qpanel]
secret = password
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.255
read = command
write = command,originate,call,agent
```

config.ini

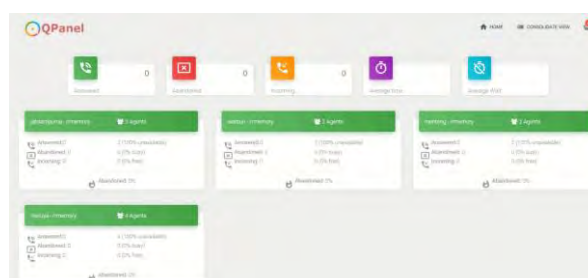
```
[manager]
user = username
password = password
port = 5038
host = localhost
interval = 5
hide = support
language = en
show_service_level = True

[users]
username = user
password = password

[user_queues]
user = meruya, menteng, warbun,
jatisampurna

[reset_stats]
meruya = daily,00:00:00
```

Tampilan layar utama *Queue Monitoring* seperti pada Gambar 12 terdapat beberapa informasi yang ditampilkan terkait dengan status *agent* yang ada dalam *call center* IP PBX Asterisk. Data yang ditampilkan berkaitan dengan jumlah panggilan yang masuk ke layanan, jumlah panggilan yang tidak terjawab, jumlah panggilan masuk yang sedang berlangsung, rata-rata waktu panggilan, rata-rata waktu antrian untuk masuk ke layanan dan status agent yang sedang bertugas.



Gambar 12. Tampilan Layar Utama Queue Monitoring

3. Hasil dan Pembahasan

Metode pengujian yang digunakan dalam penelitian ini adalah metode *Black box* yaitu merupakan sebuah metode pengujian aplikasi berdasarkan fungsi aplikasi yang dibangun. Pengujian menggunakan metode ini bertujuan untuk menemukan kesalahan fungsi yang terdapat pada aplikasi. Pengujian nantinya akan dilakukan dengan cara memberikan sejumlah masukan (valid dan tidak valid) pada aplikasi dan memperhatikan (*examine*) keluaran yang dihasilkan oleh sistem yang kemudian diproses sesuai dengan kebutuhan fungsionalnya[12].

3.1. Pengujian IP PBX Asterisk

Proses pengujian aplikasi IP PBX Asterisk dilakukan dengan melakukan tes pada fitur dasar yang ada pada sistem PABX yang telah dikonfigurasi sebelumnya. Skenario pengujian dilakukan menggunakan extension *MicroSIP* yang didaftarkan pada PC yang ada di setiap kampus yaitu Meruya, Menteng, Warung Buncit dan Jatisampurna. Tabel 6 merupakan hasil pengujian fitur dasar pada IP PBX *Asterisk*. Dari hasil pengujian didapatkan hasil bahwa fitur dasar yang dibutuhkan oleh pengguna dapat diakomodir oleh sistem IP PBX *Asterisk* yang telah dibangun.

Tabel 6. Tabel Hasil Tes Fitur IP PBX *Asterisk*

| Feature | Result |
|----------------------|---------|
| Call antar extension | Success |
| Transfer panggilan | Success |
| Voice Mail | Success |
| Hold Panggilan | Success |
| Record panggilan | Success |

3.2. Pengujian *High Availability*

Proses pengujian yang dibuat untuk menguji *high availability* dari server IP PBX *Asterisk* ini dibagi menjadi 2 yaitu *planned downtime* dan *unplanned downtime*. Skenario *downtime planned* adalah kondisi saat server dimatikan dengan sengaja untuk kebutuhan perawatan sedangkan skenario *downtime unplanned* adalah kondisi saat server mati tanpa disengaja, penyebabnya seperti *network outage*, bencana alam ataupun yang lainnya[13]. Tabel 7 merupakan hasil pengujian *high availability* server IP PBX *Asterisk* dengan kedua skenario tersebut. Skenario *downtime planned* dilakukan pada skenario No. 2,5,8 dan 11. Skenario *downtime unplanned* dilakukan pada skenario No.1,3,6,7,9 dan 10. Skenario ujicoba diatas merujuk pada skenario ujicoba yang dilakukan pada *database cluster*[14]. Dari hasil uji coba dapat dilihat HA pada IP PBX *Asterisk* berjalan dengan baik.

Tabel 7. Tabel Hasil Tes Fitur IP PBX *Asterisk*

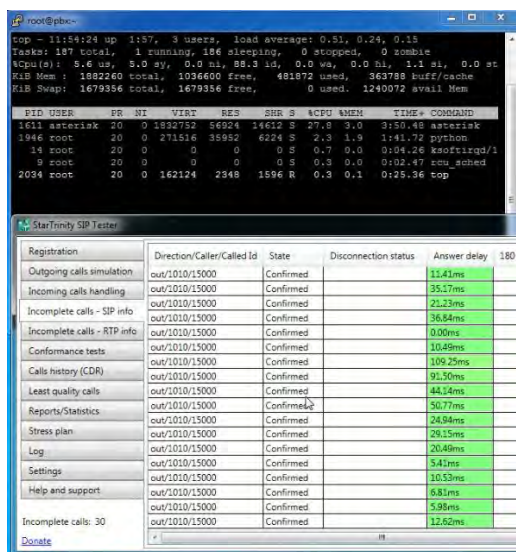
| Scenari | Type of Failure | Failure Node | Active Node | Impact |
|---------|-----------------|--------------|-------------|--------|
|---------|-----------------|--------------|-------------|--------|

| | | | | |
|----|-----------------|---|---|---|
| 1 | Service Down | 1 | 2 | Normal : Service berjalan ada node lain |
| 2 | Node Down | 1 | 2 | Normal : Service berjalan ada node lain |
| 3 | Connection Down | 1 | 2 | Normal : Service berjalan ada node lain |
| 4 | Service Down | 2 | 1 | Normal : Service berjalan ada node lain |
| 5 | Node Down | 2 | 1 | Normal : Service berjalan ada node lain |
| 6 | Connection Down | 2 | 1 | Normal : Service berjalan ada node lain |
| 7 | Service Down | 1 | 2 | Normal : Service berjalan ada node lain |
| 8 | Node Down | 1 | 2 | Normal : Service berjalan ada node lain |
| 9 | Connection Down | 1 | 2 | Normal : Service berjalan ada node lain |
| 10 | Service Down | 2 | 1 | Normal : Service berjalan ada node lain |
| 11 | Node Down | 2 | 1 | Normal : Service berjalan ada node lain |
| 12 | Connection Down | 2 | 1 | Normal : Service berjalan ada node lain |

3.2. Pengujian Aplikasi IVR

Proses pengujian aplikasi IVR dilakukan dengan melakukan *stress tes call* untuk mengetahui kapasitas IVR dalam melayani panggilan secara bersamaan dalam satu waktu. Skenario pengujian yaitu dengan menggunakan aplikasi *SIP Tester Startinity* yaitu merupakan aplikasi penguji beban pada jaringan VOIP, dimana aplikasi tersebut dapat meregistrasi banyak *extension* dari satu PC, melakukan panggilan secara bersamaan dan memonitor panggilan untuk membuat laporan hasil uji secara *realtime*[15].

Aplikasi *Startrinity* akan digunakan untuk mensimulasikan panggilan ke nomor IVR 15000, panggilan akan dilakukan otomatis secara bersamaan selama tiga menit, setiap panggilan berdurasi satu menit dan ketika masuk ke IVR akan diarahkan langsung ke operator Meruya untuk dimonitor didalam aplikasi *Queue Monitoring* dan hasil percakapan bisa dilihat *report* nya di aplikasi *CDR Viewer*. Gambar 13 merupakan tampilan aplikasi *Startrinity* merupakan proses dari pengujian *stress tes call* dengan memperhatikan berapa panggilan yang masuk ke IVR, berapa panggilan yang berhasil masuk ke *agent* dan berapa persen penggunaan CPU ketika dilakukan pengujian panggilan secara bersamaan.



Gambar 13. Tampilan Pada Proses Pengujian *Stress Tes Call*

Tabel 8 merupakan hasil pengujian *stress tes call* yang dilakukan dalam rentan waktu berdasarkan jam operasional Kampus UMB yaitu jam 9 sampai dengan jam 4 sore. Dari hasil pengujian didapatkan hasil bahwa seluruh panggilan hingga 30 panggilan bersamaan sukses ditampung oleh IVR, namun hasil tersebut nantinya akan sangat bergantung pada ketersediaan jalur PSTN yang tersedia untuk menampung panggilan masuk ke IVR yang pada sistem PABX eksisting berjumlah 28. Dari hasil pengujian juga didapatkan data panggilan yang terhubung dengan empat *agent standby* bahwa dari total 120 panggilan yang masuk ke IVR hanya 34 panggilan. Dan didapatkan pula data *CPU usage* ketika panggilan dilakukan penggunaannya masih didalam batas wajar pada saat pengujian maksimal 30 panggilan secara bersamaan tidak lebih dari 30%.

Tabel 8. Tabel Hasil *Stress Tes Call* IVR

| Time | Concurrent | Connected to IVR | Connected to Agent | Not Answered by Agent | CPU Usage |
|---------------|------------|------------------|--------------------|-----------------------|-----------|
| 06:00 - 09:00 | 10 | 10 | 0 | 0 | 5 % |
| 09:00 - 12:00 | 20 | 20 | 0 | 0 | 18,2 % |
| | 30 | 30 | 0 | 0 | 25,2 % |
| 12:00 - 16:00 | 10 | 10 | 4 | 0 | 10,3 % |
| | 20 | 20 | 7 | 3 | 18,5 % |
| 16:00 - 18:00 | 30 | 30 | 9 | 1 | 27,8 % |
| | 10 | 10 | 4 | 1 | 9,3 % |
| 18:00 - 20:00 | 20 | 20 | 5 | 6 | 17,5 % |
| | 30 | 30 | 5 | 4 | 26,2 % |
| 20:00 - 22:00 | 10 | 10 | 0 | 0 | 7,3 % |
| | 20 | 20 | 0 | 0 | 17,9 % |
| 22:00 - 00:00 | 30 | 30 | 0 | 0 | 25,5 % |

Selanjutnya melakukan pengujian IVR dengan melakukan pengecekan menu yang dikonfigurasi pada IVR untuk mengetahui kesesuaian setiap menu tersebut, hal ini untuk menghindari adanya konfigurasi yang tidak sesuai sehingga panggilan tidak diarahkan sesuai dengan menu yang dipilih oleh penelepon. Skenario pengujian yaitu dengan melakukan panggilan secara *step by step* per menu untuk menguji kesesuaian setiap menu dengan melakukan panggilan ke nomor 15000 dan menguji setiap menu yang ada.

Proses pengujian dilakukan hanya pada jam operasional kampus agar panggilan bisa melewati *filter* di *Operational Menu*. Pada *Main Menu* sistem IVR akan meminta penelpon untuk melakukan input sinyal DTMF sesuai dengan menu yang ingin dituju, proses pengujian akan dilakukan dengan kondisi masukan sesuai, tidak sesuai dan tidak melakukan input respon. Tabel 9 merupakan hasil pengujian *Main Menu* sesuai dengan kondisi yang sudah ditentukan sebelumnya, dari hasil pengujian didapatkan hasil bahwa sistem IVR dapat memilah kondisi inputan sesuai dengan kebutuhan.

Tabel 9. Tabel Hasil Pengujian *Main Menu*

| Input | Sub Menu | Condition | Expected Result | Result | Conclusion |
|-------------|--------------------|---------------------|-------------------------|-------------------------|------------|
| 1 | Menu Meruya | | Route To Menu Meruya | Route To Menu Meruya | Success |
| 2 | Menu Menteng | Agent Ready | Pickup | Pickup | Success |
| | | Agent Not Ready | Queuing | Queuing | Success |
| | | After 60s | Busy Announce | Busy Announce | Success |
| 3 | Menu Warung Buncit | Agent Ready | Pickup | Pickup | Success |
| | | Agent Not Ready | Queuing | Queuing | Success |
| | | After 60s | Busy Announce | Busy Announce | Success |
| 4 | Menu Jatisampurna | Agent Ready | Pickup | Pickup | Success |
| | | Agent Not Ready | Queuing | Queuing | Success |
| | | After 60s | Busy Announce | Busy Announce | Success |
| 0 | Menu Operator | Agent Ready | Pickup | Pickup | Success |
| | | Agent Not Ready | Queuing | Queuing | Success |
| | | After 60s | Busy Announce | Busy Announce | Success |
| 99 | | | Invalid greeting | Invalid greeting | Success |
| 6001 | Menu Extension | Extension Avail | Route To 6001 | Route To 6001 | Success |
| | | Extension Not Ready | Route To Menu Extension | Route To Menu Extension | Success |
| No Response | | | No Response Greeting | No Response Greeting | Success |

3.4 Pengujian Aplikasi *Queue Monitoring*

Proses pengujian aplikasi *Queue Monitoring* dilakukan dengan menguji fitur-fitur yang ada pada aplikasi untuk memastikan semua fitur yang ada berfungsi normal. Skenario pengujian dilakukan bersamaan dengan pengujian *stress tes call* yang masuk ke *agent call center* pada pengujian IVR sebelumnya, pada saat pengujian berlangsung aplikasi *Queue Monitoring* akan dilihat beberapa parameter seperti aplikasi dapat memperbaharui data secara *realtime*, kalkulasi waktunya sesuai dan semua fitur-fitur nya dapat digunakan.

Tabel 10 merupakan hasil pengujian fitur-fitur yang ada pada aplikasi *Queue Monitoring* pada tampilan *Normal View*. Dari hasil pengujian didapatkan hasil bahwa semua fitur berfungsi normal. Gambar 14 merupakan tampilan Aplikasi *Queue Monitoring Normal View* pada saat proses pengujian, dapat dilihat bahwa hasil jumlah panggilan secara keseluruhan sama dengan jumlah panggilan yang dilakukan pada saat proses pengujian *stress tes call* yaitu 60 panggilan per rentang waktu layanan.

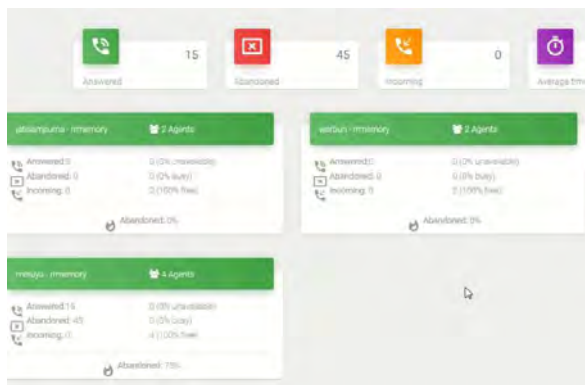
Tabel 10. Tabel Hasil Tes *Queue Monitoring Normal View*

| Menu | Sub Menu | Results |
|-------------------|----------------|---------|
| Data of All Queue | Answered Call | Success |
| | Abandoned Call | Success |
| | Incoming | Success |
| | Average Time | Success |
| | Average Wait | Success |
| Summary of Queue | Name Of Queue | Success |
| | Strategy | Success |
| | Total Agent | Success |
| | Answered | Success |
| | Abandoned | Success |
| | Incoming | Success |
| | % unavailable | Success |

Tabel 11. Tabel Hasil Tes *Search Filter CDR Viewer*

| Filter | Testing 1 | | | Testing 2 | | |
|-------------------|-----------|-----------------|---------|-------------|-----------------|---------|
| | Value | Expected Result | Result | Value | Expected Result | Result |
| Call Date | True | True | Success | False | False | Success |
| Src Channel | True | True | Success | True | True | Success |
| Source | True | True | Success | True | True | Success |
| Caller ID | True | True | Success | True | True | Success |
| Extension | True | True | Success | True | True | Success |
| DID | - | - | - | - | - | - |
| Dst Channel | True | True | Success | True | True | Success |
| Userfield | - | - | - | - | - | - |
| Account Code | - | - | - | - | - | - |
| Duration | True | True | Success | True | True | Success |
| Bill Second | True | True | Success | True | True | Success |
| Disposition | Answered | Answered | Success | No Answered | No Answered | Success |
| Ascending | True | True | Success | - | - | - |
| Descending | - | - | - | True | True | Success |
| Group by | Day | Day | Success | Disposition | Disposition | Success |
| For All Condition | True | True | Success | - | - | - |
| For Any Condition | - | - | - | True | True | Success |

% busy
% free
Success
Success



Gambar 14. Tampilan Aplikasi *Queue Monitoring Normal View*

3.5 Pengujian Aplikasi *CDR Viewer*

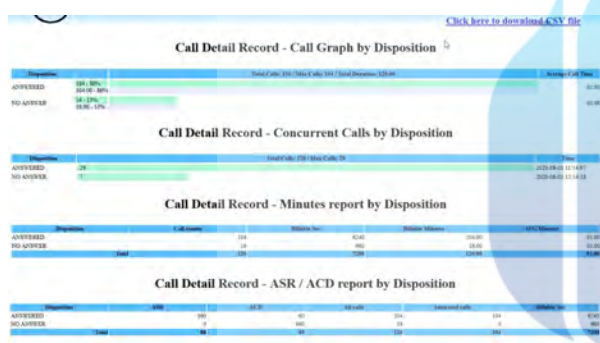
Proses pengujian aplikasi *CDR Viewer* dilakukan dengan menguji fitur-fitur dan *report* yang disediakan pada aplikasi untuk memastikan semua fitur dan *report* berfungsi normal dan sudah sesuai konfigurasi. Skenario pengujian dilakukan yaitu setelah pengujian *stress tes call* pada pengujian IVR sebelumnya, aplikasi *CDR Viewer* akan diakses untuk melihat data panggilan yang terekam dalam aplikasi *CDR Viewer* dengan melakukan *input value* pada *CDR Search Filter*, lalu akan di cek kesesuaian data panggilan dengan hasil pengujian, memutar rekaman panggilan, mengunduh rekaman panggilan dan membuat *report* panggilan. Aplikasi *CDR Viewer* memiliki dua fungsi utama yaitu *CDR Search Filter* dan *Report Type*. Tabel 11 merupakan hasil dua kali pengujian *input value* pada *CDR Search Filter*.

Setelah melakukan filter pada proses pengujian sebelumnya, hasil dari filter tersebut dapat ditampilkan dalam aplikasi *CDR Viewer* dalam enam jenis report yaitu *CDR Search*, *CSV File*, *Call Graph*, *Minutes Report* dan *ASR/ACD Report*. Tabel 12 merupakan hasil pengujian pada *Report Type* yang ada pada aplikasi *CDR Viewer*.

Tabel 12. Tabel Hasil Tes *Report Type CDR Viewer*

| Report Type | Fitur | Conclusion |
|----------------|--------------------|------------|
| CDR Search | Play Recording | Success |
| | Download Recording | Success |
| CSV File | Download CSV File | Success |
| Call Graph | | Success |
| Minutes Report | | Success |
| ASR/ACD Report | | Success |

Gambar 15 merupakan tampilan Aplikasi *CDR Viewer* pada saat proses pengujian *Report Type*, dapat dilihat bahwa hasil report yang ditampilkan secara keseluruhan sesuai dengan jumlah panggilan yang dilakukan pada *stress tes call IVR* sebelumnya.



Gambar 15. Tampilan Report *CDR Viewer*

4. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah dengan memanfaatkan aplikasi *open source* meskipun masih ada beberapa catatan dalam aplikasi yang dibangun untuk dapat diperbaiki namun sudah cukup untuk membangun sebuah *call center* yang layak untuk diimplementasikan di Kampus UMB. Dalam proses pengujian semua fungsi berfungsi normal dan layak diimplementasikan langsung pada sistem *call center* yang sesungguhnya.

Pemanfaatan IP PBX *Asterisk* untuk menyediakan fitur dasar telepon dan alur panggilan untuk menu *IVR call center*, aplikasi *CDR Viewer* untuk menyediakan data keluar masuk nya panggilan di setiap pengguna dan aplikasi *Queue Monitoring* untuk memonitor operasional *call center* sudah cukup dalam memenuhi kebutuhan fungsionalitas bagi seluruh pengguna sistem komunikasi yang ada di Kampus UMB. Selain itu desain sistem yang

menggunakan konsep *high availability* akan semakin meningkatkan pelayanan yang diberikan untuk mendapatkan kepuasan dari setiap pengguna.

Ucapan Terimakasih [jika ada]

Sebutkan nama pemberi dana dan pemberi fasilitas yang membantu.

Daftar Rujukan

- [1] F. Lagraña, *E-mail and Behavioral Changes: Uses and Misuses of Electronic Communications*. London: ISTE Ltd, 2016.
- [2] Universitas Mercu Buana, "Tentang UMB," 2019. [Online]. Available: <https://www.mercubuana.ac.id/id/profile>. [Accessed: 03-Jul-2020].
- [3] W. Sugeng, *Membangun Telepon Berbasis VoIP*. Bandung: Infomatika, 2008.
- [4] A. A. Sukmandhani and F. P. Sulisty, "Integrasi Sistem Telekomunikasi Ip-Pabx Multibrand Pada Lembaga Pendidikan," *Bul. Ekon.*, vol. 16, no. 1, pp. 13–28, 2018.
- [5] 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.
- [6] A. Meidi and I. Nurhaida, "PERANCANGAN CALL CENTER INBOUND MENGGUNAKAN IP PBX ASTERISK DI UNIVERSITAS MERCU BUANA," vol. 3, no. 1, pp. 63–72, 2019.
- [7] Asterisk Development Team, "Asterisk Administrator Guide." [Online]. Available: <https://wiki.asterisk.org/wiki/download/attachments/19005471/asterisk-admin-guide.pdf>. [Accessed: 07-Apr-2020].
- [8] Asterisk Wiki, "Realtime Database Configuration," 2013. [Online]. Available: <https://wiki.asterisk.org/wiki/display/AST/Realtime+Database+Configuration>. [Accessed: 07-Jul-2020].
- [9] A. J. Waite, *Customers: Arriving with a History Leaving with an Experience! How to build your Sales, Service or Helpdesk Call Center to Please Customer*. New York: Telecom Books, 1996.
- [10] Cisco Networking Academy, *CCNP ROUTE: Implementing IP Routing*. Cisco Systems, Inc, 2010.
- [11] J. Saputro, *Praktikum CCNA di Komputer Sendiri Menggunakan GNS3*. Jakarta: Mediakita, 2010.
- [12] J. Watkins, *Testing IT An Off The Shelf Software Testing Process*. 2001.
- [13] M. T. Devyta Asterina, Dr. Ir. Rendy Munadi, and S. T. M.T. , Ratna Mayasari, "Implementasi Dan Analisis Metode Failover Pada Sistem Redundant Dedicated Server Dan Cloud Server Untuk Layanan Voip Implementation and Analysis of Failover Method on Redundant System Dedicated Server and Cloud Server for Voip Service," *e-Proceeding Eng.*, vol. 2, no. 2, p. 3137, 2015.
- [14] H. Handoko, S. M. Isa, S. Si, and M. Kom, "High Availability Analysis with Database Cluster, Load Balancer and Virtual Router Redudancy Protocol," *3rd Int. Conf. Comput. Commun. Syst. ICCCS 2018*, pp. 51–55, 2018.
- [15] StarTrinity, "StarTrinity SIP Tester," 2020. [Online]. Available: <http://startrinity.com/VoIP/SipTester/SipTester.aspx>. [Accessed: 31-Jul-2020].

KERTAS KERJA

Ringkasan

Kertas kerja ini merupakan material kelengkapan artikel jurnal dengan judul di atas. Kertas kerja berisi semua material hasil penelitian Tugas Akhir yang tidak dimuat/atau disertakan di artikel jurnal. Di dalam kertas kerja ini disajikan: literature review, dataset yang digunakan, source code, dan hasil eksperimen secara keseluruhan.

