



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

34525 / H/09



RSE  
004. 62  
Dewa  
9-1  
2009

**TUGAS AKHIR - RE1599**

**ANALISIS PERBANDINGAN KUALITAS JARINGAN VoIP  
PADA SISTEM REAL TIME VoIP SERVICE BERDASARKAN  
PERBEDAAN PLATFORM SISTEM OPERASI**

Ade Chandra Dewa  
NRP 2205109633

Dosen Pembimbing  
Ir. Djoko Suprajitno Rahardjo

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2009

PERPUSTAKAAN ITS	
Tgl. Terima	19-2-2009
Terima Dari	H
No. Agenda Prp.	124

**ANALISIS PERBANDINGAN KUALITAS JARINGAN VoIP  
PADA SISTEM REAL TIME VoIP SERVICE BERDASARKAN  
PERBEDAAN PLATFORM SISTEM OPERASI**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Telekomunikasi Multimedia  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember**

**Menyetujui :**

**Dosen Pembimbing**



**Ir. Djoko Suprajitno Rahardjo**

**NIP. 131 651 447**



## ABSTRAK

*Voice over IP (VoIP)* adalah teknologi yang memungkinkan kemampuan melakukan percakapan telepon dengan menggunakan jalur komunikasi data pada suatu jaringan. Teknologi ini bekerja dengan jalan merubah suara menjadi format digital tertentu yang dapat dikirimkan melalui jaringan IP. Untuk komunikasi pada jaringan VoIP diperlukan *server VoIP*. Performansi dari suatu *server VoIP* dipengaruhi oleh spesifikasi *software* aplikasi *server VoIP* dan spesifikasi *hardware server* seperti kecepatan *processor*, *memory* maupun *storage*. Pada tugas akhir ini dilakukan pengukuran kinerja atau performansi jaringan VoIP pada jaringan *local area network (LAN)* memakai platform *server VoIP* yang berbeda - beda yang dijalankan pada platform sistem operasi yang berbeda pula. Selanjutnya dibandingkan hasil pengukuran performansi atau kinerja dari *server VoIP* tersebut

*Server VoIP* yang digunakan antara lain *Asterisk*, *OpenSER*, dan *Axon* yang dijalankan pada sistem operasi *FreeBSD*, *Linux*, dan *Windows* dengan *database server* sebagai tempat penyimpanan data register manajemen. Parameter pengukuran QoS yang dipakai meliputi *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss*. Untuk melakukan pengukuran QoS dilakukan dengan beberapa skenario yaitu pengukuran QoS tanpa beban, dengan pembebanan sebesar 50% dan 80% dari total bandwidth kapasitas (*fast ethernet*). Jumlah panggilan yang dilakukan terdiri dari 1 sambungan, 3 dan 6 sambungan.

Berdasarkan hasil pengukuran yang telah dilakukan, penambahan jumlah panggilan dari 1, 3 dan 6 panggilan serta pembebanan sebesar 50% dan 80% dari total bandwidth menyebabkan menurunnya QoS. Nilai *delay* yang diperoleh berkisar dari 4 ms sampai 13.3 ms. *Jitter* berkisar dari 13.9 ms sampai 17.9 ms dan *packet loss* berkisar 0.36 ms sampai 11.05 ms.

Kata kunci : *VoIP, LAN, QoS, OpenSER, Asterisk, Axon, Database*

## ABSTRACT

*Voice Over IP ( VoIP )* is a technology that have ability to do the conversation phone by using band of data communications at one particular network. The way of this technology is change voice become digital format which can be delivered through network IP. VoIP server is needed to build the communications of VoIP network. Performance from a server influenced by software VoIP server application specification platform that used VoIP server and hardware server like processor, memory and storage. At this final project doing measurement performance VoIP network in local area network using VoIP server platform that running at operating system that diference. And compared the result of performance between VoIP server platform.

VoIP server that used is Asterisk, OpenSER and Axon that running in FreeBSD, Linux and Windows operating systems and database server is used to data storage management register. QoS measurement parameters that used is call setup delay, delay, jitter, packet loss and CPU usage. For QoS measurement doing with scenario that is QoS measurement without load, with 50% and 80% packet loading from bandwidth capacity (fast ethernet). Calling number that used is 1 calling number, 3 and 6 calling number.

Based on measurement result, addition calling number from 1 number, 3 and 6 number, 50% and 80% packet loading from total bandwidth capacity is influence QoS decrease. Delay value is 4 ms to 13.3 ms. Jitter value is 13.9 ms to 17.9 ms and packet loss is 0.36 ms to 11.05 ms

Keyword : *VoIP, LAN, QoS, OpenSER, Asterisk, Axon, Database*

## KATA PENGANTAR

Puji syukur saya panjatkan kehadiran Tuhan Yang Maha Esa karena berkat rahmat dan hidayah-Nya saya dapat menyelesaikan laporan tugas akhir dengan judul

### ANALISIS PERBANDINGAN KUALITAS JARINGAN VoIP PADA SISTEM REAL TIME VoIP SERVICE BERDASARKAN PERBEDAAN PLATFORM SISTEM OPERASI

Tugas akhir yang mempunyai beban 4 SKS (Satuan Kredit Semester) ini merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan program studi Strata-1 pada Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya. Melalui kegiatan ini diharapkan mahasiswa dapat melakukan kegiatan laporan yang bersifat penelitian ilmiah dan menghubungkannya dengan teori yang telah diperoleh dalam perkuliahan.

Dalam penyusunan laporan tugas akhir ini penulis menyadari akan adanya kekurangan-kekurangan baik dalam penyusunan maupun pembahasan masalah karena keterbatasan pengetahuan penulis. Untuk itu penulis mengharapkan kritik dan saran membangun dari semua pihak agar dapat lebih baik di masa yang akan datang.

Besar harapan penulis bahwa buku tugas akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Jurusan Teknik Elektro pada khususnya.

Surabaya, Januari 2009

Penulis



## UCAPAN TERIMA KASIH

Dalam pengerjaan Tugas Akhir ini penulis banyak menerima bantuan dari berbagai pihak. Penulis sangat bersyukur kepada Allah SWT atas semua karunia yang telah dilimpahkan kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik. Tanpa menghilangkan rasa hormat penulis kepada pihak-pihak yang telah membantu penulis antara lain :

1. Dr. Ir. Mohammad Ashari, M. Eng. selaku Ketua Jurusan Teknik Elektro di Institut Teknologi Sepuluh Nopember Surabaya.
2. Bapak Ir. Djoko Suprajitno Rahardjo selaku dosen pembimbing atas petunjuk, bimbingan, serta dukungan moral yang telah diberikan mulai dari penyusunan hingga penyelesaian Tugas Akhir ini.
3. Seluruh Dosen Penguji (Bapak Dr. Ir. Achmad Affandi, DEA; Bapak Ir. Suwadi, MT; Bapak Ir. Achmad Mauludiyanto, MT; Ibu Devy Kuswidiastuti) yang telah memberi saran dan pertanyaan yang membantu dalam penyempurnaan Tugas Akhir ini.
4. Ibunda Siti Muklati, Ayahanda Ismail Hidayat dan Adinda Alicia Ayu Chandra yang telah memberikan segalanya.
5. Belahan jiwa tercinta Endang Srihartanti, ST yang selalu setia dan sabar menemani tiap perjalanan ini.
6. Keluarga besar dan kerabat yang selalu memberi semangat, dukungan dan doa.
7. Teman - teman satu angkatan dan lab B-301 atas dukungan, semangat dan segala bantuan dalam pengerjaan Tugas Akhir ini.
8. Teman - teman kos Keputih Utara 12, terutama Hardian Boby Arifianto, Ahmad Rifai dan Fajar Budi Nugroho yang selalu memberikan bantuan dan semangat.
9. Teman - teman Bumi Marina Mas, terutama Krishna Kurniawan, Candra Nurmansyah, Alfian Andri.
10. Rekan kerja di PT Indosat Surabaya yang telah membantu dalam pemberian akses penggunaan peralatan.

Serta semua pihak yang telah membantu kelancaran pelaksanaan Tugas Akhir yang tidak bisa disebutkan satu persatu. Penulis mengucapkan terima kasih yang sebesar-besarnya.

## DAFTAR ISI

<b>BAB</b>	<b>Hal.</b>
Halaman Judul.....	i
Halaman Pengesahan.....	iii
Abstrak.....	iv
Abstract.....	v
Kata Pengantar.....	vi
Ucapan Terima Kasih.....	vii
Daftar Isi.....	viii
Daftar Gambar.....	xii
Daftar Tabel.....	xiv
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah.....	3
1.4 Metodologi.....	3
1.5 Sistematika Laporan.....	4
1.6 Tujuan.....	4
1.7 Relevansi.....	5
<b>BAB II TEORI PENUNJANG</b> .....	<b>7</b>
2.1 Pengenalan Teknologi VoIP.....	7
2.2 Konsep Dasar VoIP.....	8
2.3 Protokol penunjang Jaringan VoIP.....	10
2.4 Session Initiation Protocol.....	14
2.4.1 Fungsionalitas Entitas SIP.....	16
2.4.2 Messages pada SIP.....	19
2.4.3 Komunikasi dengan SIP.....	20
2.4.4 Framework SIP.....	23
2.4.5 Aplikasi dan Kelebihan SIP.....	23
2.4.5 Implementasi SIP.....	23
2.5 Parameter Penentu Kualitas Layanan VoIP.....	24
2.5.1 Delay.....	26
2.5.2 Jitter.....	27
2.5.3 Throughput.....	27
2.5.4 Loss Packet.....	27
2.6 Server IP PBX.....	28

2.6.1	Komponen Dasar IP PBX.....	30
2.6.2	Spesifikasi Hardware.....	31
2.7	Platform Server VoIP.....	31
2.7.1	Asterisk Server.....	31
2.7.2	OpenSER Server.....	35
2.7.3	Axon Server.....	36
2.8	Peralatan VoIP.....	36
<b>BAB III PERANCANGAN DAN PEMBUATAN SISTEM.....</b>		<b>39</b>
3.1	Perancangan Topologi Jaringan VoIP.....	39
3.2	Penentuan Server dan Client.....	40
3.3	Persiapan Software dan Hardware yang Diperlukan...	40
3.3.1	Persiapan Hardware.....	40
3.3.2	Persiapan Software.....	42
3.4	Instalasi Software dan Hardware.....	43
3.4.1	Konfigurasi PC Client.....	43
3.4.1.1	Instalasi dan Konfigurasi SoftPhone ....	44
3.4.2	Konfigurasi Server VoIP.....	47
3.4.2.1	Instalasi Asterisk.....	47
3.4.2.2	Konfigurasi Asterisk pada Server.....	48
3.4.2.3	Konfigurasi SIP Asterisk.....	48
3.4.2.4	Konfigurasi Extension Asterisk.....	49
3.4.2.5	Instalasi Database pada Asterisk.....	49
3.4.2.6	Menjalankan Asterisk.....	50
3.4.2.7	Instalasi OpenSER.....	50
3.4.2.8	Instalasi Database OpenSER.....	51
3.4.2.9	Instalasi Axon.....	52
3.4.2.10	Instalasi VQManager.....	54
3.4.2.11	Instalasi Wireshark.....	55
3.5	Skenario Pengukuran.....	55
3.5.1	Kondisi tanpa Beban.....	56
3.5.2	Pembebanan 50%.....	57
3.5.3	Pembebanan 80%.....	57
<b>BAB IV ANALISA DATA DAN PEMBAHASAN.....</b>		<b>59</b>
4.1	Pengukuran Delay.....	59
4.2	Pengukuran Jitter.....	61
4.3	Pengukuran Packet Loss.....	62
4.4	Pengukuran Call Setup Delay.....	64

4.4.1	Pengukuran Call Setup Delay dengan Beban ...	65
4.4.2	Pengukuran Call Setup Delay tiap Softphone dengan Pembebanan.....	66
4.5	Pengukuran CPU Usage .....	67
<b>BAB V</b>	<b>PENUTUP</b> .....	69
5.1	Kesimpulan .....	69
5.2	Saran .....	70
<b>DAFTAR PUSTAKA</b>	.....	71
<b>LAMPIRAN</b>	<b>A</b>	
<b>LAMPIRAN</b>	<b>B</b>	
<b>BIODATA PENULIS</b>		

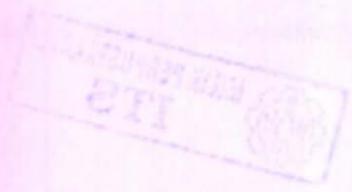
----- Halaman ini sengaja dikosongkan -----

## DAFTAR GAMBAR

Gambar		Hal.
2.1	Prinsip Dasar Jaringan VoIP .....	8
2.2	Topologi Dasar Jaringan VoIP .....	9
2.3	Mekanisme Protokol TCP/IP .....	11
2.4	Header Protokol TCP .....	12
2.5	Header Protokol UDP .....	13
2.6	Header Protokol IP .....	14
2.7	Arsitektur SIP.....	15
2.8	Proxy Server Call Flow.....	17
2.9	Redirect Server Call Flow.....	18
2.10	Basic Call Flow.....	20
2.11	Contoh Sesi Komunikasi SIP.....	22
2.12	Konfigurasi IP PBX .....	29
2.13	Station to Station.....	32
2.14	Komunikasi antar Cabang dengan VoIP .....	35
3.1	Topologi Jaringan VoIP.....	39
3.2	Tampilan Softphone.....	44
3.3	SIP Account X-Lite.....	45
3.4	Pembuatan Account pada X-Lite .....	45
3.5	Pembuatan Account pada Mizu Phone .....	46
3.6	Masuk Network Setting .....	52
3.7	Konfigurasi Network Setting .....	53
3.8	Konfigurasi Extension.....	53
3.9	Tampilan VQManager .....	55
3.10	Skenario Pengukuran Tanpa Beban .....	56
3.11	Skenario Pengukuran Beban 50%.....	57
3.12	Skenario Pengukuran Beban 80%.....	58
4.1	Delay dengan Beban .....	60
4.2	Jitter dengan Beban .....	61
4.3	Packet Loss dengan Beban.....	63
4.4	Call Setup Delay dengan Beban.....	65
4.5	Call Setup Delay pada Softphone dengan Beban.....	66
4.6	CPU Usage dengan Beban .....	67



----- Halaman ini sengaja dikosongkan -----



## DAFTAR TABEL

<b>Tabel</b>		<b>Hal.</b>
2.1	Klasifikasi QoS Class .....	24
2.2	Nilai yang Direkomendasikan ITU-T Y.1541 .....	26
4.1	Perbandingan Pengukuran dengan Rekomendasi .....	64

----- Halaman ini sengaja dikosongkan -----

# BAB I

## PENDAHULUAN

### 1.1. Latar belakang

Perkembangan teknologi dalam bidang industri telekomunikasi telah mampu mengimplementasikan teknologi broadband yang telah memiliki mekanisme pemeliharaan *Quality of Service* (QoS). Seperti saat ini, teknologi internet berbasis IP berkembang sangat pesat dan telah menjadi standar defacto untuk sistem komunikasi data secara global. IP sangat baik dalam segi skalabilitas sehingga membuat teknologi internet menjadi cukup murah.

Kemajuan teknologi internet yang diikuti dengan penyajian *local content* yang tepat bagi para pengguna memunculkan salah satu solusi komunikasi murah yaitu teknologi VoIP dengan layanan multimedia yang mulai diterapkan pada jaringan ini. VoIP kependekan dari *Voice over Internet Protocol* adalah suatu teknologi menyalurkan data berupa suara, video dan data yang sudah dikemas dalam bentuk paket ke dalam jaringan Internet atau jaringan berbasis IP. Tidak seperti pada jaringan non-VoIP, misalkan jaringan PSTN maka suara yang berupa sinyal analog tidak mengalami perubahan terlebih dahulu sebelum disalurkan. Pada VoIP, suara yang berupa sinyal analog diubah menjadi data digital, diberi penomoran dan *header*, kemudian dikirimkan dalam bentuk paket. Di bagian penerima selanjutnya paket tersebut dibongkar dan diurutkan kembali untuk di ubah menjadi sinyal analog. Dari segi kebutuhan pemakaian kanal tentunya dengan teknologi VoIP sangat hemat dibandingkan dengan teknologi telepon digital. Satu kanal suara analog pada telepon digital membutuhkan sekitar 64 Kbps, jika pada VoIP maka satu kanal suara analog bisa diubah dan dikemas dalam bentuk digital ke dalam jaringan internet atau TCP/IP menjadi 8 – 16 Kbps sehingga harganya pun menjadi cukup murah.

Untuk itulah implementasi jaringan VoIP memberikan manfaat yang cukup banyak dalam hal berkomunikasi dengan menggunakan jaringan data antara lain meningkatkan *reliabilitas* jaringan komputer, mampu menekan biaya penggunaan, efisiensi dalam pemakaian *bandwidth*, kemampuan untuk menggunakan kompresi suara, dan sebagainya. Karenanya teknologi VoIP merupakan salah satu bentuk penerapan teknologi tepat guna, dengan adanya teknologi ini dapat diperoleh suatu jaringan telepon yang lebih fleksibel dan membutuhkan biaya yang lebih murah daripada menggunakan fasilitas PSTN yang ada saat ini.

Untuk membangun jaringan VoIP dibutuhkan suatu *proxy* atau *server* VoIP yang biasa disebut dengan IP PBX. Saat ini ada berbagai macam *software* aplikasi untuk *server* VoIP yang dibuat dan sifatnya sudah ada yang *open source* atau dapat digunakan dengan bebas tanpa harus membayar lisensi untuk menggunakan produk tersebut. *Software* aplikasi *server* tersebut dapat dijalankan dengan menggunakan berbagai macam sistem operasi tergantung dari produsen pembuat *software* tersebut agar lebih *user friendly* dalam hal pengimplementasiannya oleh *end user* untuk merancang atau mengembangkan suatu jaringan VoIP.

Pada pengimplementasian suatu jaringan VoIP tentunya kinerja atau performansi dari jaringan tersebut merupakan salah satu hal yang menjadi perhatian, sebab walaupun dengan spesifikasi *hardware* rendah dan *software* aplikasi untuk *server* VoIP yang *open source* dapat memenuhi harapan menginginkan suatu kinerja yang optimum. Performansi dari suatu *server* VoIP tersebut dipengaruhi oleh beberapa hal yaitu spesifikasi dari *software* aplikasi *server* VoIP dan spesifikasi *hardware server* seperti kecepatan *processor*, *memory* maupun *storage* yang berbeda – beda antara *server* yang satu dengan yang lainnya.

Pada tugas akhir ini spesifikasi *software* aplikasi *server* VoIP yang akan digunakan antara lain *Asterisk*, *OpenSER*, dan *Axon* yang akan dijalankan pada sistem operasi yang berbeda yaitu *FreeBSD*, *Linux*, dan *Windows*. *Asterisk* akan dijalankan pada sistem operasi *FreeBSD*, *OpenSER* akan dijalankan pada sistem operasi *Linux* dan *Axon* akan dijalankan pada sistem operasi *Windows*. Untuk mengetahui performansi suatu *server* dilakukan dengan mekanisme *Quality of Service* (QoS) yang meliputi pengukuran *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss*. Dengan membandingkan performansi QoS dari ketiga macam aplikasi *server* VoIP yang dijalankan pada sistem operasi yang berbeda tersebut akan dapat diketahui sejauh mana performansi atau kinerja *server* VoIP yang paling baik antara ketiga macam *server* VoIP itu untuk memenuhi harapan seperti yang dijelaskan sebelumnya.

## 1.2. Permasalahan

Permasalahan yang akan dibahas dalam tugas akhir ini adalah pengukuran kinerja atau performansi jaringan VoIP dari platform *server* VoIP yang berbeda - beda yang dijalankan pada platform sistem operasi yang berbeda pula yaitu menggunakan *server* VoIP *Asterisk*, *OpenSER*, dan *Axon* yang akan dijalankan pada sistem operasi *FreeBSD*, *Linux*, dan *Windows* dengan *database server* sebagai tempat penyimpanan data register manajemen. Parameter pengukuran QoS yang akan digunakan

meliputi *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss*. Selanjutnya hasil pengukuran performansi atau kinerja antara ketiga macam *server* VoIP tersebut akan dibandingkan.

### 1.3. Batasan Masalah

Mengacu pada permasalahan diatas maka batasan masalah pada tugas akhir ini dibatasi dengan asumsi sebagai berikut :

1. Menggunakan komputer sebagai server jaringan VoIP
2. Menggunakan sebuah perangkat *server* untuk digunakan menjalankan aplikasi VoIP *server* beserta *database server*
3. *Server* VoIP terhubung dalam Jaringan Lokal (LAN).
4. Aplikasi *proxy* atau *server* VoIP yang akan dibandingkan antara lain *Asterisk*, *OpenSER* dan *Axon*.
5. Sistem Operasi yang akan dipakai antara lain *FreeBSD*, *Linux* dan *Windows*.
6. *Sofiphone* yang digunakan antara lain *Mizuphone*, *X-lite*, *SIP Communicator*
7. Pengukuran QoS meliputi *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss*.

### 1.4. Metodologi

Metode yang digunakan dalam pembuatan Tugas Akhir ini adalah

1. Studi Literatur  
Mempelajari, mengumpulkan referensi tentang jaringan VoIP, parameter QoS, *database server* serta *software* aplikasi *server* VoIP yaitu *Asterisk*, *OpenSER* dan *Axon*.
2. Desain Sistem  
Merencanakan pembangunan jaringan VoIP dan peralatan yang dibutuhkan baik *hardware* maupun *software* untuk diintegrasikan guna merealisasikan jaringan yang akan dibangun.
3. Pembuatan dan Pengujian Sistem  
Mengaplikasikan jaringan VoIP yang telah direncanakan dan dibangun untuk selanjutnya dilakukan pengujian traffic terhadap jaringan VoIP dan melakukan pengukuran QoS *server* VoIP.
4. Pengumpulan dan Analisa Data  
Data yang diperoleh dari hasil pengujian masing – masing *server* VoIP, selanjutnya akan dianalisa dan dibandingkan guna mengetahui performansi yang dihasilkan oleh jaringan tersebut.

## 5. Penulisan Laporan

Dalam penulisan laporan ini mengacu pada pedoman penulisan ilmiah dalam hal ini penulisan Tugas Akhir baku yang telah diatur oleh Jurusan Teknik Elektro.

### 1.5. Sistematika Laporan

Untuk memudahkan pembahasan, maka tugas akhir ini akan dibagi menjadi lima bab dengan sistematika sebagai berikut :

#### Bab I : Pendahuluan

Bab ini meliputi latar belakang, permasalahan, batasan masalah, metodologi, tujuan, sistematika laporan, dan relevansi.

#### Bab II : Teori Penunjang

Pada bab ini akan dibahas mengenai konsep jaringan VoIP baik topologi jaringan VoIP maupun protokol yang digunakan pada jaringan VoIP, *server* VoIP, parameter pengukuran QoS baik standart maupun metode yang digunakan.

#### Bab III : Perancangan Sistem

Pada bab ini akan dijelaskan desain dari jaringan VoIP dan membangun *server* VoIP yang berbeda platform, metode pengukuran QoS dan skenario pengukuran.

#### Bab IV : Analisa Data dan Pembahasan

Pada bab ini akan membahas data hasil pengukuran yang meliputi *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss* serta analisa dari data yang telah diperoleh dari hasil pengukuran.

#### Bab V : Penutup

Bab ini berisi tentang kesimpulan dan saran yang diperoleh dari hasil pengukuran dan pengujian sistem dalam tugas akhir ini.

### 1.6. Tujuan

Penelitian pada tugas akhir ini bertujuan sebagai berikut

1. Mengukur performansi jaringan komunikasi VoIP dengan mekanisme QoS pada *server* VoIP dengan platform yang berbeda - beda yang dijalankan pada sistem operasi yang berbeda pula
2. Mengevaluasi dan membandingkan performansinya untuk mengetahui *server* VoIP yang paling baik kinerjanya.

### 1.7. Relevansi

Hasil dari Tugas Akhir ini diharapkan dapat membantu dan mempermudah mahasiswa ITS pada khususnya dan masyarakat pada umumnya dalam mengimplementasikan layanan VoIP yaitu merancang dan membangun jaringan VoIP dengan platform *server* VoIP yang berbeda pada sistem operasi yang berbeda pula dengan pertimbangan dari sisi kinerja atau performansi sistem.

----- Halaman ini sengaja dikosongkan -----

## BAB II

### TEORI PENUNJANG

Pada bab ini akan menjelaskan tentang dasar teori yang melandasi pengambilan tema dalam proyek akhir ini. Beberapa hal yang berkaitan dengan jaringan VoIP yang meliputi konsep dasar jaringan VoIP itu sendiri, protokol yang digunakan pada jaringan VoIP, *server* VoIP, kualitas layanan VoIP dan platform sistem operasi yang dipakai untuk menjalankan *server* VoIP tersebut.

#### 2.1. Pengenalan Teknologi VoIP

Cikal bakal atau sejarah munculnya teknologi VoIP dimulai lebih dari 30 tahun yang lalu, dalam hal ini sistem komunikasi yang ada masih menggunakan jalur teknologi telepon PSTN (*Public Switched Telephone Network*). Selang beberapa tahun kemudian mulai berkembang teknologi yang baru. Pembuatan PC (*Personal Computer*) secara masal, sistem komunikasi telepon seluler, dan terakhir munculnya sistem berdasarkan jaringan internet yang memberikan layanan *e-mail* atau teknologi berkirim surat secara elektronik, dan lainnya. Merupakan awal revolusi secara besar – besaran dalam sistem komunikasi. Diseluruh dunia saat semua orang mulai menggunakan komputer dan jaringan internet untuk berkomunikasi satu sama lain, menukar data seperti gambar, suara dan dokumen bahkan berbicara satu sama lain menggunakan aplikasi internet phone atau *netmeeting*. Kemudian muncul penggunaan komunikasi secara *real-time* melalui jaringan internet yang disebut dengan VoIP (*Voice over IP*).

Dengan tuntutan perkembangan zaman yang cukup pesat yang jelas membutuhkan sarana pendukung yang bisa diandalkan dari faktor efisiensi, kecepatan dan biaya maka penelitian terus dilakukan yang diawali dengan kemampuan mengirimkan suara melalui jaringan internet hanya merupakan eksperimen dari beberapa orang atau perusahaan kecil. Ini dimulai dengan perusahaan seperti *Vocaltech* dan kemudian pada akhirnya diikuti oleh *Microsoft* dengan program aplikasi *netmeeting*. Pada saat itu jaringan komputer internet masih sangat lambat. Di rumah - rumah masih digunakan *dial-up* dengan kecepatan 36,6Kbps. *Backbone* internet pun masih kecil, aplikasi yang bersifat menghabiskan bandwidth misalnya suara atau video masih sangat terbatas penggunaannya di pusat penelitian yang memiliki bandwidth besar.

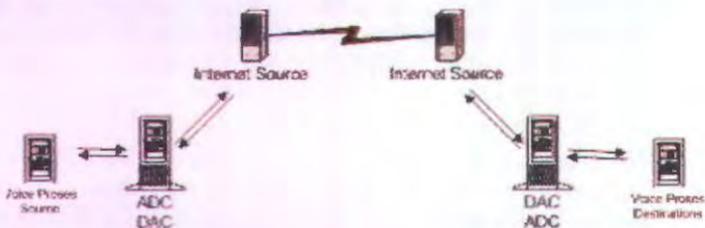
Seiring dengan berpacunya perkembangan teknologi yang cukup pesat. Aplikasi multimedia mulai dapat dilakukan pada jaringan internet,

IP telephony menjadi memungkinkan dijalankan dengan menggunakan infrastruktur internet. IP telephony dapat dibagi menjadi dua kategori yaitu internet telephony dan VoIP. Perbedaan dari keduanya adalah internet telephony menggunakan jaringan internet yang umum (*public internet*), sementara VoIP menggunakan jaringan *privat* meskipun masih menggunakan teknologi yang sama.

## 2.2. Konsep Dasar VoIP

*Voice over Internet Protocol*<sup>1</sup> – adalah teknologi transmisi suara, video dan data melalui *packet-switch* pada jaringan IP, Jaringan IP sendiri merupakan jaringan komunikasi data yang berbasis *packet-switch*, jadi dalam bertelepon menggunakan jaringan IP atau internet.

Prinsip dasar dari VoIP berupa pengiriman sebuah sinyal ke *remote destination* atau tujuan yang dapat dilakukan secara digital. Sebelum dilakukan pengiriman, data yang berupa sinyal analog terlebih dahulu diubah ke bentuk data digital dengan ADC (*analog to digital converter*), kemudian ditransmisikan dan dipenerima dilakukan konversi kembali dari data digital ke data analog dengan DAC (*digital to analog converter*). Begitu juga dengan VoIP, digitalisasi suara atau video dalam bentuk paket data, dikirimkan dan dikonversi kembali dalam bentuk suara atau video dipenerima. Seperti yang terlihat pada gambar sederhana dibawah ini.



Gambar 2.1 Prinsip Dasar Jaringan VoIP

Format digital lebih mudah dikendalikan ataupun diproses, dalam hal ini dapat dikompresi, dan dapat diubah ke format yang lebih baik dan data digital lebih tahan terhadap *noise* daripada data analog.

Infrastruktur pengaplikasian jaringan VoIP untuk pertama kalinya atau sering disebut VoIP tradisional, pada umumnya terdiri dari

<sup>1</sup> M. Iskandarsyah H., "Dasar-dasar Jaringan VoIP", <<http://www.ilmukomputer.com>>, 2005, hal.1

komponen – komponen seperti internet infrastruktur, peralatan *wireless* internet, internet teleponi *gateway* dan PABX. Setelah mengalami perkembangan dan banyak digunakan saat ini, salah satunya penggunaan cara melakukan panggilan dengan menggunakan IP phone. Pesawat telepon khusus ini kelihatannya sama dengan telepon biasa, tapi selain mempunyai konektor RJ-11 standar, IP phone juga mempunyai konektor RJ-45. IP phone menghubungkan langsung dari telepon ke router, dan didalam IP phone sudah ada semua perangkat keras maupun lunak yang terpasang didalamnya dalam menunjang melakukan pemanggilan IP.



Gambar 2.2<sup>2</sup> Topologi Dasar Jaringan VoIP

Gambar 2.2 adalah topologi dasar dari konsep jaringan VoIP. Terdapat 2 jaringan VoIP yang dipisahkan oleh jaringan publik yaitu PSTN (*Public Switch Telephone Network*) dan WAN (*Wide Area Network*). Pada tiap jaringan VoIP terdapat sebuah *host* yang bertugas memberikan layanan VoIP kepada *client*, yaitu berupa *ip phones* dan PC dengan menggunakan perangkat lunak VoIP *client* atau yang lebih dikenal dengan nama *softphone*. Untuk menghubungkan antara jaringan VoIP dibutuhkan sebuah perangkat *gateway* yang bertugas mengintegrasikan dengan jaringan Public. Perangkat *gateway* yang dibutuhkan adalah ATA (*Analog Telephone Adapter*) atau ITG (*Internet Telephone Gateway*).

Dengan bertelepon menggunakan VoIP, banyak keuntungan yang dapat diambil diantaranya dari segi biaya lebih murah dari tarif telepon tradisional. Karena jaringan IP bersifat *global*, sehingga untuk hubungan internasional dapat ditekan. Selain itu, biaya *maintenance* rendah karena *voice* dan *data network* terpisah, sehingga IP Phone dapat ditambah, dipindah dan di ubah. Hal ini karena VoIP dapat dipasang di sembarang *ethernet* dan *IP address*, tidak seperti telepon tradisional yang harus

<sup>2</sup> Ibid, hal.1

mempunyai port tersendiri di sentral atau PBX. Perkembangan teknologi internet yang sangat pesat mendorong ke arah *konvergensi* dengan teknologi komunikasi lainnya. Standarisasi protokol komunikasi pada teknologi VoIP seperti SIP atau H.323 telah memungkinkan komunikasi terintegrasi dengan jaringan komunikasi lainnya seperti PSTN.

### 2.3. Protokol Penunjang Jaringan VoIP

TCP/IP<sup>3</sup> (*Transfer Control Protocol/Internet Protocol*) merupakan sebuah protokol yang digunakan pada jaringan Internet. Protokol ini terdiri dari dua bagian besar, yaitu TCP dan IP. Konsep TCP/IP berawal dari kebutuhan DoD (*Department of Defense*) dari negara Amerika akan suatu komunikasi diantara berbagai variasi komputer yang telah ada untuk kepentingan sistem pertahanan negara. Oleh karenanya pada tahun 1969 dimulailah penelitian terhadap serangkaian protokol TCP/IP. Di antara tujuan – tujuannya sebagai berikut :

- Terciptanya protokol - protokol umum, DoD memerlukan suatu protokol yang dapat ditentukan untuk semua jaringan.
- Meningkatkan efisiensi komunikasi data.
- Dapat dipadukan dengan teknologi WAN (*Wide Area Network*) yg telah ada.
- Mudah dikonfigurasi.

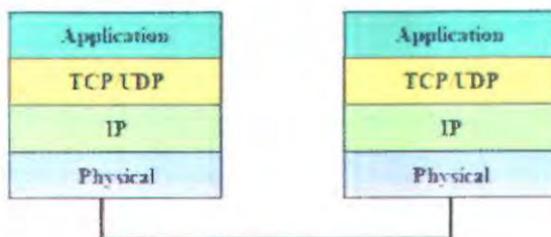
Tahun 1968 DoD ARPAnet (*Advanced Research Project Agency*) memulai penelitian yang kemudian menjadi asal mula *packet switching*. *Packet switching* inilah yang memungkinkan komunikasi antara lapisan jaringan. Protokol - protokol TCP/IP dikembangkan lebih lanjut pada awal 1980 dan menjadi protokol - protokol standar untuk ARPAnet pada tahun 1983. Protokol-protokol ini mengalami peningkatan popularitas di komunitas pemakai ketika TCP/IP digabungkan menjadi versi 4.2 dari BSD (*Berkeley Standard Distribution*) UNIX. Versi ini digunakan secara luas pada institusi penelitian dan pendidikan dan digunakan sebagai dasar dari beberapa penerapan UNIX komersial, termasuk *SunOS* dari Sun dan *Ultrix*. Karena BSD UNIX mendirikan hubungan antara TCP/IP dan sistem operasi UNIX, banyak implementasi UNIX sekarang menggabungkan TCP/IP.

Dikarenakan TCP/IP adalah serangkaian protokol yang setiap protokol melakukan sebagian dari keseluruhan tugas komunikasi jaringan, tentulah implementasinya tidak lepas dari arsitektur jaringan

---

<sup>3</sup> Dostalek, L. & Kabelova, A. "*Understanding TCP/IP a clear and Comprehensive Guide*", PACKT Publishing, 2006, hal.1

itu sendiri. Arsitektur rangkaian protokol TCP/IP mendefinisikan berbagai cara agar TCP/IP dapat saling menyesuaikan. Karena TCP/IP merupakan salah satu lapisan dari protokol OSI *layer* (*Open System Interconnections*), berarti hierarki TCP/IP merujuk kepada 7 lapisan OSI tersebut. Adapun ilustrasi pemrosesan data untuk dikirimkan dengan menggunakan protokol TCP/IP diberikan pada gambar 2.3 di bawah ini.



Gambar 2.3<sup>4</sup> Mekanisme Protokol TCP/IP

- **Application Layer**

Lapisan inilah biasa disebut lapisan akhir (*front end*) atau bisa disebut *user program*. Pada susunan protokol OSI *layer*, *application layer* berada pada susunan lapisan paling atas. Lapisan inilah yang menjadi alasan keberadaan lapisan sebelumnya. Lapisan sebelumnya hanya bertugas mengirimkan pesan yang ditujukan untuk lapisan ini. Di lapisan ini dapat ditemukan program yang menyediakan pelayanan jaringan.

Fungsi utama lapisan ini adalah pemindahan file. Perpindahan file dari sebuah sistem ke sistem lainnya yang berbeda memerlukan suatu sistem pengendalian untuk mengatasi adanya inkompatibilitas sistem file yang berbeda. Protokol ini berhubungan dengan aplikasi, salah satu contoh aplikasi yang telah dikenal misalnya HTTP (*Hypertext Transfer Protocol*) untuk web, FTP (*File Transfer Protocol*) untuk perpindahan file, TELNET menyediakan remote login dalam jaringan, SMTP untuk mengirimkan *e-mail*, DNS untuk memetakan IP address ke dalam nama tertentu, RIP dan OSPF sebagai protokol *routing*, dan lainnya. Oleh karena itu lapisan ini berhubungan langsung dengan aplikasi yang dijalankan oleh *user*.

<sup>4</sup> M. Iskandarsyah H, op.cit, hal.2

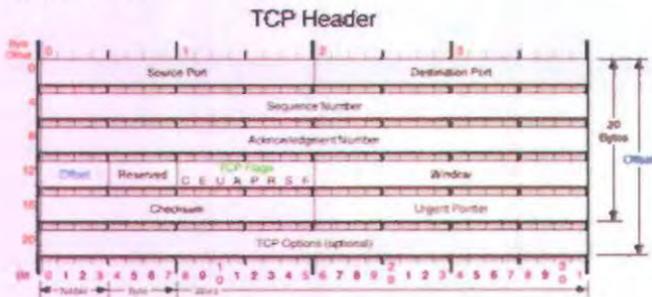
- **TCP (Transmission Control Protocol)**

Dalam transmisi data pada OSI layer *Transport* ada dua protokol yang berperan, yaitu TCP dan UDP. TCP merupakan protokol yang *connection-oriented*, yang artinya menjaga reliabilitas dan konektivitas hubungan komunikasi *end-to-end*.

Konsep dasar kerja TCP adalah mengirim dan menerima segmen-segmen informasi dengan panjang data bervariasi pada suatu datagram internet. TCP menjamin reliabilitas hubungan komunikasi karena melakukan perbaikan terhadap data yang rusak, hilang atau kesalahan pengiriman. Hal ini dilakukan dengan memberikan nomor urut pada setiap oktet yang dikirimkan dan membutuhkan sinyal jawaban positif dari penerima berupa sinyal ACK (*Acknowledgement*). Jika sinyal ACK ini tidak diterima pada interval waktu tertentu, maka data akan dikirimkan kembali. *Header* TCP berisi beberapa *field*, yaitu *source port*, *destination port*, *sequence number*, *acknowledge number*, *checksum*.

Pada sisi penerima, nomor urut tadi berguna untuk mencegah kesalahan urutan data dan duplikasi data. TCP juga memiliki mekanisme *flow control* dengan cara mencantumkan informasi dalam sinyal ACK mengenai batas jumlah oktet data yang masih boleh ditransmisikan pada setiap segment yang diterima dengan sukses.

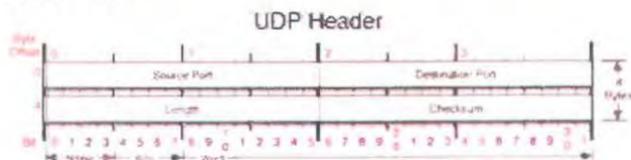
Dalam hubungan VoIP, TCP digunakan pada saat *signaling*, TCP digunakan untuk menjamin setup suatu call pada sesi *signaling*. TCP tidak digunakan dalam pengiriman data suara pada VoIP karena pada suatu komunikasi data VoIP penanganan data yang mengalami keterlambatan lebih penting daripada penanganan paket yang hilang. Berikut ini adalah gambar ilustrasi *header* dari protokol TCP.



**Gambar 2.4 Header Protokol TCP**

- **UDP (User Datagram Protocol)**

UDP merupakan salah satu protokol utama di atas IP dan lebih sederhana daripada TCP. UDP digunakan untuk situasi yang tidak mementingkan mekanisme realibilitas atau dengan kata lain menawarkan layanan datagram tanpa koneksi yang menjamin pengiriman atau pengurutan paket yang dikirimkan secara benar. *Header* UDP hanya berisi empat *field*, yaitu *source port*, *destination port*, *length*, dan *UDP checksum* dimana fungsinya hampir sama dengan TCP, namun fasilitas *checksum* pada UDP bersifat opsional. Gambar dibawah ini menunjukkan gambar ilustrasi dari *header* protokol UDP.



Gambar 2.5 Header Protokol UDP

UDP pada VoIP digunakan untuk mengirimkan audio *stream* yang dikirimkan secara terus menerus. UDP digunakan pada VoIP karena pada pengiriman audio *streaming* yang berlangsung terus menerus lebih mementingkan kecepatan pengiriman data agar tiba di tujuan tanpa memperhatikan adanya paket yang hilang walaupun mencapai 50% dari jumlah paket yang dikirimkan.

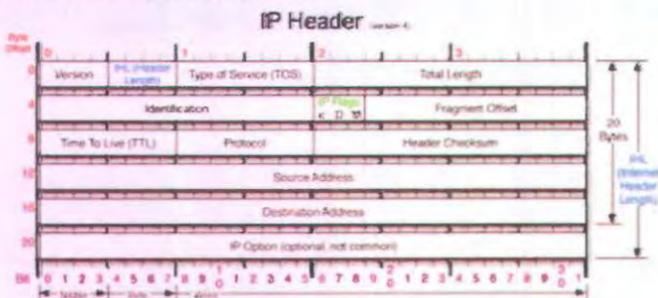
Karena UDP mampu mengirimkan data *streaming* dengan cepat, maka dalam teknologi VoIP UDP merupakan salah satu protokol penting yang digunakan sebagai *header* pada pengiriman data selain RTP (*Realtime Transport Protocol*) dan IP. Untuk mengurangi jumlah paket yang hilang saat pengiriman data (karena tidak ada mekanisme pengiriman ulang) maka pada teknologi VoIP pengiriman data banyak dilakukan pada *private network*.

- **IP (Internet Protocol)**

Internet Protocol didesain untuk interkoneksi sistem komunikasi komputer pada jaringan *packet-switched*. Pada jaringan TCP/IP, sebuah komputer diidentifikasi dengan alamat IP. Tiap-tiap komputer memiliki alamat IP yang unik, masing-masing berbeda satu sama lainnya. Hal ini dilakukan untuk mencegah kesalahan pada transfer data. Terakhir, protokol data akses berhubungan

langsung dengan media fisik. Secara umum protokol ini bertugas untuk menangani pendeteksian kesalahan pada saat transfer data.

Untuk komunikasi data, IP mengimplementasikan dua fungsi dasar, yaitu *addressing* dan *fragmentasi*. Salah satu hal penting dalam IP, pengiriman informasi adalah metode pengalamatan pengirim dan penerima. Saat ini terdapat standar pengalamatan yang sudah digunakan yaitu IPv4 dengan alamat terdiri dari 32 bit. Jumlah alamat yang diciptakan dengan IPv4 diperkirakan tidak dapat mencukupi kebutuhan pengalamatan IP sehingga muncul pengimplementasian sistem pengalamatan yang baru yaitu IPv6 yang menggunakan sistem pengalamatan 128 bit. Berikut ini gambar ilustrasi dari *header IP*.



Gambar 2.6 Header Protokol IP

#### 2.4. Session Initiation Protocol<sup>5</sup>

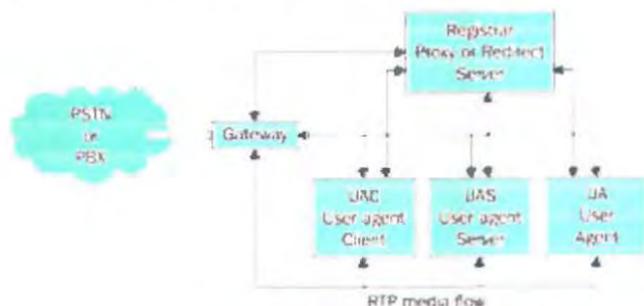
SIP merupakan standar protokol multimedia yang dikeluarkan oleh group yang tergabung dalam *Multiparty Multimedia Session Control* (MMUSIC) yang berada dalam organisasi *Internet Engineering Task Force* (IETF) yang didokumentasikan kedalam dokumen request for command (RFC) 2543 dan RFC 3261 pada bulan maret 1999.

SIP adalah suatu *signalling* protokol pada layer aplikasi yang berbasis ASCII dan berfungsi untuk membangun, memodifikasi, dan mengakhiri suatu sesi multimedia yang melibatkan satu atau beberapa pengguna. Sesi multimedia adalah pertukaran *datastream* antar pengguna yang meliputi suara, video, atau text. Datastream ini dapat dibawa melalui layer TCP maupun layer UDP.

<sup>5</sup> Goncalves, F. "Building Telephony System with OpenSER", PACKT publishing. April 2008, hal.10

Karena SIP merupakan *signalling protocol*, bukan *media transfer protocol*, sehingga SIP tidak membawa paket data *voice* atau *video* atau dengan kata lain SIP tidak menyediakan layanan secara langsung tetapi menyediakan fondasi yang dapat digunakan oleh protokol aplikasi lainnya untuk memberikan layanan yang lebih lengkap bagi pengguna, misalnya dengan RTP (*Real Time Transport Protocol*) untuk transfer data secara *real-time*, dengan SDP (*Session Description Protocol*) untuk mendeskripsikan sesi multimedia, dengan MEGACO (*Media Gateway Control Protocol*) untuk komunikasi dengan PSTN (*Public Switch Telephone Network*). Meskipun demikian, fungsi dan operasi dasar SIP tidak tergantung pada protokol tersebut. SIP juga tidak tergantung pada protokol layer transport yang digunakan. Dalam implementasi VoIP berbasis protokol SIP, RTP (*Real Time Protocol*) digunakan sebagai *media transfer protocol*. SIP menggunakan protokol UDP port 5060, sedangkan RTP menggunakan protokol UDP pada port dinamis (disarankan untuk menggunakan port antara 8000-20000).

Prinsip dasar dari protokol SIP adalah sebagai protokol yang mengatur proses inialisasi dari suatu sesi multimedia serta mengakhiri suatu proses multimedia yang telah berlangsung. Komponen dasar dari sebuah arsitektur SIP adalah *user agent*, *proxy server*, *redirect server* dan *registrar server* seperti pada gambar dibawah ini.



Gambar 2.7<sup>6</sup> Arsitektur SIP

Pada gambar 2.7 menjelaskan bagaimana arsitektur SIP dapat berjalan bersama dengan jaringan PSTN. Arsitektur SIP terdiri dari 4 bagian utama yaitu *user agent*, *proxy server*, *redirect server* dan *registrar server*. Pada topologi di atas Jika *User Agent* ingin melakukan panggilan ke jaringan PSTN, maka ada beberapa hal yang harus

<sup>6</sup> Ibid, hal.10

dijalankan. Pertama *user* harus terdaftar di registrar *server* untuk selanjutnya pada *proxy server* akan meneruskan panggilan melalui *redirect server* menuju *proxy server* pada jaringan yang berbeda, untuk selanjutnya meneruskan panggilan pada perangkat *gateway* yang bertugas menghubungkan antara jaringan VoIP dengan jaringan PSTN.

#### 2.4.1. Fungsionalitas Entitas SIP

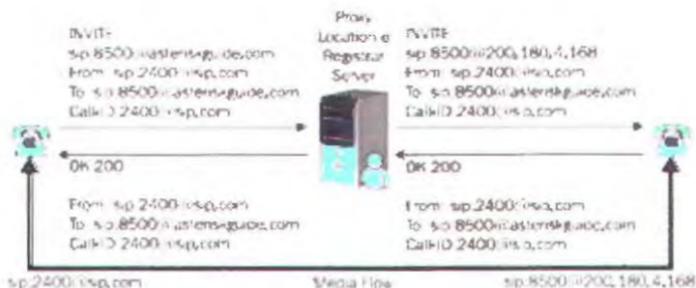
Jaringan SIP terdiri dari 4 macam entitas. Tiap entitas memiliki fungsi yang spesifik dan berpartisipasi dalam komunikasi SIP sebagai *client* (*initiates request*), sebagai *server* (*respond to request*) ataupun sebagai keduanya. Satu "*physical device*" bisa memiliki fungsionalitas lebih dari satu entitas SIP. Sebagai contoh, *network server* yang bekerja sebagai *proxy server* bisa juga berfungsi sebagai registrar pada saat yang sama. Untuk dapat berkomunikasi VoIP, minimal harus terdapat *proxy server*. Keempat macam entitas SIP tersebut antara lain :

##### 1. *User Agent*

Merupakan komponen SIP yang memulai, menerima dan menutup sesi komunikasi. *User Agent* terdiri dari 2 komponen utama, *User Agent Client* (UAC), komponen yang memulai sesi komunikasi dan *User Agent Server* (UAS), komponen yang menerima atau menanggapi sesi komunikasi. Baik UAC ataupun UAS dapat menutup sesi komunikasi. *User agent* dapat berupa software (*softphone*) ataupun hardware (*hardphone*). Contoh beberapa SIP *softphone* (SIP *User Agent* berupa software) antara lain X-Lite, SJPhone, Kphone, Ekiga, Mizu-Phone, LinPhone, SIP Communicator, WengoPhone, Zoiper, dan lain – lain. Sedangkan contoh beberapa SIP *hardphone* (SIP *User Agent* berupa hardware) antara lain seperti ATA (*Analog Telephone Adaptor*), IP Phone atau Ethernet Phone dan USB Phone.

##### 2. *Proxy server*

Merupakan komponen penengah antar *user agent*, bertindak sebagai *server* dan *client* yang menerima *request message* dari *user agent* dan menyampaikan pada *user agent* lainnya, menerjemahkan dan/atau menulis ulang *request message* sebelum menyampaikan pada *user agent* tujuan atau *proxy* lain. *Request* dapat dilayani sendiri atau disampaikan (*forward*) pada *proxy server* lain. *Proxy server* menyimpan seluruh state sesi komunikasi antara UAC dan UAS. Pada gambar 2.8 menjelaskan ilustrasi call flow *proxy sever* dalam membangun koneksi panggilan.



Gambar 2.8<sup>7</sup> Proxy Server Call Flow

Pada gambar 2.8 menjelaskan ilustrasi Proxy Server dalam sebuah komunikasi VoIP. Semua proses yang terjadi pada komunikasi VoIP akan melewati proxy server. Proses diawali ketika user 2400@sip.com melakukan panggilan dengan mengirimkan pesan INVITE yang ditujukan kepada user 8500@200.180.4..168. Selanjutnya server akan meneruskan panggilan kepada user yang dituju dengan mengirimkan pesan INVITE kepada user yang dituju, yaitu 8500@200.180.4..168. Proses akan berlanjut ketika user yang dituju telah siap melakukan panggilan yang ditandai dengan nada dering yang muncul pada perangkat yang digunakan oleh client serta dikirimkannya pesan 200 OK. Selanjutnya setelah pesan diterima oleh user 2400@sip.com

### 3. Redirect server

Merupakan komponen yang menerima request message dari user agent, memetakan alamat SIP user agent atau proxy server tujuan kemudian menyampaikan hasil pemetaan kembali pada user agent pengirim (UAC). Redirect Server tidak menyimpan state sesi komunikasi antara UAC dan UAS setelah pemetaan disampaikan pada UAC. Tidak seperti proxy server, redirect server tidak dapat memulai inisiasi request message dan tidak seperti UAS, redirect server tidak dapat menerima dan menutup sesi komunikasi. Pada gambar 2.9 menjelaskan ilustrasi alur panggilan untuk redirect server.

<sup>7</sup> Ibid, hal.13



Gambar 2.9<sup>8</sup> Redirect Server Call Flow

Perbedaan yang mendasar dari Redirect Server dengan Proxy server adalah pada proxy server, semua sesi komunikasi akan disimpan oleh proxy server sehingga semua paket RTP akan melewati proxy server tersebut. Untuk redirect server, paket INVITE akan diarahkan langsung pada user yang bersesuaian dengan alamat yang dituju, sehingga sesi komunikasi tidak melewati dari server yang melayani panggilan VoIP, atau dengan kata lain bahwa server VoIP langsung mengarahkan sesi panggilan ke user yang bersesuaian.

#### 4. Registrar server

Merupakan komponen yang menerima *request message* REGISTER. Registrar dapat menambahkan fungsi autentikasi user untuk validasi. Registrar menyimpan database user untuk autentikasi dan lokasi sebenarnya (berupa IP dan port) agar user yang terdaftar dapat dihubungi oleh komponen SIP lainnya (berfungsi sebagai Location Server juga). Registrar Server biasa disandingkan dengan Proxy Server.

Dari ketiga macam server tersebut yaitu proxy server, redirect server, dan registrar server yang ketiganya bekerja dalam satu kesatuan sebagai *network server* merupakan satu komponen aplikasi SIP server. Adapun beberapa contoh dari SIP server antara lain Asterisk, Openser, Axon, Openswitch, OnDo Brekeke, SER (SIP Express Router), Yate dan masih banyak lagi lainnya. Beberapa macam contoh SIP server tersebut ada yang dapat dijalankan pada sistem operasi keluarga Windows dan ada pula yang dapat dijalankan pada sistem operasi keluarga UNIX.

<sup>8</sup> Ibid, hal.14

#### 2.4.2. Messages Pada SIP

Hubungan yang dibangun oleh SIP pada proses *signalling* bersifat *client server*. Dengan demikian ada 2 jenis message, yaitu *request* dan *response*. SIP *Request* antara lain :

- INVITE, mengundang *user agent* lain untuk bergabung dalam sesi komunikasi.
- ACK, konfirmasi bahwa *user agent* telah menerima pesan terakhir dari serangkaian pesan INVITE
- BYE, terminasi sesi
- CANCEL, membatalkan INVITE
- REGISTER, registrasi di Registrar *Server*
- OPTIONS, meminta informasi tentang kemampuan *server*
- INFO, digunakan untuk membawa pesan informasi lainnya, seperti informasi inline DTMF

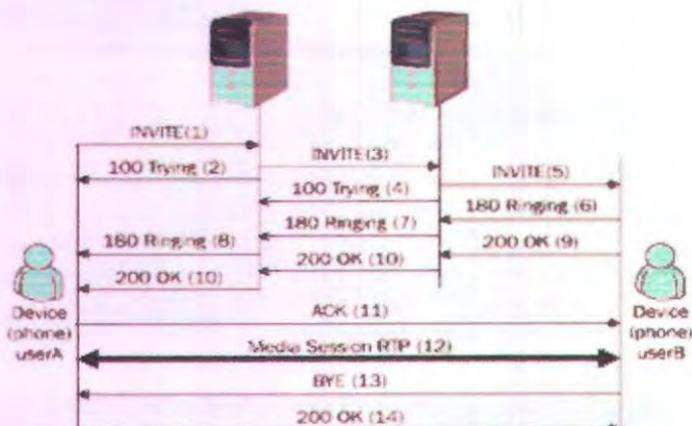
Sedangkan untuk SIP *Response* antara lain :

- 1xx - Informational Message, contoh : 180 artinya Ringing
- 2xx - Successful Response, contoh : 200 artinya OK
- 3xx - Redirection Response, contoh : 302 artinya Moved Temporarily
- 4xx - Request Failure Response, contoh : 403 artinya Forbidden
- 5xx - Server Failure Response, contoh : 504 artinya Gateway Time-Out
- 6xx - Global Failures Response, contoh : 600 artinya Busy Everywhere.

Penggunaan SIP *request* dan *response* dalam proses SIP dapat digunakan untuk mengamati proses terbentuknya sebuah koneksi VoIP serta dapat membantu melakukan pemeriksaan kesalahan konfigurasi pada sistem yang dibangun. Pada sisi *client*, pesan yang ditampilkan pada perangkat yang digunakan di sisi *client* dapat membantu memberikan hipotesa awal tentang kesalahan yang mungkin terjadi. Pada sisi *server* proses hipotesa kesalahan akan sangat membantu bila terjadi gangguan pada komunikasi VoIP yang sedang berlangsung.

Proses panggilan sederhana atau *basic call flow* adalah proses panggilan yang dilakukan antara 2 *user* yang saling terhubung dalam sistem VoIP. Pada gambar 2.10 terdapat 2 *user* yang sedang berhubungan yaitu *user A* yang melakukan panggilan terhadap *user B*.





Gambar 2.10<sup>9</sup> Basic Call Flow

Diagram aliran panggilan komunikasi VoIP dapat dilihat dari gambar 2.10. Dari gambar tersebut terlihat aliran message antara user agent A dengan user agent B. User A sebagai client yang mengawali panggilan, akan mengirimkan pesan INVITE kepada user B. Ketika Pesan tersebut diterima oleh user B, maka akan timbul nada dering pada client B seiring yang ditandai dengan dikirimkannya pesan 180 RINGING. Pesan 200 OK akan dikirimkan oleh client B yang menandakan bahwa client B siap melakukan proses komunikasi. Pesan tersebut akan dibalas oleh Client A dengan mengirimkan pesan ACK atau acknowledgement yang menandakan bahwa client A siap melakukan panggilan. Proses komunikasi ditandai dengan pengiriman paket RTP oleh masing-masing client, sampai salah satu mengakhiri panggilan dengan mengirimkan pesan BYE.

### 2.4.3. Komunikasi Dengan SIP

Pembangunan suatu komunikasi multimedia dengan SIP dilakukan melalui beberapa tahap :

- *User location*, menentukan lokasi pengguna yang akan berkomunikasi.
- *User availability*, menentukan tingkat keinginan pihak yang dipanggil untuk terlibat dalam komunikasi.

<sup>9</sup> Ibid, hal.16

- *User capability*, menentukan media maupun parameter yang berhubungan dengan media yang akan digunakan untuk komunikasi.
- *Session setup*, pembentukan hubungan antara pihak pemanggil dan pihak yang dipanggil (*ringing*).
- *Session management*, meliputi transfer, modifikasi, dan pemutusan sesi.

Komunikasi pada SIP dilakukan dengan mengirimkan message yang hampir sama dengan HTTP dalam hal header, body maupun pengalamatannya. Contoh suatu header SIP :

```
INVITE sip:deep@202.154.179.69 SIP/2.0
Via: SIP/2.0/UDP 202.137.12.237:5060
From: <sip:20014@202.137.12.237>
To: <sip:20006@202.154.179.69>
Call-ID: c2943000-e0563-2a1ce-2e323931@202.137.12.237
CSeq: 100 INVITE
Expires: 180
User-Agent: Cisco IP Phone/ Rev. 1/ SIP enabled
Accept: application/sdp
Contact: sip:20014@202.137.12.237:5060
Content-Type: application/sdp
```

Informasi diatas adalah paket INVITE yang dikirimkan dari salah satu *client* kepada *server* VoIP. Dari informasi diatas, bisa diketahui bahwa *user* dengan extensi 20014 yang terdaftar pada *server* dengan *IP address* 202.137.12.237 akan melakukan panggilan terhadap *user* deep dengan extensi 20006 yang terdaftar pada *server* VoIP yang mempunyai *IP address* 202.154.179.69. Perangkat di sisi *client* atau yang lazim disebut IP phone bisa dilihat pada informasi diatas, yaitu *Cisco IP Phone*. Format alamat yang digunakan oleh SIP mencakup *username* atau *user id* serta *hostname* atau *IP address* dari *server*.

[sip:]<username@host>

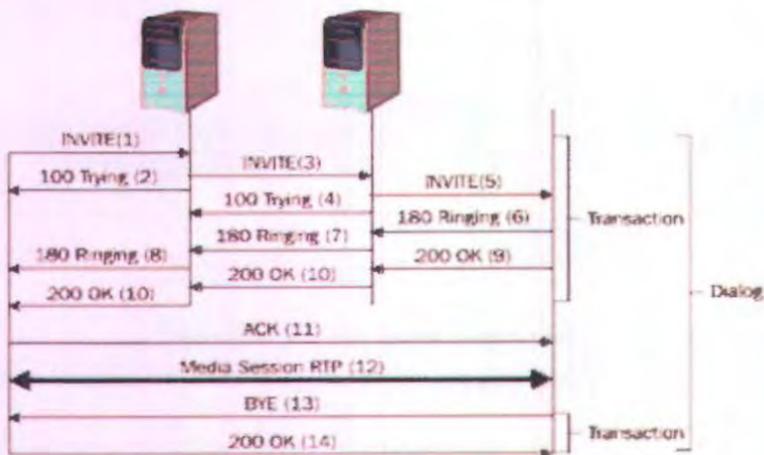
Contoh :

sip:ade@192.168.0.1

Contoh alamat tersebut adalah *user* ade terdaftar di *server* dengan *IP address* 192.168.0.1, seluruh proses panggilan baik yang berasal atau menuju *user* ade akan diproses oleh *server* tersebut, mulai proses panggilan ataupun proses pemutusan sambungan.

Konsep dasar dari komunikasi VoIP dengan menggunakan protokol SIP adalah pengaturan sesi , baik proses inialisasi ataupun

proses pemutusan hubungan atau sambungan. *User* akan mengirimkan permintaan pada *server* untuk membangun sebuah sambungan dengan *user* lain yang telah terdaftar pada *server* yang sama ataupun berbeda dengan *user* yang melakukan panggilan. Komunikasi VoIP secara lengkapnya dapat dilihat pada gambar 2.11.



Gambar 2.11<sup>10</sup> Contoh sesi komunikasi SIP

Dari gambar diatas terlihat proses atau alur komunikasi dari *client* ke *client* VoIP lainnya. Proses inialisasi untuk melakukan proses panggilan diawali dengan dikirimkannya paket INVITE oleh *user agent* yang pertama kepada *proxy server* dimana *user agent* tersebut terdaftar. Paket tersebut diperiksa oleh *server* dan paket tersebut diarahkan kepada *proxy server* dimana *user* yang dituju terdaftar. Jika pesan tersebut sudah diterima, maka *user agent* yang dituju akan membalas pesan tersebut dengan pesan 180 RINGING yang menandakan bahwa *user* tersebut siap untuk melakukan panggilan. Sebelum proses panggilan dilakukan seiring dengan dikirimkannya paket RINGING *user agent* akan mengirimkan pesan 200 OK yang selanjutnya dibalas dengan pesan ACK yang berisi pemberitahuan *user agent* siap melakukan sesi komunikasi suara. Sesi komunikasi akan berakhir ketika salah satu *user agent* mengirimkan pesan BYE yang menandakan bahwa *user agent* ingin mengakhiri sesi komunikasi suara.

<sup>10</sup> Ibid, hal.22

#### 2.4.4. Framework SIP

Framework SIP terdiri dari 3 bagian yaitu:

1. Integrasi  
Kemudahan untuk integrasi dengan protokol lain standar IETF (*Internet Engineering Task Force*).
2. Skalabilitas  
Komponen SIP dapat digabungkan secara fisik dalam *server* yang sama atau berbeda lokasi secara topologis. Pendistribusian komponen memungkinkan penambahan komponen baru tanpa mempengaruhi jaringan yang sudah ada.
3. Simplicity  
Menangani paket SIP lebih mudah dilakukan, seperti pada protokol standar IETF lainnya (HTTP atau SMTP).

#### 2.4.5. Aplikasi dan Kelebihan SIP

Aplikasi dari SIP antara lain digunakan pada VoIP, konferensi multimedia, *text-messaging*, *voice notification*, *callback notification*, *unified messaging* seperti *voicemail to email*. Sedangkan kelebihan – kelebihan protokol SIP antara lain :

- *General-purpose*, protokol SIP dapat diintegrasikan dengan protokol – protocol standar IETF lainnya untuk membuat suatu aplikasi yang berbasis SIP.
- Arsitektur yang terdistribusi dan *scalable*, arsitektur dari *network server* yang terdiri dari *proxy server*, *redirect server*, dan *registrar server* dengan adanya fungsi yang terdistribusi maka jika ada pengembangan yang dilakukan pada salah satu komponen, tidak akan mengganggu komponen yang lain.
- Sederhana, artinya pengiriman *message* berbasis HTTP (text-based), bukan *binary based*. Hal ini menyebabkan SIP mudah diimplementasikan.
- Mobilitas, seorang pengguna dapat menerima *message/call* yang ditujukan kepadanya dengan peralatan PC baik di rumah atau dikantor, *wireless phone*, atau IP phone.

#### 2.4.6. Implementasi SIP

Berkomunikasi dengan SIP-based VoIP dapat dilakukan dengan sistem VoIP yang menggunakan komponen-komponen yang mendukung protokol SIP. Misalnya untuk *server* antara lain Asterisk, Axon dan OpenSER, sedangkan untuk *softphone* antara lain X-Lite atau Mizu Phone. Contoh sistem SIP-based VoIP yang ada di Indonesia adalah jaringan VoIP rakyat yang menggunakan Asterisk sebagai *server*nya.

## 2.5. Parameter Penentu Kualitas Layanan VoIP

Jaringan IP (*Internet Protocol*) adalah jaringan komputer yang menggunakan protokol IP untuk mentransmisikan informasi. Jaringan IP sendiri adalah merupakan jaringan komunikasi data yang berbasis *packet-switch*. Terdapat 4 parameter terhadap kualitas layanan VoIP yaitu *Delay*, *Jitter*, *Throughput* dan *Packet Loss*.

Standar yang digunakan sebagai acuan dalam menentukan kualitas jaringan VoIP adalah standar ITU-T Y.1541 yaitu "*Network Performance Objectives for IP-Based Service*". Rekomendasi ini mengatur parameter dari *Quality of Service* dan kondisi dari jaringan yang berbasiskan *internet protocol*. Pada rekomendasi ini, terdapat beberapa *QoS class*<sup>11</sup> yang digunakan untuk merepresentasikan pembagian kelas pada layanan berbasis *Internet Protocol*, yaitu *QoS class 0* sampai dengan *QoS class 5* dengan aplikasi dan kondisi jaringan yang berbeda untuk tiap *class*. Beberapa parameter yang dijelaskan pada rekomedasi ini adalah :

- IPTD, kependekan dari *IP Packet Transfer Delay* atau biasa disebut dengan *delay*.
- IPDV, kependekan dari *IP Packet Delay Variation*. Parameter ini adalah variasi dari nilai *delay* pada tiap paket, biasa disebut dengan *jitter*
- IPLR, adalah kependekan dari *IP Packet Loss Ratio* atau biasa disebut dengan *packet loss*.

Ketentuan nilai minimum yang harus didapatkan untuk memenuhi kriteria QoS pada jaringan berbasiskan IP, sesuai dengan class yang dibutuhkan dijelaskan pada ITU-T Y.1451 pada tabel 2.1

Tabel 2.1<sup>12</sup> Klasifikasi QoS Class

QoS Class	Application	Node Mechanism	Network Techniques
0	Real-time, Jitter sensitive, high interaction (VoIP)	Separate Queue with preferential servicing, traffic grooming	Constrained routing and distance
1	Real-time, Jitter sensitive, high interaction (VoIP)		Less Constrained routing and distance

<sup>11</sup> ITU-T Recommendation Y.1541. "*Network Performance Objectives for IP-based services*", International Telecommunication Union, 2006, hal.11

<sup>12</sup> Ibid, hal.11

Tabel 2.1<sup>13</sup> Klasifikasi QoS Class (Lanjutan)

QoS Class	Application	Node Mechanisms	Network Techniques
2	Real-time, jitter sensitive, high interaction (VoIP)	Separate queue, drop priority	Constrained routing and distance
3	Transaction data, interactive		Less Constrained routing and distance
4	Low loss only (short transactions, bulk data, video streaming)	Long queue, drop priority	Any route path
5	Traditional applications of default IP networks	Separate queue (lowest priority)	Any route path
NOTE – Any example application listed in table 2 could also be used in class 5 with unspecified performance objectives as long as the users are willing to accept the level of performance provided during their session.			

Pada tabel 2.1 terdapat pembagian *QoS class* untuk layanan berbasis Internet Protocol (IP). Penggunaan *QoS class 0* dan *1* untuk aplikasi yang bersifat *realtime*, *jitter sensitive*, dan membutuhkan interaksi dengan intensitas yang tinggi seperti VoIP. Untuk *QoS class* yang lain digunakan sesuai dengan sifat aplikasi yang digunakan. Contohnya adalah *QoS class 2* dan *3* untuk transaksi data dalam intensitas yang tinggi, sedangkan *QoS class 4* untuk aplikasi data dengan *loss* yang kecil serta *class 5* untuk aplikasi standar jaringan IP.

Beberapa class yang didefinisikan oleh standar ITU-T Y.1541 didasarkan pada beberapa kondisi dan teknik jaringan yang digunakan. Salah satu contoh adalah *class 0* yang dipilih berdasarkan digunakannya pembatasan routing dan jaringan serta dilakukannya *traffic shaping*, yaitu pembatasan bandwidth yang digunakan dalam jaringan. Pada *QoS class 4* dan *5* bisa digunakan untuk tiap jalur dari jaringan yang digunakan, yang membedakan adalah pada *QoS class 4*, mekanisme dari tiap node adalah terdapat antrian paket yang panjang serta adanya prioritas pada proses *drop* paket, sedangkan pada *QoS class 5* digunakan antrian yang terpisah pada prioritas antrian yang terendah pada tiap paket. Pada rekomendasi ITU-T Y-1541 juga menjelaskan tentang batasan nilai kualitas dari layanan dari sebuah layanan berbasis *internet protocol*. Nilai tersebut didasarkan pada beberapa aspek pada tiap kelas

<sup>13</sup> *Ibid*, hal. 11

yang didefinisikan pada tabel 2.1. Sedangkan nilai parameter yang dianjurkan oleh rekomendasi ini dapat dilihat seperti pada tabel 2.2 dibawah ini :

Tabel 2.2<sup>14</sup> Nilai yang direkomendasikan ITU-T Y.1541

Network performance parameter	Nature of network performance objective	QoS Classes					
		Class 0	Class 1	Class 2	Class 3	Class 4	Class 5 Unspecified
IPD	Upper bound on the mean IPD	100 ms	400 ms	100 ms	400 ms	1 s	U
IPDV	Upper bound on the $1 - 10^{-5}$ quantile of IPD minus the maximum IPD	50 ms	50 ms	U	U	U	U
IPLR	Upper bound on the packet loss probability	$1 \times 10^{-3}$	U				
IPEE	Upper bound	$1 \times 10^{-4}$					U

Seperti yang terlihat pada tabel 2.2 diatas, nilai *delay* yang dianjurkan oleh ITU untuk *delay* pada aplikasi VoIP adalah 100 ms untuk *class 0* dan *class 2* serta 400 ms untuk *class 1* dan *class 4*. Untuk nilai IPDV (IP Packet Delay Variation) atau yang biasa disebut *jitter* adalah 50 ms untuk *class 0* dan *class 1*. Nilai IPLR atau IP Packet Loss Ratio yang dianjurkan pada aplikasi VoIP adalah bernilai  $1 \times 10^{-3}$ . Pada Tugas akhir ini dipilih *class 0* sebagai acuan dari nilai QoS VoIP yang diukur, karena kondisi jaringan yang sesuai dengan parameter yang digunakan oleh ITU. Nilai yang digunakan sebagai acuan dari kualitas jaringan VoIP adalah 100 ms untuk *delay*, 50 ms untuk *jitter*, dan  $1 \times 10^{-3}$  untuk nilai *packet loss*. *Class 0* dipilih sebagai acuan dalam pengerjaan tugas akhir ini, disebabkan selain karena sifatnya yang bersifat realtime, topologi jaringan dan teknik yang digunakan dibatasi dengan menggunakan teknik tertentu, sesuai dengan karakteristik dari *class 0*.

### 2.5.1. Delay

Yang dimaksud dengan *delay* adalah waktu yang dibutuhkan oleh paket untuk dikirimkan dan diterima penuh oleh penerima (RFC 2679). Dan *delay* maksimum yang direkomendasikan oleh ITU-T Y.1541 untuk

<sup>14</sup> Ibid, hal.8

aplikasi suara adalah 100 ms, yang masih dapat diterima pengguna adalah 250 ms.

Beberapa *delay* yang dapat mengganggu kualitas suara dalam VoIP adalah :

- Propagation *delay* yaitu *delay* yang terjadi akibat transmisi melalui jarak antar pengirim dan penerima
- Serialization *delay* yaitu *delay* pada saat proses peletakan bit ke dalam circuit
- Processing *delay* yaitu *delay* yang terjadi saat proses coding, compression, decompression dan decoding
- Packetization *delay* yaitu *delay* yang terjadi saat proses paketisasi digital voice sample
- Queuing *delay* yaitu *delay* akibat waktu tunggu paket sampai dilayani)
- Jitter buffer yaitu *delay* akibat adanya buffer untuk mengatasi jitter).

#### 2.5.2. Jitter

Merupakan variasi *paket delay* yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima. Untuk mengatasi *jitter* maka paket data yang datang dikumpulkan dulu dalam *jitter buffer* selama waktu yang telah ditentukan sampai paket dapat diterima pada sisi penerima dengan urutan yang benar. Nilai jitter yang direkomendasikan oleh ITU – T Y.1541 adalah dibawah 50 ms.

#### 2.5.3. Throughput

*Throughput*, adalah jumlah data per satuan waktu yang dikirimkan melalui sebuah saluran fisik atau logik, atau melewati sebuah *node* jaringan. *Throughput* biasanya dinyatakan dalam bit per second (bit/s atau bps). *System throughput* atau *aggregate throughput* adalah jumlah total *data rate* yang dikirimkan menuju semua terminal dalam jaringan. *Throughput* maksimum sinonim dengan kapasitas sebuah saluran jaringan, yaitu ketika jumlah beban (jumlah data yang datang) adalah sangat besar.

#### 2.5.4. Packet Loss

Kehilangan paket ketika terjadi *peak load* dan *congestion* (kemacetan transmisi paket akibat padatnya traffic yang harus dilayani) dalam batas waktu tertentu. *Packet Loss* dinyatakan dalam persen (%) dengan nilai yang direkomendasikan pada ITU-T Y.1541 tidak boleh lebih dari 0.1 % dengan batas toleransi 1%.

Pengamatan paket ini dilakukan dengan cara mengamati jumlah paket yang dikirimkan maupun yang diterima oleh masing-masing *client* yang melalui protokol TCP/IP dan yang sangat penting dalam pengamatan paket adalah adanya paket *loss* saat melakukan komunikasi dengan menggunakan *voip*. Karena saat terjadi *packet loss* suara dan gambar yang terjadi pada sisi penerima menjadi terputus – putus, paket *loss* disebabkan karena *delay* yang terlalu besar. Selain parameter-parameter lain yang mempengaruhi VoIP adalah Quality of Service (QoS), parameter tersebut adalah :

- Pemenuhan kebutuhan bandwidth
- Keterlambatan data (*latency*)
- Jenis kompresi data
- Interopabilitas peralatan (*vendor* yang berbeda)
- Jenis standar multimedia yang digunakan (H.323/SIP/MGCP)

Untuk berkomunikasi dengan menggunakan teknologi VoIP maka yang harus real time adalah jitter, echo dan *loss packet*.

## 2.6. Server IP PBX

*Private Branch Exchange* (PBX) merupakan teknologi komunikasi yang mengatur hubungan *telephone* antar pelanggan tanpa harus melalui sentral lokal, serta berfungsi sebagai *gateway* dalam koneksi dengan jaringan *PSTN* (*Public Switched Telephone Network*).

- *Telephone analog*  
Terminal *telephone* ini terletak pada pelanggan yang terdiri dari *mikrophone* dan *earphone*, *digit dial number* serta *interface RJ 11*.
- *Sentral PABX*  
Sentral PABX merupakan pusat koordinasi dari semua komponen jaringan PABX yang berfungsi sebagai *call processing* dan *call switching*. Serta berisi database dari pelanggan PABX tersebut.

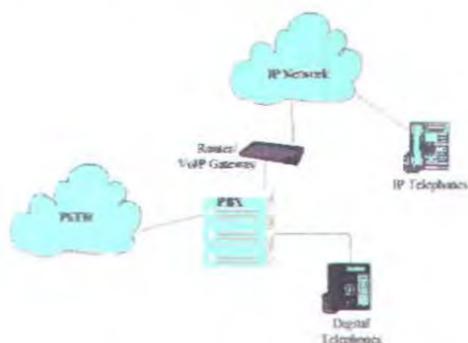
PBX dirancang dengan dua *interface* yaitu, *line* dan *trunk*, *internal call processing* berfungsi sebagai *call switching hardware* dan *software*. *Line interface* dihubungkan dengan terminal *telephone* dan *faximile*, sedangkan *trunk interface* dihubungkan dengan sentral lokal (Jaringan *PSTN*) atau sentral PBX lain. *Call processing* berfungsi untuk membandingkan digit yang ditekan dengan pola alamat yang dikonfigurasi pada sentral telepon, melakukan proses pensinyalan serta melakukan proses penyambungan pada segi hardware.

Komponen *internal PABX* yaitu *call processing* dan *call switching* berfungsi menyediakan intelegensia untuk menerjemahkan *digit* yang ditekan menjadi lokasi / alamat tujuan serta menyambungkan

panggilan yang datang dengan *interface* yang sesuai. Misalnya, seorang pelanggan *PABX* melakukan panggilan ke pelanggan lain, dari sentral *PABX* yang sama, maka sentral *PABX* tersebut harus mengenali *digit* nomer yang ditekan, menentukan *port* yang berhubungan dengan *digit* nomer yang dihubungi, memberikan nada dering pada *telephone* yang dituju, serta meghubungkan panggilan tersebut saat yang dipanggil (*called party*) menjawab.

Penerjemah *digit* nomer yang ditekan menjadi lokasi atau alamat tujuan merupakan fungsi *call processing* sedangkan proses hubungan *inbound call* dengan *outbound inteface*, merupakan fungsi *call switching*. *Switch / Router* berfungsi mengarahkan paket data yang datang ke jaringan data sesuai dengan alamat tujuannya. Secara konvensional terdapat 2 jaringan yang berbeda yaitu jaringan suara (*Circuit Network*) dan jaringan paket data (*Packet Data Network*). Akan tetapi seiring dengan perkembangan teknologi telekomunikasi, terdapat suatu teknologi yang memungkinkan komunikasi suara dan faksimili mengalami proses paketisasi dan dikirimkan melalui jaringan paket data yang dikenal dengan teknologi *Voice over Internet Protocol (VoIP)*.

Secara umum VoIP didefinisikan sebagai suatu sistem yang menggunakan jaringan internet untuk mengirimkan data paket suara dari suatu tempat ke tempat lain menggunakan perantara protokol IP. Layanan VoIP inilah yang merupakan *trend* ke depan untuk era komunikasi voice yang sebelumnya berbasiskan *circuit switching*. Dengan layanan VoIP yang terinstall pada jaringan IP, maka keperluan administrasi dan *maintenance* dapat dengan mudah dilakukan. Oleh karena itu implementasi layanan VoIP di berbagai lembaga telah banyak dilakukan terutama untuk menggantikan jaringan PBX konvensional. Layanan VoIP ini secara umum disebut VoIP PBX / IP PBX.



Gambar 2.12 Konfigurasi IP PBX

Jaringan VoIP PBX dibangun pada suatu area lokal yang dilayani oleh *server* IP PBX untuk komunikasi internal pada suatu area lokal tersebut tanpa harus melalui sentral telepon milik operator PSTN. Begitu juga apabila instansi tersebut memiliki jaringan / kantor cabang di tempat / kota lain, maka dengan sistem yang sama di tempat tersebut dapat dilakukan komunikasi *voice* melalui jaringan IP / internet tanpa harus melalui sentral telepon milik operator PSTN. Dengan demikian biaya yang dikeluarkan oleh instansi tersebut menjadi lebih murah karena untuk komunikasi *voice* internal dan antar cabang tidak perlu berlangganan ke operator PSTN melainkan hanya pada internet *provider* saja, itupun sekaligus dengan layanan koneksi internet pada *provider* tersebut.

Pada awalnya, bentuk jaringan hanya antar lokasi (satu tempat) untuk penggunaan kalangan terbatas. Namun pada perkembangannya, kebutuhan jaringan VoIP lebih kompleks, tidak hanya sebatas komunikasi suara saja. Misalnya, untuk penggunaan sambungan antar-cabang, VoIP digunakan sebagai penyambung antar PABX. Teknologi di atasnya adalah teknologi penggabungan PABX yang tidak lagi menggunakan jaringan lokal, tetapi telah menggunakan jaringan internet sebagai bentuk komunikasi antar cabang kantor atau perusahaan tersebut.

### 2.6.1 Komponen Dasar IP PBX

Komponen dasar pada IP PBX terdiri dari 2 macam yaitu *data account* dan *dial plan*. Kedua hal ini menjadi dasar yang paling fundamental dalam membuat *server* IP PBX.

- *Data Account*, terdiri dari *extension* dan *trunk*  
*Extension* merupakan data *account* yang akan digunakan oleh *extension* agar terhubung dengan IP PBX ini. *Extension* adalah sebuah nama atau nomor yang merepresentasikan user dari IP PBX ini.  
*Trunk* merupakan data *account* yang akan digunakan IP PBX untuk menghubungi *trunk*. *Trunk* adalah sebuah nama atau nomor yang merepresentasikan server lain atau IP PBX lain yang akan dihubungi oleh IP PBX ini.
- *Dial Plan*  
Merupakan aturan *dial* yang akan dimanfaatkan oleh *extension* untuk menghubungi sesama *extension* atau *trunk* dan sebaliknya.

## 2.6.2 Spesifikasi Hardware

Pada tugas akhir ini digunakan sebuah PC untuk dijadikan *server* VoIP. PC terdiri dari beberapa komponen-komponen utama yang sangat berpengaruh terhadap kinerja dari PC itu sendiri. Komponen-komponen tersebut diantaranya adalah:

- *Processor*  
Merupakan otak dari sebuah komputer. Semakin tinggi *clock* dari sebuah *processor* maka akan semakin cepat kinerja dalam mengolah data.
- *Memory*  
Merupakan media penyimpanan sementara sebelum data diolah oleh *processor*. Semakin besar ukuran dari *memory* akan semakin cepat kinerja komputer tersebut karena data dapat diolah secara cepat.
- *Storage*  
Merupakan media untuk penyimpanan data secara permanen. Data yang tersimpan dalam *hard disk* akan berpengaruh terhadap kinerja dari sebuah komputer. Semakin banyak data dalam media penyimpanan maka akan semakin memperberat kinerja komputer. *Free space* media penyimpanan juga akan berpengaruh terhadap kinerja komputer, semakin besar *free space* yang tersedia maka kita dapat memperbesar ukuran *paging file* untuk virtual memory. Virtual memory ini akan membantu kinerja dari sebuah komputer.

## 2.7 Platform Server VoIP

Platform *server* VoIP yang dimaksud disini adalah jenis model *server* VoIP atau *software* aplikasi *server* VoIP yang dipakai dan dijalankan pada sistem operasi tertentu sesuai dengan ketentuan *software* aplikasi tersebut dapat berjalan untuk membangun suatu jaringan VoIP.

### 2.7.1. Asterisk<sup>15</sup> Server

*Asterisk* adalah full fitur *software* dari IP PBX, yang dikembangkan oleh GNU/Linux untuk x86 tetapi juga dijalankan pada operating sistem yang lain seperti BSD dan Mac. Dengan menggunakan *Asterisk* kita dapat mengakses source code, memungkinkan untuk menggabungkannya dengan sistem operasi yang lain bahkan

---

<sup>15</sup> Gomillion, D. & Dempster, B. "Building telephony System with Asterisk", Packt Publishing, 2005, hal.5

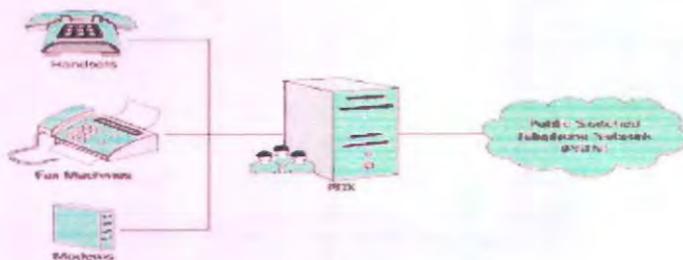
memodifikasinya. *Asterisk* memiliki banyak fitur dan mendukung terhadap protokol dalam VoIP. *Asterisk* juga menyediakan layanan *voicemail*, servis *directory*, *conferencing*, *Interactive Voice Response (IVR)* dan fitur-fitur yang lain.

*Asterisk* didesign sangat flexible dan *extensible* serta mampu menyediakan *central switching core*, *hardware interface*, *handling file format* dan *codec*. *Asterisk* (Digium) menciptakan open source *Inter Asterisk Exchange (IAX) channel protocol* (digunakan untuk *tunnel signaling* dan media antara *asterisk-compliant IP PBX*). *Asterisk* juga support terhadap protokol *SCCP Cisco*. Adapun fungsi dasar dari *Asterisk* dan beberapa fitur serta kelebihanannya seperti berikut ini

- **Fungsi dasar *Asterisk***

Pada dasarnya *Asterisk* mempunyai fungsi sebagai PBX, dan sebagai link trunking. Sebagai PBX, *Asterisk* menawarkan fitur panggilan antar pengguna (*station to station*). Dengan fungsi ini ini, pengguna dapat melakukan panggilan kepada pengguna lain yang berada dalam satu PBX. Perbedaan yang nyata antara *Asterisk* dan konvensional PBX adalah kebebasan pada *handset*. Pada sistem konvensional untuk mempunyai dukungan terhadap *multiple phone* dan *line* dengan memasang *outgoing line* pada tiap-tiap telepon.

Pemanfaatan *Asterisk* untuk menjalankan aplikasi yang selama ini sudah berjalan seperti panggilan telepon dan sistem fax dapat dilihat pada ilustrasi gambar 2.13. Selain memenuhi kebutuhan dasar untuk melakukan panggilan telepon dengan menggunakan jaringan IP, *asterisk* dapat dimanfaatkan untuk mengirimkan pesan fax melalui modem yang sudah terpasang.



Gambar 2.13<sup>16</sup> Station to station

<sup>16</sup> Ibid, hal.6

Gambar 2.13 adalah salah satu contoh fungsi station to station pada *Asterisk*. Semua ekstensi yang terdaftar pada PBX, dapat melakukan langsung antar keduanya melalui PBX, sebagai contoh adalah komunikasi dari modem ke fax. Fungsi dasar kedua dari *Asterisk* adalah line trunking, dimana membagi akses ke beberapa line telepon. Line telepon tersebut biasanya dihubungkan dengan jaringan telepon global atau yang biasa disebut dengan *Public Switched Telephone Network* (PSTN), atau bisa juga terhubung sistem telepon yang lain. Koneksi tersebut bias berupa *single trunk*, *multiple analog trunk* atau koneksi digital dengan kapasitas yang besar.

- **Fitur Telco**

*Asterisk* mendukung semua fitur standar yang terdapat pada perangkat telekomunikasi. *Asterisk* mendukung pengiriman dan penerimaan menggunakan *caller ID*, dan dukungan terhadap routing jaringan berdasarkan *caller ID*. Selain itu *Asterisk* juga mempunyai fitur-fitur standar, seperti transfer panggilan, *call forwarding*, dan berbagai fitur lainnya.

- **Advance Call Distribution**

*Asterisk* dapat melakukan fungsi standar telekomunikasi, yaitu menerima panggilan, memeriksa atribut dari panggilan, dan menentukan routing panggilan berdasarkan atribut panggilan. Setelah route panggilan ditentukan, kita dapat mengirimkan data *single extension*, *grup extension*, rekaman suara, atau sekelompok *host* yang dapat melakukan *roaming* dari satu telepon ke telepon yang lain. Pada *asterisk* bisa juga digunakan fitur antrian telepon untuk membuat operaional lebih efektif. Fleksibilitas inilah yang memungkinkan kita untuk mewujudkan solusi yang efektif lewat telepon. Salah satu yang memembedakan antara *asterisk* dengan PBX konvensional yang mempunyai dukungan terhadap ACD adalah tidak diperlukannya lisensi tambahan, batasan hanya ditentukan oleh perangkat yang kita gunakan.

- **Call Recording**

*Asterisk* memberikan keleluasaan bagi kita untuk merekam panggilan yang melewati PBX. Kita dapat menggunakan rekaman untuk material latihan, sebagai contoh saat panggilan sedang bagus atau dalam kondisi yang buruk. Rekaman panggilan dapat juga digunakan sebagai bukti kualitas layanan kepada pelanggan.

- **Sistem IVR pada Asterisk**

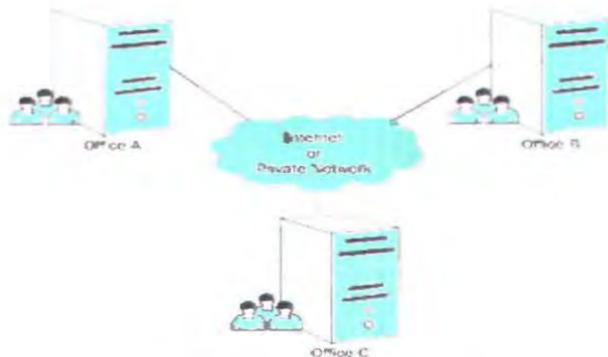
IVR (*interactive voice response*), adalah salah satu metode pada sistem telekomunikasi yang mampu merespon input dari pelanggan secara otomatis dengan menggunakan suara operator yang telah direkam sebelumnya. Dengan IVR ini kita bisa menggantikan peran operator dalam memproses panggilan dari pengguna. Kita bisa menggunakan *asterisk* dalam 24 jam tanpa harus tergantung terhadap tenaga manusia. Dalam fitur ini *asterisk* mempunyai kemampuan untuk memainkan kembali sebuah file (*playback*), membaca file text dan membaca/menerima informasi dari sebuah database. Sistem ini pula yang memungkinkan *asterisk* untuk digunakan dalam sistem telephone banking. Dengan sistem ini kita bisa lebih berhemat dengan menggunakan suatu sistem

- **Voicemail pada Asterisk**

*Asterisk* mempunyai dukungan penuh terhadap sistem voicemail. Sistem voicemail pada *asterisk* mendukung beberapa voicemail dalam *server* yang sama. Dukungan terhadap beberapa zona waktu menambah keunggulan *asterisk* sebagai *softswitch* PBX.

Kuntungan yang lain adalah semakin murah harga perangkat komputer dalam beberapa tahun belakangan ini. Hal ini sesuai dengan semakin banyaknya permintaan pasar terhadap teknologi (terutama bidang teknologi informasi), sementara itu harga perangkat telekomunikasi cenderung stagnan dan tidak bisa dikatakan murah. Keuntungan lain adalah tingkat fleksibilitas yang tinggi pada jaringan VoIP dibandingkan dengan jaringan telepon konvensional. Yang dibutuhkan pada jaringan IP adalah konektivitas dengan *server*, yang tentu saja sudah tercukupi ketika kita membangun sebuah jaringan komputer.

Sistem komunikasi VoIP juga mempunyai tingkat fleksibilitas yang tinggi, bahkan ketika *user* sedang berada di rumah atau berada jauh dari tempat semula, tentu saja selama pengguna mempunyai koneksi broadband dan *server* VoIP mempunyai IP *public* yang bisa diakses melalui jaringan internet. Tidak itu saja, VoIP juga mempunyai keuntungan ketika kita menginginkan pada tiap cabang mempunyai koneksi local (*local exchange*) dan antar tiap cabang terhubung dengan *server asterisk*. Tentu saja hal ini akan sangat menghemat pengeluaran perusahaan dari sisi telekomunikasi, karena suara dilewatkan melalui jaringan internet yang sudah tersedia. Contoh kasus bisa kita lihat pada gambar 2.14.



Gambar 2.14<sup>17</sup> Komunikasi antar cabang dengan VoIP

Pada gambar 2.13 adalah suatu sistem komunikasi VoIP yang digunakan untuk meningkatkan efektifitas pada sebuah perusahaan dengan skala nasional. Perusahaan ini mempunyai beberapa cabang yang tersebar di beberapa lokasi. Penggunaan VoIP dapat digunakan sebagai nilai tambah atau *value added service* dari jaringan IP yang sudah terbangun untuk menghubungkan tiap cabang. Untuk sebuah sistem VoIP yang sederhana, sebuah *server* VoIP bisa dipasang di salah satu cabang dengan mendaftarkan tiap nomor pada cabang sebagai *client* dari *server* tersebut. Untuk pengembangan lebih lanjut pada tiap cabang, baik cabang A,B atau C, jaringan VoIP bisa diintegrasikan dengan jaringan telepon konvensional dengan menggunakan perangkat tambahan seperti PABX, ATA (Analog Telephone Adapter), ataupun ITG (Internet Telephone Gateway).

### 2.7.2 Openser<sup>18</sup> Server

Openser adalah salah satu *software* aplikasi Voip *server* yang dapat dijalankan pada sistem operasi menggunakan Linux. *Server* VoIP ini merupakan SIP router atau SIP proxy yaitu menggunakan protokol SIP, sama seperti pada *Asterisk*. Openser merupakan salah satu *free open-source* VoIP *server* yang berbasis pada protokol SIP, merupakan sebuah signalling atau *application layer control* protokol untuk membuat, mengubah, dan mengirimkan session dengan jumlah user satu

<sup>17</sup> Ibid, hal.10

<sup>18</sup> Goncalves, F. op.cit, hal.31

atau lebih misalkan pada panggilan dengan internet telepon, multimedia conference atau multimedia distribusi.

SER awalnya dikembangkan oleh FhG Fokus dalam sebuah riset di suatu institut di Berlin, Jerman dan dikeluarkan dibawah lisensi GPL. Beberapa nama pengembang SER antara lain Andrei Pelinescu-Onciul, Bogdan-Andrei Iancu, Daniel Constantin Mierla, Jan Janak, dan Jiri Kuthan. Beberapa orang kontibutor yang bergabung dalam proyek ini yaitu dalam penyusunan bagian infrastruktur penyusun SER antara lain Juha Heinamen (RADIUS, ENUM, DOMAIN, URI), Greg Fausak (POSTGRES), Maxim Sobolev (NATHELPER), Adrian Georgescu (MEDIAPROXY), Elena Ramona Modroiu (XLOG, DIAMETER, AVPOPS, SPEEDDIAL), Miklos Tirpak (Permissions), dan masih banyak yang lainnya. Openser masih dalam bagian proyek SER. Pada tahun 2004 FhG Fokus mulai memisahkan proyek pembuatan SER. Pada tahun 2005 varian dari proyek itu terjual kepada TEKELEC. Beberapa anggota pengembang (*core development*) dibagi menjadi 2 bagian yaitu Bogdan-Andrei Iancu dan Daniel Constantin Mierla melanjutkan kembali proyek OpenSER yang dimulai pada tahun itu juga dan pada tahun 2008 OpenSER sudah banyak digunakan.

OpenSER dibangun dengan menggunakan bahasa C untuk Linux atau Unix dengan arsitektur yang memungkinkan performansi tinggi yaitu mampu meng-*handle* sampai 4 juta user pada 1 prosessor *server*. Keunggulan dari OpenSER salah satunya adalah *high scalability* yaitu mampu sampai 1000 panggilan tiap detik dalam transaksi paket *throughput*, fleksibilitas yang tinggi dalam hal *routing*.

### 2.7.3 Axon Server

*Axon* merupakan jenis aplikasi *server* VoIP yang dijalankan pada sistem operasi menggunakan *Windows*. Seperti jenis aplikasi *server* VoIP lainnya, aplikasi *server* VoIP ini juga termasuk *open source*. *Axon* adalah virtual PBX untuk sistem operasi menggunakan *Windows* yang didesain untuk mengakomodir panggilan – panggilan dalam dunia usaha atau call center. *Software* aplikasi ini bekerja sebagai telepon dengan banyak fitur sebagai *switch connecting* ke sambungan telepon dan ekstensinya menggunakan teknologi VoIP.

## 2.8 Peralatan VoIP

Sistem VoIP mempunyai variasi bentuk yang sangat banyak. Seperti komputer saat ini sudah menyediakan VoIP untuk beberapa platform. Untuk platform *windows* sudah menyediakan Microsoft Netmeeting sedangkan untuk Apple menyediakan iChat sebagai layanan

VoIP. Linux mempunyai beberapa aplikasi layanan VoIP untuk dipilih. Pada umumnya VoIP dapat digunakan pada peralatan yang menyediakan kemampuan untuk melakukan panggilan nomor telepon dan berkomunikasi dengan peralatan VoIP lainnya bahkan dengan telepon analog tradisional. Peralatan VoIP tersebut seperti dibawah ini :

- *Softphone*, merupakan *software* yang diinstallkan pada komputer milik *user* yang digunakan untuk menerima dan melakukan panggilan seperti halnya pada telepon biasa. *Softphone* dapat diinstall juga pada ponsel atau PDA yang support wifi/WLAN. Sehingga aplikasi *softphone* dijalankan pada ponsel/PDA lalu akses ke jaringan VoIP via WLAN.
- *IP Phone* memiliki bentuk seperti telepon biasa. Untuk menggunakannya harus terhubung dengan jaringan IP tanpa melalui perangkat lain. Untuk wifi lebih sering dikenal dengan *wifi phone*
- *USB Phone* memiliki bentuk menyerupai telepon seluler. Untuk menggunakannya, *USB Phone* harus dihubungkan ke komputer melalui port USB.
- *Internet Telephony Gateway* (ITG) adalah *user agent* VoIP yang memiliki dua jenis port, yaitu port FXS (terhubung ke telepon biasa) dan FXO (terhubung ke PSTN langsung atau bisa melalui PBX).

----- Halaman ini sengaja dikosongkan -----

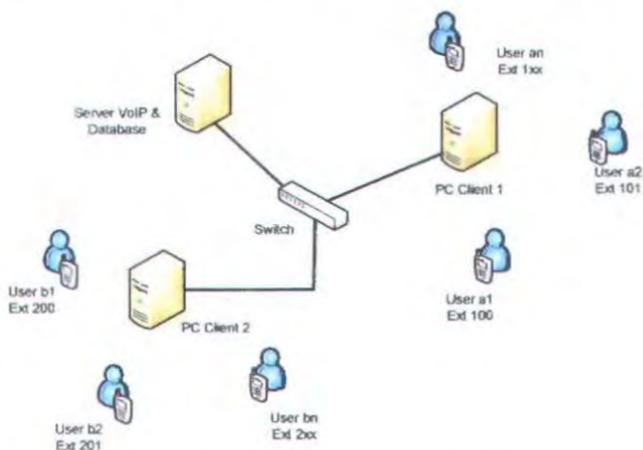
## BAB III

### PERANCANGAN DAN PEMBUATAN SISTEM

Bab ini membahas tentang perencanaan sistem yang dibuat untuk memberikan layanan VoIP pada jaringan LAN, dimulai dengan perencanaan topologi jaringan, persiapan *hardware* dan *software* yang diperlukan, jaringan, instalasi, konfigurasi, serta persiapan pengujian untuk membandingkan kinerja dari beberapa *server* VoIP tersebut.

#### 3.1. Perancangan Topologi Jaringan VoIP

Sistem VoIP yang digunakan pada tugas akhir ini adalah VoIP LAN. Setiap host dalam topologi ini terhubung dengan koneksi standar *fast ethernet*. Topologi jaringan dapat digambarkan sebagai berikut :



Gambar 3.1 Topologi Jaringan VoIP

Untuk *server* VoIP yang akan digunakan antara lain OpenSER, Asterisk dan Axon. Masing – masing *server* VoIP tersebut dijalankan pada sistem operasi Linux, FreeBSD dan Windows. *Server* VoIP OpenSER akan dijalankan pada sistem operasi Linux, *server* VoIP Asterisk akan dijalankan pada sistem operasi FreeBSD dan *Server* VoIP Axon akan dijalankan pada sistem operasi Windows. *Server* VoIP dan *database* menjadi satu bagian dalam 1 PC *server*. Pengukuran dengan mekanisme QoS akan dilakukan satu persatu pada *server* VoIP tersebut dengan jumlah *user* yang melakukan panggilan sebanyak 1 user, 3 user dan 6 user secara bersamaan, masing – masing dengan pengkondisian

jaringan tanpa beban dan dengan pemberian beban yang akan ditentukan besar prosentasenya terhadap total bandwidth yang dimiliki. Untuk masing – masing *client* akan menggunakan sistem operasi Windows.

### 3.2. Penentuan Server dan Client

Sistem yang akan digunakan terdiri dari 3 buah *server* VoIP dan masing - masing akan dibandingkan performansinya dengan mekanisme QoS. Yang pertama adalah *server* VoIP memakai OpenSER yang dijalankan pada sistem operasi Linux Debian Etch (stable), *server* VoIP Asterisk yang dijalankan pada sistem operasi FreeBSD dan *server* VoIP Axon yang dijalankan pada sistem operasi Windows XP. Masing – masing *server* VoIP menggunakan 1 PC dan untuk *database* menjadi satu bagian dengan PC *server* VoIP. Untuk *client* akan memakai sistem operasi Windows XP pada masing-masing *client*. Alasan dipilihnya Windows XP untuk *client*nya adalah alasan kemudahan penggunaan dan kelengkapan fitur *softphone* yang akan dipakai yaitu X-Lite, Mizu Phone dan SIP Communicator. Jumlah *client* menyesuaikan dengan jumlah *client* yang akan melakukan panggilan.

### 3.3. Persiapan Software dan Hardware yang diperlukan

Dalam pembuatan sistem VoIP ini diperlukan beberapa software yang harus diinstall pada PC untuk *server* VoIP dan beberapa PC sebagai *client*. Selain itu terdapat beberapa hardware yang harus disiapkan diantaranya adalah sebagai berikut .:

#### 3.3.1 Persiapan Hardware

Kebutuhan hardware dalam pelaksanaan tugas akhir ini terdiri dari 1 buah PC yang digunakan sebagai *server* VoIP, 1 PC akan dipakai secara bergantian untuk menjalankan *server* VoIP tersebut. Sebagai *client* digunakan 4 buah Laptop dengan sistem operasi Windows XP.

Komputer *server* yang digunakan sebagai VoIP *server* memegang peranan penting dalam menyediakan layanan VoIP untuk pengguna pada jaringan LAN, yaitu menyangkut jumlah kanal yang dapat dilayani untuk layanan VoIP. Komputer yang bertindak sebagai VoIP *server* mempunyai spesifikasi sebagai berikut :

- *Server* Voip 1

Nama Komputer	: <i>Server</i> VoIP1 (OpenSER)
Processor	: Intel Pentium IV 2.40 GHz
Motherboard	: DFI
Memory	: DDR 1 GB
NIC	: RealTek Fast Ethernet

Sound Card : Realtek AC97 Audio  
Hard Disk : Seagate 40 GB  
Swap Area : 512 MB  
Sistem Operasi : Linux Debian Etch 4.0

- *Server Voip 2*

Nama Komputer : *Server VoIP2 (Asterisk)*  
Processor : Intel Pentium IV 2.40 GHz  
Motherboard : DFI  
Memory : DDR 1 GB  
NIC : RealTek Fast Ethernet  
Sound Card : Realtek AC97 Audio  
Hard Disk : Seagate 40 GB  
Swap Area : 512 MB  
Sistem Operasi : FreeBSD

- *Server Voip 3*

Nama Komputer : *Server VoIP3 (Axon)*  
Processor : Intel Pentium IV 2.40 GHz  
Motherboard : DFI  
Memory : DDR 1 GB  
NIC : RealTek Fast Ethernet  
Sound Card : Realtek AC97 Audio  
Hard Disk : Seagate 40 GB  
Swap Area : 512 MB  
Sistem Operasi : Windows XP SP2

Komputer *client* yang digunakan terdiri dari 4 buah Laptop. Semua omputer *client* menggunakan sistem operasi Windows XP SP2. Jumlah panggilan yang akan disediakan tergantung dari jumlah *softphone* yang akan diinstall pada masing – masing komputer *client*. Komputer yang bertindak sebagai komputer *client* mempunyai spesifikasi sebagai berikut :

- *Client PC 1*

Nama Komputer : PC1  
Processor : Intel Pentium IV 2.40 GHz  
Motherboard : DFI  
Memory : DDR 512 MB  
NIC : RealTek Fast Ethernet  
Sound Card : Realtek AC97 Audio  
Hard Disk : Seagate 40 GB  
Swap Area : 512 MB

- *Client PC2*
  - Nama Komputer : PC2
  - Processor : Intel Core 2 Duo 2.0 GHz
  - Motherboard : DFI
  - Memory : DDR 1 GB
  - NIC : RealTek Fast Ethernet
  - Sound Card : Realtek AC97 Audio
  - Hard Disk : Seagate 80 GB
  - Swap Area : 512 MB
  
- *Client PC3*
  - Nama Komputer : PC3
  - Processor : Intel Core Duo 2.0 GHz
  - Motherboard : DFI
  - Memory : DDR 1 GB
  - NIC : Marvel Yukon
  - Sound Card : Realtek Audio
  - Hard Disk : Seagate 80 GB
  - Swap Area : 512 MB
  
- *Client PC4*
  - Nama Komputer : PC4
  - Processor : Intel Pentium IV 2.6 GHz
  - Motherboard : DFI
  - Memory : DDR 512 MB
  - NIC : RealTek Fast Ethernet
  - Sound Card : Realtek Audio
  - Hard Disk : Seagate 40 GB
  - Swap Area : 512 MB

### 3.3.2 Persiapan Software

Software yang digunakan dalam tugas memiliki berbagai macam tugas, mulai dari sebagai penyedia layanan VoIP, sebagai alat ukur kualitas jaringan VoIP dan untuk melakukan analisa terhadap paket yang dikirimkan. Adapun Software yang digunakan pada tugas akhir ini adalah :

- *OpenSER, Asterisk dan Axon*

Merupakan software yang digunakan untuk menyediakan layanan VoIP. OpenSER, Asterisk dan Axon merupakan VoIP *server* yang populer digunakan saat ini, dikarenakan software ini bersifat gratis, *open source*, dan memiliki fitur yang beragam untuk layanan VoIP.

- **Iperf**

*Iperf* adalah software berbasis *console* yang digunakan untuk membangkitkan trafik UDP atau TCP. *iperf* berjalan pada model *client-server*, *server iperf* membangkitkan trafik UDP atau TCP pada sisi *client*.

- **X-lite, Mizu Phone dan SIP Communicator**

Adalah jenis - jenis *softphone* yang dapat digunakan untuk layanan VoIP berbasis SIP Protocol. Kesemua jenis *softphone* tersebut akan berjalan pada sistem operasi Windows dikomputer *client*.

- **VQManager**

Merupakan program monitoring VoIP *call* berbasis web. VQManager juga tersedia dalam dua platform yaitu versi Windows dan versi Linux. Program ini merupakan programs shareware tetapi dari adventnet sendiri menyediakan versi gratis maupun versi trial. Untuk versi gratis dan trial tidak semua fitur disediakan. Program ini menyediakan fitur monitoring panggilan voip, *delay*, *packet loss*, *jitter*, *bandwidth* maupun MOS.

- **Wireshark**

Adalah *tool* yang digunakan untuk mengamati paket data yang dikirimkan pada jaringan. Jika *tool* ini dijalankan, maka *interface* jaringan akan bekerja dalam mode *promiscuous*, dimana *interface* tersebut akan menangkap tiap paket yang masuk ke *interface* tersebut, meskipun paket tersebut tidak ditujukan pada host yang bersangkutan. Pada tugas akhir ini *tool* ini berguna untuk menghitung *Call Setup Delay* serta mengamati bila terjadi kesalahan pada jaringan VoIP.

### 3.4. Instalasi Software dan Hardware

Setelah melalui tahap perancangan dan telah didapatkan semua persiapannya, maka selanjutnya dilakukan instalasi *software* dan *hardware*nya. Untuk instalasi *hardware* tidak terdapat perlakuan khusus, karena sebagian besar perangkat keras yang dibutuhkan sudah terintegrasi didalam komputer. Untuk instalasi jaringan dan pengkabelan disesuaikan dengan kondisi dan tempat. Untuk instalasi perangkat lunak akan dijelaskan sebagai berikut :

#### 3.4.1. Konfigurasi PC Client

*Client* dalam Tugas Akhir ini menggunakan sistem operasi Windows XP untuk memudahkan dalam instalasi aplikasi *softphone*

- **Iperf**

*Iperf* adalah software berbasis *console* yang digunakan untuk membangkitkan trafik UDP atau TCP. *iperf* berjalan pada model *client-server*, *server iperf* membangkitkan trafik UDP atau TCP pada sisi *client*.

- **X-lite, Mizu Phone dan SIP Communicator**

Adalah jenis - jenis *softphone* yang dapat digunakan untuk layanan VoIP berbasis SIP Protocol. Kesemua jenis *softphone* tersebut akan berjalan pada sistem operasi Windows dikomputer *client*.

- **VQManager**

Merupakan program monitoring VoIP *call* berbasis web. VQManager juga tersedia dalam dua platform yaitu versi Windows dan versi Linux. Program ini merupakan programs shareware tetapi dari adventnet sendiri menyediakan versi gratis maupun versi trial. Untuk versi gratis dan trial tidak semua fitur disediakan. Program ini menyediakan fitur monitoring panggilan voip, *delay*, *packet loss*, *jitter*, *bandwidth* maupun MOS.

- **Wireshark**

Adalah *tool* yang digunakan untuk mengamati paket data yang dikirimkan pada jaringan. Jika *tool* ini dijalankan, maka *interface* jaringan akan bekerja dalam mode *promiscuous*, dimana *interface* tersebut akan menangkap tiap paket yang masuk ke *interface* tersebut, meskipun paket tersebut tidak ditujukan pada host yang bersangkutan. Pada tugas akhir ini *tool* ini berguna untuk menghitung *Call Setup Delay* serta mengamati bila terjadi kesalahan pada jaringan VoIP.

### 3.4. Instalasi Software dan Hardware

Setelah melalui tahap perancangan dan telah didapatkan semua persiapannya, maka selanjutnya dilakukan instalasi *software* dan *hardware*nya. Untuk instalasi *hardware* tidak terdapat perlakuan khusus, karena sebagian besar perangkat keras yang dibutuhkan sudah terintegrasi didalam komputer. Untuk instalasi jaringan dan pengkabelan disesuaikan dengan kondisi dan tempat. Untuk instalasi perangkat lunak akan dijelaskan sebagai berikut :

#### 3.4.1. Konfigurasi PC Client

*Client* dalam Tugas Akhir ini menggunakan sistem operasi Windows XP untuk memudahkan dalam instalasi aplikasi *softphone*

yang digunakan antara lain seperti X-Lite, Mizu Phone dan SIP Communicator yang bersifat *freeware*. Dalam Tugas Akhir ini menggunakan *softphone* yang berbasis pada protokol SIP.

### 3.4.1.1. Instalasi dan Konfigurasi Softphone

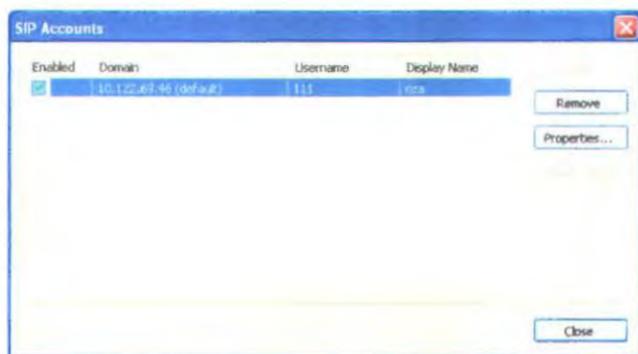
*Softphone* yang digunakan antara lain X-Lite3.0, MizuPhone dan SIP Communicator 1.0 Alpha3. Instalasi *softphone* tersebut sangat mudah, seperti dengan instalasi *software - software* di Windows. Proses instalasi dapat dilakukan dengan mengikuti *wizard*. Setelah ter-install *softphone* tersebut tetapi masih belum dapat digunakan, untuk itu perlu dilakukan beberapa konfigurasi Pada gambar 3.2 adalah tampilan utama dari *softphone* MizuPhone, X-Lite dan SIP Communicator.



Gambar 3.2. Tampilan softphone

Gambar 3.2 adalah tampilan utama dari *softphone* yang digunakan untuk melakukan panggilan dalam suatu sistem komunikasi VoIP. Untuk konfigurasi dari *softphone* tersebut akan dijelaskan sebagai berikut

Untuk *softphone* X-Lite, klik kanan pada window X-Lite, pada menu yang muncul, pilih "SIP Account Settings", maka akan muncul window "SIP Accounts" seperti pada gambar 3.3.



Gambar 3.3. SIP Account X-Lite

Klik "Add" untuk menambah SIP account, dan pada window berikutnya yang muncul dan perlu di-setting adalah :



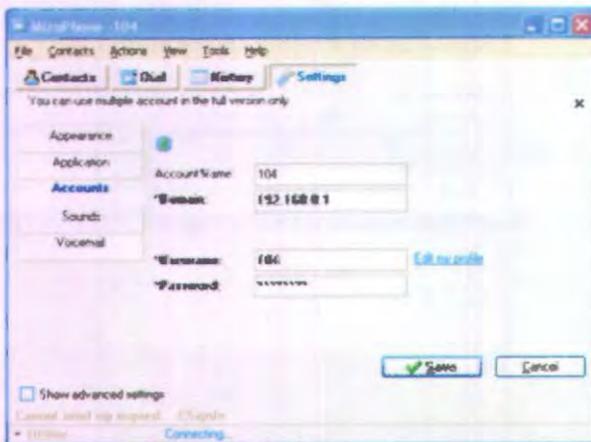
Gambar 3.4. Pembuatan Account pada X-lite

Pada menu *account* SIP setting terdapat beberapa pilihan yang harus diisi agar X-lite mampu berkomunikasi dengan *softphone* SIP lainnya. Parameter yang harus diisi antara lain seperti di bawah ini :

- **Display Name**  
Nama yang akan muncul pada window X-Lite.
- **User Name**  
Username dari SIP account pada server

- **Password**  
Merupakan kata kunci yang digunakan oleh user untuk melakukan registrasi dengan *server* VoIP. Kata kunci ini harus terdaftar pada konfigurasi SIP pada *asterisk* .
- **Authorization username**  
Merupakan nama otoritas dari user. Field ini diisi sama seperti *username*
- **Domain**  
Merupakan domain dari SIP Proxy sever VoIP. Field ini bisa diisi langsung dengan nama domain ataupun dengan nomor IP dari *server* VoIP.

Sedangkan untuk *softphone* Mizu Phone, klik menu “File” disebelah kiri atas lalu pilih meni “Basic Setting” maka akan muncul window seperti pada gambar dibawah ini dan ada beberapa parameter yang harus diisi yaitu *account name*, *domain*, *username* dan *password*.



Gambar 3.5. Pembuatan Account pada Mizu Phone

Setelah pembuatan SIP account selesai, misalkan pada *softphone* X-Lite maka otomatis akan me-register account tersebut ke *server* SIP. Bila berhasil, maka akan muncul pesan : "Ready, Your username isi: <username>" pada jendela X-Lite. Sedangkan bila gagal, maka pesan berganti dengan : "Registration error : 403 – Forbidden" atau muncul pesan Time Out bila user tidak bisa teregister pada *server* VoIP dalam waktu tertentu. Setelah semua *client* ter-register pada server, maka mereka dapat melakukan panggilan satu dengan lainnya dengan

memanggil nomor ekstensi sesuai dengan yang ada pada file `extensions.conf`

### 3.4.2. Konfigurasi server VoIP

Untuk menyediakan layanan VoIP pada jaringan, digunakan Openser, Asterisk dan Axon sebagai VoIP Server. Openser, Asterisk dan Axon merupakan software IP PBX *open source* dengan performa yang baik dan populer digunakan saat ini. Masing – masing *server* VoIP tersebut tersedia untuk dijalankan dalam versi Windows, FreeBSD atau Linux.

#### 3.4.2.1. Instalasi Asterisk

Untuk membangun sebuah *server* VoIP dengan menggunakan *Asterisk*, hal pertama yang harus dilakukan adalah dengan melakukan instalasi. Metode instalasi *Asterisk* sendiri terdapat beberapa cara. Cara yang pertama adalah melakukan instalasi langsung dari source. Cara ini digunakan ketika membutuhkan atau menghilangkan fitur-fitur tertentu pada paket *Asterisk*. Pada tugas akhir ini digunakan metode yang kedua, yaitu instalasi paket *Asterisk* melalui repository, di mana repository yang digunakan adalah menggunakan ITS-NET ([mirror.its.ac.id](http://mirror.its.ac.id)). Setelah memastikan koneksi dengan mirror telah terbangun, bisa menggunakan perintah :

```
apt-get install asterisk
```

Jika tidak terdapat pesan error, maka proses instalasi akan berjalan dengan baik. Setelah paket *Asterisk* berhasil, diperlukan beberapa paket lain yang diperlukan untuk menunjang paket *Asterisk* yang telah berhasil terinstall. Beberapa paket yang dibutuhkan untuk mendukung kerja dari *Asterisk* antara lain paket *gcc compiler*, *bison*, *gcc-c++*, *kernel*, *openssl*, *make* dan untuk keperluan paket databasenya yaitu *unixODBC*, *libtool*, dan *mysql*. Berikut ini cara untuk instalasi paket – paket tersebut seperti dibawah ini :

```
apt-get install gcc-c++
apt-get install bison
apt-get install make
apt-get install unixODBC
```

seperti yang telah dicontohkan diatas cara instalasi dari paket – paket yang dibutuhkan untuk mendukung kerja dari *Asterisk*.

### 3.4.2.2. Konfigurasi Asterisk pada server

*Asterisk* agar dapat berjalan dengan baik sebagai *server* memerlukan beberapa konfigurasi. Tidak banyak yang harus dilakukan untuk melakukan konfigurasi *Asterisk* dengan tujuan untuk meng-authentifikasi user, mengkonfigurasi dial plan dan mengenalkan ENUM, Seluruh proses konfigurasi merupakan proses editing file-file yang ada pada folder */etc/asterisk*. File yang perlu diperhatikan untuk melakukan hal ini adalah:

- sip.conf (untuk melakukan autentifikasi user dengan nomor telepon dan password)
- extensions.conf (untuk mengatur dial plan)
- enum.conf (untuk memperkenalkan nomor +62XX)

Konfigurasi tersebut dilakukan untuk instalasi minimal *Asterisk*. Jika ingin menggunakan *Asterisk* lebih jauh lagi masih terdapat file-file lain yang perlu dikonfigurasi pada folder */etc/asterisk*. Fitur-fitur lain di dalam *Asterisk* juga bisa dimanfaatkan dengan mengubah beberapa konfigurasi pada *Asterisk* seperti voice mailbox maupun pesan interactive dari server.

### 3.4.2.3. Konfigurasi SIP Asterisk

Pada Tugas akhir ini protocol SIP digunakan untuk melakukan inisialisasi dan terminasi komunikasi VoIP. Konfigurasi protocol SIP bisa dilakukan pada file sip.conf. Pada file ini mengatur masalah registrasi, autentifikasi dan codec pada sistem komunikasi VoIP. Berikut konfigurasi sederhananya (tanda titik koma menunjukkan baris komentar):

```
port=5060
;<<port UDP standar SIP yaitu 5060>>
bindaddr=0.0.0.0
;<<proses berjalan pada alamat IP 0.0.0.0 (semua alamat
yang dimiliki server)>>
context=panggilan
;<<nama konteks jaringan yang dipakai>>
;---- eda 301-----;
[301]
type = friend
context = internal
username = 301
secret = password
host = dynamic
dtmfmode = rfc2833
;----- dae 302-----
[302]
type = friend
context = internal
```

```
username 302
secret = password
host = dynamic
dtmfmode = rfc2833
```

#### 3.4.2.4. Konfigurasi Extension Asterisk

Konfigurasi `extension.conf` merupakan konfigurasi untuk mengatur alur panggilan atau biasa disebut *dial plan* pada perangkat sentral telepon. Blok *text* yang digunakan untuk perintah konfigurasi `extension.conf` adalah :

```
exten =>_30XX,1,Dial(SIP/${EXTEN},30,rt)
exten =>_30XX,2,HangUp
misal
exten =>3000,1,Dial(SIP/3000,30,rt)
exten =>3000,2,Hang Up
```

Cara membaca perintah di atas adalah :

Pada kondisi pertama, jika ada orang yang menelepon ke ekstensi 30XX, langkah pertama yang harus dilakukan adalah memanggil (*dial*) Extensi tersebut menggunakan SIP, lalu tunggu sampai 20 detik. Setelah 20 detik, jika tidak diangkat atau dijawab, panggilan dianggap *time out* (rt). Pada kondisi kedua, *asterisk* harus melakukan pemutusan hubungan (*HangUp*). Pada setiap nomor ekstensi yang telah diregistrasikan pada `sip.conf`, maka perlu dikonfigurasi pula *dial plan* – nya pada `extension.conf` dengan perintah seperti di atas

#### 3.4.2.5. Instalasi Database pada Asterisk

Yang perlu dilakukan pertama kali adalah meng-install SQL *database server* lalu stsr *database* yang membutuhkan beberapa detik untuk inialisasi

```
yum install -y postgresql-server
service postgresql start
```

Selanjutnya untuk membuat user panggilan *Asterisk* harus terhubung dan masuk pada *database*, seperti dibawah ini *command* yang harus dijalankan

```
# su - postgres
$ createuser -P
Enter name of user to add: asterisk
Enter password for new user:
Enter it again:
Shall the new role be a superuser? (y/n) n
Shall the new user be allowed to create databases? (y/n) y
```

```
Shall the new user be allowed to create more new users?  
(y/n) n  
CREATE USER
```

### 3.4.2.6. Menjalankan Asterisk

*Asterisk* bekerja sebagai *daemon*. Untuk menjalankan *daemon asterisk*, kita harus menjalankan *script asterisk* :

```
# /etc/init.d/asterisk start
```

Pada *Asterisk*, terdapat terminal yang bisa digunakan untuk melakukan administrasi pada *Asterisk*. Terminal ini bisa digunakan untuk mengaktifkan modul, ataupun memeriksa *log* pada *Asterisk* dengan mengaktifkan fitur *debug*. Untuk dapat mengakses *Asterisk* terminal, bisa dilakukan dengan perintah

```
# asterisk -r
```

Dengan terminal ini proses administrasi bisa dilakukan secara detail, baik melakukan administrasi sistem, pemeriksaan *log* dan menjalankan atau mematikan sebuah modul.

### 3.4.2.7. Instalasi OpenSER

Untuk instalasi *server* VoIP menggunakan OpenSER ada beberapa paket yang dibutuhkan seperti halnya dengan instalasi pada *Asterisk*. Untuk OpenSER yang akan dijalankan pada sistem operasi Linux Debian, setelah menginstall *debian* maka ada beberapa paket yang dibutuhkan untuk instalasi OpenSER yaitu :

```
apt-get install gcc  
apt-get install bison  
apt-get install flex  
apt-get install make  
apt-get install openssl  
apt-get install libmysqlclient-dev  
apt-get install libradiusclient-ng2  
apt-get install libradiusclient-ng-dev  
apt-get install mysql-server
```

Kemudian dekompres untuk instalasi OpenSER menggunakan *command* berikut ini :

```
cd /usr/src  
tar -zxvf openser-1.2.2-tls_src.tar.gz
```

Melakukan edit Makefile pada Linux editor dengan melakukan beberapa perintah seperti dibawah ini :

```
cd /usr/src/openser-1.2.2-tls/  
vi Makefile
```

File before making changes:

```
exclude_modules?= jabber cpl-c mysql pa postgres osp unixodbc \  
    avp_radius auth_radius \  
    group_radius uri_radius xmpp \  
    presence pua pua_mi pua_usrloc \  
    mi_xmlrpc perl snmpstats
```

File after making changes:

```
exclude_modules?= jabber cpl-c pa postgres osp unixodbc \  
    xmpp \  
    presence pua pua_mi pua_usrloc \  
    mi_xmlrpc perl snmpstats
```

Untuk meng-*compile* dan meng-*install core* dan module dilakukan dengan *command* berikut :

```
cd openser-1.2.2-tls  
make prefix=/ all  
make prefix=/ install
```

Untuk melakukan *starting*, *stopping* dan *restarting* OpenSER menggunakan initskrips sebagai berikut :

```
/etc/init.d/openser start|stop|restart
```

### 3.4.2.8. Instalasi Database pada OpenSER

Dalam membangun *database* pada OpenSER ada beberapa hal yang harus diperhatikan yaitu mengkonfigurasi *database* MySQL terlebih dahulu dan memastikan bahwa dalam OpenSER sudah terinstall MySQL kemudian meng-*compile* module tersebut. Untuk memastikan sudah terdapat modul *mysql.so* pada *directory*, dapat dilakukan dengan *command* berikut ini :

```
ls /lib/openser/modules/mysql.so
```

Untuk membuat tabel MySQL dilakukan dengan menggunakan *openser\_mysql.sh*. Beberapa script akan dibuat pada tabel MySQL dengan beberapa ketentuan parameter sebagai berikut :

```
DBNAME="openser"  
DBHOST="localhost"  
DBRWUSER="openser"
```

```

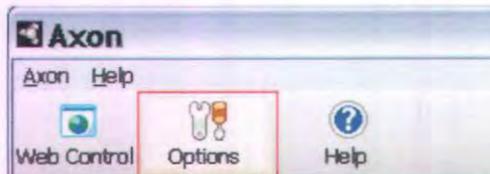
DBRWPW="openserw"
DBROUSER="openserro"
DBROPW="openserro"
DBROOTUSER="root"
cd/sbin
./openser_mysql.sh create
MySQL password for root:
Enter password:
Enter password:
creating database openser ...
Core OpenSER tables succesfully created.
Install presence related tables ?(y/n):y
creating presence tables into openser ...
Presence tables succesfully created.
Install extratables - imc,cpl,siptrace,domainpolicy
?(y/n):y
creating extra tables into openser ...
Extra tables succesfully created.
Install SERWEB related tables ?(y/n):n
Domain (realm) for the default user 'admin':
voffice.com.br

```

### 3.4.2.9. Instalasi Axon

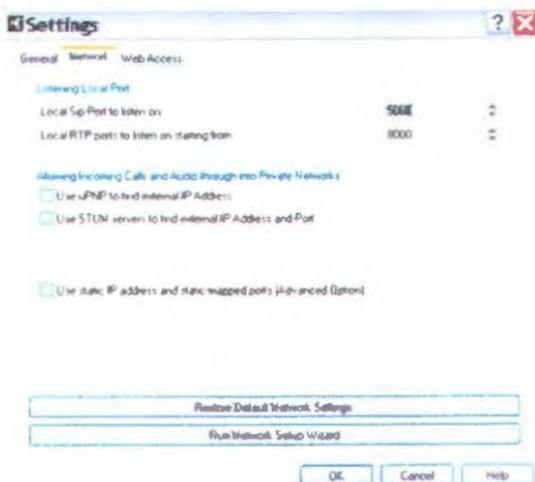
*Axon* merupakan jenis aplikasi *server* VoIP yang dijalankan pada sistem operasi menggunakan *Windows*. Seperti jenis aplikasi *server* VoIP lainnya, aplikasi *server* VoIP ini juga termasuk *open source*. Instalasi *Axon* lebih mudah karena dijalankan pada sistem operasi *Windows* yang biasa dikenal sehari – hari. Berikut ini beberapa langkah instalasi *server* VoIP *Axon* pada sistem operasi *Windows* :

- Download dan install *software* *Axon* virtual PBX pada sistem operasi *Windows* dengan menjalankan tipe file aplikasinya.
- Untuk melakukan konfigurasi setting jaringannya maka jalankan *Axon* PBX pada PC dan tekan tombol “Options”



Gambar 3.6. Masuk Network Setting

Kemudian setelah terbuka setting windows, maka dilakukan beberapa konfigurasi seperti yang ditunjukkan pada gambar 3.7



Gambar 3.7. Konfigurasi Network Setting

- Setting untuk ekstensi ada beberapa hal yang perlu dilakukan yaitu mengisi ekstensi ID (*username*), *display name*, *password*. Berikut ini ilustrasi gambar untuk konfigurasi ekstensinya.



Gambar 3.8. Konfigurasi Extension

### 3.4.2.10. Instalasi VQManager

VQManager adalah salah satu software yang dapat digunakan untuk mengukur kualitas suara. Beberapa parameter yang bisa diukur dengan menggunakan perangkat ini adalah *jitter*, *packet loss*, *delay* dan *MOS (Mean Opinion Score)*. Perangkat lunak ini bersifat *web-based*, dengan hanya dibutukannya web browser pada *client* untuk melakukan pengukuran. Proses instalasi VQManager adalah sebagai berikut :

- Ubah permission dari file `Manae_Engine_VQManager.bin` menjadi executable dengan perintah :

```
# chmod +x Manage_Engine_VQManager.bin
```

- Jalankan file tersebut dengan perintah

```
# ./Manage_Engine_VQManager.bin
```

- Setelah perintah tersebut dijalankan, maka akan terdapat panduan untuk melakukan instalasi. Pertama akan terdapat pilihan untuk instalasi paket VQManager. Selanjutnya akan terdapat pilihan port dan proses menjalankan VQManager.
- Setelah proses instalasi selesai, untuk menjalankan VQManager dapat dilakukan dengan menggunakan perintah :

```
# /etc/init.d/VQManager-service start
```

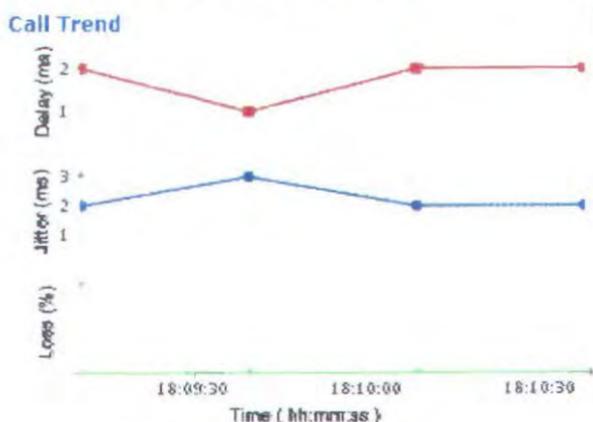
Untuk melakukan pemeriksaan bahwa VQManager telah berhasil terinstal dan dapat dijalankan, bisa dilakukan perintah sebagai berikut :

```
# ps -aux | grep VQmanager
```

Jika terdapat proses VQManager dalam daftar aplikasi yang sedang berjalan, maka VQmanager sudah terinstall dan berjalan.

- Untuk menjalankan VQManager dari *client*, cukup masukkan *IP address / hostname* dari *server* dengan menggunakan port 8647. Contoh <http://localhost:8647>.
- Default login dari VQManager adalah admin dengan password admin. Untuk selanjutnya *password* dari admin harus diganti untuk menghindari adanya pembajakan dari *server*.
- Untuk mengetahui apakah VQManager sudah berjalan, bisa dilakukan pemeriksaan pada proses yang sedang berjalan.
- Pilih menu *sniffer* dan masukkan perangkat keras yang digunakan. Pastikan bahwa *interface* jaringan yang digunakan adalah *interface* yang akan dilewati paket RTP.

- Jika semua berjalan dengan baik, maka *interface* jaringan akan mampu menangkap paket RTP yang dikirimkan kepada server, untuk selanjutnya dihitung nilai QoS dari VoIP oleh VQManager.
- Dari monitor yang terdapat VQManager terdapat informasi tentang paket yang masuk ke jaringan.



Gambar 3.9. Tampilan VQ Manager

Hasil pengukuran dengan VQManager seperti yang tersaji pada gambar 3.9 menunjukkan nilai *delay* dan *jitter* dalam satuan *milliseconds(ms)*, serta *packet loss* dalam satuan persen. Pada gambar diatas waktu pengambilan data ditunjukkan pada nilai yang tertera pada sumbu x. Dari contoh gambar 3.9 waktu pengambilan data adalah pukul 18:09:30 dan data diambil tiap 30 detik selama pengukuran berlangsung.

### 3.4.2.11. Instalasi Wireshark

Wireshark adalah program yang memiliki fungsi sama dengan program *Tshark*, yaitu program yang berfungsi untuk menangkap paket yang masuk ke *interface* jaringan.

Untuk melakukan instalasi paket Wireshark, cukup menjalankan program aplikasinya karena sudah ada tipe Wireshark portable yang lebih memudahkan dalam penggunaannya.

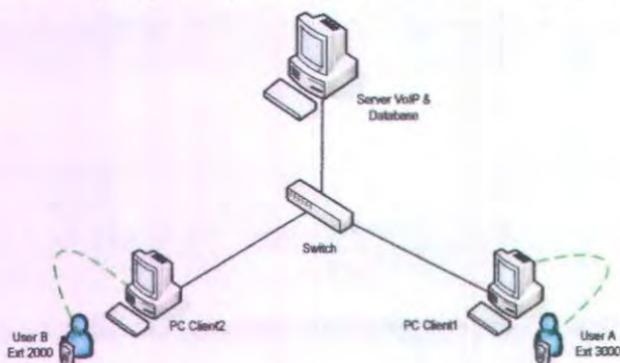
### 3.5. Skenario Pengukuran

Untuk melihat performansi masing - masing VoIP *server*, dilakukan beberapa skenario pengukuran dengan beberapa kondisi, yaitu kondisi tanpa beban, kondisi dengan beban sebesar 50% dari total

kapasitas bandwidth yang dimiliki dan kondisi beban 80% dari total bandwidth yang dimiliki oleh jaringan. Pada tiap kondisi pembebanan, akan dilakukan pembebanan dengan pembebanan paket UDP untuk tip – tiap panggilan yaitu 1 user, 3 user dan 6 user.

### 3.5.1. Kondisi Tanpa Beban

Skenario pengukuran ini dilakukan secara bergantian untuk masing – masing *server* VoIP yaitu dengan melakukan sambungan VoIP dari *client* 1 kepada *client* 2 yang identitas kedua *client* tersebut terdaftar pada *server* VoIP. Setelah pengukuran 1 user panggilan selesai maka dilanjutkan untuk pengukuran 3 user panggilan dan 6 user panggilan. Ilustrasi kondisi jaringan tanpa beban seperti disajikan pada gambar 3.10



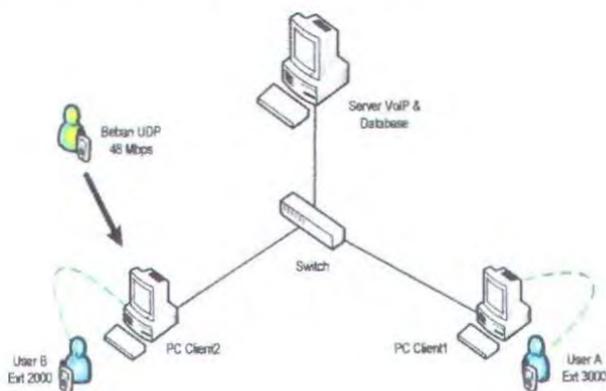
Gambar 3.10. Skenario Pengukuran Tanpa Beban

Untuk mendapatkan hasil parameter yang diinginkan, pada salah satu *client* dijalankan aplikasi VQManager untuk melakukan proses sniffing dari paket RTP yang berasal dan menuju *server*. Selanjutnya dari paket RTP yang berhasil ditangkap, VQManager akan melakukan penghitungan nilai kualitas suara yang dihasilkan, yaitu *delay*, *jitter* dan *packet loss*. Untuk mengukur seberapa banyak utilitas yang diperlukan oleh *server* untuk menangani proses swithing, maka dijalankan aplikasi TOP. Proses pengukuran sendiri dilakukan dengan melakukan *remote access* dengan menggunakan protokol SSH pada masing-masing *server* yang saling berhubungan. Aplikasi TOP akan dijalankan selama proses komunikasi untuk mendapatkan nilai CPU *usage* pada masing-masing *server* yang melakukan proses komunikasi VoIP. Pengukuran call setup delay menggunakan suatu aplikasi *sniffer* yang berguna untuk menangkap paket yang lewat dijalankan. Aplikasi

yang digunakan adalah aplikasi *wireshark* yang bekerja dengan mode teks dan tanpa menggunakan *desktop environment* dan dijalankan pada tiap *server* untuk mendapatkan waktu dimulainya panggilan sampai dering telepon berbunyi di sisi penerima.

### 3.5.2. Pembebanan 50%

Pada skenario ini dilakukan pembebanan pada jaringan VoIP dengan beban UDP. Pembebanan dilakukan dengan menjalan *iperf* yang akan membangkitkan trafik yang ditujukan pada jalur trafik agar mendapat beban sebesar 50% dari total bandwidth (100 Mbps). Proses pembebanan adalah ketika *server* sedang melakukan switching panggilan antara 2 *client*, beserta itu pula dikirimkan paket UDP sebesar 50% yaitu 50 MB ke kanal trafik. Proses pembebanan dilakukan selama proses komunikasi berlangsung, ilustrasi pengukuran dengan menggunakan seperti tersaji pada gambar 3.11



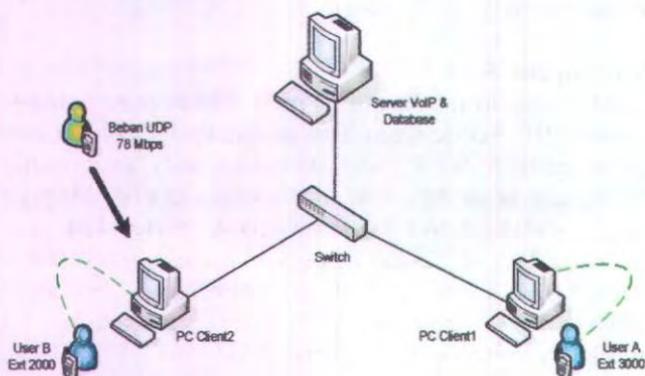
Gambar 3.11. Skenario Pengukuran Beban 50%

Aplikasi yang dijalankan pada skenario ini sama dengan skenario tanpa beban. Parameter yang didapatkan pada pengukuran ini adalah : *jitter*, *Delay*, *Packet Loss*, *CPU Usage* dan *Call Setup Delay*.

### 3.5.3. Pembebanan 80%

Proses pengukuran pada skenario ini sama dengan skenario dengan pembebanan sebelumnya. Pada jaringan VoIP akan dibangkitkan trafik UDP dalam skala tinggi, mengingat skenario ini mewakili kondisi jaringan pada saat trafik dalam kondisi beban tinggi. Proses pengukuran dengan beban ini diawali dengan dikirimkannya paket UDP dari *iperf* sebesar 80 MB. Dengan pembebanan pada saat sedang melakukan sesi

komunikasi, diharapkan pengkondisian jaringan dapat diterapkan, yaitu kondisi jaringan pada saat terdapat trafik yang masuk ke jaringan sesuai dengan protokol yang digunakan.



**Gambar 3.12.** Skenario Pengukuran Beban 50%

Aplikasi yang dijalankan pada skenario ini sama dengan skenario pembebanan sebelumnya. Parameter yang didapatkan pada pengukuran ini adalah : *jitter*, *Delay*, *Packet Loss*, dengan menggunakan VQManager, aplikasi TOP untuk pengukuran *CPU usage* dan aplikasi Wireshark untuk mendapatkan nilai *Call Setup Delay*.

## BAB IV

### ANALISA DATA DAN PEMBAHASAN

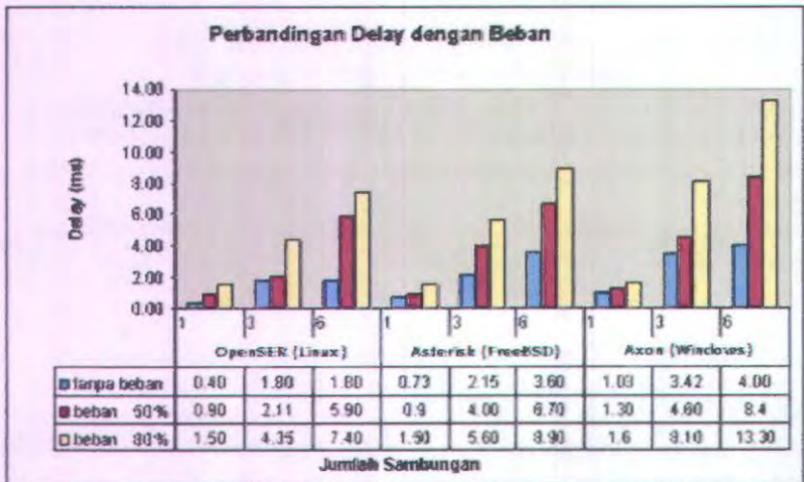
Bab ini membahas tentang analisa pengujian kinerja VoIP *server*. Pengujian dilakukan dengan membandingkan kinerja *server* VoIP pada sistem, terutama pada saat *server* dalam kondisi tanpa beban maupun pada saat jaringan diberi beban yaitu beban berupa paket UDP. Parameter kualitas *server* terhadap jaringan VoIP yang diukur meliputi *delay*, *jitter*, *packet loss* dan *call setup delay*. Selain itu parameter yang diukur untuk mengetahui kinerja *server* berdasarkan spesifikasi *hardware* yang dimiliki adalah *CPU usage*.

#### 4.1. Pengukuran *Delay*

*Delay* yang diukur pada pengukuran ini merupakan selisih waktu saat paket mulai dikirimkan *client* hingga diterima oleh *client* yang dituju atau dengan kata lain adalah waktu yang dibutuhkan oleh paket untuk dikirimkan dan diterima penuh oleh penerima, tentu saja dalam hal ini paket suara telah melalui proses switching pada *server* VoIP. Dan *delay* maksimum yang direkomendasikan oleh ITU-T Y.1541 untuk aplikasi suara adalah 100 ms, yang masih dapat diterima pengguna adalah 250 ms. Pengukuran *delay* ini dilakukan dalam kondisi tanpa beban, dan juga pembebanan dengan paket UDP, dengan beberapa kondisi pada jaringan, tanpa beban, beban menengah dan beban tinggi. Pengukuran *delay* juga dilakukan pada saat jaringan terdapat 1 sambungan, 3 dan 6 sambungan komunikasi VoIP. Untuk menghitung *delay*, dilakukan beberapa skenario, yaitu panggilan dari *user* (ext : 101) ke *user* yang terdaftar di *server* 1 (ext 201), kemudian data suara tersebut dicapture oleh software *VQManager*. Dilakukan *testcall* dari *client* 1 ke *client* 2 menggunakan beberapa perangkat lunak VoIP *client* atau lazim disebut *softphone*. *Testcall* dilakukan selama kurang lebih 90-180 detik.

Skenario awal yang dijalankan adalah melakukan pengukuran *delay* pada saat jaringan berada dalam kondisi tanpa beban, Pengukuran *delay* dengan pembebanan paket UDP dilakukan dalam skenario pengukuran yang sama seperti pengukuran *delay* tanpa beban. Pembebanan dilakukan dengan variabel nilai sebesar 50% dan 80% dari total bandwidth kapasitas. Hal ini dimaksudkan agar didapatkan hasil perbandingan antara kondisi tanpa beban dan dengan pembebanan

terhadap *delay*. Dari pengukuran yang dilakukan didapatkan grafik seperti pada gambar 4.1 dibawah ini.



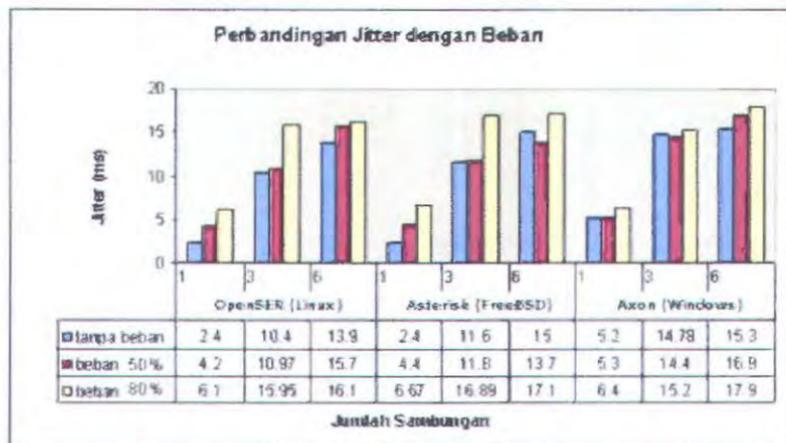
**Gambar 4.1.** Delay dengan Beban

Dari hasil pengukuran yang terlihat pada gambar 4.1, terlihat bahwa *server 1* (OpenSER) mempunyai nilai *delay* pada paket RTP stream yang dikirimkan paling rendah untuk tiap – tiap skenario pengukuran. Dengan pembebanan paket UDP sebesar 80%, nilai *delay*-nya sebesar 7.40 ms. Sedangkan *server 3* (Axon) mempunyai nilai *delay* yang paling tinggi untuk tiap-tiap skenario pengukuran. Nilai *delay* tertinggi pada pengukuran ini adalah sebesar 13.30 ms, yang terjadi pada *server 3* (Axon) dengan pembebanan paket UDP sebesar 80% dari total *bandwidth* yang tersedia pada jaringan. *Delay* yang didapatkan untuk pengukuran dengan pembebanan paket UDP ini juga masih tergolong baik karena dibawah standart ITU-T Y.1541, yaitu sebesar 100 ms. Kenaikan nilai *delay* yang terjadi dikarenakan adanya pembebanan simultan yang dikirimkan ke *host destination* dan bersamaan itu juga sedang dikirimkan paket data suara dari *host source* ke *host destination*. Pembebanan trafik yang besar secara simultan pada kanal akan membuat trafik pada kanal menjadi semakin tinggi, sehingga untuk menghindari terjadinya tabrakan paket data yang dikirim maka pada switch menjalankan fitur *colusion detection* yaitu kondisi sistem antrian. Dengan adanya antrian ini tentunya akan memberikan kontribusi nilai *delay* pada pengiriman paket data.

## 4.2 Pengukuran Jitter

*Jitter* adalah variasi *delay* yang terjadi karena *delay* yang terjadi selama proses koneksi VoIP atau dengan kata lain merupakan variasi *paket delay* yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima, adapun nilai dari *delay* ini tidaklah konstan melainkan bervariasi. *Jitter* yang diukur pada pengukuran ini merupakan variasi *delay* total dari komunikasi VoIP jadi bukan hanya variasi *delay* dari pengiriman paket. Pengukuran *jitter* dilakukan dengan skenario yang sama seperti pada pengukuran *delay*. Untuk mendapatkan *jitter*, pengukuran dilakukan dengan menggunakan *VQ Manager*. *Jitter* yang diukur mempunyai satuan *mili seconds* (ms). Nilai *jitter* yang direkomendasikan oleh ITU-T Y.1541 adalah dibawah 50 ms

Proses pengukuran diawali dengan pengukuran pada kondisi tanpa beban. Kemudian dilakukan pembebanan pada kanal trafik yang diarahkan pada panggilan kepada *client* yang dituju. Pembebanan sendiri akan membuka port 5001 dan protokol yang digunakan adalah UDP. Setelah *client VQmanager* diaktifkan, data *jitter* yang didapatkan dari trafik RTP yang melewati jaringan akan diolah untuk menghasilkan data sebagai berikut :



Gambar 4.2. Jitter dengan Beban

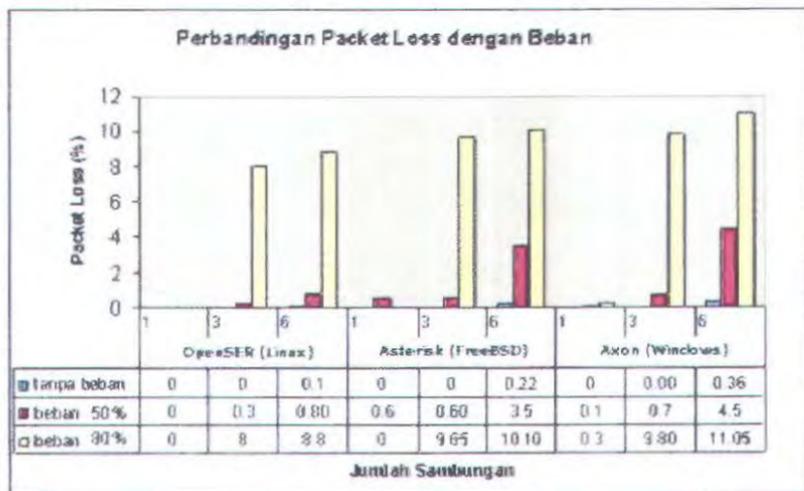
Dari gambar 4.2 menunjukkan bahwa terjadi peningkatan nilai *jitter* yang signifikan ketika jumlah panggilan dinaikkan dari 1 panggilan menjadi 3 panggilan dan 6 panggilan, baik untuk kondisi tanpa beban, beban UDP senilai 50% dari total *bandwidth* yang tersedia ataupun beban UDP 80% dari total *bandwidth* yang tersedia. *Server 1*

(OpenSER) memiliki nilai *jitter* yang paling rendah dibandingkan dengan *server* Asterisk dan Axon. Untuk kondisi pada pembebanan sebesar 80% dari total bandwidth, pada *server* 1 (OpenSER) nilai *jitter*-nya sebesar 16.1 ms, Sedangkan pada *server* Asterisk dan Axon nilai *jitter*-nya sebesar 17.1 ms dan 17.9 ms. Nilai *jitter* yang didapatkan untuk semua skenario pembebanan memenuhi rekomendasi dari ITU-T Y-1541 sebesar 50 ms. Hal ini disebabkan karena adanya *routing* yang terbatas sesuai dengan topologi jaringan yang digunakan. Kenaikan nilai *jitter* yang terjadi berkaitan pula dengan kenaikan *delay* seperti pada pembahasan tentang pengukuran *delay* sebelumnya. Karena terjadi kenaikan *delay* yang disebabkan adanya pembebanan simultan yang juga dikirimkan ke *host destination*, maka nilai *jitter* yang didapatkan juga akan mengalami kenaikan sebab *jitter* merupakan variasi *delay* yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima.

#### 4.2 Pengukuran Packet Loss

Pengiriman paket dari *client* 1 ke *client* 2 tidak semuanya dapat diterima oleh *client* 2. Terkadang dalam beberapa waktu terdapat paket-paket yang hilang pada saat transmisi, paket-paket yang hilang tersebut disebut dengan *packet loss*. Kehilangan paket tersebut biasanya terjadi ketika *peak load* dan *congestion* (kemacetan transmisi paket akibat padatnya traffic yang harus dilayani) dalam batas waktu tertentu. Untuk mengetahui parameter *packet loss*, pengukuran juga dilakukan dengan menggunakan software *VQManager*, pengukuran dilakukan pada saat jaringan dalam kondisi tanpa beban, maupun dalam kondisi dibebani. Skenario pengukuran sama dengan pengukuran *delay* dan *jitter*.

Pembebanan yang diberikan berupa paket UDP pada sistem VoIP *server* bertujuan untuk mengkondisikan jaringan berada pada kondisi terdapat banyak trafik UDP yang masuk pada jaringan. Protokol UDP banyak digunakan untuk aplikasi yang bersifat *real-time*. Salah satu protokol transport yang digunakan pada sistem komunikasi VoIP, yaitu RTP (Real Time Transfer Protokol) juga adalah salah satu dari bagian protokol UDP. Skenario yang dijalankan pada pengukuran ini, diawali dengan pengukuran tanpa beban pada sistem, kemudian dilakukan penambahan jumlah sambungan pada tiap-tiap kondisi pembebanan trafik UDP. Hasil pengujian sistem dengan pembebanan, yaitu *packet loss* yang terjadi pada sistem komunikasi VoIP ditunjukkan pada gambar 4.3 berikut ini.



**Gambar 4.3.** *Packet loss* dengan Beban

Dari gambar 4.3 menunjukkan pengaruh pembebanan terhadap nilai *packet loss* pada sistem komunikasi VoIP. Nilai *packet loss* meningkat ketika dilakukan pembebanan 50% dan penambahan beban menjadi 80%, nilai *packet loss* naik secara signifikan. Hal ini dipengaruhi oleh masuknya trafik UDP dalam jumlah yang besar yang dihasilkan oleh pembebanan, dalam hal ini adalah iperf pada client yang membangkitkan trafik UDP menuju *client destination*. Salah satu alasan banyaknya *packet loss* yang timbul adalah terjadi ketika *peak load* dan *congestion* (kemacetan transmisi paket akibat padatnya traffic yang harus dilayani) dan sifat dari protokol UDP sendiri, yaitu yang lebih mementingkan faktor kecepatan dalam pengiriman data tanpa memprioritaskan *reability* dari paket data yang dikirimkan. Penambahan jumlah sambungan turut mempengaruhi nilai *packet loss* yang dihasilkan. Penyebab dari korelasi penambahan sambungan dengan bertambahnya nilai *packet loss* adalah penggunaan protokol transport dari sistem komunikasi VoIP, yaitu RTP yang merupakan protokol transport yang berbasis UDP untuk pengiriman paket data. Dari hasil beberapa pengukuran sebelumnya menunjukkan adanya korelasi antara *delay*, *jitter* dan *packet loss*. *Delay* yang besar maka *jitter*-nya juga besar dan terjadinya *packet loss* juga besar.

Dari percobaan pengukuran terhadap pengukuran kualitas dari layanan VoIP, didapatkan hasil yang memenuhi standar ITU-T Y.1541, yaitu pada parameter *delay* dan *packet loss*, sedangkan pada pengukuran *packet loss* dengan beban tertinggi tidak memenuhi standar yang

ditetapkan ITU-T Y.1541. Rekapitulasi nilai kualitas jaringan dengan nilai yang dianjurkan oleh ITU-T Y.1541 seperti pada tabel 4.1

**Tabel 4.1** Perbandingan hasil pengukuran dengan rekomendasi ITU-T Y.1541

Parameter	Hasil Pengukuran	Rekomendasi ITU-T Y.1541	Keterangan
Delay	13.30 ms	100 ms	Memenuhi rekomendasi
Jitter	17.90 ms	50 ms	Memenuhi rekomendasi
acket Loss	11.05 %	$1 \times 10^{-3}$	Tidak Memenuhi rekomendasi

Tabel 4.1 berisi perbandingan antara hasil pengukuran dengan nilai yang direkomendasikan ITU-T Y.1541 pada tiga parameter yang diukur yaitu, *delay*, *jitter* dan *packet loss*. Hasil pengukuran adalah nilai yang didapatkan ketika kondisi jaringan berada pada beban paling tinggi serta merupakan nilai tertinggi dari tiap parameter yang digunakan. Dari hasil perbandingan yang dilakukan nilai dari parameter *delay* dan *jitter* memenuhi rekomendasi yang dianjurkan untuk kondisi beban tertinggi, sedangkan untuk parameter *packet loss* nilai yang didapatkan tidak memenuhi rekomendasi. Nilai dari *delay* dan *jitter* yang didapatkan pada pengukuran mempunyai nilai yang jauh lebih baik dari yang dianjurkan oleh rekomendasi ITU-T Y.1541 dan nilai *packet loss* yang didapatkan pada pengukuran mempunyai nilai diatas rekomendasi ITU-T Y.1541. Hal ini disebabkan adanya keterbatasan dari path dan jalur yang dilalui oleh packet serta *class*<sup>19</sup> yang digunakan yaitu *class 0* yang digunakan untuk tipe jaringan yang dibatasi dan dikondisikan dengan suatu pembebanan trafik.

### 4.3 Pengukuran Call Setup Delay

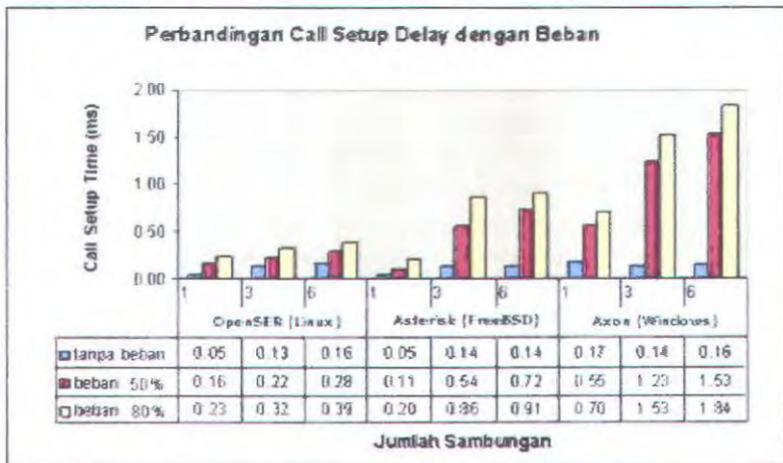
Call setup dalam komunikasi VoIP adalah adalah proses terbentuknya suatu komunikasi antara *server* dan *client*. Proses ini diawali dengan dikirimkannya pesan *INVITE* dari *client* kepada *server* dan dibalasa dengan pesan *ACK* sampai akhirnya terjadi proses "Ringing" pada *client* yang dituju. Pengukuran call setup time pada tugas akhir ini menggunakan tools *wireshark*. Dengan tools ini paket SIP yang digunakan untuk membangun koneksi VoIP dapat ditangkap untuk selanjutnya dihitung waktu terbangunnya koneksi VoIP.

<sup>19</sup> ITU-T recommendation Y-1541, op.cit, hal.8

#### 4.4.1. Pengukuran Call Setup Delay dengan Pembebanan

Pengukuran call setup delay pada VoIP dilakukan dengan skenario yang sama dengan pengukuran sebelumnya. Dalam pengukuran ini dilakukan sampai 6 panggilan yang dilakukan secara berurutan dengan menggunakan tiga buah *softphone* sebagai client. client 1 akan melakukan panggilan terhadap client 2. Berikut ini adalah hasil dari pengukuran call setup delay dengan kondisi tanpa beban, beban UDP sebesar 50% dan 80% dari total bandwidth yang dimiliki oleh jaringan yang digunakan selama pengukuran yaitu *fast ethernet* dengan bandwidth sebesar 100 Mbps.

Pada gambar 4.4 didapatkan hasil pengukuran *call setup delay* yang terjadi pada tiap server VoIP yang sedang melakukan komunikasi antara satu dengan yang lain sampai dengan 6 sambungan pada tiap server.



Gambar 4.4. Call Setup Delay dengan Beban

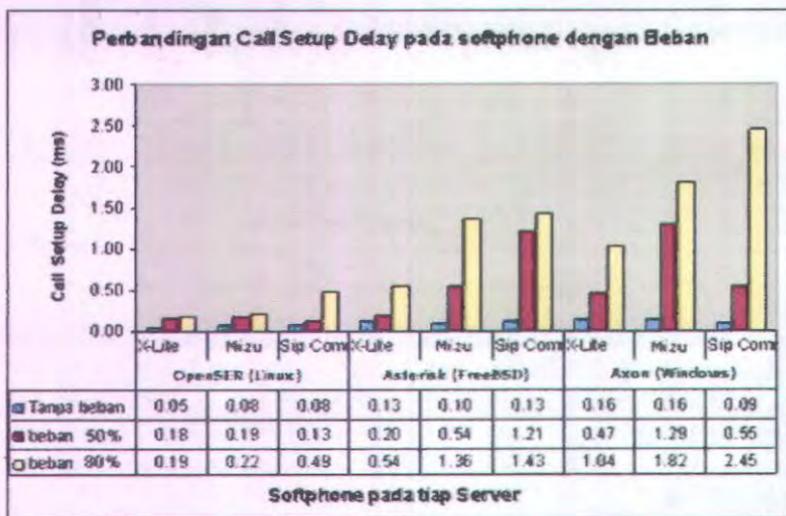
Pada gambar 4.4 adalah merupakan hasil pengukuran call setup delay dengan pembebanan. Dari data diatas dapat terlihat terjadi lonjakan *call setup delay* yang signifikan ketika sistem dibebani dengan beban UDP. Salah satu contoh adalah pada *server 2* (Asterisk) terdapat tiga sambungan yang dilakukan, dalam kondisi tanpa beban *call setup delay* sebesar 0.14 ms. Ketika *server* dibebani dengan beban UDP 50% memakai iperf nilai *call setup delay* naik menjadi sebesar 0.54 ms. Secara keseluruhan untuk *call setup delay* hasil yang lebih baik didapatkan dengan menggunakan *server 1* (OpenSER). Kenaikan nilai

*call setup delay* disebabkan karena pembebanan yang dilakukan. Semakin besar pembebanan yang diberikan maka kondisi trafik akan semakin padat sehingga *call setup*-nya juga akan semakin lama.

Pada pengukuran dengan pembebanan UDP terjadi beberapa kali kegagalan panggilan. Kegagalan ini banyak diakibatkan karena banyaknya traffic UDP yang masuk ke *interface* jaringan dari *server* sehingga menyebabkan terjadinya packet loss yang cukup tinggi. Jika nilai packet loss meningkat, maka peluang untuk berhasilnya suatu sambungan VoIP akan menjadi semakin kecil. Hal ini dibuktikan dengan nilai packet loss yang tinggi ketika terjadi sambungan VoIP serta banyaknya packet data SIP yang hilang dalam proses panggilan.

#### 4.4.2. Pengukuran Call Setup Delay pada tiap *softphone* dengan Pembebanan.

Pengukuran ini dilakukan dengan menjalankan 3 panggilan dengan 3 *softphone* yang berbeda. Masing-masing *softphone* akan digunakan untuk melakukan panggilan VoIP kepada *client* yang terdaftar pada *server*. Pada jaringan akan dibebani trafik UDP yang dibangkitkan dari salah satu *client* dengan menggunakan iperf sebesar 50% dan 80% dari total *bandwidth* yang tersedia. Hasil pengukuran disajikan pada gambar 4.5

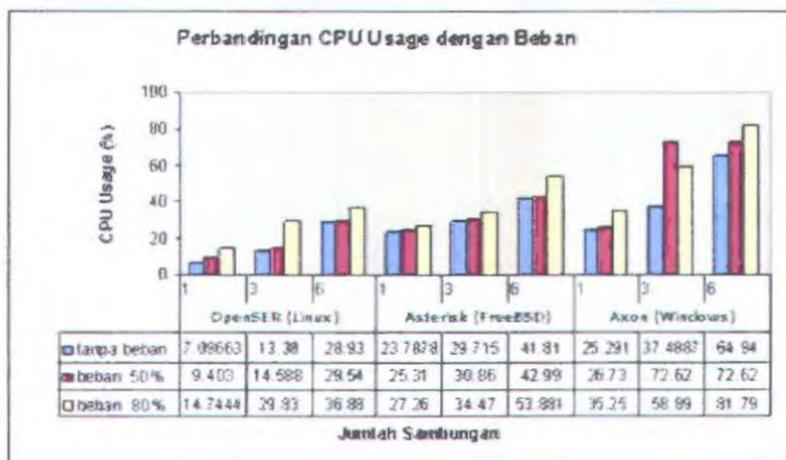


Gambar 4.5. Call Setup Delay Pada Tiap *Softphone* dengan Beban

Hasil pada gambar 4.5 adalah hasil dari selisih antar paket INVITE yang dikirimkan oleh client 1 dimana panggilan berasal sampai dengan pesan RINGING diterima. Hasil pengukuran menunjukkan bahwa pembebanan dengan paket UDP berpengaruh terhadap call setup delay, softphone X-Lite mempunyai nilai call setup delay terpendek. Sedangkan SIP communicator memiliki nilai call setup delay terbesar. Penyebab lain performa SIP communicator yang kurang baik dibandingkan dengan softphone yang lain adalah bahasa pemrograman yang digunakan adalah *java* yang cenderung membutuhkan *resource* yang besar dari *client* yang menjalankan aplikasi tersebut.

#### 4.4 Pengukuran CPU Usage

Skenario pengukuran utilitas dari CPU pada serve VoIP sama dengan pengukuran sebelumnya. Perbedaan yang mendasar adalah proses pengambilan data pada *server* yang sedang melakukan komunikasi. Selama proses komunikasi VoIP berjalan, pada client dilakukan remote access menuju *server* VoIP yang sedang berhubungan untuk menjalankan aplikasi TOP untuk memperoleh nilai CPU usage pada tiap-tiap *server*. Tools TOP, yaitu tools berbasis teks yang digunakan pada sistem operasi dengan platform UNIX. Penambahan jumlah panggilan ditambahkan untuk tiap – tiap pengukuran, seiring dengan ditambahkan beban UDP sebesar 50% hingga 80% dari total bandwidth yang dimiliki. Hasil yang didapatkan dari pengukuran utilitas CPU seperti yang disajikan pada gambar 4.6 dibawah ini.



Gambar 4.6. CPU Usage dengan Beban

Nilai yang ditunjukkan pada pengukuran CPU *usage* dengan pembebanan menunjukkan bahwa *server* 3 (Axon) mempunyai nilai yang terbesar di antara ke dua *server* yang lain. Untuk pembebanan UDP sebesar 80% dari total bandwidth yang dimiliki, yaitu sebesar 100 Mbps selama terjadi panggilan dengan 6 sambungan aktif pada *server*, *server* 3 mengeluarkan 81.79% dari total utilitas CPU yang dimiliki. Hal ini tentu saja mempengaruhi kinerja dari *server* dalam menangani proses switching, karena dengan semakin tinggi nilai dari CPU *usage*, maka kemungkinan terjadi *crash* pada *server* semakin tinggi. Ketika *crash* terjadi, hal yang terjadi adalah kegagalan dalam melayani permintaan dari *client*. Nilai CPU *usage* yang tinggi seiring beban yang diberikan untuk menjalankan komunikasi antar *client*.

## BAB V PENUTUP

### 5.1. Kesimpulan

Dari hasil pengujian dan analisa perbandingan *Quality of Service* dari beberapa platform *server* VoIP dengan melakukan beberapa jenis skenario pengukuran, dapat ditarik kesimpulan sebagai berikut :

1. Secara keseluruhan *server* VoIP OpenSER lebih unggul dibandingkan dengan *server* VoIP Asterisk dan Axon dari segi QoS. Pada bandwidth 100 Mbps (*fast ethernet*) memiliki *call setup delay*, *delay*, *jitter* dan *packet loss* paling rendah dibandingkan dengan *server* VoIP Asterisk dan Axon, baik saat tidak dilakukan pembebanan maupun saat dilakukan pembebanan paket UDP. Keunggulan performansi OpenSER dibandingkan *server* VoIP Asterisk dan Axon terletak pada proses pengiriman paket RTP saat 2 user saling berkomunikasi. Pada OpenSER proses tersebut tidak melalui *server* tetapi langsung antara user 1 dan user 2, *server* hanya berperan ketika proses *signalling* saja. Berbeda dengan *server* Asterisk dan Axon, untuk proses *signalling* dan pengiriman paket RTP harus melalui *server* terlebih dahulu.
2. Secara keseluruhan nilai *packet loss* yang tinggi terjadi ketika dilakukan pembebanan pada jaringan, yaitu 4.5 % pada saat jaringan dibebani dengan beban 50% dari bandwidth yang dimiliki dan nilai *packet loss* 11.05 % ketika jaringan dibebani 80% dari total bandwidth yang dimiliki.
3. Penambahan beban pada jaringan serta penambahan sambungan pada saluran menyebabkan nilai *delay* bertambah. Nilai *delay* yang tinggi ikut mempengaruhi nilai *jitter* yang terjadi pada saat komunikasi berlangsung dan terjadinya *packet loss* juga akan ikut bertambah.
4. Nilai Kualitas Jaringan VoIP yang didapatkan pada percobaan masih memenuhi standar ITU-T Y.1541, yaitu 100 ms untuk *delay* dan 50 ms untuk *jitter*. Hal ini disebabkan oleh adanya jarak yang dekat antar tiap *server* dalam cluster pada waktu dilakukan percobaan serta topologi dari jaringan yang digunakan. Pada tugas akhir ini digunakan Topologi LAN, untuk mengukur performa *server* VoIP.

5. Softphone yang mempunyai performa paling bagus adalah X-lite dengan didapatkannya nilai *call setup delay* terkecil dalam tiap kondisi jaringan. Untuk kondisi beban tertinggi nilai call setup delay bernilai 1.04 ms. Softphone yang memiliki performa paling rendah adalah SIP Communicator dengan nilai call setup delay yang tertinggi yaitu 2.45 ms pada saat jaringan dibebani dengan beban tertinggi.

## 5.2. Saran

Dari hasil pengamatan dan analisa yang telah dilakukan, beberapa saran yang dapat diberikan antara lain :

1. Pada pengukuran QoS untuk tugas akhir ini hanya dilakukan pada jaringan LAN dengan area yang sempit, untuk selanjutnya diharapkan adanya pengembangan dengan area yang lebih luas dan diadakan dengan jumlah *client* yang lebih banyak.
2. Sebagai *platform* yang akan dibandingkan mungkin dapat dibandingkan dengan *platform* server VoIP yang dalam bentuk hardware.

## DAFTAR PUSTAKA

- [1] Dostalek, L. & Kabelova, A. "*Understanding TCP/IP a clear and Comprehensive Guide*", PACKT Publishing. 2006
- [2] Gomillion, D. & Dempster, B. "*Building telephony System with Asterisk*", Packt Publishing. 2005
- [3] Goncalves, F. "*Building Telephony System with OpenSER*", PACKT publishing. April 2008
- [4] ITU-T Recommendation Y.1541. "*Network Performance Objectives for IP-based services*", International Telecommunication Union, 2006
- [5] Jackson, B. & Clark, C. "*Asterisk Hacking*", Syngress Publishing. April 2007
- [6] M. Iskandarsyah H. "*Dasar-dasar Jaringan VoIP*", <http://www.ilmukomputer.com>, 2005
- [7] M Riza Pahlevi, "*Analisa Perbandingan Quality of Service pada server VoIP Berdasarkan Perbedaan Platform*", Bidang Studi Telekomunikasi Multimedia Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember, 2008
- [8] Raharja A., "*Membuat Server VoIP Axon di Windows*", Jakarta Center of ICT Studies
- [9] Singh, K. & Schulzrine, H. "*Failover and load balancing in SIP telephony*", Department of Computer Science, Columbia University . 2005
- [10] Van Meggelen, J. "*Asterisk The Future of Telephony*", Oreilly. Agustus 2007

----- Halaman ini sengaja dikosongkan -----

## LAMPIRAN A

### Proposal Tugas Akhir

Jurusan Teknik Elektro - FTI  
Institut Teknologi Sepuluh Nopember

RE.1599 TUGAS AKHIR - 4 SKS

Nama Mahasiswa : Ade Chandra Dewa  
Nomor Induk : 2205 109 633  
Bidang Studi : Teknik Telekomunikasi Multimedia  
Tugas Diberikan : Semester Ganjil 2008/2009  
Dosen Pembimbing : Ir. Djoko Suprajitno Rahardjo  
Judul Tugas Akhir : Analisis Perbandingan Kualitas Jaringan VoIP pada Sistem Real Time VoIP Service Berdasarkan Perbedaan Platform Sistem Operasi.  
(Comparison Analysis of VoIP Network Quality on VoIP Service Real Time System Based on Operating System Platform Differences).

26 SEP 2008

#### Uraian Tugas Akhir :

Perkembangan teknologi internet dan komunikasi telepon yang sangat pesat memunculkan teknologi yang disebut dengan VoIP yaitu suatu teknologi yang mampu melewatkan data, suara dan video dalam bentuk paket ke dalam jaringan berbasis IP. Jaringan IP sangat baik dalam segi skalabilitas sehingga pengimplementasian teknologi VoIP menjadi cukup murah dalam hal biaya operasionalnya dibandingkan dengan teknologi telephony konvensional. Dalam membangun jaringan VoIP diperlukan server VoIP atau biasa dikenal dengan IP PBX. Jenis software aplikasi proxy atau server VoIP yang dipakai tentunya memiliki spesifikasi tertentu yang dapat mempengaruhi kinerja suatu jaringan VoIP. Selain itu spesifikasi hardware server seperti kecepatan processor, memory maupun storage dapat menghasilkan kinerja yang berbeda-beda antara server yang satu dengan yang lain. Untuk mengetahui performansi suatu server dilakukan dengan mekanisme Quality of Service (QoS).

Pada tugas akhir ini permasalahan yang akan dibahas yaitu membandingkan performansi dari platform aplikasi proxy atau server VoIP yang berbeda-beda yaitu Asterisk, OpenSER, dan Axon pada platform Operating System (OS) yang berbeda-beda yaitu FreeBSD-Asterisk, Linux-OpenSER dan Windows-Axon. Performansi Quality of Service (QoS) yang akan dianalisa meliputi Call Setup Delay, CPU Usage, Delay, Jitter, dan Packet Loss. Untuk data register manajemen dalam akses hubungan antara tiap-tiap user pada layanan server VoIP akan digunakan satu server sebagai database server menggunakan MySQL Server atau SQL Server.

Kata kunci : VoIP, QoS, Asterisk, OpenSER, Axon, Database

Surabaya, 19 September 2008

Menyetujui,  
Dosen Pembimbing

Ir. Djoko Suprajitno Rahardjo  
NIP. 131 651 447

Menyetujui,  
Bidang Studi Telekomunikasi  
Koordinator

Mengetahui,  
Jurusan Teknik Elektro FTI-ITS  
Multimedia Ketua

DR. Ir. Mochamad, Ashari, M.Eng.  
NIP. 131 918 688

Ir. M. Aries Paramono  
NIP. 130 532 040

## A. JUDUL TUGAS AKHIR

Analisis Perbandingan Kualitas Jaringan VoIP pada Sistem Real Time VoIP Service Berdasarkan Perbedaan Platform Sistem Operasi.  
(Comparison Analysis of VoIP Network Quality on VoIP Service Real Time System Based on Operating System Platform Differences).

## B. RUANG LINGKUP

1. Jaringan Komputer
2. Jaringan Manajemen Telekomunikasi
3. Rekayasa Teknik Telekomunikasi
4. Sentral dan Jaringan Telekomunikasi

## C. LATAR BELAKANG

Kemajuan teknologi internet yang diikuti dengan penyajian *local content* yang tepat bagi para pengguna memunculkan salah satu solusi komunikasi murah yaitu teknologi VoIP. VoIP kependekan dari *Voice over Internet Protocol* adalah suatu teknologi menyalurkan data berupa suara, video dan data yang sudah dikemas dalam bentuk paket ke dalam jaringan Internet atau jaringan berbasis IP. Tidak seperti pada jaringan non-VoIP, misalkan jaringan PSTN maka suara yang berupa sinyal analog tidak mengalami perubahan terlebih dahulu sebelum disalurkan. Pada VoIP, suara yang berupa sinyal analog diubah menjadi data digital, diberi penomoran dan *header*, kemudian dikirimkan dalam bentuk paket – paket. Di bagian penerima selanjutnya paket – paket tersebut dibongkar dan diurutkan kembali untuk di ubah menjadi sinyal analog.

Implementasi jaringan VoIP memberikan manfaat yang cukup banyak dalam hal berkomunikasi dengan menggunakan jaringan data antara lain meningkatkan *reliabilitas* jaringan komputer, mampu menekan biaya penggunaan, efisiensi dalam pemakaian *bandwidth*, kemampuan untuk menggunakan kompresi suara, dan sebagainya. Karenanya teknologi VoIP merupakan salah satu bentuk penerapan teknologi tepat guna, dengan adanya teknologi ini dapat diperoleh suatu jaringan telepon yang lebih fleksibel dan membutuhkan biaya yang lebih murah daripada menggunakan fasilitas PSTN yang ada saat ini.

Untuk membangun jaringan VoIP dibutuhkan suatu *privy* atau *server* VoIP yang biasa disebut dengan IP PBX. Saat ini ada bermacam – macam *software* aplikasi untuk *server* VoIP yang dibuat dan salinnya sudah *open source* atau dapat digunakan dengan bebas tanpa harus membayar lisensi untuk menggunakan produk tersebut. *Software* – *software* aplikasi *server* tersebut dapat dijalankan dengan menggunakan bermacam – macam sistem operasi tergantung dari produsen pembuat *software* tersebut agar lebih *user friendly* dalam hal pengimplementasiannya oleh *end user* untuk merancang atau mengembangkan suatu jaringan VoIP.

Performansi dari suatu *server* VoIP dipengaruhi oleh beberapa hal yaitu spesifikasi dari *software* aplikasi *server* VoIP dan spesifikasi *hardware server* seperti kecepatan *processor*, *memory* maupun *storage* yang berbeda – beda antara *server* yang satu dengan yang lainnya. Pada tugas akhir ini spesifikasi *software* aplikasi *server* VoIP yang akan digunakan antara lain *Asterisk*, *OpenSER*, dan *Axon* yang akan dijalankan pada sistem operasi yang berbeda – beda yaitu *FreeBSD*, *Linux*, dan *Windows*. *Asterisk* akan dijalankan pada sistem operasi *FreeBSD*, *OpenSER* akan dijalankan pada sistem operasi *Linux* dan *Axon* akan dijalankan pada sistem operasi *Windows*. Untuk mengetahui performansi suatu *server* dibuktikan dengan mekanisme *Quality of Service (QoS)* yang meliputi pengukuran *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss*. Dengan membandingkan performansi *Quality of Service (QoS)* dari ketiga macam

---

aplikasi server VoIP yang dijalankan pada sistem operasi yang berbeda - beda tersebut akan dapat diketahui sejauh mana performansi atau kinerja server VoIP yang paling baik antara ketiga macam server VoIP itu.

#### D. PERMASALAHAN

Permasalahan yang akan dibahas dalam tugas akhir ini adalah bagaimana membangun jaringan VoIP menggunakan software aplikasi server VoIP dengan platform server VoIP yang berbeda - beda yang dijalankan pada platform sistem operasi yang berbeda - beda pula yaitu menggunakan Asterisk, OpenSER, dan Aeon yang akan dijalankan pada sistem operasi FreeBSD, Linux, dan Windows serta membangun database server sebagai tempat penyimpanan data register manajemen. Parameter *Quality of Service (QoS)* yang akan diukur meliputi *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss*. Selanjutnya hasil pengukuran performansi atau kinerja antara ketiga macam server VoIP tersebut akan dibandingkan.

#### E. BATASAN MASALAH

Mengacu pada permasalahan diatas maka batasan masalah pada tugas akhir ini dibatasi dengan asumsi sebagai berikut :

1. Menggunakan komputer sebagai server jaringan VoIP
2. Menggunakan sebuah perangkat server untuk digunakan menjalankan beberapa aplikasi VoIP server dan sebuah database server
3. Server VoIP terhubung ke database server dalam Jaringan Lokal (LAN)
4. Aplikasi proxy atau server VoIP yang akan dibandingkan antara lain Asterisk, OpenSER dan Aeon
5. Sistem Operasi yang akan dipakai antara lain FreeBSD, Linux dan Windows
6. *Quality of Service (QoS)* yang akan diukur meliputi *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss*

#### F. TUJUAN

Penelitian pada tugas akhir ini bertujuan untuk mengukur dan menganalisa perbandingan unjuk kerja atau performansi dengan mekanisme *Quality of Service (QoS)* dari platform aplikasi server VoIP yang berbeda - beda yang dijalankan pada sistem operasi yang berbeda - beda pula sehingga dari ketiga macam server VoIP tersebut dapat diketahui server VoIP yang paling baik kinerjanya.

#### G. PENELITIAN STUDI

##### *Voice over Internet Protocol (VoIP)*

Merupakan teknologi yang menjadikan media internet untuk melakukan komunikasi jarak jauh secara langsung, baik berupa komunikasi suara ataupun video. Sinyal suara analog ketika berkomunikasi di telepon diubah menjadi data digital dan dikirimkan melalui jaringan berupa paket data secara *real time*.

Teknologi VoIP diperkenalkan setelah internet mulai berkembang sekitar tahun 1995. Pada mulanya kemampuan mengirimkan suara melalui internet hanya merupakan eksperimen dari beberapa orang atau perusahaan kecil. Ini dimulai dengan perusahaan seperti *Vocaltech* dan kemudian pada akhirnya diikuti oleh *Microsoft* dengan program *Netmeeting*. Pada saat itu jaringan komputer internet masih sangat lambat, masih digambarkan *dial-up* dengan kecepatan 36,6 Kbyte. Aplikasi yang bersifat menghabiskan *bandwidth* seperti suara atau video, masih sangat terbatas penggunaannya.

Dalam komunikasi VoIP, pemakai melakukan hubungan telepon melalui terminal yang berupa PC atau telepon biasa. Dengan menggunakan VoIP, banyak keuntungan yang dapat diambil diantaranya adalah dari segi biaya lebih murah dari tarif telepon

konvensional. Jaringan IP yang bersifat global memungkinkan melakukan hubungan internasional dengan biaya yang juga lebih murah. Selain itu biaya pemeliharannya dapat ditekan karena teknologi VoIP dapat dipasang pada sembarang *ethernet* dan *IP address*, hal ini berbeda dengan telepon konvensional yang harus mempunyai port tersendiri di sentral atau PBX.

#### **Komponen Utama VoIP**

Untuk mengimplementasikan VoIP diperlukan beberapa komponen utama VoIP yang harus bekerja sama, antara lain :

1. **User Agent**

Ada yang berupa *software* dan ada yang berupa hardware. Merupakan end point dari jaringan VoIP seperti *Softphone*, *USBphone*, *IPphone*, *Internet Telephony Gateway (ITG)*, dan *Analog Telephone Adapter (ATA)*.

2. **Proxy**

Merupakan jembatan penghubung dari jaringan VoIP berupa *hardswitch* ataupun *softswitch* seperti *Asterisk*, *OpenSER* dan *Axon*

3. **Protokol**

Merupakan suatu aturan yang harus dipenuhi agar komunikasi VoIP dapat melewati jaringan internet. Dalam komunikasi VoIP mengenal tiga macam protokol tambahan selain protokol standar internet TCP/IP antara lain H.323 dari ITU-T, SIP dari IETF dan IAX.

4. **CoDec (Coder-Decoder)**

Dikembangkan untuk memampatkan atau mengompresikan suara atau video agar dapat menggunakan *bandwidth* secara hemat tanpa mengorbankan kualitas suara atau video dalam jaringan VoIP seperti GSM, ilBC dan lain-lain.

#### **Aplikasi Server VoIP**

##### **Asterisk**

*Asterisk* dibuat oleh Digium Corp. sebagai aplikasi *proxy* pada protokol H.323, SIP yang digunakan untuk membuat server VoIP. *Asterisk* merupakan VoIP server yang populer digunakan saat ini, dikarenakan *software* ini bersifat gratis, *open source*, dan memiliki fitur yang beragam untuk menyediakan layanan VoIP pada jaringan.

*Asterisk* bisa diimplementasikan sebagai:

1. **PABX**

*Asterisk* memungkinkan untuk terciptanya suatu PABX yang dapat menadangi fitur dan fungsi dari PABX circuit switch yang telah beredar di pasaran. *Asterisk* dapat dapat diintegrasikan dengan perangkat standard dari pesawat telepon yang berbasis IP atau perangkat lunak untuk telepon.

2. **Interactive Voice Response**

*Asterisk IVR* merupakan kemampuan untuk mengijinkan pemakai untuk saling berhubungan dengan database menggunakan menu yang sudah direkam sebelumnya.

3. **Auto-Attendant**

Suatu penerima otomatis yang meliputi salam, nada tunggu, pesan suara maupun catatan pesan.

4. **Conference Bridge**

Merupakan suatu program dalam *Asterisk* yang menawarkan fitur conference yang memungkinkan untuk beberapa orang saling berhubungan secara bersamaan.

---

5. VoIP dan Gateway Protocol

Dukungan Asterisk sangat luas terhadap switch circuit *Time Division Multiplex* (TDM) dan protokol VoIP yang dapat memungkinkan fleksibilitas gateway antar jenis saluran yang berbeda.

**OpenSER**

*OpenSER* juga merupakan salah satu jenis aplikasi server VoIP yang baru muncul setelah Asterisk berkembang yang dikembangkan oleh *Enhanced Telecommunication*. Termasuk jenis server VoIP yang menggunakan protokol SIP.

**Axon**

*Axon* merupakan jenis aplikasi server VoIP yang dijalankan pada sistem operasi menggunakan *Windows*. Seperti jenis aplikasi server VoIP lainnya, aplikasi server VoIP ini juga termasuk *open source*.

**Quality of Service (QoS)**

Standar yang digunakan sebagai acuan dalam menentukan kualitas jaringan VoIP adalah standar ITU-T Y.1541 yaitu "*Network Performance Objectives for IP-Based Service*". Rekomendasi ini mengatur parameter dari *Quality of Service* dan kondisi dari jaringan yang berdasarkan internet protokol. Terdapat beberapa parameter yang digunakan dalam mengukur performansi dari jaringan VoIP yaitu *Call Setup Delay*, *CPU Usage*, *Delay*, *Jitter* dan *Packet Loss*.

1. *Call Setup Delay*

Waktu yang dibutuhkan untuk membangun sebuah koneksi VoIP. Yaitu waktu yang dibutuhkan dari mulai menekan tombol tujuan sampai terdengar dering disisi penerima. Proses *call setup* ini membutuhkan waktu karena terdiri dari beberapa proses handshaking antara *caller* dan *callee* sampai keduanya setuju untuk melakukan koneksi VoIP.

2. *CPU Usage*

Besarnya nilai *CPU Usage* pada tiap server akan bergantung pada spesifikasi dari perangkat keras yang digunakan pada server VoIP.

3. *Delay*

*Delay* adalah waktu yang diperlukan oleh paket VoIP mulai dikirimkan dari *caller* sampai diterima oleh *callee*. *Delay* maksimum yang direkomendasikan oleh ITU-T Y.1541 untuk aplikasi suara adalah 100 ms, yang masih dapat diterima pengguna adalah 250 ms.

4. *Jitter*

*Jitter* adalah variasi delay yang terjadi karena delay yang terjadi selama proses koneksi VoIP tidaklah konstan melainkan bervariasi. Nilai *jitter* yang direkomendasikan oleh ITU-T Y.1541 adalah dibawah 75 ms.

5. *Packet Loss*

Paket yang dikirim dari *caller* ke *callee* tidak semuanya diterima oleh *callee*. Ada paket-paket yang tidak sampai atau hilang dalam perjalanan dan disebut dengan *packet loss*.

**Konsep Softswitching**

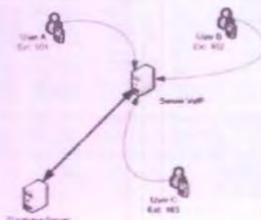
Sebagai konsep yang baru, *softswitching* juga diharapkan dapat memberikan solusi yang lebih baik bagi berbagai permasalahan yang timbul pada PSTN, terutama dalam hal masalah teknis yang menyangkut pengembangan layanan dan integrasi antar jejaring. Untuk membangun hubungan komunikasi pada jaringan VoIP, dibutuhkan *softswitching* yang berperan sebagai pusat *switching* atau sebagai jembatan untuk menghubungkan proses paketisasi, persinyalan dan pengkodean yang berbeda dalam

---

jaringan VoIP. Sebagai platform pengembangan layanan, *softswitch* memiliki keunggulan dalam hal kecepatan, kemudahan dan sifat terbuka dalam hal pengembangan layanan yang dilakukan dengan basis *application server*. Disamping itu *softswitch* juga mampu mengakomodasi layanan eksisting bahkan mengembangkan layanan tersebut lebih baik dari sistem sebelumnya. Konsep layanan yang dikembangkan dalam platform teknologi *softswitch* adalah dukungan konvergensi layanan masa datang, dengan integrasinya layanan suara dan data dalam satu platform jaringan.

#### Sistem Real Time VoIP Service

Pada tugas akhir ini dalam membangun jaringan VoIP, sebagai VoIP service akan digunakan sistem *real time*. Sistem *real time* dalam hal ini ditujukan pada akses data register manajemen, untuk itu dibuat sebuah *database server*. *Database server* berfungsi untuk menyimpan semua data – data register dari semua *user* yang dibutuhkan untuk akses sebelum melakukan panggilan. Dibawah ini dapat dilihat gambar topologi dari sistem *real time* VoIP service.



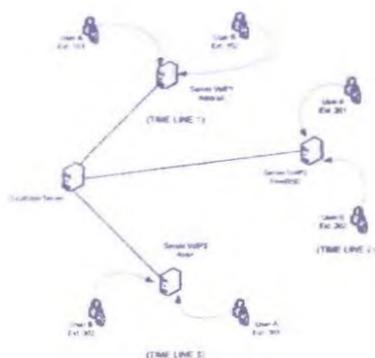
Gambar 1. Topologi Sistem Real Time VoIP Service

Pada gambar 1 dapat dilihat model sistem *real time* VoIP service sebagai contoh, misalkan *userA* akan melakukan panggilan ke arah *userB* maka sebelum *userA* melakukan panggilan ke arah *userB*, *server* VoIP akan mengambil data register atau melakukan *query data* ke *database server* pada *table user*. Setelah proses registrasi selesai maka *database server* akan *mc-reply* kembali ke *server* VoIP, menyambungkan *userA* dan *user B* untuk berkomunikasi. Komunikasi antara *server* VoIP dan *database server* terjadi dalam jaringan lokal (LAN) dengan port tertentu agar masing – masing *device* dapat berkomunikasi.

## H. METODOLOGI

Metodologi yang digunakan dalam pembahasan tugas akhir ini terdiri dari :

1. Studi Literatur
  - Mempelajari dan mengumpulkan referensi tentang jaringan VoIP, parameter QoS, *database server MySQL* serta *software aplikasi server* VoIP yaitu *Asterisk*, *OpenSER* dan *Axon*.
2. Desain Sistem
  - Merencanakan pembangunan jaringan VoIP dan peralatan yang dibutuhkan baik *hardware* maupun *software* untuk diintegrasikan guna merealisasikan jaringan yang akan dibangun.



Gambar 2. Desain Sistem Jaringan VoIP

3. Pembuatan dan Pengujian Sistem  
Menerapkan jaringan VoIP yang telah direncanakan untuk selanjutnya dilakukan pengujian terhadap jaringan VoIP yang telah dibangun dan menguji QoS server VoIP
4. Pengumpulan dan Analisa Data  
Dari hasil pengujian akan didapatkan data yang berupa QoS masing-masing server VoIP, kemudian dari data QoS akan dianalisa dan dibandingkan untuk mengetahui server VoIP yang lebih baik.
5. Penulisan Laporan  
Dalam penulisan laporan ini mengacu pada pedoman penulisan ilmiah dalam hal ini penulisan Tugas Akhir baku yang telah diatur oleh Jurusan Teknik Elektro.

#### I. JADWAL KEGIATAN

KEGIATAN	MINGGU															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a. Studi Literatur																
b. Desain Sistem																
c. Pembuatan dan Pengujian Sistem																
d. Pengumpulan dan Analisa Data																
d. Penyusunan Buku																

#### J. RELEVANSI

Dari Tugas Akhir ini diharapkan dapat digunakan sebagai referensi dalam perencanaan dan pembuatan jaringan VoIP. Serta menjadi referensi dalam merencanakan platform aplikasi server VoIP dengan sistem operasi apa yang paling sesuai untuk server VoIP berdasarkan spesifikasinya.

#### K. DAFTAR PUSTAKA

1. Gomillion D, Dempster B "Building telephony System with Asterisk" Packt Publishing, 2005
  2. Jackson B, Clark C, "Asterisk Hacking" Syngress Publishing, April 2007
  3. Goncalves F, "Building Telephony System with OpenSER", PACKT publishing, April 2008
  4. Singh K, Schulzrinc H "Failover and load balancing in SIP telephony" Department of Computer Science, Columbia University, 2005.
  5. ITU-T Recommendation Y.1541, "Network Performance Objectives for IP-based services", International Telecommunication Union, 2006
  6. M Riza Pahlevi, "Analisa Perbandingan Quality of Service pada server VoIP Berdasarkan Perbedaan Platform", Bidang Studi Telekomunikasi Multimedia Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember, 2008
-

## LAMPIRAN B

### Tabel Hasil Pengukuran Kualitas VoIP dengan Pembebanan

#### Pengukuran Delay

	Server 1 (OpenSER)			Server 2 (Asterisk)			Server 3 (Axon)		
	1	3	6	1	3	6	1	3	6
tanpa beban	0.40	1.80	3.60	0.7	2.40	3.60	1.20	3.40	3.90
beban 50%	0.90	2.10	5.90	1.7	3.50	6.70	1.70	4.60	9.10
beban 80%	1.50	4.30	7.40	2.20	5.60	8.90	2.6	8.10	13.30

#### Pengukuran Jitter

	Server 1 (OpenSER)			Server 2 (Asterisk)			Server 3 (Axon)		
	1	3	6	1	3	6	1	3	6
tanpa beban	2.4	10.4	12.6	2.6	11	13.9	5.2	14.60	15.7
beban 50%	4.2	13.20	15.7	3.6	12.20	15.7	5.3	15.10	16.9
beban 80%	6.1	15.10	16.1	6.5	16.80	17.1	6.4	15.70	18.4

#### Pengukuran Packet Loss

	Server 1 (OpenSER)			Server 2 (Asterisk)			Server 3 (Axon)		
	1	3	6	1	3	6	1	3	6
tanpa beban	0	0	0.1	0	0	0.2	0	0.00	0.4
beban 50%	0	0.30	0.80	0	0.60	3.5	0.1	0.70	4.5
beban 80%	0	3.20	7.1	0.1	5.40	9.40	0.3	9.80	11.6

#### Pengukuran CPU Usage

	Server 1 (OpenSER)			Server 2 (Asterisk)			Server 3 (Axon)		
	1	3	6	1	3	6	1	3	6
tanpa beban	7.08	13.38	29.47	19.13	28.36	32.26	26.98	41.09	55.58
beban 50%	9.22	17.29	28.42	24.73	29.76	36.6	27.29	45.51	66.58
beban 80%	14.35	28.42	38.4	27.03	35.16	58.49	34.58	59.46	75.48



### Pengukuran Call Setup Delay

	Server 1 (OpenSER)			Server 2 (Asterisk)			Server 3 (Axon)		
	1	3	6	1	3	6	1	3	6
tanpa beban	0.05	0.13	0.16	0.05	0.14	0.14	0.17	0.14	0.16
beban 50%	0.16	0.22	0.28	0.11	0.54	0.72	0.55	1.23	1.53
beban 80%	0.23	0.32	0.39	0.20	0.86	0.91	0.70	1.53	1.84

### Pengukuran Call Setup Delay pada tiap Softphone

	Server 1 (OpenSER)			Server 2 (Asterisk)			Server 3 (Axon)		
	X-Lite	Mizu	Sip Comm	X-Lite	Mizu	Sip Comm	X-Lite	Mizu	Sip Comm
Tanpa beban	0.05	0.08	0.08	0.13	0.10	0.13	0.16	0.16	0.09
beban 50%	0.18	0.19	0.13	0.20	0.54	1.21	0.47	1.29	0.55
beban 80%	0.19	0.22	0.49	0.54	1.36	1.43	1.04	1.82	2.45

### Pengukuran Delay Pada Server VoIP 1

Server 1			
Pengukuran Delay			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	2	3
2	0	1	2
3	1	1	2
4	1	1	1
5	1	1	1
6	0	1	1
7	1	1	1
8	0	0	1
9	0	1	2
10	0	0	1
Rata-Rata	0.4	0.9	1.5

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	1	2	4
2	2	3	4
3	2	2	6
4	2	3	4
5	2	2	6
6	2	2	3
7	2	2	3
8	2	2	5
9	2	1	4
10	1	2	4
Rata-Rata	1.8	2.10	4.30

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	2	2	6
2	2	4	6
3	4	4	7
4	4	2	8
5	4	4	8
6	4	7	8
7	5	7	8
8	3	10	8
9	5	10	7
10	3	9	8
Rata-Rata	3.6	5.9	7.4

## Pengukuran Delay Pada Server VoIP 2

Server 2			
Pengukuran Delay			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	2	2
2	0	3	2
3	1	2	3
4	1	2	1
5	1	2	3
6	1	1	3
7	1	1	2
8	1	1	2
9	0	2	2
10	1	1	2
Rata-Rata	0.7	1.7	2.2

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	1	4	5
2	2	4	5
3	2	3	5
4	2	3	6
5	2	4	6
6	2	4	6
7	2	3	6
8	2	2	6
9	3	4	5
10	6	4	6
Rata-Rata	2.4	3.50	5.60

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	2	4	7
2	2	4	7
3	4	4	8
4	4	4	8
5	4	4	9
6	4	7	10
7	6	9	12
8	6	12	12
9	2	10	8
10	2	9	8
Rata-Rata	3.6	6.7	8.9

### Pengukuran Delay Pada Server VoIP 3

Server 3			
Pengukuran Delay			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	1	2	2
2	1	2	3
3	2	2	4
4	2	2	3
5	1	1	2
6	1	2	2
7	1	1	3
8	1	1	3
9	1	2	2
10	1	2	2
Rata-Rata	1.2	1.7	2.6

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	2	2	6
2	4	4	10
3	3	6	9
4	3	6	7
5	3	5	7
6	5	4	8
7	4	5	6
8	4	4	8
9	5	6	12
10	1	4	8
Rata-Rata	3.4	4.60	8.10

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	3	11	14
2	3	8	14
3	4	8	15
4	4	9	13
5	5	12	11
6	5	8	12
7	4	9	14
8	4	11	14
9	4	8	12
10	3	7	14
Rata-Rata	3.9	9.1	13.3

## Pengukuran Jitter Pada Server VoIP 1

Server 1			
Pengukuran Jitter			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	1	4	17
2	1	5	15
3	2	3	10
4	4	4	3
5	2	6	2
6	3	6	2
7	4	5	3
8	3	3	2
9	2	3	3
10	2	3	4
Rata-Rata	2.4	4.2	6.1

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	11	12	13
2	9	12	14
3	11	14	14
4	10	14	15
5	11	13	15
6	12	14	16
7	11	12	18
8	10	13	16
9	10	16	12
10	9	12	18
Rata-Rata	10.4	13.20	15.10

6 panggilan			
Percobaan	Tanpa Beban	Beban	Beban
		50%	80%
1	12	11	10
2	13	8	10
3	10	10	15
4	12	13	18
5	12	13	18
6	14	20	17
7	12	22	21
8	14	27	19
9	14	15	17
10	13	18	16
Rata-Rata	12.6	15.7	16.1

### Pengukuran Jitter Pada Server VoIP 2

Server 2			
Pengukuran Jitter			
1 panggilan			
Percobaan	Tanpa Beban	Beban	Beban
		50%	80%
1	2	2	5
2	2	6	10
3	3	4	7
4	2	2	9
5	3	6	7
6	3	2	7
7	3	6	4
8	2	2	6
9	4	2	5
10	2	4	5
Rata-Rata	2.6	3.6	6.5

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	11	13	16
2	9	13	18
3	11	14	17
4	11	12	17
5	12	11	17
6	13	12	16
7	10	13	17
8	12	12	17
9	11	11	16
10	10	11	17
Rata-Rata	11	12.20	16.80

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	13	11	11
2	16	8	13
3	14	10	17
4	12	13	18
5	12	13	18
6	14	20	18
7	12	22	21
8	16	27	19
9	15	15	18
10	15	18	18
Rata-Rata	13.9	15.7	17.1

### Pengukuran Jitter Pada Server VoIP 3

Server 3			
Pengukuran Jitter			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	4	11	6
2	3	6	10
3	5	4	6
4	6	4	6
5	6	6	6
6	5	4	6
7	6	4	6
8	6	6	6
9	5	4	6
10	6	4	6
Rata-Rata	5.2	5.3	6.4

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	12	14	15
2	16	15	17
3	14	15	15
4	15	15	16
5	14	16	16
6	15	14	16
7	17	14	15
8	16	18	15
9	15	15	17
10	12	15	15
Rata-Rata	14.6	15.10	15.70

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	17	19	19
2	15	19	21
3	15	18	19
4	16	19	17
5	14	16	21
6	17	15	18
7	17	14	18
8	15	15	17
9	15	15	18
10	16	19	16
Rata-Rata	15.7	16.9	18.4

#### Pengukuran Packet Loss Pada Server VoIP 1

Server 1			
Pengukuran Packet Loss			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
Rata-Rata	0	0	0.0

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	0	0
2	0	0	3
3	0	1	3
4	0	0	2
5	0	1	1
6	0	1	1
7	0	0	4
8	0	0	5
9	0	0	6
10	0	0	7
Rata-Rata	0	0.30	3.20

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	0	5
2	0	2	5
3	0	2	8
4	0	0	7
5	1	0	10
6	0	0	8
7	0	1	5
8	0	1	7
9	0	0	9
10	0	2	7
Rata-Rata	0.1	0.8	7.1

## Pengukuran Packet Loss Pada Server VoIP 2

Server 2			
Pengukuran Packet Loss			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	1
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
Rata-Rata	0	0	0.1

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	0	6
2	0	1	6
3	0	2	5
4	0	1	4
5	0	1	5
6	0	1	6
7	0	0	4
8	0	0	7
9	0	0	7
10	0	0	4
Rata-Rata	0	0.60	5.40

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	0	11
2	0	1	9
3	0	3	9
4	1	1	11
5	0	0	11
6	1	0	8
7	0	0	11
8	0	6	8
9	0	10	7
10	0	14	9
Rata-Rata	0.2	3.5	9.4

### Pengukuran Packet Loss Pada Server VoIP 3

Server 3			
Pengukuran Packet Loss			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	0	0
2	0	0	0
3	0	0	0
4	0	1	2
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	1
9	0	0	0
10	0	0	0
Rata-Rata	0	0.1	0.3

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	0	0	8
2	0	1	8
3	0	1	11
4	0	0	9
5	0	0	9
6	0	0	9
7	0	1	9
8	0	1	12
9	0	1	11
10	0	2	12
Rata-Rata	0	0.70	9.80

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	1	2	12
2	1	5	10
3	0	6	8
4	0	7	12
5	0	5	10
6	1	5	14
7	0	3	14
8	1	4	8
9	0	6	13
10	0	2	15
Rata-Rata	0.4	4.5	11.6

## Pengukuran CPU Usage Pada Server VoIP 1

Server 1			
Pengukuran CPU Usage			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	6.6	7.6	9.3
2	6.7	7.9	8.3
3	6.7	8.6	12.3
4	6.7	9.3	14.3
5	6.7	9.9	14.9
6	7	10.3	16.3
7	7	11	17.2
8	7	10.3	20.9
9	8.2	8.7	13.7
10	8.2	8.6	16.3
Rata-Rata	7.08	9.22	14.4

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	10.6	12.6	20.9
2	11	12.6	23.2
3	11.7	14.3	23.2
4	12	14.3	25.6
5	12.9	15.2	31.1
6	13	16.1	52.8
7	13.7	20.7	19.9
8	14	22.7	26.8
9	16.2	26.1	29.6
10	18.7	18.3	31.1
Rata-Rata	13.38	17.29	28.42

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	23.8	28.2	34.4
2	27.3	28.3	34.6
3	29.9	25.5	36.2
4	30.6	29	36.3
5	32.3	30.2	33.9
6	34.4	30.2	33.9
7	38.6	31.3	59.2
8	24.2	31.6	34.7
9	25.9	24.4	45.6
10	27.7	25.5	35.2
Rata-Rata	29.47	28.42	38.4

#### Pengukuran CPU Usage Pada Server VoIP 2

Server 2			
Pengukuran CPU Usage			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	15.9	16.9	21.3
2	16.6	22	30.8
3	18.6	31.2	33.1
4	19.3	17.5	14.9
5	19.9	16.6	25.2
6	19.9	32.7	32.8
7	20	37.2	33.1
8	20.3	22.5	24.8
9	20.3	19.5	17.2
10	20.5	31.2	37.1
Rata-Rata	19.13	24.73	27.0

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	13.6	39.1	25.7
2	15.6	25.4	25.7
3	20.3	23.4	37.9
4	23.5	23.4	43.6
5	25.6	32.5	45.4
6	28.6	43.9	37.9
7	33.8	24.8	21.5
8	36.6	23.4	40.9
9	37.7	32.5	35.1
10	48.3	29.2	37.9
Rata-Rata	28.36	29.76	35.16

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	21.2	28.9	57.4
2	27.2	30.2	53.6
3	31.9	34.6	68.9
4	35.8	31	43
5	36.5	28.5	42.9
6	39.1	45.2	57.4
7	40.4	46.9	65.3
8	29.7	61.2	72
9	28.7	28.5	76.5
10	32.1	31	47.9
Rata-Rata	32.26	36.6	58.49

### Pengukuran CPU Usage Pada Server VoIP 3

Server 3			
Pengukuran CPU Usage			
1 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	16.3	14.6	32.3
2	17.2	26.3	32.7
3	18.9	31.2	32.8
4	19.1	27.7	33.4
5	23.8	28.1	33.6
6	29.4	28.3	34
7	31.1	28.7	34
8	41.7	30.2	35.1
9	51	28.9	35.1
10	21.3	28.9	42.8
Rata-Rata	26.98	27.29	34.6

3 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	34.9	37.5	51.2
2	37.5	36.2	51.6
3	55.9	58.4	53
4	62.4	63.4	63.2
5	29.7	48	65
6	29.1	38.9	68.2
7	28.1	35.6	77.5
8	35.1	34.1	82.6
9	56.6	50.2	44.6
10	41.6	52.8	37.7
Rata-Rata	41.09	45.51	59.46

6 panggilan			
Percobaan	Tanpa Beban	Beban 50%	Beban 80%
1	49.2	81.2	76.8
2	51.2	73.2	66
3	52.8	62.8	90.4
4	53.3	60.7	80.1
5	54.5	60.7	80.1
6	55.3	73.2	65.6
7	56.4	54.4	63.8
8	57.6	83.4	63.8
9	58.8	66.4	87.5
10	66.7	49.8	80.7
Rata-Rata	55.58	66.58	75.48

## RIWAYAT HIDUP PENULIS



Ade Chandra Dewa dilahirkan di kota Surabaya pada tanggal 19 September 1984 merupakan anak pertama dari dua bersaudara dari pasangan Ismail Hidayat dan Siti Muktati. Memulai pendidikan di SDN Cangkir I Driyorejo pada tahun 1990, SLTPN 1 Driyorejo pada tahun 1996, dan SMU 6 Surabaya pada tahun 2002. Melanjutkan jenjang studi perguruan tinggi di PENS-ITS jurusan telekomunikasi dan lulus pada tahun 2005. Pada awal tahun 2006 bersamaan memulai bekerja di PT

Indosat Surabaya penulis melanjutkan pendidikan tingkat sarjana di Institut Teknologi Sepuluh Nopember pada tahun 2005 di Jurusan Teknik Elektro ITS, bidang studi Telekomunikasi Multimedia.

email : [adechandra\\_dewa@yahoo.co.id](mailto:adechandra_dewa@yahoo.co.id)