



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN DAN UNJUK KERJA  
QOS OBJEKTIF VOIP BERBASIS SIP  
MENGUNAKAN *CODEC* G-711  $\mu$ -LAW DAN GSM 06.10**

**SKRIPSI**

**DEWI ASRI TIARA PUTRI  
040503029X**

**FAKULTAS TEKNIK  
PROGRAM TEKNIK ELEKTRO  
DEPOK  
JULI 2009**



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN DAN UNJUK KERJA  
OBJEKTIF QOS VOIP BERBASIS SIP  
MENGUNAKAN *CODEC* G-711  $\mu$ -LAW DAN GSM 06.10**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**DEWI ASRI TIARA PUTRI**

**040503029X**

**FAKULTAS TEKNIK  
PROGRAM TEKNIK ELEKTRO  
DEPOK  
JULI 2009**

## **HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Dewi Asri Tiara Putri**

**NPM : 040503029X**

**Tanda Tangan :**

**Tanggal : Juli 2009**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Dewi Asri Tiara Putri  
NPM : 040503029X  
Program Studi : Teknik Elektro  
Judul Skripsi : Rancang Bangun dan Unjuk Kerja Objektif QoS  
VoIP Berbasis SIP Menggunakan *Codec* G.711  $\mu$ -  
Law dan GSM 06.10

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia**

### DEWAN PENGUJI

Pembimbing : Ir. Arifin Djauhari, MT ( )

Penguji : Dr. Ir. Fitri Yuli Zulkifli ( )

Penguji : Ir. Rochmah N Sukardi Ny, MT ( )

Ditetapkan di : Depok

Tanggal : Juli 2009

## KATA PENGANTAR

Puji syukur ke hadirat Allah Azza wa Jalla yang telah memberikan hidayah serta kemudahan kepada penulis dalam menyelesaikan skripsi ini. Skripsi ini juga dapat terselesaikan atas bantuan serta dukungan banyak pihak. Pertama, penulis mengucapkan terima kasih kepada:

**Ir. Arifin Djauhari, MT.**

sebagai dosen pembimbing yang telah banyak meluangkan waktunya untuk memberikan saran, bimbingan serta pengarahan sehingga skripsi ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada:

- (1) Kedua orang tua penulis beserta keluarga besar di Bandar Lampung yang selalu memberikan doa serta dukungan selama ini.
- (2) Agus Awaludin selaku administrator PPSI yang telah memberikan pengarahan dan izin penggunaan fasilitas PPSI dalam pengerjaan skripsi ini
- (3) Teman-teman putri Elektro UI 2005 yang selalu memotivasi dan menyayangi penulis: Astatine, Citra, Dita, Khotimah, Mayang, Nia, Rinda dan juga Amel.
- (4) Teman-teman seangkatan Elektro UI 2005 yang banyak membantu dan mengarahkan khususnya Ahmad Fauzi, Arif R., Faizal, Ferdinand, Gatot, Mario dan Okki.
- (5) Teman-teman SMUNDA Balam terutama Herika P., Irvan T., dan Yoga S.
- (6) Teman-teman asisten 2005 & 2006 Laboratorium Telekomunikasi UI yang selalu memberikan semangat serta dukungan.
- (7) Teman-teman di PPSI yaitu Dimas Poltek, Mas Ewoh dan Mas Ridwan.

Penulis berharap Allah Azza wa Jalla berkenan membalas segala kebaikan semua pihak yang telah membantu hingga skripsi ini dapat diselesaikan dengan baik. Semoga penulisan skripsi ini dapat memberikan kontribusi dalam mengembangkan ilmu pengetahuan dan teknologi khususnya telekomunikasi.

Depok, Juli 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Dewi Asri Tiara Putri  
NPM : 04050329X  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

**RANCANG BANGUN DAN UNJUK KERJA OBJEKTIF QOS VOIP**

**BERBASIS SIP MENGGUNAKAN CODEC G-711  $\mu$ -LAW DAN GSM 06.10**

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia / format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada Tanggal : 4 Juli 2009

Yang menyatakan

(Dewi Asri Tiara Putri)

## ABSTRAK

Nama : Dewi Asri Tiara Putri  
Program Studi : Teknik Elektro  
Judul : Rancang Bangun dan Unjuk Kerja Objektif QoS pada Jaringan VoIP Berbasis SIP Menggunakan *Codec* G.711  $\mu$ -Law dan GSM 06.10

Skripsi ini membahas perbandingan QoS objektif pada jaringan VoIP berbasis SIP dengan variasi berupa *codec* yang digunakan yaitu G.711  $\mu$ -Law dan GSM 06.10 yang diuji di dua jaringan yang berbeda yaitu LAN UI dan Indosat. Parameter QoS objektif yang diukur adalah *delay*, *bandwidth* dan *packet loss*. Dari hasil pengukuran diperoleh bahwa kedua *codec* memenuhi QoS objektif baik di sisi *uplink* dan *downlink* pada jaringan LAN UI. Sedangkan pada jaringan Indosat kedua *codec* memiliki QoS objektif yang baik hanya di sisi *downlink* sementara di sisi *uplink* hanya *codec* GSM 06.10 yang memenuhi syarat QoS objektif untuk aplikasi layanan suara.

Kata kunci :  
VoIP, SIP, QoS, G.711  $\mu$ -Law, GSM 06.10

## ABSTRACT

Name : Dewi Asri Tiara Putri  
Study Program: Electrical Engineering  
Title : Development and Performance of Objective QoS in SIP-based  
VoIP Network Using Codec G.711  $\mu$ -Law and GSM 06.10

The focus of this study is comparison of the objective QoS in SIP-based VoIP network with the variation in the codec, they are G.711  $\mu$ -Law and GSM 06.10 that were tested in two different network, LAN UI and Indosat. Parameters of objective QoS that were measured are delay, bandwidth, and packet loss. From the measurement, the both of codecs fulfill the criteria for the objective QoS in uplink and downlink side in LAN UI network. But in Indosat network, both of the codec have good objective QoS only in downlink side but in uplink side only GSM 06.10 that fulfill the criteria of objective QoS for voice application service.

Key words :  
VoIP, SIP, QoS, G.711  $\mu$ -Law, GSM 06.10



# DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR .....	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	v
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB 1. PENDAHULUAN .....	1
1.1 LATAR BELAKANG .....	1
1.2 TUJUAN .....	4
1.3 PEMBatasan MASALAH.....	4
1.4 METODOLOGI PENELITIAN .....	4
1.5 SISTEMATIKA PENULISAN .....	5
BAB 2. KONSEP DASAR VOIP BERBASIS SIP .....	7
2.1 VOICE OVER INTERNET PROTOCOL .....	7
2.1.1 Arsitektur VoIP .....	7
2.1.2 Protokol pada VoIP .....	10
2.1.2.1 Internet Protocol (IP) .....	11
2.1.2.2 Transfer Control Protocol (TCP) .....	13
2.1.2.3 User Datagram Protocol .....	14
2.1.2.4 <i>Real Time Transport Protocol</i> (RTP) .....	14
2.1.2.5 <i>Real Time Control Protocol</i> (RTCP) .....	16
2.2 SESSION INITIATION PROTOCOL .....	16
2.2.1 Pengantar SIP .....	16
2.2.2 Arsitektur SIP .....	18
2.2.3 Nama dan Pengalamatan pada SIP .....	20

2.2.4 SIP Messages .....	21
2.2.4.1 Request Message .....	21
2.2.4.2 Response Message .....	22
2.3 CODER-DECODER (CODEC) .....	24
2.3.1 G.711 $\mu$ -Law .....	25
2.3.2 GSM 06.10 .....	26
2.3.3 Voice Activity Detection (VAD) .....	27
2.4 QUALITY OF SERVICE (QoS) .....	27
2.5.1 Subjektif QoS .....	28
2.5.2 Objektif QoS .....	28
2.5.2.1 Packet loss .....	29
2.5.2.2 Delay .....	29
2.5.2.3 Jitter .....	30
2.5.2.4 Bandwidth .....	30
<b>BAB 3 PERANCANGAN VOIP BERBASIS SIP MENGGUNAKAN G.711</b>	
$\mu$ -LAW DAN GSM 06.10 .....	32
3.1 KONFIGURASI SERVER .....	32
3.1.1 Console .....	32
3.1.2 Web .....	37
3.2 KONFIGURASI CLIENT .....	41
3.3 PuTTY .....	42
3.4 Wireshark .....	43
<b>BAB 4 UJI COBA, DATA DAN ANALISIS</b> .....	
4.1 SKENARIO UJI COBA .....	45
4.2 UJI COBA INTERNAL LAN UI .....	46
4.2.1 User1 ke User2 .....	46
4.2.1.1 Uplink .....	46
4.2.1.2 Downlink .....	47
4.2.2 User2 ke User .....	49
4.2.2.1 Uplink .....	49
4.2.2.2 Downlink .....	50

4.3 UJI COBA LAN UI – INDOSAT .....	53
4.3.1 User1 ke User3 .....	53
4.3.1.1 Uplink .....	53
4.3.1.2 Downlink .....	55
4.3.2. User3 ke User1 .....	56
4.3.2.1 Uplink .....	56
4.3.2.2 Downlink .....	58
BAB 5 KESIMPULAN .....	63
DAFTAR ACUAN .....	64
DAFTAR PUSTAKA .....	65

## DAFTAR GAMBAR

Gambar 2.1. Arsitektur jaringan VoIP .....	8
Gambar 2.2 Jaringan non-VoIP yang dihubungkan dengan jaringan IP .....	8
Gambar 2.3 Perbandingan protokol VoIP di <i>layer</i> TCP/IP .....	11
Gambar 2.4 Format Datagram IP .....	12
Gambar 2.5 Struktur Paket TCP .....	13
Gambar 2.6 Format data dan header pada model TCP/P .....	13
Gambar 2.7 Struktur paket UDP .....	14
Gambar 2.8 Kedudukan protokol RTP di antara protokol lain dalam VoIP .....	15
Gambar 2.9 <i>Header</i> RTP .....	15
Gambar 2.10 Arsitektur SIP .....	18
Gambar 2.11 <i>Call flow</i> menggunakan SIP Proxy .....	19
Gambar 2.12 <i>Call flow</i> dengan menggunakan SIP Redirect .....	19
Gambar 2.13 Proses registrasi SIP .....	20
Gambar 2.14 Contoh bentuk <i>request message</i> SIP .....	22
Gambar 2.15 Response message SIP .....	23
Gambar 2.16 Contoh sesi komunikasi lengkap pada SIP .....	24
Gambar 2.17 Subjektif vs Objektif QoS .....	28
Gambar 2.18 Macam-macam <i>delay</i> saat transmisi VoIP .....	30
Gambar 3.1 Tampilan awal instalasi Briker IPPBX .....	33
Gambar 3.2 Pemeriksaan drive CD-ROM .....	33
Gambar 3.3 Pemeriksaan hardware .....	34
Gambar 3.4 Formating hard disk .....	34
Gambar 3.5 Instalasi <i>software</i> aplikasi .....	34
Gambar 3.6 Instalasi GRUB boot loader .....	34
Gambar 3.7 Console login di Briker .....	35
Gambar 3.8 Konfigurasi network address default pada Briker .....	36
Gambar 3.9 Network address setelah dikonfigurasi .....	36
Gambar 3.10 Tampilan <i>login</i> pada Briker di alamat 152.118.148.231 .....	37
Gambar 3.11 Menu untuk mengubah password administrator .....	38

Gambar 3.12 Tampilan pada halaman IPPBX Administration .....	38
Gambar 3.13 Memilih jenis protokol extension .....	39
Gambar 3.14 Membuat extension baru .....	40
Gambar 3.15 Empat buah SIP extension yang telah dibuat .....	41
Gambar 3.16 Tampilan <i>softphone</i> .....	41
Gambar 3.17 Cara konfigurasi Xlite .....	42
Gambar 3.18 Akses <i>server</i> melalui PuTTY di Windows .....	43
Gambar 3.19 Memilih interface yang ingin di-capture melalui Wireshark .....	44
Gambar 4.1 Topologi jaringan yang diuji coba .....	45
Gambar 4.2 Grafik Delay <i>Uplink</i> User1 ke User2 di Internal LAN UI .....	47
Gambar 4.3 Grafik <i>Bandwidth Uplink</i> User1 ke user2 Internal LAN UI .....	47
Gambar 4.4 Grafik Delay <i>Downlink</i> User1 ke User2 Internal LAN UI .....	48
Gambar 4.5 Grafik <i>Bandwidth Downlink</i> User1 ke User2 Internal LAN UI .....	48
Gambar 4.6 Grafik Delay <i>Uplink</i> User2 ke User1 Internal LAN UI .....	49
Gambar 4.7 Grafik <i>Bandwidth Uplink</i> User2 ke User1 Internal LAN UI .....	50
Gambar 4.8 Grafik Delay <i>Downlink</i> User2 ke User1 Internal LAN UI .....	51
Gambar 4.9 Grafik <i>Bandwidth Downlink</i> User2 ke User1 Internal LAN UI .....	51
Gambar 4.10 Grafik Delay <i>Uplink</i> User1 ke User3 LAN UI – Indosat .....	53
Gambar 4.11 Grafik <i>Bandwidth Uplink</i> User1 ke User3 LAN UI – Indosat .....	54
Gambar 4.12 Grafik Delay <i>Downlink</i> User1 ke User3 LAN UI – Indosat .....	55
Gambar 4.13 Grafik <i>Bandwidth Downlink</i> User1 ke User3 LAN UI – Indosat .....	56
Gambar 4.14 Grafik Delay <i>Uplink</i> User3 ke User1 LAN UI – Indosat .....	57
Gambar 4.15 Grafik <i>Bandwidth Uplink</i> User3 ke User1 LAN UI – Indosat .....	58
Gambar 4.16 Grafik Delay <i>Downlink</i> User3 ke User1 LAN UI – Indosat .....	59
Gambar 4.17 Grafik <i>Bandwidth Downlink</i> User3 ke User1 LAN UI – Indosat .....	59

## DAFTAR TABEL

Tabel 1.1 Tiga kelas kualitas end-to-end telepon .....	2
Tabel 1.2 Pembagian <i>one way transmission delay</i> oleh ITU-T .....	3
Tabel 1.3 Hasil Penelitian Performansi VoIP di Laboratorium Telematika ITB ...	3
Tabel 2.1 Beberapa jenis <i>speech encoding</i> pada VoIP .....	25
Tabel 2.2 Distribusi bit yang digunakan pada codec GSM <i>Full Rate</i> .....	26
Tabel 2.3 Nilai MOS dan kualitasnya .....	28
Tabel 4.1 <i>Delay</i> Internal LAN UI .....	52
Tabel 4.2 <i>Bandwidth</i> Internal LAN UI .....	52
Tabel 4.3 <i>Packet loss</i> Internal LAN UI .....	52
Tabel 4.4 <i>Delay</i> LAN UI-Indosat .....	60
Tabel 4.5 <i>Bandwidth</i> LAN UI-Indosat .....	60
Tabel 4.6 <i>Packet Loss</i> LAN UI-Indosat .....	60

# BAB 1

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Perkembangan teknologi di bidang telekomunikasi yang begitu cepat saat ini telah mampu menawarkan berbagai macam layanan komunikasi dengan teknik dan metode yang beragam. Layanan yang ditawarkan antara lain suara (*voice*), pesan (*messaging*) hingga multimedia (audio, video dan data). Metode dan teknik yang digunakan juga bervariasi mulai dari yang paling sederhana yaitu analog dan kabel (*wire*) sampai ke digital dan nirkabel (*wireless*).

Meskipun sudah terdapat banyak tipe layanan yang ditawarkan, layanan telekomunikasi yang paling fundamental tetaplah layanan teleponi. Metode untuk layanan suara ini pada awalnya hanya berupa terminal analog berbasis *circuit switching* dan menggunakan kabel atau yang biasa dikenal dengan *Public Switched Telephone Network* (PSTN). Metode *circuit switching* sendiri adalah metode yang mengalokasikan sebuah sirkuit atau kanal yang khusus (*dedicated*) di antara simpul (*node*) dan terminal untuk digunakan pengguna (*user*) dalam berkomunikasi. Sirkuit yang *dedicated* tidak dapat digunakan oleh penelepon lain sampai sirkuit itu dilepaskan.

Perkembangan teknologi selanjutnya adalah mulai digunakannya teknologi digital dalam telekomunikasi. Akibat digitalisasi ini akhirnya muncul metode *packet switching* yaitu sebuah sirkuit tidak ditujukan khusus untuk satu *user* tapi bisa digunakan secara beramai-ramai. Hal ini dimungkinkan karena informasi yang disalurkan melalui sirkuit tersebut berupa paket-paket kecil yang waktu pendudukannya relatif sangat singkat sehingga banyak pasangan *node* yang dapat melakukan komunikasi secara hampir simultan pada kanal yang sama. Metode *packet switching* ini pada telekomunikasi digunakan pada teknologi *Integrated Service Digital Network* (ISDN) dan telepon nirkabel atau seluler (*wireless mobile telephony*) dimana terminal (*handset*) telah memiliki fungsi mengubah sinyal suara analog menjadi sinyal digital yang kemudian baru ditransmisikan. Pada

teknik ini, meskipun jaringan sirkuit digunakan bersama (*shared networks*) namun jaringan tersebut hanya digunakan untuk keperluan telekomunikasi saja.

Perkembangan selanjutnya dari teknik layanan suara adalah *packet switching* yang menggunakan jaringan umum yang tidak hanya ditujukan untuk keperluan telekomunikasi saja yaitu jaringan internet. Layanan suara yang menggunakan fasilitas internet ini kemudian dikenal sebagai *Voice over Internet Protocol* atau VoIP. Teknologi internet dapat digunakan untuk melewati layanan suara karena merupakan jaringan umum yang benar-benar berbasis digital dan dapat dilalui oleh berbagai tipe paket yang bukan hanya bertipe data saja tapi juga suara dan gambar. Penggunaan jaringan internet yang semakin meluas di seluruh dunia juga membuat layanan VoIP semakin diminati karena internet membuat orang-orang dapat saling terhubung dengan mudah dan murah tanpa memperdulikan seberapa jauh jarak yang memisahkan. Selain itu, perkembangan VoIP juga didukung dengan fenomena bermigrasinya hampir semua teknologi transport network pada jaringan telekomunikasi menjadi berbasis internet *protocol* (IP) dalam rangka mewujudkan pembentukan *Next Generation Network* (NGN).

Meskipun teknik yang digunakan untuk memberikan layanan suara sudah berkembang sedemikian pesat namun kualitas atau *Quality of Service* (QoS) suara yang terbaik ternyata masih dipegang oleh PSTN. Hal ini disebabkan jaringan PSTN masih memakai kabel serta teknik *circuit switching* yang bersifat *dedicated circuit* sehingga layanan suara yang diberikan mutunya dapat terjamin. Beberapa parameter QoS dalam sistem teleponi antara lain adalah waktu tunda transmisi satu arah (*one way delay transmission*) yang masih diperbolehkan dalam suatu sambungan dan *loss probability*. Parameter *delay* distandardisasi oleh ITU-T yaitu pada ITU-T G.114. Perbedaan nilai pada parameter-parameter ini membuat layanan suara terbagi menjadi tiga kelas QoS yang dapat dilihat di Tabel 1.1.

Tabel 1.1. Tiga kelas kualitas *end-to-end* telepon [1]

	Class A	Class B	Class C
ITU-T G.114	< 100 ms	< 150 ms	< 400 ms
Loss probability	$\leq 0.15$	$\leq 0.15$	$\leq 0.15$
Comparison Example	Fixed line telephone (PSTN)	Mobile phone	Lowest level for voice communication



Sementara itu, pada VoIP yang diprediksi akan menjadi teknologi telepon di masa depan karena berbagai keunggulannya seperti pengurangan biaya, kemudahan *konektivitas*, menambah penjumlahan panggilan, dan menambah fungsionalitas, ternyata QoS masih menjadi kendala. Masalah QoS pada VoIP antara lain adalah bagaimana meminimalisir paket suara yang hilang saat pentransmisi melalui jaringan internet yang bersifat *best effort* (tidak menjamin reliabilitas) serta bagaimana menjamin agar trafik paket untuk suara tidak tertunda atau terputus pada layanan VoIP yang bersifat real-time. Untuk memberikan layanan yang *real time* (waktu nyata), waktu tunda transmisi satu arah (*one way transmission delay*) pada VoIP hendaknya kurang dari 150 ms, seperti standar yang ditunjukkan pada Tabel 1.2 di bawah ini.

Tabel 1.2. Pembagian *one way transmission delay* oleh ITU-T [2]

Class No.	Delay by direction	Comments
1	0 to 150 ms	Acceptable for most conversations; only some highly interactive tasks may experience degradation.
2	150 to 300 ms	Acceptable for low-interactivity calls (satellite 250 ms per hop).
3	300 to 700 ms	Practically a half-duplex call.
4	Above 700 ms	Unusable unless the callers are well-versed in the art of half-duplex conversation (as used in the military).

Pada VoIP, beberapa parameter yang diperhatikan dalam QoS selain *delay* adalah *jitter*, *packet loss* dan *bandwidth* [2]. *Delay* adalah waktu tunda saat pengiriman paket, *jitter* adalah variasi *delay* dalam pengiriman paket, *packet loss* adalah paket yang hilang saat pengiriman, sedangkan *bandwidth* adalah besarnya *resource* jaringan yang diperlukan. Masing-masing faktor berhubungan untuk menentukan apakah suatu kualitas layanan VoIP baik atau tidak. Sebuah penelitian mengenai performansi jaringan VoIP berdasarkan parameter QoS di atas pernah dilakukan di laboratorium DSP Telematika ITB dengan hasilnya ditunjukkan pada Tabel 1.3 berikut.

Tabel 1.3 Hasil Penelitian Performansi VoIP di Laboratorium DSP Telematika ITB

Parameter	G.711 $\mu$ -Law	GSM 06.10
<i>Delay</i> (ms)	27.73	49.93
<i>Bandwidth</i> (Kbps)	72.79	26.97
Packet Loss (%)	1.1	1.43

Terdapat beberapa cara untuk memenuhi syarat QoS pada VoIP antara lain adalah pemilihan protokol *signaling* dan pemilihan jenis *codec* (*coder-decoder*) yang digunakan. Pemilihan protokol *signaling* berpengaruh pada proses set-up sambungan dan pengontrolannya sedangkan pemilihan *codec* berpengaruh pada besarnya *bandwidth* yang dikonsumsi untuk pentransmisian. Penelitian dalam skripsi ini akan membahas mengenai hubungan kualitas suara dengan parameter-parameter QoS VoIP. Protokol *signaling* yang dipilih adalah SIP dengan parameternya adalah *codec* G.711- $\mu$ Law dan GSM 06.10 yang diuji melalui dua jaringan yang berbeda yaitu LAN UI dan Indosat. Kedua *codec* ini dipilih karena keduanya memiliki kualitas yang cukup baik tapi nilai *bit-rate* keduanya cukup jauh berbeda. Selain itu, kedua *codec* ini tidak memerlukan lisensi sehingga dapat digunakan secara *free*.

## 1.2 TUJUAN

Tujuan skripsi ini adalah merancang dan membangun jaringan VoIP berbasis SIP lalu membandingkan dan menganalisis objektif QoS antara menggunakan *codec* G.711- $\mu$ Law dan GSM 06.10 dengan parameter yang dilihat adalah *delay*, *bandwidth* dan *packet loss*.

## 1.3 PEMBATAAN MASALAH

Beberapa batasan dalam skripsi ini antara lain:

1. Aplikasi : *Voice over Internet Protocol* (VoIP)
2. Protokol : SIP
3. *Codec* : G.711- $\mu$ Law dan GSM 06.10
4. Sistem operasi : Ubuntu 8.04, Windows XP SP 2
5. *Software* : Briker
6. Parameter QoS : *delay*, *bandwidth* dan *packet loss*.

## 1.4 METODOLOGI PENELITIAN

Metoda penelitian yang digunakan di dalam penyusunan tugas akhir ini adalah sebagai berikut:

#### 1. Studi Literatur

Penelitian dilakukan dengan cara membaca berbagai tulisan dan dokumentasi (baik *online* maupun *offline*) tentang VoIP dan QoS serta *software* yang digunakan, dan mengikuti forum yang berisi komunitas orang yang mengembangkan aplikasi VoIP.

#### 2. Pembangunan Jaringan

Membangun jaringan VoIP berbasis SIP dengan *proxy server* SIP ditempatkan di ruangan *server* PPSI UI

#### 3. Pengukuran

Pengukuran kualitas layanan telepon internet dilakukan secara kuantitatif dengan parameter-parameter QoS yaitu *delay*, *bandwidth* dan *packet loss* pada jaringan VoIP yang dibangun.

#### 4. Analisis

Jaringan yang dibangun diuji coba, kemudian hasil pengukuran yang didapatkan dari uji coba dianalisis.

## 1.5 SISTEMATIKA PENULISAN

Penulisan skripsi ini terdiri dari 5 bab dengan sistematika penulisan sebagai berikut:

### BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang, tujuan, pembatasan masalah, metodologi penelitian dan sistematika penulisan.

### BAB 2 DASAR TEORI

Bab ini membahas dasar teori mengenai jaringan VoIP, arsitektur SIP, *codec* yang digunakan, dan parameter-parameter QoS pada VoIP.

### BAB 3 PERANCANGAN

Bab ini membahas mengenai perancangan jaringan VoIP yang berbasis SIP dengan *codec* G.711 dan GSM 06.10. Perancangan terdiri dari konfigurasi *server* VoIP melalui *shell* atau terminal, konfigurasi *server* berbasis web,

konfigurasi *software client* dan konfigurasi cara pengukuran parameter QoS dari VoIP.

#### BAB 4 UJI COBA DAN ANALISIS

Bab ini membahas mengenai hasil pengujian terhadap jaringan VoIP yang telah dibuat dan analisa performansinya.

#### BAB 5 KESIMPULAN

Bab ini menjelaskan mengenai beberapa kesimpulan yang diperoleh dari hasil penelitian dan analisisnya.

## BAB 2

# KONSEP DASAR VOIP BERBASIS SIP

### 2.1 VOICE OVER INTERNET PROTOCOL (VOIP)

#### 2.1.1 Arsitektur VoIP

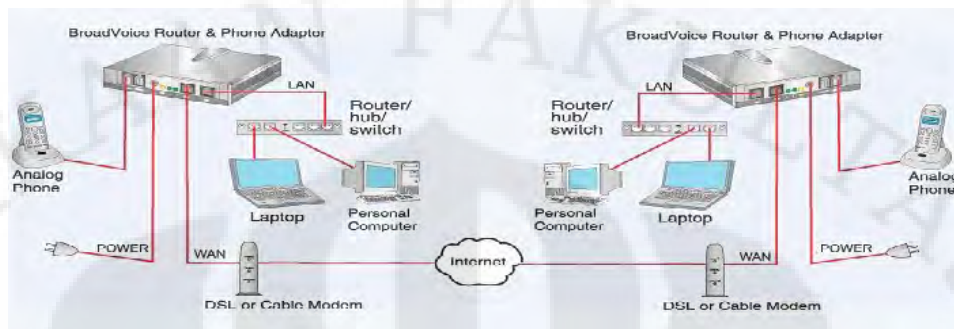
Menurut Wikipedia Indonesia, *Voice over Internet Protocol* (juga disebut VoIP, *IP Telephony*, *Internet telephony* atau *Digital Phone*) adalah teknologi yang memungkinkan percakapan suara jarak jauh melalui media internet. Data suara diubah menjadi kode digital dan dialirkan melalui jaringan yang mengirimkan paket-paket data, dan bukan lewat sirkuit analog telepon biasa [3]. Menurut Anton Raharja, VoIP adalah teknologi yang memanfaatkan *Internet Protocol* untuk menyediakan komunikasi suara secara elektronik dan *real-time* [4].

Suatu sistem VoIP akan mendijitalisasi suara menggunakan suatu teknik kompresi audio, membagi suara terdijitalisasi ke dalam paket, dan mengirim paket melalui jaringan IP ke suatu tujuan. Seluruh paket dirutekan tanpa jaminan bahwa paket tersebut akan melewati jalur yang sama. Selama transmisi, paket-paket tersebut kemungkinan mengalami *delay*, *lost*, maupun *error*. Setelah paket-paket ditransmisikan dan tiba di tujuan, paket-paket tersebut disatukan kembali dan didekompresi untuk membentuk data ke bentuk aslinya.

Semua kemampuan yang ada pada telepon konvensional juga terdapat pada VoIP bahkan dengan beberapa fitur-fitur tambahan lain seperti *voicemail*, *instant messaging*, *presence*, dan *video conference*. Bentuk nomor telepon pada VoIP dapat berupa beberapa digit nomor seperti di PSTN, alamat-alamat, atau suatu alias. Pada PSTN, digit nomor telepon digunakan langsung untuk menemukan lokasi telepon yang dituju. Sedangkan pada VoIP, nomor telepon tersebut ditranslasikan terlebih dahulu ke alamat IP sebanyak 32 *bit* atau 128 *bit* yang baru kemudian akan disambungkan dengan tujuan.

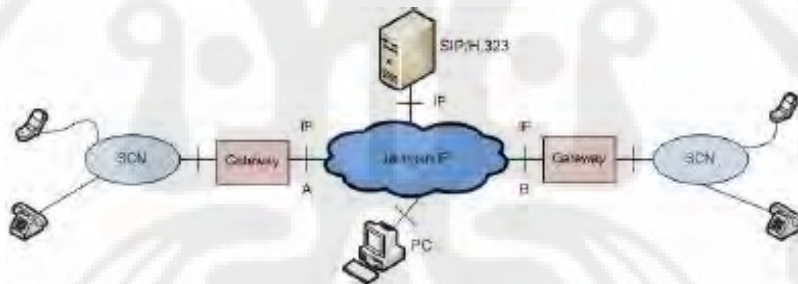
*User* atau *end-point* pada VoIP dapat berupa *hardphone* yang bentuknya seperti telepon biasa seperti *IP Phone* maupun telepon analog atau berupa *softphone* yang diinstal pada PC (*personal computer*). Oleh karena itu terdapat empat mode

komunikasi dalam VoIP yaitu *Phone to Phone*, *Phone to PC*, *PC to Phone*, dan *PC to PC*. Gambar 2.1 berikut adalah gambaran umum dari jaringan VoIP.



Gambar 2.1. Arsitektur jaringan VoIP

Pada Gambar 2.1 di atas, terlihat bahwa komputer maupun telepon analog dapat terhubung dengan jaringan VoIP. Untuk komputer dan IP *phone*, tidak harus terhubung dengan *Broad Voice Router*, yang penting adalah memiliki akses ke jaringan internet. Sedangkan untuk menyambungkan telepon biasa dengan jaringan VoIP diperlukan alat khusus yang disebut ATA (*Analog Telephone Adapter*). Sementara itu, agar VoIP bisa terhubung dengan *user* di jaringan lain seperti PSTN, maka VoIP perlu disambungkan ke *gateway*.



Gambar 2.2 Jaringan non-VoIP yang dihubungkan dengan jaringan IP

Dari Gambar 2.2 terlihat bahwa untuk mengakses VoIP dapat disambungkan langsung dengan jaringan IP jika menggunakan *softphone* pada PC, sedangkan jika telepon berada pada jaringan SCN (*Switched Communication Network*) yang dapat berupa jaringan kabel atau nirkabel, seperti PSTN, ISDN, atau GSM, maka harus dihubungkan dengan *gateway* terlebih dahulu.

Seperti teknologi lainnya, VoIP memiliki beberapa kelebihan dan juga kekurangan. Dengan bertelepon menggunakan VoIP, banyak keuntungan yang dapat diambil antara lain [3]:

- Biaya lebih rendah untuk sambungan langsung jarak jauh. Penekanan utama dari VoIP adalah biaya. Dengan dua lokasi yang terhubung dengan internet maka biaya percakapan menjadi sangat rendah.
- Memanfaatkan infrastruktur jaringan data yang sudah ada untuk suara. Berguna jika perusahaan sudah mempunyai jaringan. Jika memungkinkan jaringan yang ada bisa dibangun jaringan VoIP dengan mudah. Tidak diperlukan tambahan biaya bulanan untuk penambahan komunikasi suara.
- Penggunaan *bandwidth* yang lebih kecil daripada telepon biasa. Dengan majunya teknologi penggunaan *bandwidth* untuk *voice* sekarang ini menjadi sangat kecil. Teknik pemampatan data memungkinkan suara hanya membutuhkan sekitar 8kbps *bandwidth*.
- Memungkinkan digabung dengan jaringan telepon lokal yang sudah ada. Dengan adanya *gateway* bentuk jaringan VoIP bisa disambungkan dengan PABX (*Private Automatic Branch Exchange*) yang ada di kantor. Komunikasi antar kantor bisa menggunakan pesawat telepon biasa
- Berbagai bentuk jaringan VoIP bisa digabungkan menjadi jaringan yang besar. Contoh di Indonesia adalah VoIP Rakyat.
- Variasi penggunaan peralatan yang ada, misal dari PC sambung ke telepon biasa, *IP phone handset*

Sedangkan beberapa kelemahan VoIP antara lain adalah:

- Kualitas suara tidak sejernih PSTN sebagai efek dari kompresi suara dengan *bandwidth* kecil maka akan ada penurunan kualitas suara dibandingkan jaringan PSTN konvensional. Namun jika *koneksi* internet yang digunakan adalah *koneksi* internet pita-lebar atau broadband seperti Telkom Speedy, maka kualitas suara akan jernih bahkan lebih jernih dari sambungan PSTN dan tidak terputus-putus.
- Ada jeda dalam berkomunikasi. Proses perubahan data menjadi suara, jeda jaringan, membuat adanya jeda dalam komunikasi dengan menggunakan VoIP kecuali jika menggunakan *koneksi* Broadband.
- Belum ada jaminan kualitas jika VoIP melewati internet.
- Peralatan relatif mahal. Peralatan VoIP yang menghubungkan antara VoIP dengan PABX relatif berharga mahal. Diharapkan dengan makin

populernya VoIP ini maka harga peralatan tersebut juga mulai turun harganya.

- Berpotensi menyebabkan jaringan terhambat jika pemakaian VoIP semakin banyak, maka ada potensi jaringan data yang ada menjadi penuh jika tidak diatur dengan baik. Pengaturan *bandwidth* adalah perlu agar jaringan di perusahaan tidak menjadi jenuh akibat pemakaian VoIP.
- Penggabungan jaringan tanpa dikoordinasi dengan baik akan menimbulkan kekacauan dalam sistem penomoran.

Secara ringkas, berikut adalah beberapa komponen penyusun jaringan VoIP [4]:

- *User Agent*
  - *Softphone* (Idefisk, SJphone, X-Lite, Netmeeting)
  - *Hardphone* (IP Phone, USB Phone, ATA, ITG)
- *Proxy*
  - Softswitch (Asterisk, OpenSER, SER, Yate, Gnugk)
- *Protocol*
  - *Signaling Protocol* (SIP, H.323, IAX2, Skinny, MGCP, Megaco)
  - *Media Transfer Protocol* (RTP)
  - *Codec* (Encoder-Decoder)
    - Standar industri (G.711, G.723.1, G.729)
    - Open Source* (GSM 06.10, iLBC, Speex)

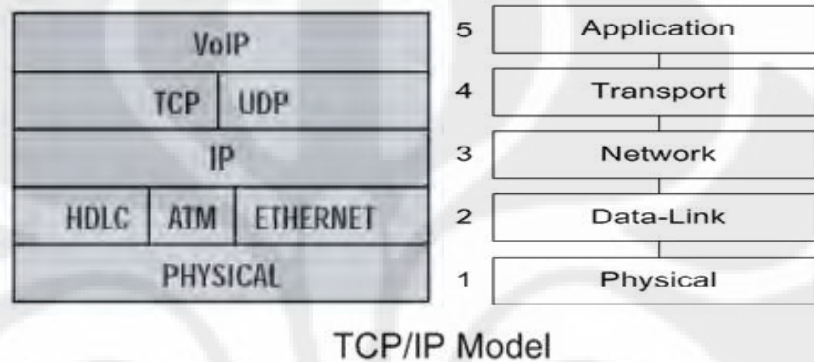
### 2.1.2 Protokol pada VoIP

Protokol pada *Voice over IP* (VoIP) secara umum dibagi 2 bagian, yaitu kontrol (*signaling*) dan data *voice*. Kontrol VoIP adalah trafik yang berfungsi untuk menghubungkan dan menjaga trafik yang sebenarnya yaitu berupa data *voice*. Selain itu, kontrol VoIP menjaga seluruh operasi jaringan (*router to router communications*). Kontrol VoIP dikenal dengan istilah *Packet Signaling* dan protokol *signaling* yang biasa digunakan adalah SIP dan H.323. Sedangkan data *voice* adalah trafik *user* berupa informasi yang disampaikan *end-to-end* yang dikenal juga sebagai *Packet Voice*.

Pada model *layer TCP/IP* (*Transmission Control Protocol/ Internet Protocol*), VoIP berada di *layer* aplikasi dengan menggunakan protokol SIP yang



dikeluarkan oleh IETF (*Internet Engineering Task Force*) atau sistem H.323 yang dikeluarkan oleh ITU-T (*International Telecommunication Union-Telecommunication Standardization Sector*) sebagai *signaling* dan menggunakan RTP (*Real Time Protocol*) untuk membawa paket suara. Selain itu VoIP juga menggunakan beberapa protokol yang ada di *layer* bawah aplikasi yaitu TCP dan UDP (*User Datagram Protocol*) pada *layer* transport serta IP pada *layer* internet. Gambar 2.3 berikut menunjukkan protokol-protokol pada aplikasi VoIP.



Gambar 2.3 Perbandingan protokol VoIP di *layer* TCP/IP

Selanjutnya akan dibahas beberapa protokol yang dipakai dalam VoIP mulai dari *layer* bawah ke atas yaitu IP, TCP, UDP, dan RTP. Sedangkan SIP akan dibahas dibagian tersendiri.

#### 2.1.2.1 Internet Protocol (IP)

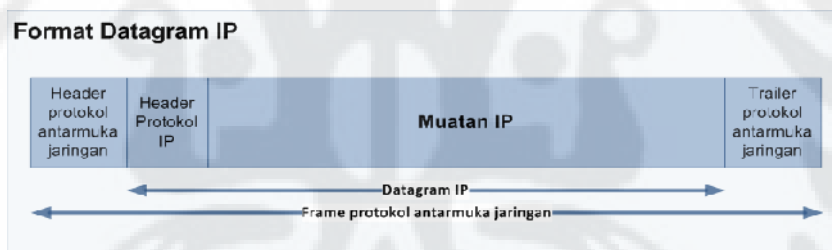
*Internet Protocol* adalah protokol lapisan jaringan (*network layer* dalam *OSI Reference Model*) atau protokol lapisan *internetwork* (*internetwork layer* dalam *DARPA Reference Model*) yang digunakan oleh protokol TCP/IP untuk melakukan pengalamatan dan *routing* paket data antar *host-host* di jaringan komputer berbasis TCP/IP [5]. Versi IP yang banyak digunakan adalah IP versi 4 (IPv4) yang didefinisikan pada RFC (*Request for Comments*) 791 dan dipublikasikan pada tahun 1981, tetapi akan digantikan oleh IP versi 6 pada beberapa waktu yang akan datang. *Internet Protocol* didesain untuk menghubungkan komunikasi komputer pada jaringan *packet-switched*. IP menyediakan pengiriman data yang bersifat *connectionless* dan *best effort*. *Connectionless* berarti tidak ada pembentukan hubungan antara satu titik dengan

titik lain sebelum proses pengiriman data. *Best effort* berarti sedapat mungkin IP akan mengirimkan data ketujuan, tetapi IP tidak menjamin data akan benar-benar sampai ke tujuan. Pada VoIP, IP digunakan untuk membungkus segmen dari *layer* atasnya yaitu TCP dan UDP.

Paket-paket data dalam protokol IP dikirimkan dalam bentuk datagram. Sebuah datagram IP terdiri atas *header* IP dan muatan IP (*payload*), sebagai berikut [5]:

- *Header* IP: Ukuran *header* IP bervariasi, yakni berukuran 20 hingga 60 byte, dalam penambahan 4-byte. *Header* IP menyediakan dukungan untuk memetakan jaringan (*routing*), identifikasi muatan IP, ukuran *header* IP dan datagram IP, dukungan fragmentasi, dan juga IP Options.
- Muatan IP: Ukuran muatan IP juga bervariasi, yang berkisar dari 8 byte hingga 65515 byte.

Sebelum dikirimkan di dalam saluran jaringan, datagram IP akan "dibungkus" dengan *header* protokol lapisan antarmuka jaringan dan *trailer*-nya, untuk membuat sebuah *frame* jaringan. Format datagram IP dapat dilihat di Gambar 2.4.

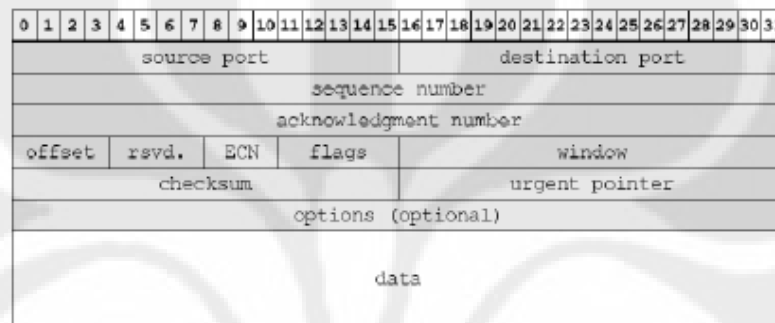


Gambar 2.4 Format Datagram IP

### 2.1.2.2 TCP (*Transmission Control Protocol*)

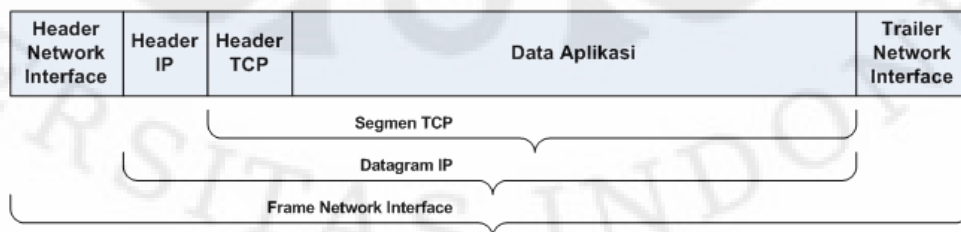
TCP adalah suatu protokol yang berada di lapisan transpor yang berorientasi sambungan (*connection-oriented*) yang berarti menjaga reliabilitas hubungan komunikasi *end-to-end* [6]. Konsep dasar kerja TCP adalah mengirim dan menerima informasi yang berbentuk segmen-segmen informasi dengan panjang data bervariasi pada suatu datagram internet. Selama transmisi, TCP membangun koneksi yang kuat antara *host* pengirim dan *host* penerima. TCP menjamin realibilitas hubungan komunikasi karena melakukan perbaikan terhadap data yang rusak, hilang atau kesalahan kirim. Hal ini dilakukan dengan memberikan nomor urut pada setiap paket yang dikirimkan dan membutuhkan sinyal jawaban positif dari penerima berupa sinyal ACK (*acknowledgment*). Jika sinyal ACK ini tidak diterima pada interval

waktu tertentu, maka data akan dikirimkan kembali. 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 paket data yang masih boleh ditransmisikan pada setiap segmen yang diterima dengan sukses. Karena sifatnya ini, dalam VoIP, TCP digunakan pada saat *signaling* yaitu untuk membawa (*transport*) protokol SIP. TCP digunakan untuk menjamin *setup* suatu *Call* pada sesi *signaling*. Gambar 2.5 berikut menunjukkan struktur paket TCP.



Gambar 2.5 Struktur Paket TCP

Segmen-segmen TCP akan dikirimkan sebagai datagram-datagram IP (datagram merupakan satuan *protocol data unit* pada lapisan *internetwork*). Sebuah segmen TCP terdiri atas sebuah *header* dan segmen data (*payload*), yang dienkapsulasi dengan menggunakan *header IP* dari protokol IP. Proses enkapsulasi data protokol TCP/IP: Data aplikasi + *header TCP* + *header IP* + *header network interface* (*Ethernet, Token Ring, dll*) + *trailer network interface*. Sebuah segmen dapat berukuran hingga 65495 byte:  $2^{16}$ -(ukuran *header IP* terkecil (20 byte)+ukuran *header TCP* terkecil (20 byte)). *Datagram IP* tersebut akan dienkapsulasi lagi dengan menggunakan *header protokol network interface* (lapisan pertama dalam *TCP/IP Reference Model*) menjadi *frame* lapisan *Network Interface* [7]. Gambar 2.6 mengilustrasikan data yang dikirimkan ke sebuah *host*.



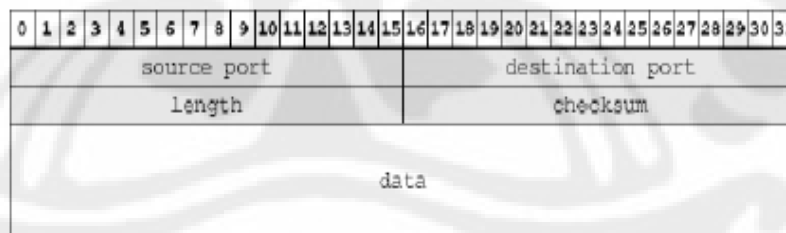
Gambar 2.6 Format data dan header pada model TCP/P

### 2.1.2.3 UDP (*User Datagram Protocol*)

UDP merupakan salah satu protokol pada *layer transport*. UDP mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara *host* dalam jaringan yang menggunakan TCP/IP [8]. UDP pada VoIP digunakan untuk mengirimkan *audio stream* yang dikirimkan secara terus-menerus.

VoIP menggunakan UDP untuk membawa data *voice* RTP karena pada pengiriman *audio streaming* yang berlangsung terus-menerus lebih mementingkan kecepatan pengiriman data agar tiba di tujuan tanpa memperhatikan apakah semua paket berhasil sampai atau ada beberapa yang hilang. Data dikirimkan dari *host* pengirim ke *host* penerima tanpa memandang hambatan dan kemampuan penerima. UDP tidak pernah melakukan pengiriman ulang, sehingga penerima tidak akan pernah menerima paket ketika terjadi kerusakan atau *packet lost* pada jalur antara pengirim dan penerima.

Pada aplikasi *streaming*, pengiriman ulang paket tidaklah bermanfaat karena adanya pembatasan waktu oleh aplikasi *streaming*. Paket-paket yang memuat data *streaming* dan telah dikirimkan dua kali tidak akan tiba pada waktu yang sama. Pengiriman ulang bukan hanya tidak berguna tetapi juga dapat mengakibatkan hal yang berbahaya bagi unjuk kerja aplikasi *streaming*. Gambar 2.7 menunjukkan struktur paket UDP.

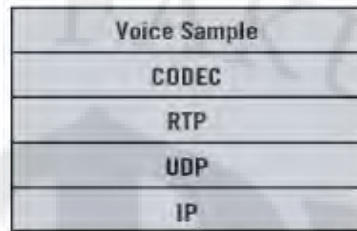


Gambar 2.7 Struktur paket UDP

### 2.1.2.4 RTP (*Real Time Transport Protocol*)

RTP merupakan format standarisasi paket untuk pengiriman suara dan video melalui jaringan internet [9]. RTP merupakan protokol pada lapisan aplikasi yang berperan mentransmisikan data *real time*. Besarnya ukuran paket tergantung pada penggunaan *codec*. RTP digunakan berpasangan dengan TCP atau UDP. Namun RTP biasa digunakan berpasangan dengan UDP. Hal ini bertujuan dalam penggunaannya untuk *streaming* data yang bersifat *real time*. Oleh karena itu,

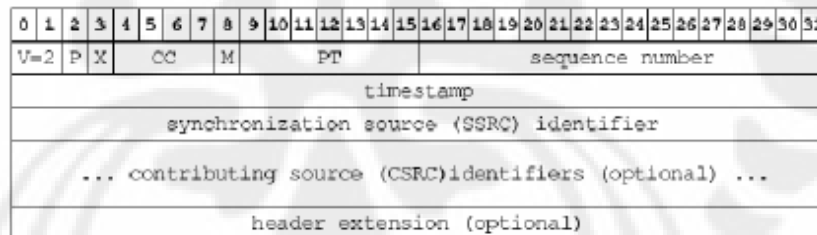
protokol ini tidak menjamin adanya mekanisme QoS untuk pengiriman *real time*. Gambar 2.8 berikut memperlihatkan susunan *protocol* RTP pada *layer* TCP/IP.



Gambar 2.8 Kedudukan protokol RTP di antara protokol lain dalam VoIP

Informasi RTP dienkapsulasi dalam packet UDP. Jika paket RTP hilang atau didrop di jaringan, maka RTP tidak akan melakukan *retransmission* (sesuai standard *protocol* UDP). Hal ini agar *user* tidak terlalu lama menunggu (*long pause*) atau *delay*, dikarenakan permintaan *retransmission*. Namun pada RTP terdapat aplikasi *real time* dengan layanan pengiriman *end to end* seperti *payload type identification* dan *delivery monitoring*. Pesan RTP juga membolehkan aplikasi mendeteksi paket hilang, duplikasi paket, transport *delay* dan paket tiba tidak secara berurutan.

Sebuah pesan RTP mengandung RTP *header* yang diikuti dengan RTP *payload*. Pesan RTP versi 2 ditunjukkan di Gambar 2.9



Gambar 2.9 Header RTP

Gambar 2.3 menunjukkan *header* RTP yang memiliki struktur sebagai berikut [6]:

- *Version (V)*: *Field* ini merupakan versi dari RTP.
- *Padding (P)*: *Field* ini digunakan jika *media stream* di-enkripsi.
- *Extension (X)*: *Field* ini merupakan ekstensi tambahan yang mengikuti *header* yang dibuat oleh tipe *payload* tertentu.
- *CSCR count (CC)*: *Field* ini memuat nomor CSRC.
- *Marker (M)*: berfungsi menandai awal dari *frame* baru pada video.

- *Payload Type (PT)*: merupakan *7-bit field* yang menandai *codec* yang digunakan.
- *Sequence Number*: berukuran *16 bit* yang berfungsi untuk mendeteksi hilangnya paket karena jumlah *sequence number* akan bertambah untuk setiap paket RTP yang dikirim.
- *Timestamps*: berupa *32 bit* yang mengindikasikan waktu relatif ketika *payload* di-sample.
- *Synchronization Source Identifier (SSRCI)*: mengidentifikasi *sender* dari paket RTP.
- *CSCR Contributing Source Identifier*: *field* ini hanya ada jika paket RTP telah dikirim oleh *mixer*.

#### **2.1.2.5 RTCP (Real Time Control Protocol)**

RTCP berperan mengontrol paket-paket secara berkala ke semua bagian yang terlibat dalam sebuah sesi. Protokol ini memungkinkan *endpoint* mengatur *call* secara *real time*. Hal ini bertujuan untuk meningkatkan kualitas suara dan memberikan *feedback* pada kualitas distribusi data. Protokol ini memungkinkan partisipan pada suatu sesi RTP saling mengirimkan laporan kualitas. Fungsi laporan tersebut adalah mengetahui kualitas dari *koneksi* yang dibuat termasuk informasi seperti jumlah paket yang dikirim dan diterima, jumlah paket yang hilang dan *jitter* dari paket.

## **2.2 SESSION INITIATION PROTOCOL(SIP)**

### **2.2.1 Pengantar SIP**

SIP adalah protokol kontrol multimedia *layer* aplikasi dan distandardisasi di bawah IETF RFC 2543. Selain SIP, sebenarnya terdapat beberapa lagi *protocol* signaling pada VoIP seperti H.323 yang dikeluarkan oleh ITU-T dan Skinny yang merupakan *proprietary* dari Cisco. Namun, pada saat ini yang dianggap paling berkembang dan juga dijadikan standar untuk pengembangan NGN adalah SIP. Alasan utama mengapa SIP lebih berkembang adalah sifatnya yang sederhana berupa protokol persinyalan yang ramping dan

berbasis teks, yang dioptimalkan agar mudah dikembangkan bersama aplikasi-aplikasi internet

Sama seperti kebanyakan protokol *World Wide Web* (WWW), SIP memiliki sintaks berbasis *American Standard Code for Information Interchange* (ASCII) yang sangat mirip *HyperText Transfer Protocol* (HTTP). SIP dapat membuat, memodifikasi, dan mengakhiri sesi multimedia seperti konferensi multimedia, panggilan telepon internet, dan aplikasi-aplikasi yang serupa. Protokol SIP didukung oleh beberapa protokol antara lain, antara lain RSVP (*Resource Reservation Protocol*) untuk melakukan pemesanan pada jaringan, RTP (*Real time Transport Protocol*) dan RTCP (*Real time Transport Control Protocol*) untuk melakukan transmisi media dan mengetahui kualitas layanan, serta SDP (*Session Description Protocol*) untuk mendeskripsikan sesi media. SIP menggunakan *protocol* UDP port 5060, sedangkan RTP menggunakan *protocol* UDP pada port dinamis (disarankan untuk menggunakan port antara 8000-20000).

SIP terdiri atas 5 segi pembuatan dan penghentian komunikasi multimedia, yaitu [10]:

- *User location*: mekanisme untuk menentukan sistem terakhir yang digunakan untuk komunikasi.
- *User capabilities*: mekanisme untuk menentukan media dan parameter media yang digunakan.
- *User availability*: mekanisme untuk menentukan jika pihak terpanggil sedang akan ikut serta di dalam komunikasi.
- *Call setup*: pembuatan parameter panggilan, baik di sisi pemanggil dan pihak terpanggil.
- *Call handling*: kemampuan untuk mengatur panggilan sisipan dan kontrol panggilan pihak ketiga seperti *call transfer* dan *call waiting*, setelah panggilan awal sudah diatur, dan untuk mengatur penghentian panggilan-panggilan ini.

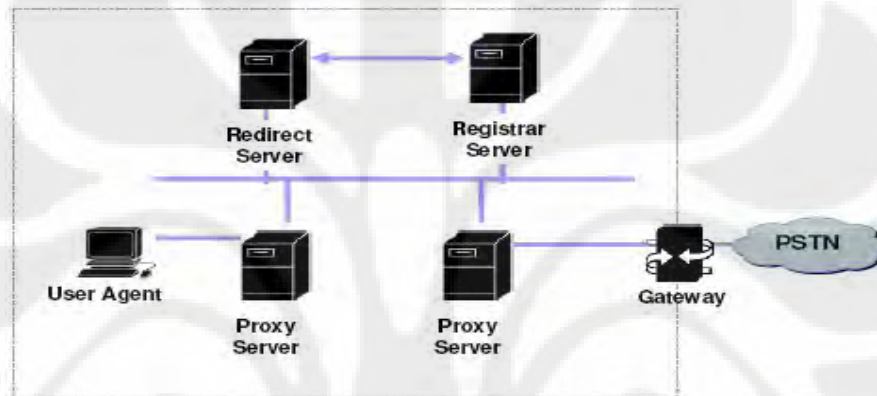
Dari penjelasan di atas, dapat disimpulkan beberapa fungsi dari SIP yaitu [4]:

- *Call initiation*
  - Membangun sebuah sesi komunikasi
  - Negosiasi media transfer *protocol*
  - Mengundang *user agent* lain untuk bergabung di dalam sesi komunikasi

- *Call modification*
  - Bila perlu, SIP dapat memodifikasi sesi komunikasi
- *Call termination*
  - Menutupi sesi komunikasi

### 2.2.2 Arsitektur SIP

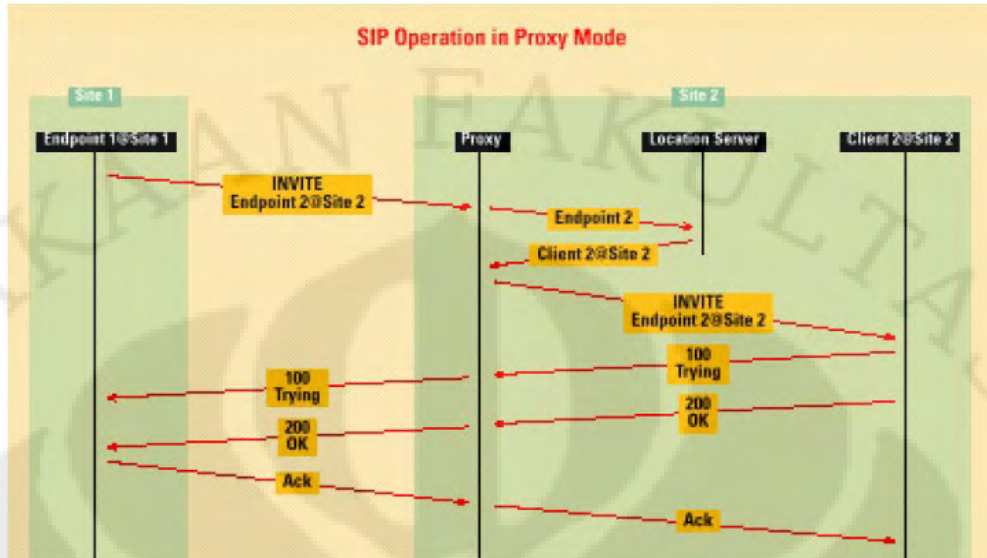
Arsitektur SIP terdiri dari 4 komponen yaitu: *user agent*, *proxy server*, *redirect server*, dan *registrat server*, yang dapat dilihat pada Gambar 2.10 berikut



Gambar 2.10 Arsitektur SIP

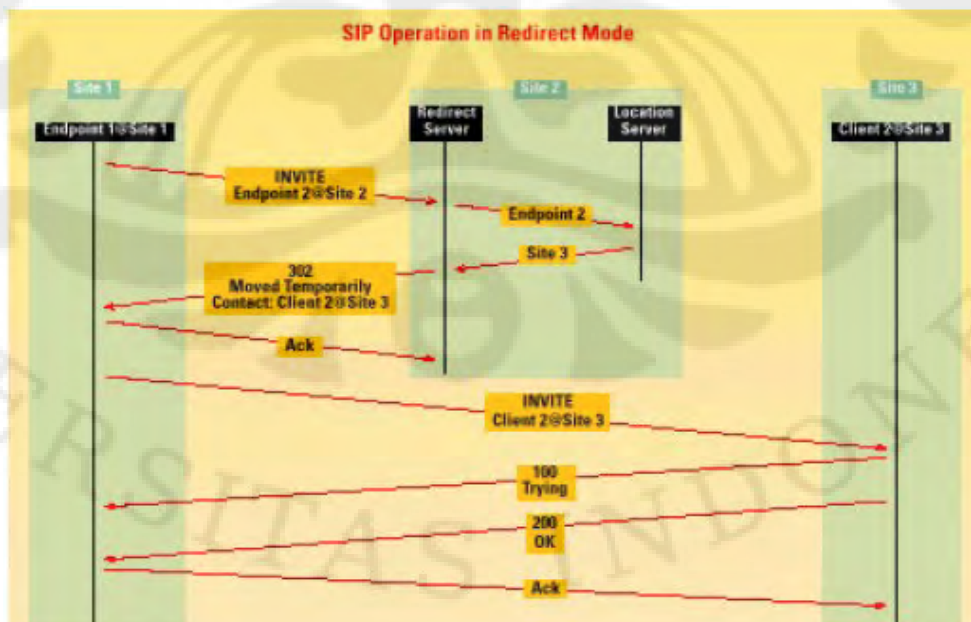
- **User Agent (UA):** adalah sebuah *internet endpoint*, seperti telepon atau PC, yang digunakan untuk membangun, memodifikasi, dan terminasi sebuah sesi. *User Agent* terdiri dari *User Agent Server (UAS)* dan *User Agent Client (UAC)*. UAC adalah aplikasi pemanggil yang memulai dan mengirimkan *SIP request* sedangkan UAS berperan dalam menerima dan menjawab *SIP request*. Baik UAC ataupun UAS dapat menutup sesi komunikasi. *User agent* ini dapat berupa *software (softphone)* ataupun hardware (*hardphone*)
- **Proxy Server:** Komponen penengah antar *user agent*, bertindak sebagai *server* yang menerima *request message* dari *user agent* dan menyampaikan pada *user agent* lainnya. Berikut pada Gambar 2.11 adalah *call flow* dengan menggunakan *SIP Proxy*. *Request* dapat dilayani sendiri atau disampaikan (*forward*) pada *proxy* lain atau *server* lain. Menerjemahkan dan/atau menulis ulang *request message* sebelum menyampaikan pada *user agent* tujuan atau *proxy* lain. *Proxy server* menyimpan state sesi komunikasi antara UAC dan UAS.





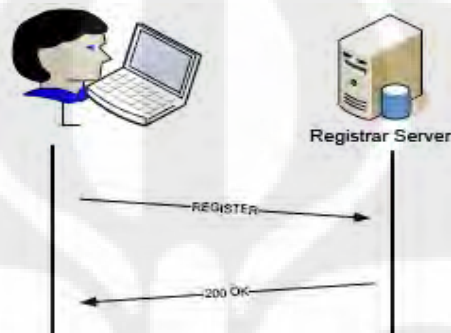
Gambar 2.11 Call flow menggunakan SIP Proxy

- **Redirect server:** Komponen yang menerima *request message* dari *user agent*, memetakan alamat SIP *user agent* atau *proxy* 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*. Tidak seperti UAS, *redirect server* tidak dapat menerima dan menutup sesi komunikasi. Pada Gambar 2.12 berikut adalah *call flow* dengan SIP *Redirect*.



Gambar 2.12 Call flow dengan menggunakan SIP Redirect

- **Registrar:** Komponen yang menerima *request message* register. Registrar dapat menambahkan fungsi otentikasi *user* untuk validasi. Registrar menyimpan database *user* untuk otentikasi dan lokasi sebenarnya (berupa IP dan port) agar *user* yang terdaftar dapat dihubungi oleh komponen SIP lainnya. Komponen ini biasa disandingkan dengan *Proxy Server*. Gambar 2.13 berikut menunjukkan proses registrasi pada SIP.



Gambar 2.13 Proses registrasi SIP

### 2.2.3 Nama dan Pengalamatan pada SIP

Setiap komponen SIP memiliki pengalamatan masing-masing. Sebagai contoh, setiap pengguna SIP teridentifikasi oleh sebuah SIP *Uniform Resource Identifier* (URI). Bentuk identifikasi URI yang terdeskripsikan dengan penamaan hampir sama dengan bentuk pengalamatan sebuah *email* yang terdiri atas *user name* dan *host name*. sebagai contoh, seorang SIP *user* bernama *user1* yang terdaftar di *host* 152.118.148.231, maka *user1* memiliki SIP URI:

```
sip:user1@152.118.148.231
```

dengan 152.118.148.231 sebagai *domain* SIP *server*. Sebuah *secure* SIP URI yang biasa disebut SIPS URI menandakan sebuah cara yang aman dan ter-enkripsi untuk mengirimkan SIP *message*. Format dari SIPS URI sama dengan SIP URI akan tetapi bagian sip diganti dengan sips. Selain itu itu, SIP URI dapat mengandung *password user*. Secara umum, bentuk SIP URI memiliki format:

```
sip:user:password@hostname:port
```

Bagian pertama bisa berupa sip: atau sips:. Berikutnya adalah *userinfo* yang terdiri atas *username* dan *password* dengan format *user:password@hostname*. Sebuah URI dikirim dalam format *cleartext* sehingga walaupun dibolehkan, kehadiran *password* pada URI tidak dianjurkan. Satu hal

yang harus diperhatikan adalah bagian *userinfo* (*user* dan *password*) bersifat *case sensitive*, sedangkan bagian lain tidak kecuali ada pengaturan lain [11].

## 2.2.4 SIP Messages

SIP *message* terdiri dari 3 jenis yaitu *request*, *response* dan *acknowledgement*. Berikut adalah penjelasan dari tiap SIP *messages*.

### 2.2.4.1 Request Messages

Sebuah *request message* dikirim dari *client* (UAC) ke *server* (UAS) [11] dan ditulis dalam *cleartext*. Oleh karena itu, SIP merupakan *text-based protocol*. Berikut adalah beberapa jenis *request message* yang didefinisikan pada SIP [4]:

- *INVITE*: digunakan oleh *user* untuk membangun sebuah sesi SIP yaitu mengundang *user agent* lain untuk bergabung dalam sesi komunikasi
- *ACK*: digunakan untuk konfirmasi bahwa *user agent* telah menerima pesan terakhir dari serangkaian pesan *INVITE*.
- *CANCEL*: digunakan untuk membatalkan sebuah SIP *request* atau membatalkan *INVITE*.
- *OPTIONS*: digunakan untuk menanyakan kemampuan *server*.
- *REGISTER*: digunakan oleh *user* untuk Registrasi di Registrar *Server*
- *INFO*: digunakan untuk membawa sesi yang berhubungan dengan kontrol informasi atau untuk membawa membawa pesan informasi lainnya, seperti informasi inline DTMF.

Bentuk *message* pada SIP terdiri dari beberapa bagian yaitu sebuah *start-line*, satu atau lebih *header*, sebuah baris kosong dan *message body* (*optional*) [11]. *Start-line* digunakan untuk membedakan antara *request message* dan *response message*. *Request message* ditandai dengan adanya *request-line* yang berisi nama metode, SIP URI, dan versi protokol SIP. *Header field* digunakan untuk membawa informasi yang dibutuhkan untuk mengatur sebuah sesi SIP. Sebuah *header field* berisi nama *field* lalu diikuti titik dua (“:”) dan isi dari *field*. Beberapa *field* yang umum dipakai SIP antara lain seperti *To:*, *From:*, *Subject:*, *Via:*, *Contact:*, *Max-forward:*, *Cseq:*, *Call-ID:*, *User-Agent:*, dan lain sebagainya. Beberapa dari *header field* hanya digunakan pada *request message* dan beberapa

hanya digunakan untuk *response message*. Gambar 2.14 berikut adalah contoh sebuah *request message*.

```

INVITE sip:ken@ee.uottawa.ca SIP/2.0
Via: SIP/2.0/UDP gw11.uottawa.ca;branch=834E
maddr=137.128.16.254;nl=16
Via: SIP/2.0/UDP gw11.uottawa.ca
Record-Route: gw11.uottawa.ca
From: Bill Gate <sip:bill@microsoft.com>
To: Ken Chan <sip:ken@uottawa.ca>
Contact: Ken Chan <sip:ken@site.uottawa.ca>
Call-ID: 56258002159@site.uottawa.ca
CSeq: 1 INVITE
Subject: SIP will be discussed, too
Content-Type: application/sdp
Content-Length: 187

v=0
o=bill 32653763 2353687637 IN IP4 224.116.3.4
s=RTP Audio
i=Discussion of Net
c=IN IP4 224.2.0.1/127
t=0 0
m=audio 3456 RTP/AVP 0

```

Gambar 2.14 Contoh bentuk *request message* SIP

Suatu *Request-URI* menamakan tujuan permintaan yang sedang berlangsung. Secara umum, *Request-URI* memiliki nilai yang sama sebagai *header field* “To”, tetapi mungkin berbeda jika pemanggil diberikan alamat tersimpan (*cached*) yang menawarkan banyak jalur langsung ke pihak terpanggil melalui *field* “Contact”. *Header field* “From” dan “To” mengindikasikan alamat registrasi masing-masing dari pemanggil dan pihak terpanggil. *Header field* “Via” bersifat pilihan dan mengindikasi suatu jalur bahwa permintaan sudah berjalan jauh. Ini mencegah putaran (*looping*) permintaan dan memastikan balasan mendapat jalur yang sama seperti permintaan, yang membantu di dalam *firewall traversal* dan situasi *routing* yang tidak biasa lainnya. Hanya *proxy* yang melampirkan alamatnya sebagai nilai *header* “Via” ke pesan permintaan.

#### 2.2.4.2 Response Message

*Response message* pada SIP bentuknya berupa *status-code* yang terdiri dari 3 *digit integer* yang berfungsi untuk mengindikasikan respon dari sebuah *request*. Terdapat enam kelas SIP *response* dengan *digit* pertama dari *status-code* mengindikasikan kelas dari SIP *response* tersebut. Berikut ini adalah kelas-kelas dari SIP *response*:

- 1XX → *Informational Message*

Mengindikasikan sebuah *request* sedang dalam proses (100: *trying*, 180: *ringing*, 183: *progress*).

- 2XX → *Successful Response*

Mengindikasikan metode pada sebuah *request* telah diterima (200: OK)

- 3XX → *Redirection Response*

Tindakan lebih jauh perlu dilakukan oleh pengirim untuk melengkapi *request*-nya (302: *moved temporarily*, 305: menggunakan *proxy*).

- 4XX → *Request Failure Response*

*Request* memiliki kesalahan *syntax* sehingga tidak dapat dipenuhi (403: *forbidden*).

- 5XX → *Server Failure Resonse*

*Server* gagal memenuhi *request* yang *valid* (500: *server internal error*, 501: tidak diimplementasikan).

- 6XX → *Global Failures Response*

*Request* tidak dapat dipenuhi di semua *server* (606: *not acceptable*).

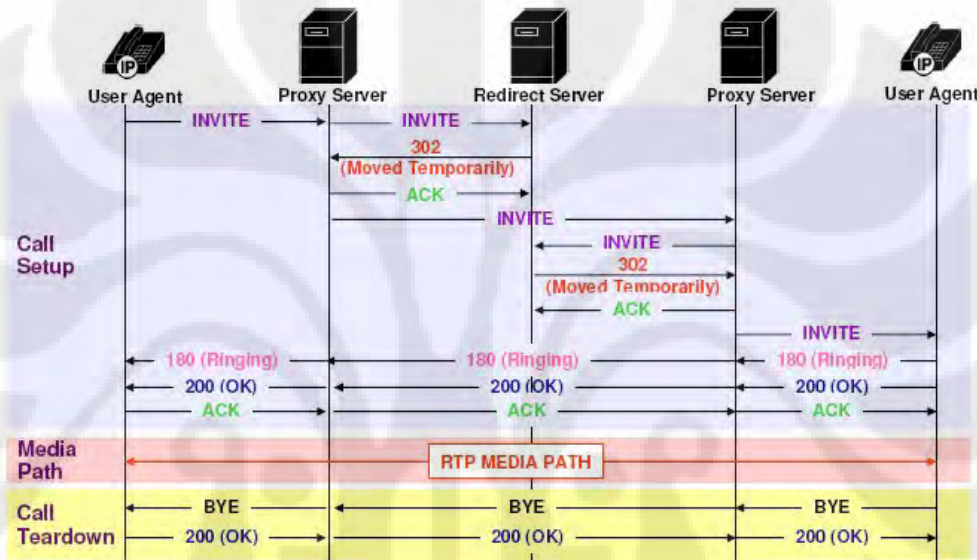
```
OK 200 SIP/2.0
Via: SIP/2.0/UDP gwj1.uottawa.ca;branch=8345
maddr=137.128.16.254;nl=16
Record-Route: gwj.es.uottawa.ca
From: Bill Gate <sip:bill@microsoft.com>
To: Ken Chan <sip:ken@uottawa.ca>
Contact: Ken Chan <sip:ken@site.uottawa.ca>
Call-ID: 56258002189@site.uottawa.ca
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Length: 187
```

Gambar 2.15 Response message SIP

Pada *response*, *message* ditandai oleh *Status line* yang berisi *Status -code* numerik dan penjelasan tekstualnya diikuti oleh versi protokol SIP. Ketika pesan respon yang sesuai tiba di *proxy*, *proxy* mungkin akan menghilangkan *header* “Via” terhubung dari *header* pesan respon. Permintaan “Record Route” dan *header field* respon adalah *field* pilihan dan ditambahkan ke suatu permintaan oleh *proxy* manapun di jalur permintaan selanjutnya untuk suatu *call leg* yang sama. Itu mengandung sebuah *Request URI* terjangkau secara global yang mengidentifikasi *proxy server*. “Call Id” merepresentasikan sebuah pengidentifikasi unik secara global untuk sesi panggilan saat ini. *Command Sequence* (CSeq) terdiri dari *transaction id* unik dan

metoda permintaan terhubung. Itu membolehkan *user agent* dan *proxy* melacak dan penyesuaian respon pesan terhubung ke suatu transaksi. *Header field* “Content Type” dan “Content Length” mengindikasikan tipe dari isi *body* pesan dan panjang *body* pesan yang diukur di dalam byte.

Pesan-pesan *request*, *response* dan *acknowledgment* (ack) ini semuanya digunakan untuk membangun sebuah sesi komunikasi SIP yang utuh. Gambar 2.16 berikut menunjukkan sebuah contoh sesi komunikasi menggunakan SIP lengkap dengan *message request*, *message response* dan *acknowledgement*.



Gambar 2.16 Contoh sesi komunikasi lengkap pada SIP

### 2.3 CODER-DECODER (CODEC)

*Codec* adalah sebuah *device* atau program komputer (*software*) yang dapat melakukan proses encoding dan atau decoding pada aliran data digital atau sinyal. Kata “*codec*” sendiri awalnya merupakan gabungan dua kata “*compressor-decompressor*”, namun sekarang lebih umum diartikan sebagai “*coder-encoder*”.

*Codec* yang digunakan pada aplikasi VoIP merupakan *speech encoding*, yaitu kompresi data audio digital yang mengandung *speech* (pembicaraan). *Speech encoding* ini sendiri termasuk jenis *audio codecs* pada kategori *lossy data compression* yang berarti data terkompresi ketika nantinya di-dekompresi akan sedikit berbeda dengan data asli tapi masih dianggap cukup merepresentasikan.

Untuk implementasi VoIP, banyak sekali jenis *speech encoding* yang dapat dipakai antara lain yang distandradisasi oleh ITU-T seperti G.711, G.723.1,

G.726, G.728, dan G.729 dan non-ITU seperti GSM, Speex dan iLBC. *Codec-codec* ini umumnya memiliki *bit rate* dan kualitas yang berbeda satu dengan yang lain. Besarnya *bit rate* biasanya mempengaruhi kerja prosesor, yaitu jika semakin kecil nilai *bit rate*, maka kerja prosesor akan semakin berat begitu juga sebaliknya. Namun, penggunaan *bit rate* yang kecil dibutuhkan untuk menghemat pengonsumsi *bandwidth*. Tabel 2.1 berikut, menunjukkan beberapa jenis *speech encoding* beserta *bit rate*-nya.

Tabel 2.1 Beberapa jenis *speech encoding* pada VoIP

Codec	Data bitrate (Kbps)	License required?
G.711	64 Kbps	No
G.726	16, 24, 32, or 40 Kbps	No
G.729A	8 Kbps	Yes (no for passthrough)
GSM	13 Kbps	No
iLBC	13.3 Kbps (30-ms frames) or 15.2 Kbps (20-ms frames)	No
Speex	Variable (between 2.15 and 22.4 Kbps)	No

Pada skripsi ini, ada dua buah *speech encoding* yang akan digunakan dalam jaringan VoIP yaitu G.711 dan GSM. Dari Tabel di atas, terlihat bahwa kedua *speech encoding* tersebut memiliki *bit rate* yang cukup jauh berbeda sehingga QoS antara keduanya dapat dibandingkan.

### 2.3.1 G.711 $\mu$ -Law

G.711 adalah *codec* standar internasional untuk kompresi audio yang dikeluarkan oleh ITU untuk digunakan pada teleponi digital seperti ISDN dan sambungan E1 atau T1. G.711 ini menggunakan teknik PCM (*Pulse Code Modulation*) dalam pengiriman suara dan mempunyai *bitrate* yang tinggi yaitu sebesar 64 kbps, berasal dari penggunaan kecepatan sampel sebesar 8000 sampel/detik dengan *coding 8 bit* untuk setiap sampel. *Codec* ini memiliki payload sebesar 160 byte dan waktu untuk satu frame adalah 20 ms. Sedangkan ukuran header totalnya adalah 40 byte yang berasal dari 20 byte *header IP*, 12 byte header UDP dan 8 byte header RTP.

G.711  $\mu$ -Law merupakan standar G.711 yang menggunakan algoritma  $\mu$ -Law untuk melakukan *companding* (*compression and expanding*). Algoritma *companding* dapat mereduksi rentang dinamik dari sebuah sinyal audio. Pada

sistem analog, algoritma *companding* dapat meningkatkan *signal-to-noise ratio* (SNR) yang diperoleh selama transmisi. Sedangkan pada domain digital, teknik ini dapat mengurangi eror akibat kuantisasi sehingga meningkatkan nilai *signal to quantization noise*. Dalam aplikasi VoIP, algoritma  $\mu$ -Law yang digunakan adalah versi digital terkuantisasi. Pada G.711 $\mu$ -Law, digunakan kompresi logaritmis yang meng-kompres tiap 16 *bit* menjadi 8 *bit* sehingga diperoleh rasio kompresi sebesar 1:2 [12]. *Codec* ini dapat digunakan secara bebas (*free*) pada aplikasi VoIP karena tidak ada biaya lisensi dan paling baik digunakan pada jaringan area lokal yang memiliki *bandwidth* cukup besar. Keuntungan lain dari penggunaan *codec* ini adalah implementasi yang sederhana sehingga tidak memerlukan daya CPU yang terlalu besar.

### 2.3.2 GSM 06.10

GSM 06.10 adalah jenis *codec* GSM yang memiliki kecepatan penuh atau disebut *Full Rate* (FR) yang dikeluarkan oleh ETSI. Standar ini mendefinisikan sebuah referensi konfigurasi untuk rantai transmisi suara terutama pada telekomunikasi digital selular. Input bagi *encoder* suara ini adalah signal PCM 13 *bit uniform*. *Encoder* ini memiliki 3 bagian utama yaitu: *Linear prediction analysis (short-term prediction)*, *Long-term prediction* dan *Excitation analysis* [13]. Distribusi *bit* dalam setiap *block* dibagi ke dalam tiga bagian utama tersebut dengan komposisi seperti pada Tabel 2.2 berikut.

Tabel 2.2 Distribusi *bit* yang digunakan pada *codec* GSM *Full Rate*

		Bits per 5 ms block	Bits per 20 ms block
<b>LPC filter</b>	8 parameters		36
<b>LTP filter</b>	Delay parameter	7	28
	Gain parameter	2	8
<b>Excitation signal</b>	Subsampling phase	2	8
	Maximum amplitude	6	24
	13 samples	39	156
<b>Total</b>			260 bits

GSM 06.10 merupakan *speech encoding* yang didasarkan pada algoritma RPE-LTP (*Regular Pulse Excitation-Long Term Prediction*). GSM 06.10 mempunyai *bit rate* sebesar 13 kbps dengan 160 sampel dan sampel *rate* 8 KHz. *Encoder* memproses *block* suara sebesar 20 ms yang tiap *block* berisi 260 *bit* sehingga dihasilkan kecepatan 13 Kbps ( $260 \text{ bits}/20 \text{ ms}=13.000 \text{ bits/s}= 13\text{kbits/s}$ ).



Ukuran *payload*-nya adalah 33 byte dengan ukuran header seperti G.711 yaitu 40 byte.

### 2.3.3 VAD (*Voice Activity Detection*)

Suara dalam percakapan biasanya berisi 35-50 persen *silence*. Pada jaringan suara pada zaman dulu, semua panggilan suara masih menggunakan *bandwidth* tetap sebesar 64 Kbps yang tidak bergantung pada perbandingan antara banyaknya percakapan dan *silence* [14]. Namun pada jaringan suara yang modern seperti VoIP, semua percakapan dan *silence* di-*packetized*. Dengan menggunakan *voice activity detection* (VAD), paket yang berisi *silence* dapat ditekan (*suppress*), sehingga memungkinkan percakapan melalui VoIP menjadi lebih efisien dalam penggunaan *bandwidth*.

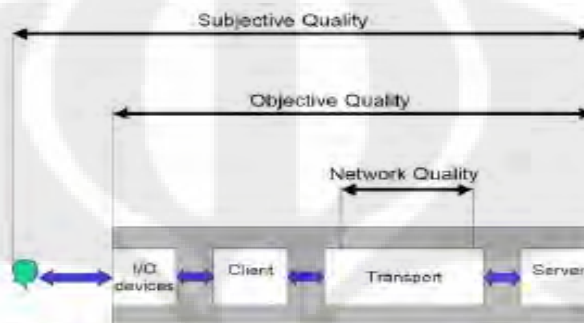
Sistem yang memungkinkan transmisi diskontinyu didasarkan pada algoritma VAD dan algoritma *Comfort Noise Generator* (CNG) yang memungkinkan penyisipan *noise* buatan saat tidak ada percakapan. Penyisipan *noise* ini cukup penting agar percakapan pada sistem yang menggunakan VAD tetap terasa seperti percakapan biasa. Hal ini disebabkan karena ketika menggunakan VAD, maka tidak akan ada paket yang ditransmisikan ketika tidak ada percakapan sehingga pendengar dapat mengira bahwa percakapan telah terputus dan akhirnya menutup sambungannya. Penggunaan CNG di sisi pendengar akan mencegah terjadinya hal yang tidak diinginkan seperti ini.

## 2.4 QUALITY OF SERVICE (QoS)

Sebagai jaringan yang didesain sejak awal sebagian jaringan komunikasi data, jaringan internet mempunyai karakteristik yang berbeda dibandingkan dengan jaringan telpon. Data-data yang mengalir di internet memperebutkan *bandwidth* yang ada. Kecepatan sampainya data tergantung pada beberapa hal seperti besarnya paket dan tingkat kepadatan *traffic* yang ada di dalam jaringan. Kondisi jaringan seperti ini berbeda dengan jaringan telepon biasa, dimana satu kanal hanya dikhususkan untuk satu pembicara telepon. Tidak terdapat perebutan *bandwidth* disana sehingga kualitas suara pun dapat terjaga.

Kualitas yang diberikan oleh suatu servis atau sering disebut juga sebagai *quality of service* (QoS) dapat diartikan sebagai tingkat performa yang

dibutuhkan suatu aplikasi agar berfungsi dengan baik. Pada VoIP, QoS merupakan tantangan yang timbul disebabkan sifat alami dari jaringan internet itu sendiri. Meskipun begitu, VoIP diharapkan sebisa mungkin dapat memiliki QoS yang setara atau mendekati QoS pada jaringan telepon konvensional (PSTN). Terdapat dua tipe QoS yaitu subjektif QoS dan objektif QoS [15]. Gambar 2.18 berikut menunjukkan dua tipe QoS.



Gambar 2.17 Subjektif vs Objektif QoS

### 2.5.1 Subjektif QoS

Subjektif QoS, berhubungan dengan tingkat kualitas yang berdasarkan perspektif masing-masing pengguna sehingga tiap pengguna mungkin memberikan nilai QoS yang berbeda untuk sebuah aplikasi yang sama. Teknik yang paling umum untuk mengukur subjektif QoS pada VoIP adalah dengan menggunakan MOS di mana dilakukan uji coba langsung penggunaan VoIP oleh beberapa *user* yang kemudian memberikan penilaian atas layanan yang dirasakan. Rentang nilai yang diberikan oleh *user* adalah 1 sampai dengan 5 dimana nilai 1 adalah yang terburuk dan nilai 5 untuk kualitas yang terbaik. Tabel 2.3 berikut menunjukkan urutan nilai MOS beserta kualitasnya.

Tabel 2.3 Nilai MOS dan kualitasnya

Nilai	Kualitas Suara
5	Baik sekali
4	Baik
3	Cukup
2	Kurang
1	Jelek

### 2.5.2 Objektif QoS

Sedangkan yang dimaksud dengan objektif QoS adalah QoS yang diukur secara konkrit dan kuantitatif. Tipe QoS yang digunakan pada skripsi ini adalah

tipe objektif QoS. Beberapa faktor dalam jaringan internet yang dapat mempengaruhi nilai objektif QoS pada VoIP yaitu:

#### 2.5.2.1 Packet loss

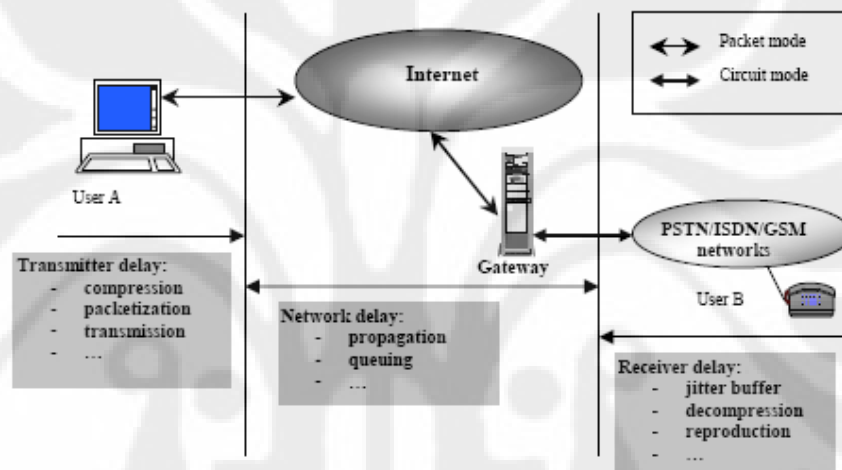
Merupakan hilangnya paket data yang sedang dikirimkan. Hilangnya data ini bisa disebabkan karena adanya permasalahan di perangkat-perangkat jaringan seperti *router* yang terlalu sibuk dan jalur komunikasi yang terlalu padat yaitu ketika terjadi *peak load* dan *congestion* (kemacetan transmisi akibat padatnya *traffic* yang harus dilayani) dalam batas waktu tertentu, sehingga *frame* (terdiri dari *header* dan *payload*) suara yang ditransmisikan akan dibuang sama seperti perlakuan terhadap *frame* data lainnya pada jaringan berbasis IP. Selain itu, hilangnya paket juga dapat disebabkan oleh habisnya *lifetime* paket (TTL = 0), *delay* di penerima yang lebih besar dari nilai *jitter buffer* dan paket yang invalid akibat kesalahan transmisi. Agar diperoleh kualitas suara yang masih dapat diterima maka paket yang hilang harus kurang dari 15% [1].

#### 2.5.2.2 Delay

*Delay* disebut juga sebagai waktu tunda merupakan waktu total yang diperlukan sebuah paket mulai dari ketika di proses di sisi pengirim sampai akhirnya diterima dan dapat didengarkan di sisi penerima. Dalam perancangan jaringan VoIP, *delay* merupakan suatu permasalahan yang harus diperhitungkan karena kualitas suara bagus tidaknya tergantung dari waktu *delay*. Besarnya *delay* maksimum yang direkomendasikan oleh ITU untuk aplikasi suara adalah 150 ms, sedangkan *delay* maksimum dengan kualitas suara yang masih dapat diterima pengguna adalah 300 ms sementara *delay* dalam jaringan internet berkisar antara 50 – 500 ms. *Delay end to end* adalah jumlah dari *transmitter delay* ditambah *network* (jaringan) *delay* pada saat waktu tertentu. Beberapa komponen dari *transmitter delay* yaitu [2]:

- Digitalisasi dan coding, waktu yang diperlukan kartu suara (*sound card*) untuk mendigitalisasi dan mengkodekan sebuah sinyal analog.
- *Processing delay*, *delay* yang terjadi saat proses *framing*, *coding*, *compression* dan sebaliknya.

- *Packetization delay*, *delay* yang terjadi saat proses pakettisasi *digital voice sample* (pembuatan *header* dan penyisipan data).
  - *Serialization delay*, *delay* pada saat proses peletakan *bit* ke dalam *circuit*
- Sedangkan komponen *network delay*, yaitu:
- *Propagation delay*, *delay* yang terjadi akibat transmisi melalui jarak antar pengirim dan penerima. Pada jaringan kabel, kecepatan propagasi adalah  $2 \times 10^8$  m/s
  - *Routing dan queuing delay*, *delay* akibat routing dan waktu tunggu antrian paket sampai dilayani.



Gambar 2.18 Macam-macam *delay* saat transmisi VoIP

### 2.5.2.3 Jitter

Merupakan variasi waktu tunda atau variasi *delay* yang terjadi akibat adanya selisih waktu atau interval antar-kedatangan paket di penerima. Untuk mengatasi *jitter* maka paket data yang datang di sisi penerima terlebih dahulu dikumpulkan dengan menggunakan mekanisme *jitter buffer* selama beberapa saat sampai paket tersusun dengan urutan yang benar di sisi penerima.

### 2.5.2.4 Bandwidth

*Bandwidth* yang besar merupakan hal utama yang harus disediakan untuk memperoleh QoS yang baik. Dengan alokasi *bandwidth* ini, setiap aliran paket data yang berisi suara mendapatkan jatah *bandwidth* yang tetap dan tidak perlu berkompetisi dengan paket data lain. *Bandwidth* sendiri adalah kecepatan

maksimum yang dapat digunakan untuk melakukan transmisi data antar komputer pada jaringan IP atau internet. Dalam perancangan VoIP, *bandwidth* merupakan suatu yang harus diperhitungkan agar dapat memenuhi kebutuhan pelanggan yang dapat digunakan menjadi parameter untuk menghitung jumlah peralatan yang dibutuhkan dalam suatu jaringan. Perhitungan ini juga sangat diperlukan dalam efisiensi jaringan dan biaya serta sebagai acuan pemenuhan kebutuhan untuk pengembangan di masa mendatang.

## BAB 3

### PERANCANGAN VOIP BERBASIS SIP MENGUNAKAN G.711 $\mu$ -LAW DAN GSM 06.10

Jaringan VoIP yang dibangun dalam penelitian skripsi ini adalah jaringan VoIP yang berbasis SIP. Pada jaringan ini digunakan sebuah PC dengan *software* Briker IPPBX yang merupakan salah satu distro Linux berbasis Ubuntu 8.04 Hardy Heron yang difungsikan sebagai *SIP proxy server* sekaligus registrar untuk proses autentikasi. Komputer *server* ini diletakkan di dalam *de-militerized zone* (DMZ) di kantor Pengembangan dan Pelayanan Sistem Informasi (PPSI) UI. Briker IPPBX ini merupakan gabungan dari beberapa *software* untuk membangun sebuah jaringan VoIP seperti Asterisk dan FreePBX serta beberapa *software* untuk menampilkan layanan berbasis web (*web-based*) seperti Apache, PHP dan MySQL. Selain itu, di dalam jaringan juga digunakan beberapa buah PC yang berfungsi sebagai *client* yang telah diinstal *softphone* sehingga dapat saling berkomunikasi seperti menggunakan telepon biasa melalui jaringan VoIP.

Sementara itu, terdapat beberapa *software* lain yang juga digunakan dalam penelitian skripsi ini yaitu untuk melakukan pengukuran objektif QoS digunakan *software* bernama Wireshark yang berfungsi untuk menangkap paket-paket VoIP dalam jaringan. Selain itu juga digunakan PuTTY untuk melakukan konfigurasi *server* jarak jauh dengan menggunakan SSH pada port 22 dan Vnc4server untuk *remote desktop* melalui port 5901. Briker, Wireshark, PuTTY dan Vnc4server yang digunakan dalam skripsi ini adalah *software* yang bersifat *opensource*.

### 3.1 KONFIGURASI SERVER

#### 3.1.1 Console

Proses pertama dalam konfigurasi *server* adalah menginstal *software* Briker pada sebuah *central processing unit* (CPU). Untuk mendapatkan *software* ini dapat mengunduh langsung *file* ISO-nya di alamat <http://www.briker.org>. Setelah *file* ISO di-unduh lalu bakar (*burn*) *file* ISO tersebut ke dalam CD dengan tujuan agar dapat di-booting dari CDROM untuk proses instalasi Briker IPPBX ke

dalam CPU. Untuk membakar *file* ISO Briker IPPBX, dapat digunakan program Nero (Windows) atau K3B (Linux) lalu memilih opsi “Burn Image to Disk”. Setelah proses *burning* ISO selesai, lalu install Briker dengan mengubah dahulu prioritas urutan booting komputer yang semula dari harddisk hda0 (ATA) atau sda0 (SATA) menjadi booting dari CDROM. Caranya, sesaat setelah komputer melakukan booting, tekan tombol “Delete” pada keyboard untuk masuk ke dalam konfigurasi BIOS (*Basic Input-Output System*). Tombol ini dapat berbeda-beda untuk masing-masing produsen motherboard komputer. Secara umum, konfigurasi prioritas urutan booting ada pada bagian Advanced Settings atau tab Boot. Masuk ke bagian tersebut, lalu ubah prioritas *booting* dari CDROM menjadi urutan yang pertama (First boot). Setelah itu, menyimpan konfigurasi ini dengan masuk ke bagian Exit dan pilih Exit and Save Settings. Setelah berhasil booting melalui CDROM maka akan muncul tampilan seperti pada Gambar 3.1 berikut ini.

```

Welcome to Briker 1.0.2 "OWP" installer menu.
- Please type 'install' and press Enter to install Briker
- Please type 'check' and press Enter to detect any defect on installer CD
- Please type 'rescue' and press Enter to enter rescue session
- Please type 'memtest' and press Enter for memory test
- Please type 'hd' and press Enter to boot from first harddisk

Getting started guide and user manuals available at http://www.briker.org

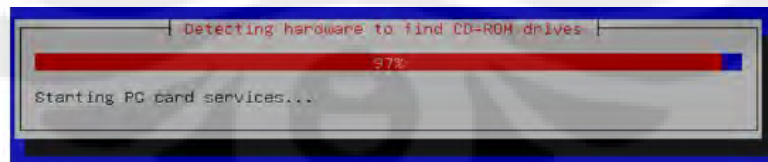
Information:
Please be informed that this installer will wipe out your disk drive.

boot: _

```

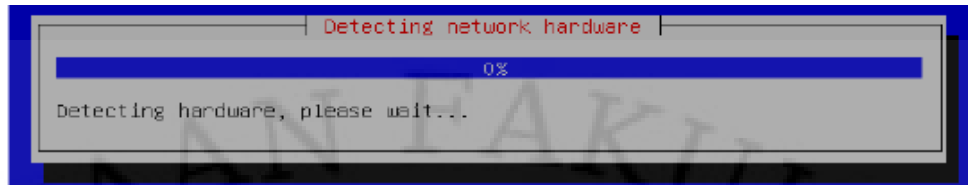
Gambar 3.1 Tampilan awal instalasi Briker IPPBX

Setelah muncul tampilan ini, lalu ketik “install” (tanpa tanda petik) pada cursor di depan kata “boot:”. Selanjutnya Briker otomatis memeriksa hardware yang terpasang dengan pertama kali memeriksa CDROM seperti pada Gambar 3.2 ini.



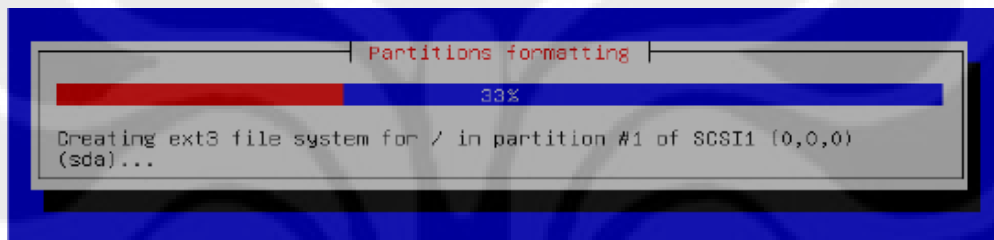
Gambar 3.2 Pemeriksaan drive CD-ROM

Briker lalu otomatis memeriksa perangkat keras jaringan, lalu mengkonfigurasi alamat IP secara otomatis seperti tampak di Gambar 3.3 ini.



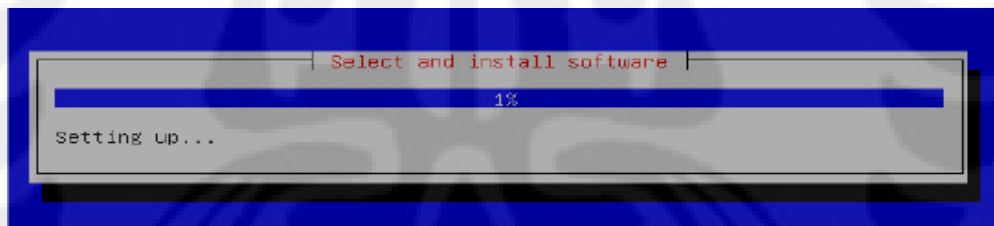
Gambar 3.3 Pemeriksaan hardware

Jika menggunakan perintah “install” maka dapat menginstal Briker tanpa ditunggu (*unattended*) namun Briker akan otomatis menghapus (*format*) hardisk dan menggunakan semua isi hardisk, sedangkan jika menggunakan pilihan perintah “wizad” maka kita harus menunggu proses instalasi karena akan muncul beberapa pertanyaan mengenai pembagian partisi hard disk yang ingin dibuat. Pada skripsi ini, instalasi Briker menggunakan pilihan “install”.

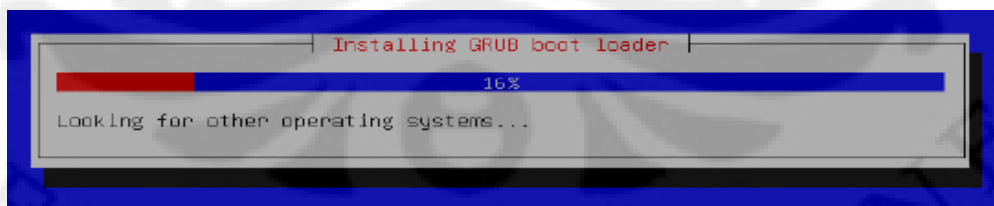


Gambar 3.4 Formatting hard disk

Briker lalu otomatis *install base system* dan *software* lainnya seperti Gambar 3.5

Gambar 3.5 Instalasi *software* aplikasi

Pada tahap terakhir, Briker akan menginstal *GRUB (Grand Unified Bootloader)*.



Gambar 3.6 Instalasi GRUB boot loader

Instalasi sistem selesai, CD Briker akan otomatis keluar dari CDROM dan komputer akan *restart*. Setelah *restart*, maka akan muncul tampilan seperti pada Gambar 3.7 berikut untuk melakukan *console login*.



```

uctel2xp.
uct1xxp.
uctc1xp.
uctdm24xxp.
ucfxo.
ystdm8xx.
ystdm16xx.
uctdn.
ucusb.
xpp_usb.
No hardware timing source found in /proc/zaptel, loading ztdummy
Running ztcfg: done.
 * Starting periodic command scheduler crond [ OK ]
 * Starting web server apache2 [ OK ]
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
 * Running local boot scripts (/etc/rc.local) [ OK ]
nohup: appending output to 'nohup.out' [ OK ]

Briker 1.0.2 "OWP" ippbx tty1
ippbx login: support
Password: _

```

Gambar 3.7 Console login di Briker

Setelah proses instalasi selesai, sistem akan membuat *password default* untuk *console login* dan *web login*, serta mengkonfigurasi alamat IP *default*.

- *Default console login* (SSH port 22):

*Username* : support

*Password* : Briker

- *Default web login* (HTTP port 80):

*Username* : administrator

*Password* : Briker

- *Alamat IP default*:

*IP address* : 192.168.2.2

*Subnet mask* : 255.255.255.0

Perintah-perintah pada *console login* hanya dapat dilakukan setelah melakukan otentikasi sebagai *user root*. Jalankan perintah berikut untuk otentikasi sebagai *user root*.

```
$ sudo su -
```

*Password* yang dimasukkan setelah perintah diatas adalah *password* yang sama dengan *user support* (*password default*). Demi keamanan, sebaiknya mengganti *password default console login* dengan cara menjalankan perintah sebagai berikut:

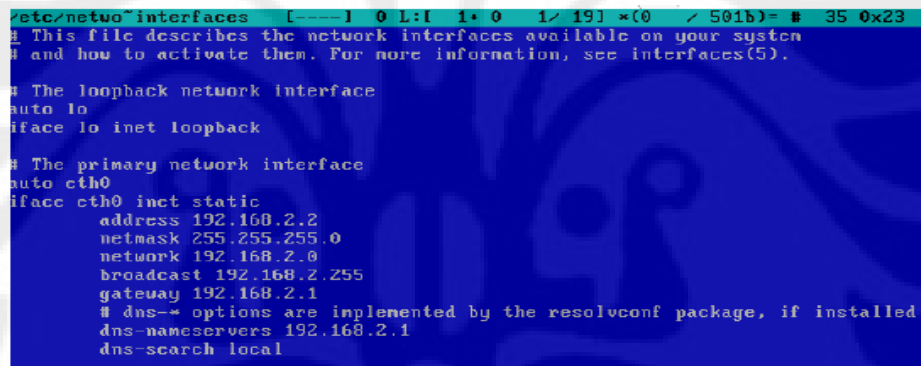
```
# passwd
```

Perlu diperhatikan bahwa jika sebelum *command* terdapat tanda “\$” maka berarti *user* sedang dalam posisi sebagai *user* biasa sedangkan jika yang muncul adalah tanda “#” maka berarti *user* telah berada dalam posisi sebagai *user root* dan dapat melakukan berbagai macam konfigurasi melalui *console*. *Console* atau terminal atau *shell* pada Linux adalah aplikasi tempat mengetikkan *command line interface* (CLI) untuk perintah-perintah konfigurasi.

Seperti telah dijelaskan sebelumnya, bahwa alamat IP *default* dari Briker adalah 192.168.2.2 dan ini kemudian diganti sesuai dengan IP yang diberikan oleh pihak PPSI untuk sever VoIP ini yaitu 152.18.148.231. Perintah untuk mengubah alamat IP dan informasi lain terkait dengan network address adalah sebagai berikut:

```
# mcedit /etc/network/interfaces
```

Tampilan awal yang muncul dari perintah ini ditunjukkan di Gambar 3.8 berikut.



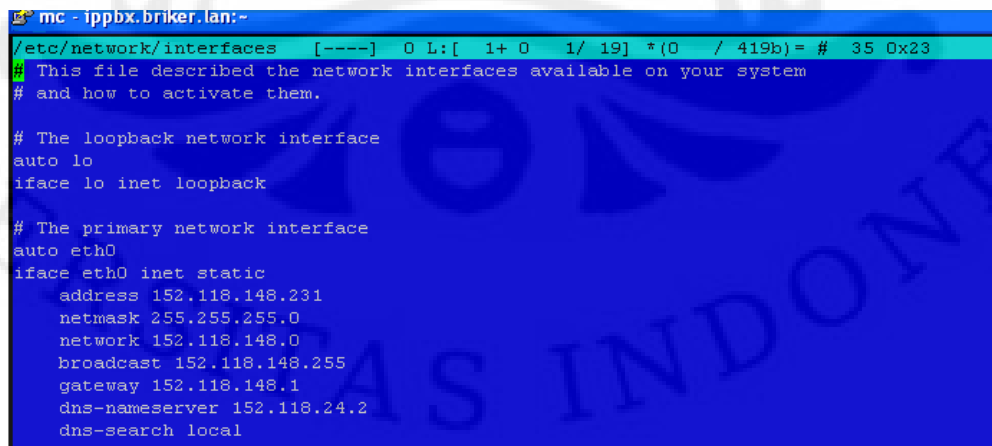
```
/etc/network/interfaces [----] 0 L:[ 1+ 0 1/ 19] *(0 / 501b)= # 35 0x23
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.2.2
    netmask 255.255.255.0
    network 192.168.2.0
    broadcast 192.168.2.255
    gateway 192.168.2.1
# dns-* options are implemented by the resolvconf package, if installed
dns-nameservers 192.168.2.1
dns-search local
```

Gambar 3.8 Konfigurasi network address *default* pada Briker

Kemudian alamat IP untuk *interface* auto eth0 kita ubah sesuai dengan IP jaringan Universitas Indonesia yang ditunjukkan pada Gambar 3.9 berikut.



```
mc - ippbx.briker.lan:-
/etc/network/interfaces [----] 0 L:[ 1+ 0 1/ 19] *(0 / 419b)= # 35 0x23
# This file described the network interfaces available on your system
# and how to activate them.

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 152.118.148.231
    netmask 255.255.255.0
    network 152.118.148.0
    broadcast 152.118.148.255
    gateway 152.118.148.1
    dns-nameserver 152.118.24.2
    dns-search local
```

Gambar 3.9 Network address setelah dikonfigurasi

Setelah men-set *network address* yang baru lalu disimpan dengan menekan tombol F2 dan keluar dari editor dengan menekan tombol F10. Selanjutnya adalah me-restart layanan networking untuk mengaktifkan konfigurasi dengan mengetikkan perintah sebagai berikut lalu enter.

```
# /etc/init.d/networking restart
```

Lalu setelah itu adalah mengatur jam dan tanggal melalui perintah sebagai berikut.

```
# date -s "2009-05-01 08:00:00"
```

Perintah tersebut adalah jika kita ingin mengatur tanggal menjadi 01 Mei 2009 pada waktu 08.00.

### 3.1.2 Web

Salah satu kelebihan dari *software* Briker IPPBX ini adalah adanya GUI yang *web-based* sehingga untuk melakukan konfigurasi terhadap *software* Asterisk yang berfungsi sebagai *server* VoIP dapat dilakukan dengan lebih mudah tanpa harus mengetikkan *command* melalui terminal *console*. Cara untuk mengakses GUI ini yaitu browse alamat IP Briker melalui web browser, setelah itu akan muncul halaman untuk *login* seperti Gambar 3.10 berikut. Sebagai *username default* masukan *administrator* dan *password default Briker* setelah itu tekan *Login*.



Gambar 3.10 Tampilan *login* pada Briker di alamat 152.118.148.231

Setelah *login* maka dapat mengubah *password default* Briker menjadi *password* yang lebih aman sesuai yang diinginkan oleh administrator. Pada menu di sebelah

kiri pilih *My Account* lalu *Preferences* dan wajib diisi box yang memiliki tanda bintang merah lalu tekan “Save”. Tampilan dapat dilihat di Gambar 3.11 berikut.

Home | IPPBX Administration | Billing | CDR | ACD Statistics | User Portal

**Logged in: administrator**  
**Status: Administrator**

[Home](#)  
[My Account](#)  
[Preferences](#)  
[Administration](#)  
[Logout](#)

## Preferences

### Login Information

Username : administrator  
 Email (\*) : dewi.asri.tp@gmail.com  
 Password :   
 Re-Type Password :

### Personal Information

Name (\*) : Administrator  
 Address (\*) : Jakarta  
 City :   
 State/Province :   
 Country (\*) : Indonesia  
 Zipcode :

Gambar 3.11 Menu untuk mengubah *password* administrator

Langkah selanjutnya untuk mengkonfigurasi adalah pilih IPPBX Administration yang terletak di menu sebelah atas. Pada halaman ini tersedia menu-menu untuk mengatur fitur IPPBX dari Briker, antara lain pengaturan *extensions*, *trunks* dan *routes*. Extension adalah nomor pengguna (*user*) yang tersambung dengan *server* ini sedangkan trunks adalah jika ingin menyambungkan sebuah *server* dengan *server* yang lain sedangkan routes adalah untuk mengatur mekanisme panggilan dari dan atau ke jaringan selain VoIP (seperti ke PSTN). Gambar 3.12 berikut menunjukkan menu-menu pada halaman IPPBX Administration.

The screenshot displays the IPPBX Administration interface with several panels:

- IPPBX Status**:
  - IPPBX Notices**: No new notifications show.
  - IPPBX Statistics**:
 

Telnet active calls	0
Internal calls	0
External calls	0
Telnet active channels	0
IPPBX Connections	0
IP Phones On Line	0
  - Uptime**:
 

System Uptime	1 week, 2 days, 27 hours, 40 minutes
Asterisk Uptime	5 days, 18 hours, 7 minutes
Last Reload	0 minutes
- System Statistics**:
 

Processor	
Load Average	0.10
CPU	1%
Memory	
Free Memory	5%
Swap	0%
Disks	
Free	1%
Network	1%
Networks	1%
Mail	1%
Mailbox	1%
Networks	
eth0 receive	0.00 KB/s
eth0 transmit	0.00 KB/s
- Server Status**:
 

Apache	OK
MySQL	OK
Web Server	OK
CGI Server	OK

Gambar 3.12 Tampilan pada halaman IPPBX Administration

Gambar di atas adalah tampilan saat melakukan *web login* dan masuk ke dalam menu *IPPBX Administration*. *IPPBX Status* menampilkan *System Statistics* yang menunjukkan persentasi *Load Average*, *CPU*, *Memory* dan *Swap* yang terpakai, penggunaan ruang *harddisk* dan kecepatan *Receive* dan *Transmit Ethernet*. Terdapat pula *IPPBX Statistics* yang menampilkan *Total Active Calls*, *Internal Calls*, *External Calls*, *Total Active Channels*, serta informasi *Uptime* Briker. Karena proses pengambilan data-data nya *realtime* dan menggunakan *CPU resource* yang tidak sedikit maka tidak disarankan untuk terus menerus membuka halaman ini. Konfigurasi utama fitur-fitur IPPBX dapat ditemui pada menu-menu di sebelah kiri.

Selanjutnya pada halaman ini pilih menu *Extensions* yang ada di sisi sebelah kiri untuk membuat *account user* VoIP. Setelah itu pilih *Add Extensions* dan pilih protokol *signaling* yang ingin digunakan yaitu SIP lalu tekan “Submit” seperti terlihat pada Gambar 3.13 berikut.



Gambar 3.13 Memilih jenis protokol untuk extension

Setelah itu akan muncul tampilan seperti pada Gambar 3.14 ini dan kemudian isikan informasi mengenai extension yang ingin dibuat.

Gambar 3.14 Membuat extension baru

Dari Gambar 3.14, yang wajib diisi hanyalah kolom *User Extensions*, *Display Name*, *Secret* dan *Accountcode*, berikut adalah keterangan masing-masing kolom:

- *User Extensions* : Nomor extension, misal 1001. Biasanya hanya numerikal
- *Display Name* : Nama yang akan digunakan sebagai *Caller ID* ketika melakukan panggilan
- *Secret* : *Password* yang digunakan *user* untuk proses otentikasi saat registrasi extension pada *User Agent*
- *Accountcode* : Kode khusus yang dipakai untuk mengidentifikasi *user log*

Dalam skripsi ini dibuat 3 *extension* SIP yang bertujuan untuk mensimulasikan beberapa panggilan melalui beberapa jaringan yang berbeda. Gambar 3.15 berikut menunjukkan extension yang telah dibuat.



Gambar 3.15 Tiga buah SIP extension yang telah dibuat

## 3.2 KONFIGURASI CLIENT

Di dalam uji coba jaringan ini, *client* menggunakan *softphone* yang diinstal ke dalam PC atau laptop seperti terlihat pada Gambar 3.16 berikut.



Gambar 3.16 Tampilan *softphone*

Meskipun berupa *software* namun *softphone* ini berfungsi layaknya telepon biasa seperti adanya dialpad, on-hook dan off-hook. *Softphone* yang digunakan adalah Xlite keluaran Counterpath yang dapat diunduh dari <http://www.counterpath.com>. Agar dapat digunakan, *softphone* harus dikonfigurasi terlebih dahulu sehingga bisa teregistrasi dan tersambung ke *server*. Pertama klik tombol *Show Menu* di bagian atas lalu pilih *SIP Account Settings* dan pilih *Add* lalu masukkan data salah satu extension yang sebelumnya telah didaftarkan di *server*. Tampilannya dapat dilihat pada Gambar 3.17 berikut.



Gambar 3.17 Cara konfigurasi Xlite

### 3.3 PuTTY

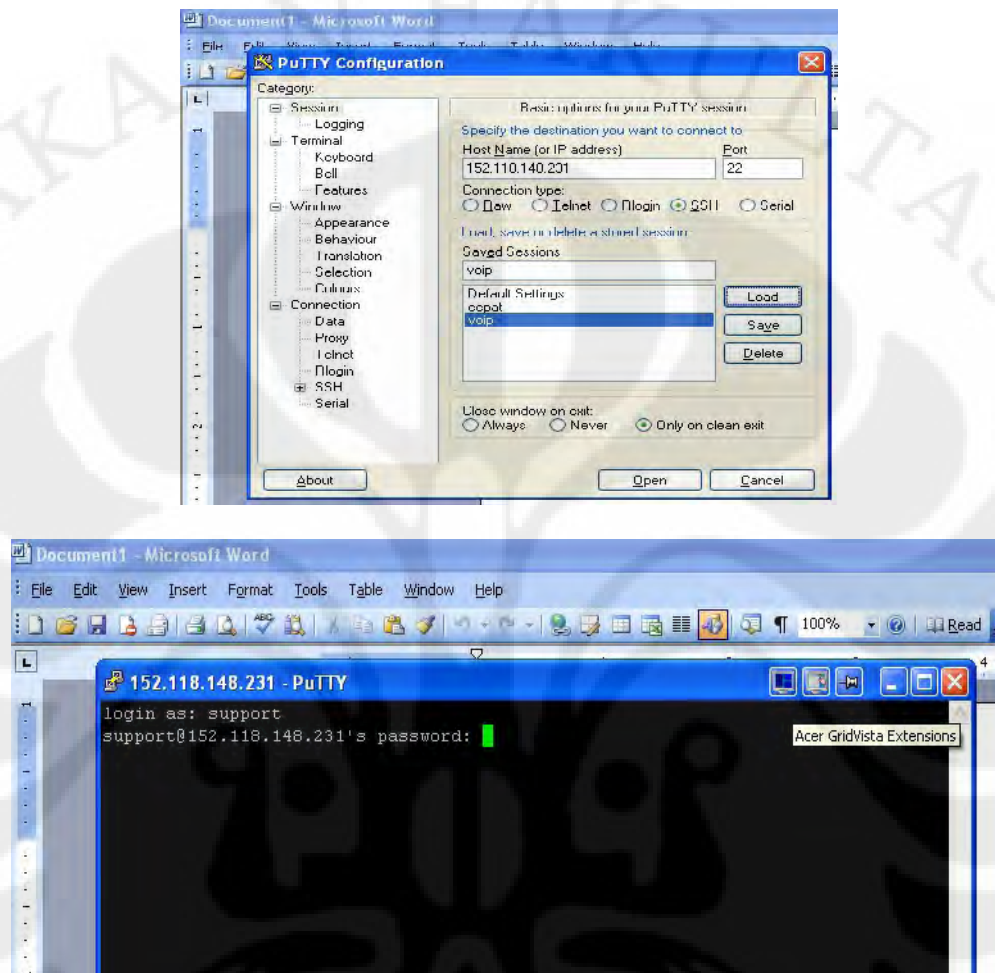
PuTTY adalah *software* yang berfungsi untuk mengakses *server* dari jarak jauh melalui SSH port 22. Hal ini diperlukan agar tetap bisa mengkonfigurasi *server* meskipun tidak berada di DMZ PPSI yang letaknya cukup jauh. PuTTY dapat diinstal di PC atau laptop yang berbasis Linux Ubuntu maupun Windows XP. Untuk menginstall PuTTY pada Ubuntu cukup mengetikkan perintah berikut.

```
# sudo apt-get install putty
```

Sedangkan pada Windows, PuTTY dapat terlebih dahulu diunduh dari alamat <http://the.earth.li/~sgtatham/putty/0.60/x86/putty.exe>. Meskipun *server* yang



digunakan berbasis Ubuntu, tapi tetap bisa diakses melalui PuTTY yang berada di laptop berbasis Windows. Gambar 3.18 adalah tampilan PuTTY di Windows.



Gambar 3.18 Akses server melalui PuTTY di Windows

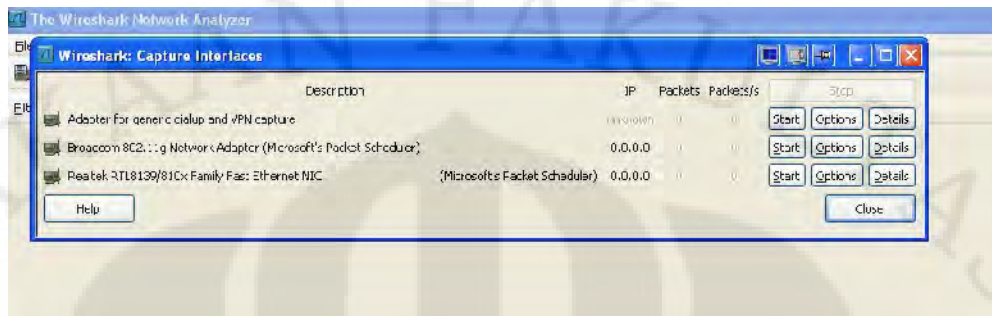
### 3.4 WIRESHARK

Wireshark adalah *software* yang digunakan untuk pengukuran objektif QoS untuk jaringan VoIP yang dibangun pada penelitian skripsi ini. Seperti halnya PuTTY, *Wireshark* juga dapat diinstal baik di Linux Ubuntu maupun di Windows. Untuk menginstal Wireshark di Ubuntu cukup dengan mengetikkan perintah berikut di terminal atau *shell*.

```
# apt-get install wireshark
```

Sedangkan pada Windows dapat mengunduh installernya terlebih dahulu di alamat <http://www.wireshark.org> lalu install di drive yang diinginkan. Setelah itu

buka Wireshark dan pilih menu Capture lalu pilih Interfaces sehingga muncul seperti Gambar 3.19 berikut lalu pilih *interface* yang diinginkan lalu tekan Start.



Gambar 3.19 Memilih *interface* yang ingin di-capture melalui Wireshark

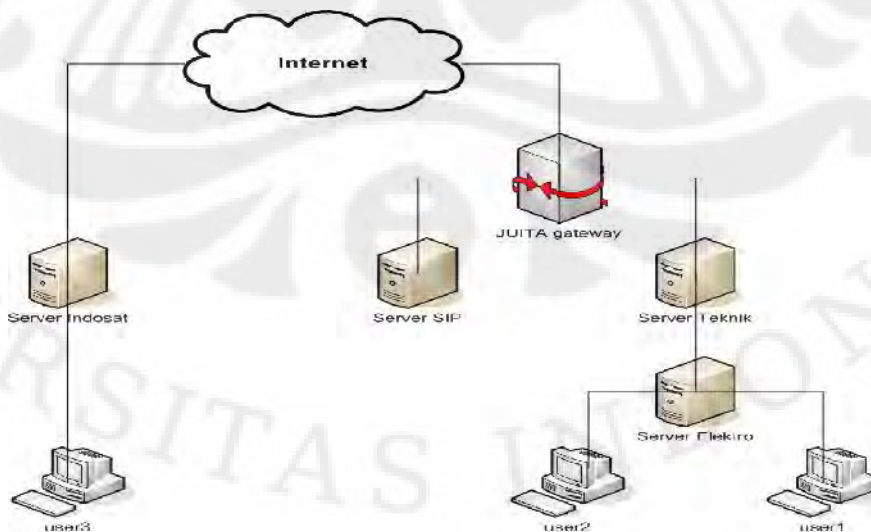
Karena Wireshark pada skripsi ini digunakan untuk menangkap panggilan VoIP, maka paket yang ditangkap berjenis RTP. Untuk mendapatkan paket ini dilakukan dengan cara menekan tombol *Statistics* yang ada di menu bagian atas lalu pilih RTP kemudian *Show All Streams*. Selanjutnya pilih salah satu aliran paket data (*stream*) yang sesuai dengan *source IP* dan *destination IP* yang diinginkan.

## BAB 4

### UJI COBA, DATA DAN ANALISIS

#### 4.1. SKENARIO UJI COBA

Uji coba unjuk kerja jaringan VoIP berbasis SIP yang telah dibangun dilakukan dengan menjalankan perangkat lunak X-lite versi 3.0 sebagai SIP *softphone* yang digunakan oleh *user* untuk saling berkomunikasi. Terdapat dua macam jaringan yang digunakan oleh *user* yaitu jaringan LAN UI dan jaringan Indosat. Pada uji coba ini, *server* berkedudukan di PPSI UI Depok, dua *user* yaitu user1 dan user2 berada di jaringan LAN UI dan satu *user* yaitu user3 berada di jaringan Indosat. *User* yang ada di jaringan LAN UI keduanya memiliki IP publik yang berada di belakang proxy UI sedangkan *user* yang ada di jaringan Indosat memiliki IP privat yang berada di belakang NAT (*Network Address Translator*) milik Indosat. Pada jaringan UI, user1 dan user2 memiliki IP address dan SIP URI masing-masing adalah 152.118.101.178 (sip:1001@152.118.148.231) dan 152.118.101.167 (sip:1002@152.118.148.231). Sedangkan pada jaringan Indosat, user3 memiliki IP address privat 10.207.44.49 (sip:1003@152.118.148.231) dengan IP NAT adalah 202.93.36.78. Uji coba dilakukan dengan menggunakan dua macam jaringan karena kedua jaringan tersebut memiliki besar *bandwidth* yang berbeda. Jaringan LAN UI memiliki *bandwidth* sebesar 100 Mbps sedangkan jaringan Indosat yang digunakan memiliki *bandwidth* sebesar 256 Kbps. Gambar 3.1 berikut adalah topologi jaringan VoIP yang diuji coba.



Gambar 4.1 Topologi jaringan yang diuji coba

Parameter pengujian yang diambil datanya dan kemudian dianalisis adalah *delay*, *bandwidth* dan *packet loss*. Variabel yang divariasikan pada pengujian ada dua yaitu variasi jenis *codec* antara G.711  $\mu$ -Law dan GSM serta variasi jaringan yang digunakan yaitu antara LAN UI dan Indosat. Hasil uji coba setiap percobaan dideskripsikan melalui beberapa grafik yang menunjukkan hubungan antara nomor paket dan *delay* serta hubungan antara nomor paket dan *bandwidth*. Sedangkan untuk parameter *packet loss* hanya disebutkan dalam bentuk persentase dari total paket yang diterima. Nomor paket merupakan penomoran setiap paket yang menggambarkan urutan paket.

## 4.2 UJI COBA INTERNAL LAN UI

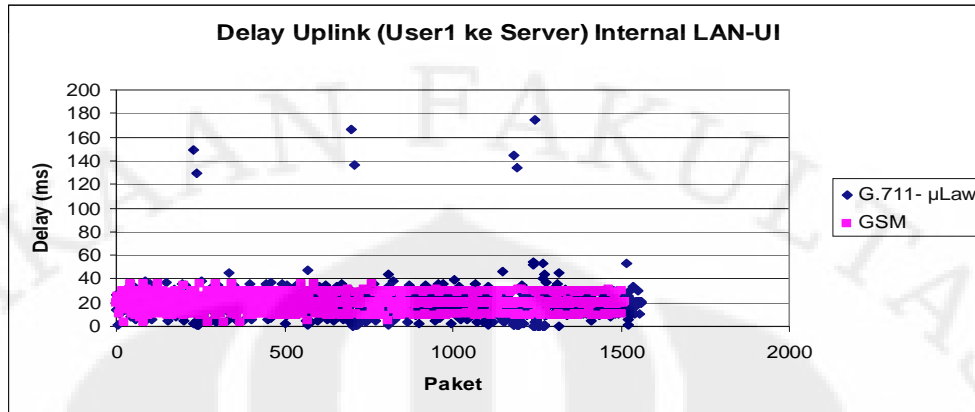
Pada jaringan LAN UI, pengujian dilakukan dua kali yaitu dari user1 ke user2 dan begitu juga sebaliknya dari user2 ke user1 dengan tujuan untuk melihat kualitas VoIP dari kedua arah.

### 4.2.1 User1 ke User2

Pada pengujian ini, pengukuran dilakukan pada dua sisi yaitu *uplink* dari user1 ke *server* serta *downlink* dari *server* ke user2. Pengukuran kedua sisi ini diperlukan untuk memperoleh nilai *one way transmission delay* user1 ke user2.

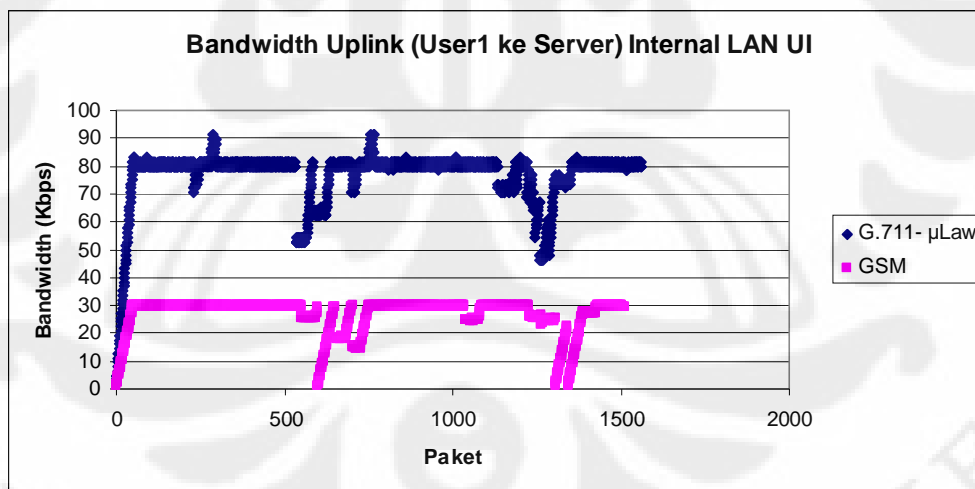
#### 4.2.1.1 Uplink

Pada sisi *uplink*, *delay* rata-rata ketika menggunakan *codec* G.711 adalah sebesar 19,99 ms dengan *delay* maksimum mencapai 174,95 ms. Sedangkan ketika menggunakan *codec* GSM, *delay* rata-rata adalah 19,98 ms dengan *delay* maksimum adalah 36,32 ms. Meskipun kedua *codec* memiliki *bit rate* yang jauh berbeda tapi *delay* keduanya hampir sama. Hal ini disebabkan karena tes dilakukan di jaringan internal LAN UI yang *bandwidth*nya cukup tinggi sehingga tetap dapat men-support dengan baik *codec* G.711. Namun karena *bit rate* G.711 tinggi sehingga membuat *delay*-nya menjadi lebih tidak stabil. Terdapat beberapa paket G.711 dengan *delay* cukup tinggi yang terlihat dari Gambar 4.2 berikut ini.



Gambar 4.2 Grafik *Delay Uplink* User1 ke User2 di Internal LAN UI

Sedangkan untuk parameter *bandwidth*, G.711 memiliki *bandwidth* rata-rata sebesar 76,72 Kbps dengan *bandwidth* maksimum mencapai 91.2 Kbps. Sedangkan pada GSM, *bandwidth* rata-rata nya adalah sebesar 26,54 Kbps dengan *bandwidth* maksimum mencapai 29,78 Kbps. Perbedaan konsumsi *bandwidth* antara kedua *codec* ini cukup jauh di mana konsumsi *bandwidth* GSM jauh lebih kecil dibandingkan dengan G.711. Namun begitu, baik pada G.711 maupun GSM tidak ada yang mengalami *packet loss*. Gambar 4.3 menunjukkan perbandingan *bandwidth* antara G.711 dan GSM pada sisi *uplink* user1 ke user2.

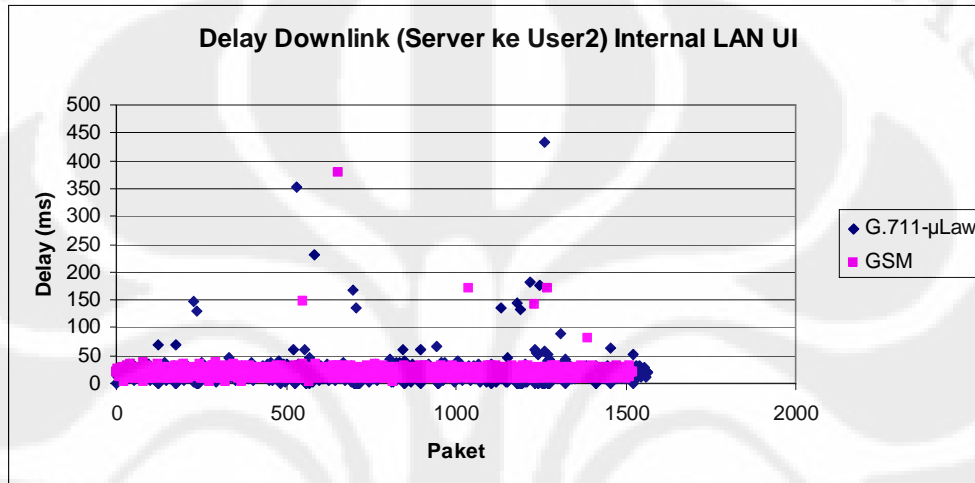


Gambar 4.3 Grafik *Bandwidth Uplink* User1 ke user2 Internal LAN UI

#### 4.2.1.2 Downlink

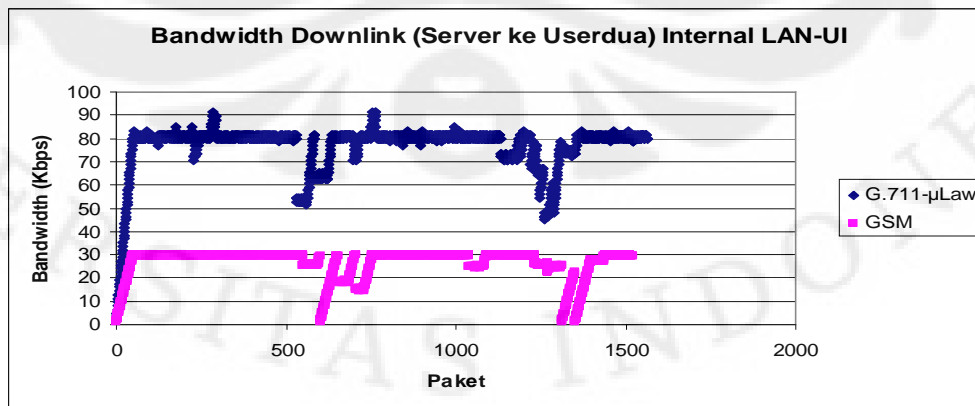
Pada sisi *downlink*, *delay* rata-rata ketika menggunakan *codec* G.711 adalah sebesar 20.85 ms dengan *delay* maksimum mencapai 434.44 ms. Sedangkan ketika menggunakan *codec* GSM, *delay* rata-rata adalah 20.62 ms

dengan *delay* maksimum adalah 379.97 ms. Meskipun *delay* rata-rata kedua *codec* pada sisi *downlink* sedikit lebih tinggi dibandingkan saat di sisi *uplink* tapi nilainya tidak jauh berbeda. Pada sisi *downlink* ini, ada beberapa paket GSM yang *delay*-nya juga cukup tinggi yang mungkin disebabkan proses dekompresi yang kurang baik di sisi penerima. Grafik *delay* pada sisi *downlink* ditunjukkan pada Gambar 4.4 berikut ini.



Gambar 4.4 Grafik *Delay Downlink* User1 ke User2 Internal LAN UI

Sementara untuk parameter *bandwidth* di sisi *downlink*, G.711 memiliki *bandwidth* rata-rata sebesar 76,67 Kbps dengan maksimum *bandwidth* adalah 91,2 Kbps. Sedangkan pada GSM memiliki *bandwidth* rata-rata sebesar 26.52 Kbps dengan *bandwidth* maksimum adalah 29.78 Kbps. Jika dibandingkan, *bandwidth* pada sisi *downlink* hampir sama persis dengan *bandwidth* pada sisi *uplink*, hal ini karena di sisi *downlink* juga tidak ada *packet loss* sehingga semua paket dari user1 ke user2 berhasil ditransmisikan dengan sempurna seperti pada Gambar 4.5 ini.



Gambar 4.5 Grafik *Bandwidth Downlink* User1 ke User2 Internal LAN UI

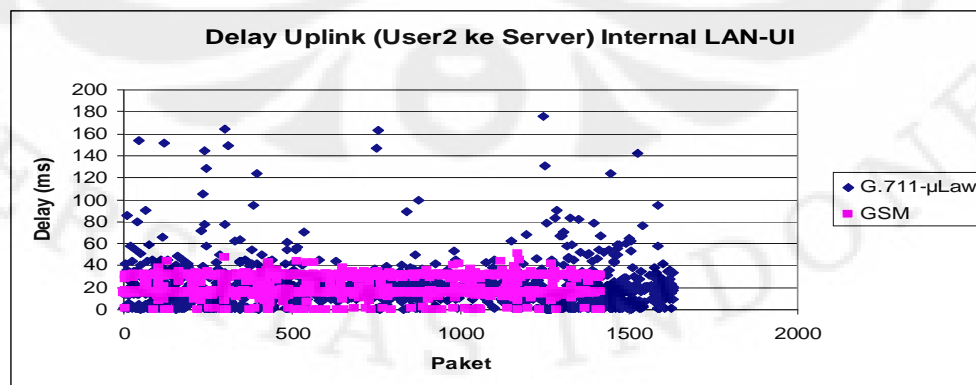
Dari penjelasan mengenai *uplink* dan *downlink* dapat diperoleh nilai total dari *one way transmission delay* pada arah user1 ke user2. Untuk G.711, *delay* totalnya adalah 40,84 ms sedangkan pada GSM *delay* total sebesar 40,6 ms. Dari data tersebut dapat dikatakan bahwa kedua jenis *codec* ini dapat digunakan dengan baik pada arah user1 ke user2 karena memenuhi syarat *one way transmission delay* yang kurang dari 150 ms. Selain itu, untuk kedua *codec* juga tidak terdapat *packet loss* sama sekali (0%) karena *bandwidth* yang diperlukan oleh kedua *codec* terpenuhi oleh jaringan.

#### 4.2.2 User2 ke User1

Pada pengujian ini, pengukuran dilakukan juga pada dua sisi yaitu *uplink* dari user2 ke *server* serta *downlink* dari *server* ke user1.

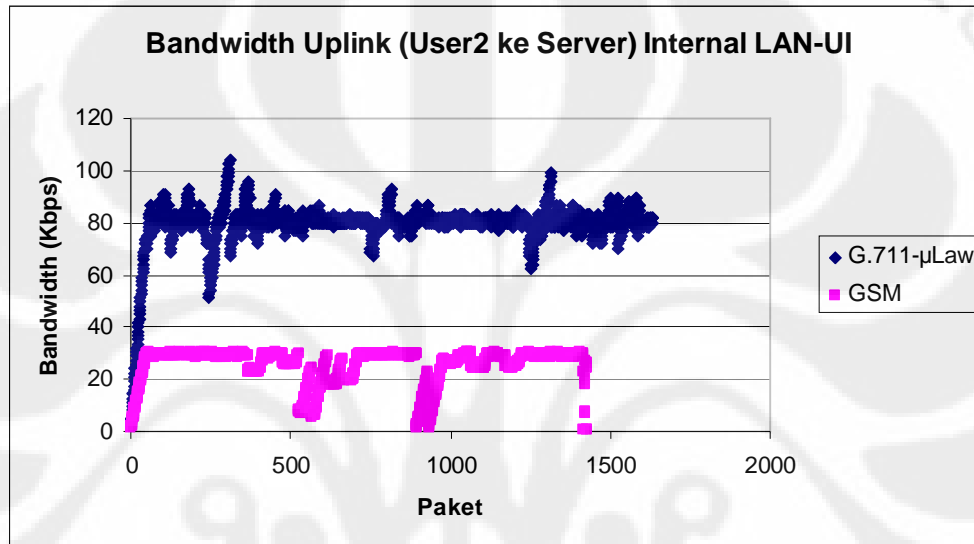
##### 4.2.2.1 Uplink

Pada sisi *uplink*, *delay* rata-rata ketika menggunakan G.711 adalah 19,99 ms dengan *delay* maksimum mencapai 176.29 ms. Sedangkan ketika menggunakan GSM, *delay* rata-rata adalah 20.05 ms dengan *delay* maksimum 50.3 ms. Nilai *delay uplink* ini hampir sama dengan *uplink* pada arah user1 ke user2. Hal ini karena kedua user berada di jaringan yang sama yaitu internal LAN UI sehingga performansi untuk kedua arah tidak jauh berbeda. Namun *uplink* pada arah user2 ke user1 memiliki *delay* yang lebih tidak stabil dibandingkan pada arah user1 ke user2. Hal ini karena jalur perutingan yang digunakan untuk kedua arah tidak sama. Grafik *delay* ini ditunjukkan oleh Gambar 4.6 berikut.



Gambar 4.6 Grafik *Delay Uplink* User2 ke User1 Internal LAN UI

Sedangkan untuk *bandwidth*, G.711 memiliki *bandwidth* rata-rata sebesar 79,42 Kbps dengan *bandwidth* maksimum mencapai 104 Kbps. Sedangkan pada GSM, *bandwidth* rata-rata nya adalah sebesar 25,8 Kbps dengan *bandwidth* maksimum mencapai 30,37 Kbps. *Bandwidth uplink* user2 ke user1 ini tidak jauh berbeda dengan nilai *bandwidth* pada *uplink* user1 ke user2. Gambar 4.7 berikut adalah grafik *bandwidth uplink* user2 ke user1.



Gambar 4.7 Grafik *Bandwidth Uplink* User2 ke User1 Internal LAN UI

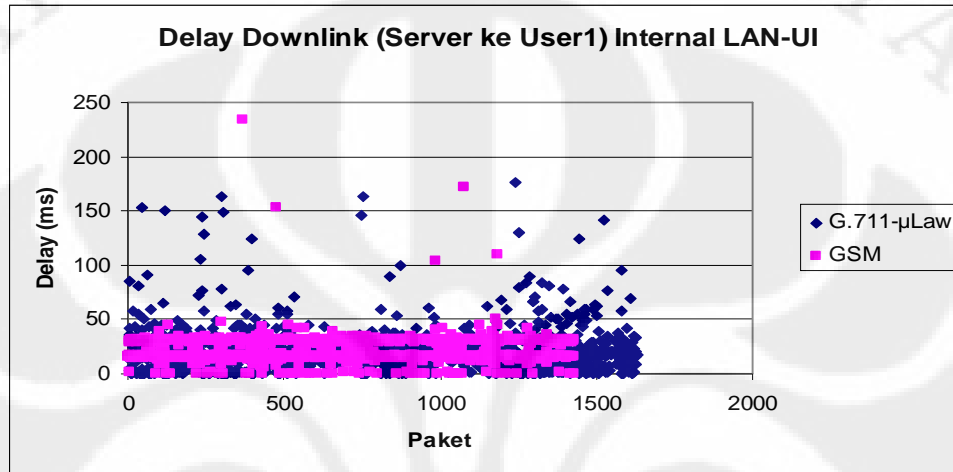
Meskipun *bandwidth* hampir sama besar dengan *uplink* pada arah user1 ke user2 tapi terdapat sebuah paket G.711 yang hilang ketika *uplink* pada arah user2 ke user1 ini. Hal ini seperti juga telah disebutkan tadi dapat terjadi karena paket G.711 memiliki *bit rate* yang tinggi dan kedua arah memiliki jalur perutingan yang berbeda.

#### 4.2.2.2 Downlink

Pada sisi *downlink*, *delay* rata-rata *codec* G.711 adalah sebesar 20.02 ms dengan *delay* maksimum mencapai 176,43 ms. Sedangkan pada *codec* GSM, *delay* rata-rata adalah 20.52 ms dengan *delay* maksimum adalah 233,53 ms. Berbeda dengan sebelum-sebelumnya, pada sisi *downlink* ini *delay* GSM baik rata-rata maupun maksimum nilainya sedikit lebih tinggi dibandingkan dengan G.711. Meskipun begitu, paket-paket G.711 tetap memiliki *delay* yang lebih tidak stabil

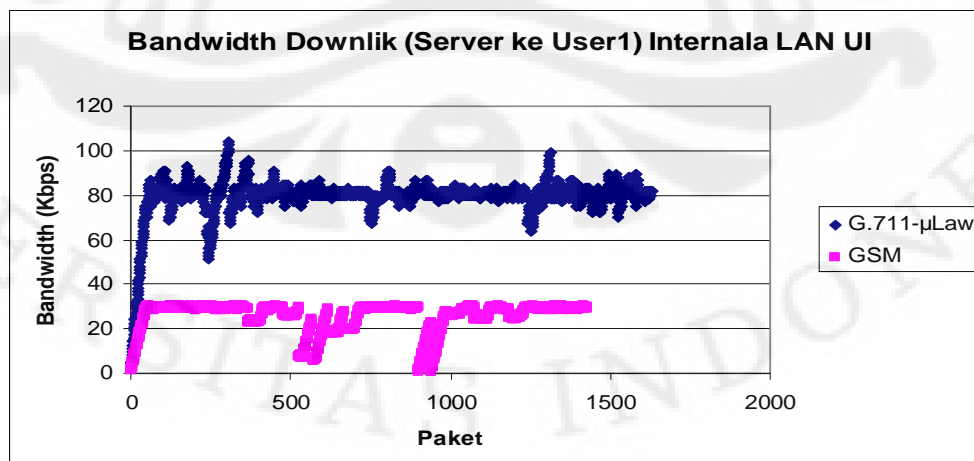


dibandingkan dengan pada GSM. Pada GSM, paket yang memiliki *delay* yang cukup tinggi hanya beberapa, tapi nilainya cukup besar sehingga membuat nilai *delay* rata-rata GSM sedikit lebih tinggi dibandingkan *delay* rata-rata G.711. Gambar 4.7 berikut menunjukkan perbandingan *delay* G.711 dan GSM pada sisi *downlink*.



Gambar 4.8 Grafik *Delay Downlink* User2 ke User1 Internal LAN UI

Sementara pada *bandwidth* di sisi *downlink*, seperti terlihat pada Gambar 4.8, G.711 memiliki *bandwidth* rata-rata sebesar 79,42 Kbps dengan maksimum *bandwidth* adalah 104 Kbps. Sedangkan pada GSM memiliki *bandwidth* rata-rata sebesar 25,75 Kbps dengan *bandwidth* maksimum adalah 30,37 Kbps. Jika dibandingkan, *bandwidth* pada sisi *downlink* hampir sama persis dengan *bandwidth* pada sisi *uplink*, hal ini karena di sisi *downlink* tidak ada *packet loss* sehingga semua paket yang diterima oleh *server* dari user2 diteruskan ke user1.



Gambar 4.9 Grafik *Bandwidth Downlink* User2 ke User1 Internal LAN UI

Dari penjelasan mengenai *uplink* dan *downlink* dapat diperoleh nilai total *one way transmission delay* pada arah user1 ke user2. Untuk G.711, *delay* totalnya adalah 40 ms sedangkan pada GSM, *delay* total sebesar 40,57 ms. Dari data di tersebut dapat dikatakan bahwa kedua jenis *codec* ini dapat digunakan dengan baik pada arah user2 ke user1 karena memenuhi syarat yaitu *one way transmission delay* yang kurang dari 150 ms. Meskipun terdapat *packet loss* pada G.711 saat *uplink*, tapi karena hanya satu paket yang hilang maka hal ini masih berada dalam toleransi nilai *packet loss* maksimum yang diperbolehkan yaitu sebesar 15% dari total paket yang dikirimkan.

Berdasarkan pemaparan pada arah user1 ke user2 maupun user2 ke user1 dapat disimpulkan bahwa kedua *codec* dapat digunakan dengan baik pada kedua arah dalam jaringan internal LAN UI yang memiliki *bandwidth* tinggi. Tabel 4.4, 4.5, dan 4.6 berikut adalah nilai dari parameter *delay*, *bandwidth* dan *packet loss* di jaringan internal LAN UI pada arah user1 ke user2 dan arah user2 ke user1.

Tabel 4.1 Delay Internal LAN UI

Codec	User1 ke User2				User2 ke User1			
	Uplink		Downlink		Uplink		Downlink	
	Rata-rata	Maks.	Rata-rata	Maks.	Rata-rata	Maks.	Rata-rata	Maks.
G.711 $\mu$ -Law	19,99 ms	174,95 ms	20,85 ms	434,44 ms	19,99 ms	176,29 ms	20,02 ms	176,43 ms
GSM 06.10	19,98 ms	36,32 ms	20,62 ms	379,97 ms	20,05 ms	50,3 ms	20,52 ms	233,53 ms

Tabel 4.2 Bandwidth Internal LAN UI

Codec	User1 ke User2				User2 ke User1			
	Uplink		Downlink		Uplink		Downlink	
	Rata-rata	Maks.	Rata-rata	Maks.	Rata-rata	Maks.	Rata-rata	Maks.
G.711 $\mu$ -Law	76,72 Kbps	91,2 Kbps	76,67 Kbps	91,2 Kbps	79,42 Kbps	104 Kbps	79,42 Kbps	104 Kbps
GSM 06.10	26,54 Kbps	29,78 Kbps	26,52 Kbps	29,78 Kbps	25,8 Kbps	30,37 Kbps	25,75 Kbps	30,37 Kbps

Tabel 4.3 Packet loss Internal LAN UI

Codec	User1 ke User2		User2 ke User1	
	Uplink	Downlink	Uplink	Downlink
G.711 $\mu$ -Law	0%	0%	0,1 %	0%
GSM 06.10	0%	0%	0%	0%

### 4.3 UJI COBA LAN UI - INDOSAT

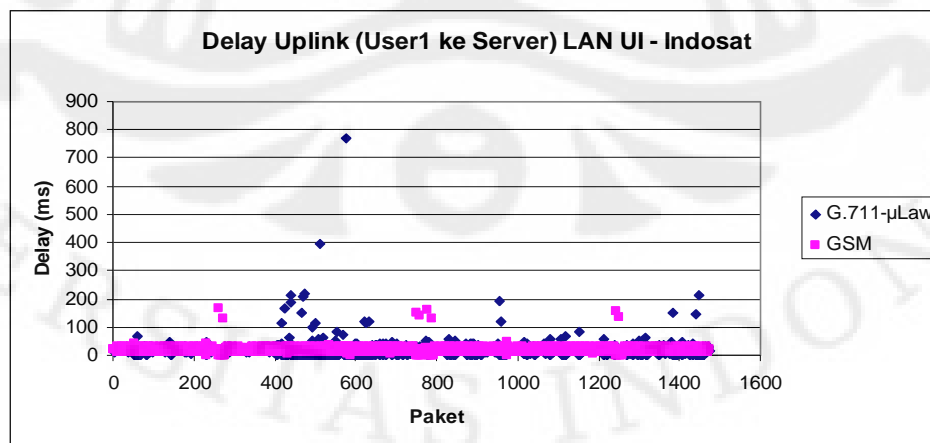
Pada pengujian ini kedua user berada di dua jaringan yang berbeda yaitu user1 di jaringan LAN UI sedangkan user3 berada di jaringan Indosat. Meskipun *server* VoIP berada di jaringan UI, user dapat berada di luar jaringan LAN UI karena sistem VoIP berbasis SIP memiliki kelebihan dapat menembus NAT dengan mudah. Pengujian dilakukan pada kedua arah yaitu dari user1 ke user3 dan begitu juga sebaliknya dari user3 ke user1.

#### 4.3.1 User1 ke User3

Seperti pada pengujian sebelumnya, pengukuran dilakukan pada dua sisi yaitu *uplink* dan *downlink*. Sisi *uplink* adalah dari user1 ke *server* sedangkan sisi *downlink* adalah dari *server* ke user3.

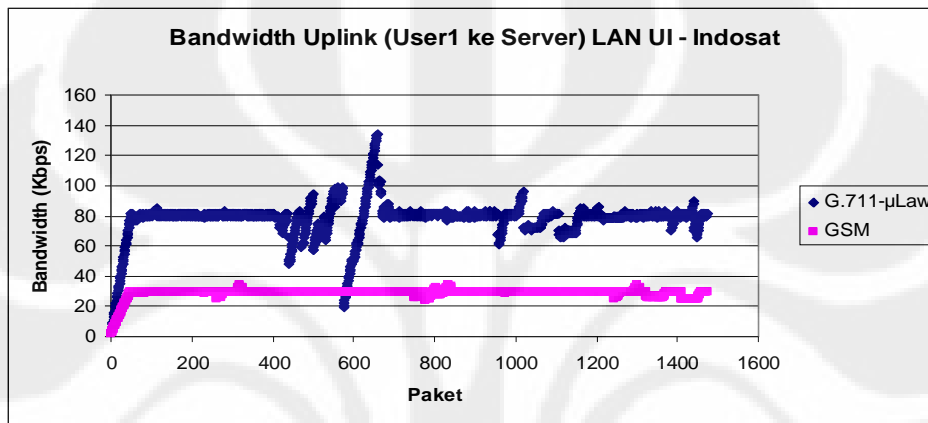
##### 4.3.1.1 Uplink

Pada sisi *uplink*, *delay* rata-rata ketika menggunakan *codec* G.711 adalah sebesar 20,11 ms dengan *delay* maksimum mencapai 769,08 ms. Sedangkan ketika menggunakan *codec* GSM, *delay* rata-rata adalah 19,99 ms dengan *delay* maksimum adalah 164 ms. *Delay* maksimum pada G.711 yang sangat tinggi ini dapat disebabkan oleh jaringan yang mulai sibuk ketika pengujian dilakukan. Namun karena hanya ada dua paket yang *delay*-nya di atas 250 ms maka tidak terlalu berpengaruh pada *delay* rata-rata keseluruhan. Grafik *delay uplink* user1 ke user3 ini dapat dilihat pada Gambar 4.10



Gambar 4.10 Grafik *Delay Uplink* User1 ke User3 LAN UI – Indosat

Sementara untuk *bandwidth*, G.711 memiliki *bandwidth* rata-rata sebesar 78,02 Kbps dengan *bandwidth* maksimum mencapai 134,4 Kbps. Sedangkan pada GSM, *bandwidth* rata-rata nya adalah sebesar 28,83 Kbps dengan *bandwidth* maksimum mencapai 33,87 Kbps. *Bandwidth* ini tidak terlalu berbeda dengan *bandwidth uplink* di pengujian internal LAN UI karena dari user1 ke *server* berada di internal LAN UI. Grafik *bandwidth uplink* ditunjukkan oleh Gambar 4.11

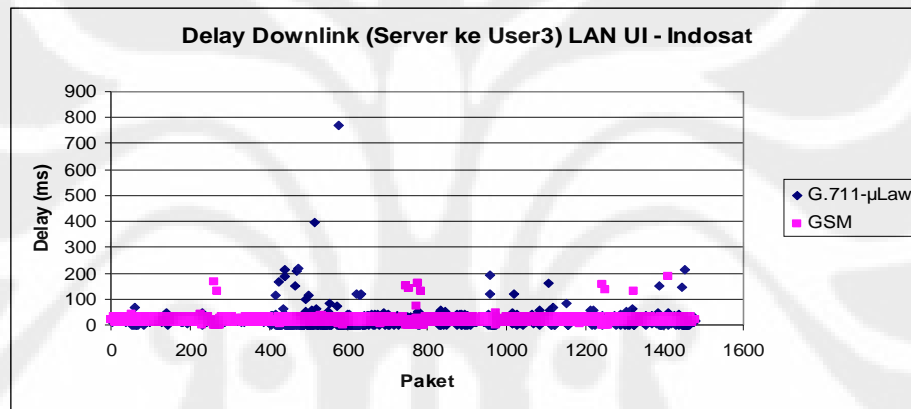


Gambar 4.11 Grafik *Bandwidth Uplink* User1 ke User3 LAN UI – Indosat

Namun pada *uplink* ini, untuk G.711 terdapat 6 buah paket (0,4%) yang hilang dan 5 buah paket diterima oleh *server* dengan urutan yang salah (*sequence error*). Paket yang hilang ini membuat grafik *bandwidth* G.711 di Gambar 4.11 tampak ada bagian yang terputus. Karena adanya paket yang hilang ini juga menyebabkan nilai *bandwidth* paket selanjutnya tiba-tiba langsung menjadi sangat kecil yaitu hanya sebesar 19,2 Kbps yang kemudian terus merangkak naik hingga mencapai 134,4 Kbps. Jika diperhatikan, bagian tengah grafik G.711 setelah terjadinya paket yang hilang menyerupai bentuk grafik G.711 di bagian awal yang menunjukkan proses inisialisasi jaringan terlebih dahulu. Hal ini disebabkan karena tidak adanya paket yang dikirimkan sebelumnya atau telah terjadi paket yang hilang sebelumnya. Hilangnya beberapa paket dapat disebabkan oleh trafik jaringan yang mulai padat sehingga menyebabkan beberapa paket G.711 yang *bit rate*-nya jauh lebih tinggi dibandingkan GSM tidak dapat dikirimkan karena kurang tersedianya *bandwidth*.

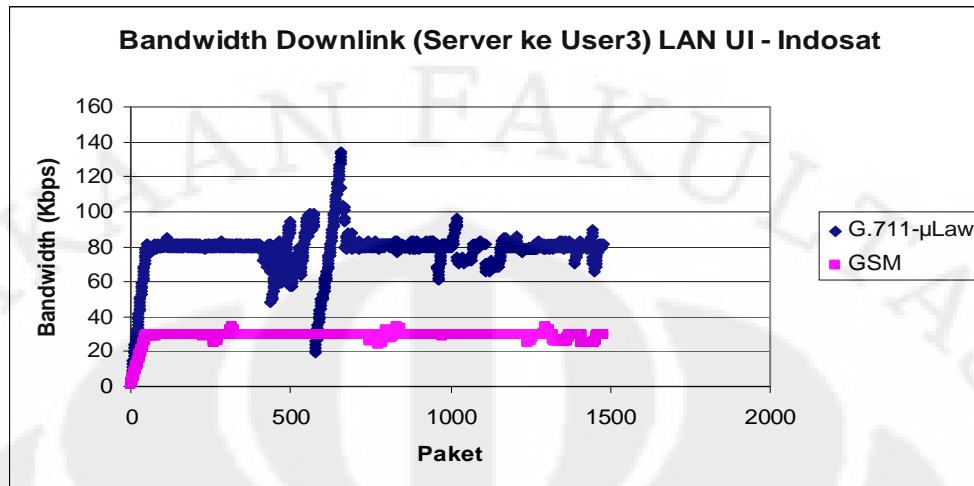
#### 4.3.1.2 Downlink

Pada sisi *downlink*, *delay* rata-rata ketika menggunakan *codec* G.711 adalah sebesar 20.29 ms dengan *delay* maksimum mencapai 769,1 ms. Sedangkan ketika menggunakan *codec* GSM, *delay* rata-rata adalah 20.19 ms dengan *delay* maksimum adalah 187,73 ms. *Delay* rata-rata kedua *codec* pada sisi *downlink* hampir sama nilainya dengan sisi *uplink*. Begitu juga dengan *delay* maksimum untuk kedua *codec* pada sisi *downlink* tidak jauh berbeda dengan sisi *uplink* karena paket yang terlambat sampai ke *server* akan mengalami keterlambatan juga saat sampai ke user3. *Delay uplink* ini ditunjukkan pada Gambar 4.12 berikut.



Gambar 4.12 Grafik *Delay Downlink* User1 ke User3 LAN UI – Indosat

Sementara itu, konsumsi *bandwidth* rata-rata pada sisi *downlink* untuk G.711 adalah 78 Kbps dengan *bandwidth* maksimum sebesar 134,4 Kbps. Sedangkan untuk GSM, *bandwidth* rata-rata adalah sebesar 28,81 Kbps dengan *bandwidth* maksimum sebesar 33,87 Kbps. Pada sisi *downlink* ini tidak ada paket yang hilang sehingga persentase *packet loss* adalah 0% yang menyebabkan besar *bandwidth* rata-rata untuk kedua *codec* hampir sama antara sisi *uplink* dan *downlink*. Hal ini membuat bentuk grafik *bandwidth downlink* pada Gambar 4.13 berikut sama seperti grafik *bandwidth uplink* pada Gambar 4.11 sebelumnya karena semua paket dari sisi *uplink* diteruskan oleh *server* ke user3 di sisi *downlink*. Oleh karena itu, pada sisi *downlink* tampak seolah-olah ada paket yang hilang seperti pada sisi *uplink* meskipun pada kenyataannya tidak ada paket hilang di sisi *downlink*.



Gambar 4.13 Grafik *Bandwidth Downlink* User1 ke User3 LAN UI – Indosat

Dari penjelasan mengenai *uplink* dan *downlink* dapat diperoleh nilai total *one way transmission delay* pada arah user1 ke user3. Untuk G.711, *delay* totalnya adalah 40,4 sedangkan pada GSM, *delay* total sebesar 40,57 ms. Dari data tersebut dapat dikatakan bahwa kedua jenis *codec* ini dapat digunakan dengan baik pada arah user1 ke user3. Meskipun terdapat *packet loss* sebesar 0,4% pada G.711 saat *uplink* tapi hal ini masih masuk ke dalam batas toleransi maksimum *packet loss* yaitu sebesar 15% dari total paket yang dikirimkan.

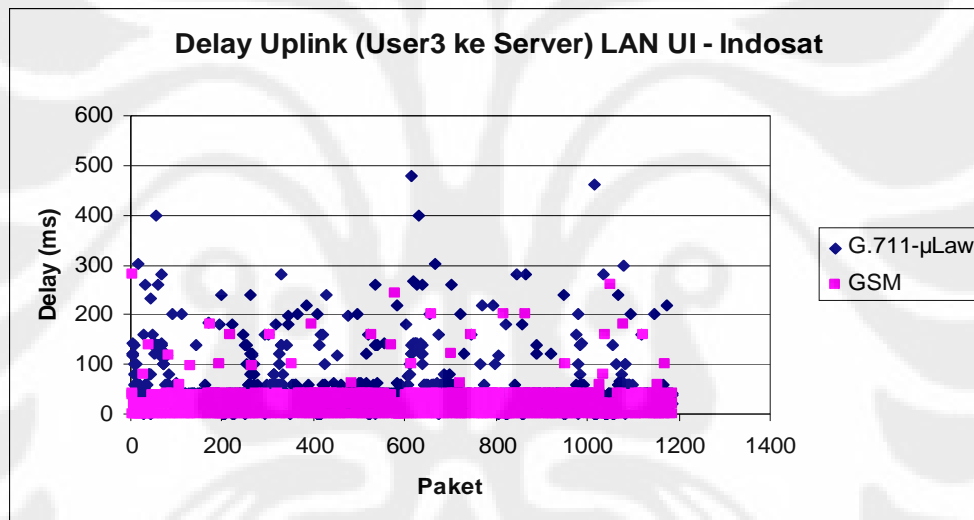
### 4.3.2. User3 ke User1

Pada pengujian yang berlawanan arah, pengukuran juga dilakukan pada kedua sisi baik *uplink* yaitu dari user3 ke *server* serta *downlink* yaitu dari *server* ke user1.

#### 4.3.2.1 Uplink

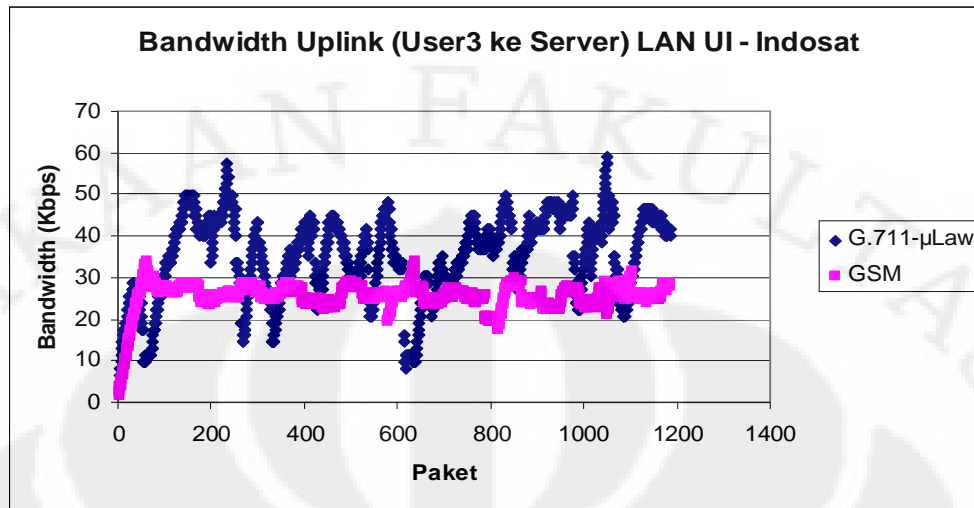
Pada sisi *uplink*, *delay* rata-rata ketika menggunakan *codec* G.711 adalah 52,28 ms dengan *delay* maksimum mencapai 479,11 ms. Sedangkan ketika menggunakan *codec* GSM, *delay* rata-rata adalah 22,33 ms dengan *delay* maksimum sebesar 281,62 ms. Nilai rata-rata *delay uplink* untuk G.711 pada arah user3 ke user1 sangat jauh berbeda dengan rata-rata *delay uplink* untuk G.711 pada arah user1 ke user3. Hal ini karena pada arah user3 ke user1, proses *uplink* terjadi di jaringan Indosat yang memiliki kecepatan *uplink* lebih rendah dari

kecepatan *downlink*-nya yang juga hanya 256 Kbps. Kecepatan atau *bandwidth* yang kecil pada sisi *uplink* menyebabkan terjadinya *delay* yang tinggi saat menggunakan G.711 yang ber-*bit rate* tinggi. Sedangkan saat menggunakan GSM, nilai rata-rata *delay uplink* pada arah user3 ke user1 hanya sedikit lebih tinggi dibandingkan pada arah user1 ke user3. Perbedaan yang tidak terlalu signifikan ini karena GSM memiliki *bit rate* yang cukup rendah sehingga tetap dapat digunakan dengan baik pada jaringan dengan *bandwidth* yang rendah. Selain itu, pada Gambar 4.14 juga terlihat bahwa penggunaan G.711 pada jaringan Indosat menimbulkan *delay* yang jauh lebih tidak stabil dibandingkan dengan saat menggunakan GSM.



Gambar 4.14 Grafik *Delay Uplink* User3 ke User1 LAN UI – Indosat

Sementara untuk *bandwidth* di sisi *uplink* ini, G.711 memiliki rata-rata *bandwidth* sebesar 34,97 Kbps dengan *bandwidth* maksimum hanya sebesar 59,2 Kbps. Sedangkan pada GSM, *bandwidth* rata-rata nya adalah sebesar 25,4 Kbps dengan *bandwidth* maksimum adalah 33,87 Kbps. *Bandwidth uplink* untuk G.711 pada arah user3 ke user1 ini jauh lebih rendah dibandingkan dengan kecenderungan *bandwidth* G.711 di skenario pengujian lainnya yang rata-rata bernilai sekitar 78 Kbps. Hal ini disebabkan oleh alasan yang sama seperti pada *delay* yaitu kecepatan yang sangat terbatas di sisi *uplink* dalam jaringan Indosat. *Bandwidth* pada sisi *uplink* ini dapat dilihat pada Gambar 4.15.



Gambar 4.15 Grafik *Bandwidth Uplink* User3 ke User1 LAN UI – Indosat

Sedangkan pada GSM, *bandwidth* yang terpakai tidak jauh berbeda dengan hasil pengujian skenario yang lain yaitu sekitar 30 Kbps. Selain itu, melihat bahwa *bandwidth* maksimum G.711 hanya bisa mencapai 59,2 Kbps, diperkirakan *bandwidth uplink* pada jaringan Indosat yang dipakai hanya sebesar 64 Kbps.

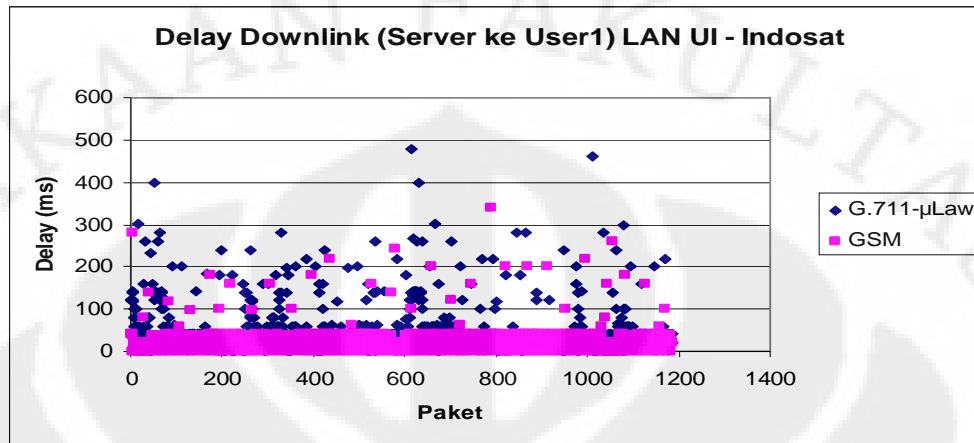
Pada skenario pengujian ini terjadi *packet loss* yang cukup besar. Pada G.711 terdapat 365 paket yang hilang atau sebesar 23,5% dari total paket. Sedangkan pada GSM, terdapat 160 paket yang hilang yang berarti sekitar 10,8% dari total jumlah paket. Selain *packet loss*, pada sisi *uplink* ini juga terjadi *sequence error* pada paket G.711 dan juga pada GSM. Untuk G.711, paket dengan *sequence error* berjumlah 160 paket sedangkan pada GSM berjumlah 29 paket.

#### 4.3.2.2 Downlink

Pada sisi *downlink*, *delay* rata-rata untuk G.711 adalah sebesar 52,3 ms dengan *delay* maksimum mencapai 479,11 ms. Sedangkan pada GSM, *delay* rata-rata adalah 23,01 ms dengan *delay* maksimum adalah 339,75 ms. *Delay* dari kedua *codec* ini tidak jauh berbeda dengan *delay* pada sisi *uplink* sebelumnya karena paket yang terlambat sampai ke *server* akan mengalami keterlambatan juga untuk sampai ke user1. Seperti juga pada hasil pengujian skenario lainnya, *delay* untuk G.711 lebih tidak stabil dibandingkan dengan *delay* untuk GSM. Hal ini dikarenakan G.711 memiliki *bit rate* tinggi yang pengirimannya memerlukan *bandwidth* yang cukup besar sementara kondisi jaringan yang dinamis membuat

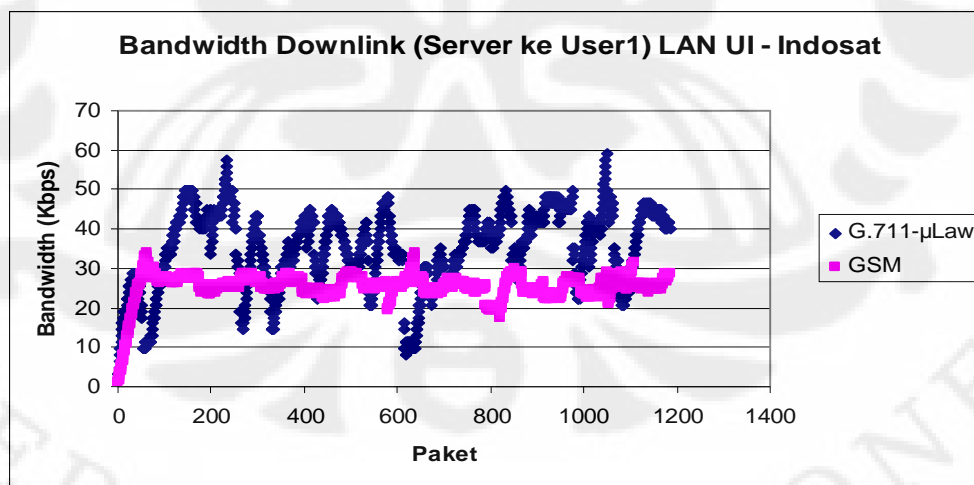


*bandwidth* besar yang diperlukan tidak selalu tersedia. Gambar 4.16 berikut adalah grafik *delay downlink* user3 ke user1.



Gambar 4.16 Grafik *Delay Downlink* User3 ke User1 LAN UI – Indosat

Sementara untuk *bandwidth* di sisi *downlink* ini, G.711 memiliki rata-rata *bandwidth* sebesar 34,96 Kbps dengan maksimum *bandwidth* adalah 59,2 Kbps. Sedangkan pada GSM, *bandwidth* rata-rata sebesar 25,38 Kbps dengan *bandwidth* maksimum adalah 33,87 Kbps. Jika dibandingkan, *bandwidth* pada sisi *downlink* hampir sama persis dengan *bandwidth* pada sisi *uplink*, hal ini karena di sisi *downlink* ini tidak ada *packet loss* sehingga semua paket yang diterima oleh *server* dari user3 diteruskan ke user1 yang ditunjukkan pada Gambar 4.17 berikut.



Gambar 4.17 Grafik *Bandwidth Downlink* User3 ke User1 LAN UI – Indosat

Dari penjelasan mengenai *uplink* dan *downlink* dapat diperoleh nilai total *one way transmission delay* pada arah user3 ke user1. Untuk G.711, *delay* totalnya adalah 104,58 ms sedangkan pada GSM, *delay* total sebesar 40,57 ms. Meskipun

total *delay* G.711 tidak melebihi 150 ms, namun nilai *packet loss*-nya lebih dari 15% yaitu mencapai 23,5% yang berarti *codec* ini tidak baik untuk digunakan pada arah user3 ke user1 yang proses *uplink*-nya terjadi di jaringan Indosat yang *bandwidth*-nya rendah. Sedangkan saat menggunakan GSM, *delay* totalnya adalah 45,34 ms dengan nilai *packet loss* masih di bawah 15% yaitu sebesar 10,8%. Oleh karena itu, untuk komunikasi pada arah user3 ke user1 jenis *codec* yang lebih baik untuk digunakan adalah GSM. Berdasarkan pemaparan di atas, dapat disimpulkan bahwa pada arah user1 ke user3, kedua jenis *codec* dapat digunakan dengan baik. Sedangkan pada arah user3 ke user1, *codec* yang baik untuk digunakan adalah GSM. Pada Tabel 4.4, 4.5, dan 4.6 berikut menunjukkan nilai dari parameter *delay*, *bandwidth* dan *packet loss* di jaringan LAN UI-Indosat pada arah user1 ke user3 dan arah user3 ke user1.

Tabel 4.4 *Delay* LAN UI-Indosat

<i>Codec</i>	User1 ke User3				User3 ke User1			
	<i>Uplink</i>		<i>Downlink</i>		<i>Uplink</i>		<i>Downlink</i>	
	Rata-rata	Maks.	Rata-rata	Maks.	Rata-rata	Maks.	Rata-rata	Maks.
G.711 $\mu$ -Law	20,11 ms	769,08 ms	20,29 ms	769,1 ms	52,28 ms	479,11 ms	52,3 ms	479,11 ms
GSM 06.10	19,99 ms	164 ms	20,19 ms	187,73 ms	22,33 ms	281,62 ms	23,01 ms	339,75 ms

Tabel 4.5 *Bandwidth* LAN UI-Indosat

<i>Codec</i>	User1 ke User3				User3 ke User1			
	<i>Uplink</i>		<i>Downlink</i>		<i>Uplink</i>		<i>Downlink</i>	
	Rata-rata	Maks.	Rata-rata	Maks.	Rata-rata	Maks.	Rata-rata	Maks.
G.711 $\mu$ -Law	78,02 Kbps	134,4 Kbps	78 Kbps	134,4 Kbps	34,97 Kbps	59,2 Kbps	34,96 Kbps	59,2 Kbps
GSM 06.10	28,83 Kbps	33,87 Kbps	28,81 Kbps	33,87 Kbps	25,8 Kbps	25,4 Kbps	25,38 Kbps	33,87 Kbps

Tabel 4.6 *Packet Loss* LAN UI-Indosat

<i>Codec</i>	User1 ke User3		User3 ke User1	
	<i>Uplink</i>	<i>Downlink</i>	<i>Uplink</i>	<i>Downlink</i>
G.711 $\mu$ -Law	0,4 %	0%	23,5 %	0%
GSM 06.10	0%	0%	10,8 %	0%

Seperti telah dijelaskan pada bab 2 bahwa *delay end-to-end* total dalam komunikasi VoIP dipengaruhi oleh beberapa faktor, yaitu *delay processing* (terdiri dari *delay coder*, *delay* paketasasi, *delay algorithmic* dan *delay* antrian), *delay*

*switching*, *delay* transmisi dan *delay* propagasi. Sedangkan besar *bandwidth* yang dikonsumsi oleh suatu *codec* dalam suatu jaringan (khususnya jaringan VoIP) sangat dipengaruhi oleh besar *payload* dalam paket VoIP dan *bitrate* dari *speech codec* yang digunakan. Semakin besar *bitrate speech codec* yang digunakan maka akan semakin besar pula *bandwidth* yang digunakan oleh *codec* tersebut. Besarnya *packet loss* yang masih bisa ditolerir dalam komunikasi VoIP adalah sebesar 15%, karena apabila besar *packet loss* melebihi 15%, maka kualitas suara menjadi jelek dikarenakan banyak paket yang akan dibuang. *Packet loss* ini terjadi karena adanya kongesti pada jaringan, dimana lalu lintas paket pada jaringan yang dilewati oleh paket sangat besar. Terjadinya *packet loss* juga dipengaruhi oleh rute jaringan yang dilewati oleh paket. Apabila pada rute yang dilewati tersebut terdapat kegagalan di media fisiknya, maka paket yang melewati rute jaringan tersebut akan dibuang dan terjadilah *packet loss*.

Dari hasil pengukuran QoS objektif VoIP pada jaringan LAN UI dan Indosat, nilai parameter-parameter QoS ini ada yang terpenuhi dan ada yang tidak. Tidak terpenuhinya parameter tersebut disebabkan oleh dua hal yaitu *bandwidth* jaringan yang terbatas dan beban trafik yang tinggi di jaringan tersebut. *Bandwidth* jaringan yang terbatas merupakan salah satu permasalahan besar dalam implementasi jaringan VoIP oleh karena itu diperlukan solusi yang mampu menghemat pemakaian *bandwidth* jaringan. Salah satunya adalah dengan cara melakukan kompresi pada *overhead* VoIP yang sebesar 40 byte menggunakan protokol bernama cRTP (*Compressed Real-Time Protocol*). Kompresi tersebut mampu menekan ukuran *overhead* VoIP hanya menjadi 2 atau 4 byte. Selain itu, bisa juga dengan cara menambah ukuran *payload* pada paket VoIP. Karena ukuran *overhead* satu paket VoIP adalah tetap dan ukuran *payload* paket VoIP dapat diubah-ubah, maka dengan menambah ukuran *payload* pada paket VoIP, dapat menurunkan konsumsi *bandwidth* dari paket VoIP tersebut. Akan tetapi, di sisi lain, penambahan *payload* akan menaikkan *delay* paketisasi yang akan menaikkan *delay* keseluruhan. Oleh karena itu, penambahan *payload* pada paket VoIP juga harus memperhatikan dampaknya pada *delay* paketisasi. Sementara untuk beban trafik yang tinggi akan menyebabkan adanya kongesti yang bisa menyebabkan terjadinya *packet loss* pada jaringan. Salah satu cara untuk

mengurangi beban trafik tersebut adalah dengan menggunakan *speech codec* yang memiliki *bitrate* rendah, seperti GSM 06.10. *Speech codec* GSM ini mampu menghemat pemakaian *bandwidth* jaringan secara signifikan. Di samping itu, penggunaan protokol seperti RSVP (*Resource Reservation Protocol*) yang melakukan reservasi *bandwidth* terlebih dahulu sebelum paket dikirimkan, atau penggunaan metode DiffServ (*Differentiated Service*) yang menerapkan prioritas pengiriman pada paket-paket suara dapat meminimalisir terjadinya *packet loss*.



## BAB 5

### KESIMPULAN

1. Pada pengujian internal LAN UI, nilai objektif QoS pada sisi *uplink* dan *downlink* hampir sama pada G.711- $\mu$ Law dan GSM 06.10 untuk parameter total *delay* rata-rata yaitu antara 40 ms sampai 40,84 ms dengan *packet loss* hampir 0%. Sedangkan untuk *bandwidth* rata-rata, G.711  $\mu$ -Law berkisar antara 76,72 Kbps sampai 79,42 Kbps sedangkan *bandwidth* GSM 06.10 berkisar antara 25,82 Kbps sampai 26,52 Kbps.
2. Pada pengujian sisi *downlink* jaringan Indosat, nilai *delay* rata-rata antara G.711  $\mu$ -Law dan GSM 06.10 hampir sama yaitu antara 20,19 ms sampai 20,29 ms dan *packet loss* keduanya adalah 0%. Sedangkan untuk parameter *bandwidth* rata-rata, G.711- $\mu$ Law bernilai 78 Kbps sedangkan GSM 06.10 bernilai 28,81 Kbps.
3. Pada pengujian sisi *uplink* jaringan Indosat, nilai rata-rata *delay* untuk G.711  $\mu$ -Law mencapai 52,28 ms sedangkan GSM 06.10 hanya 22,33 ms. Sementara untuk rata-rata *bandwidth*, pada G.711  $\mu$ -Law bernilai 34,97 Kbps sedangkan pada GSM 06.10 bernilai 25,8 Kbps. *Packet loss* pada G.711  $\mu$ -Law mencapai 23,5% sedangkan pada GSM 06.0 mencapai 10,8%.
4. G.711  $\mu$ -Law memiliki *bit rate* yang jauh lebih tinggi dibandingkan GSM sehingga hanya baik digunakan pada jaringan internet dengan *bandwidth* tinggi di kedua sisi, baik *downlink* dan *uplink*. Sedangkan *codec* GSM 06.10 memiliki *bandwidth* yang lebih rendah sehingga lebih fleksibel untuk digunakan pada jaringan internet dengan *bandwidth* terbatas.
5. Kualitas VoIP dipengaruhi oleh *bandwidth* yang tersedia yaitu semakin besar *bandwidth* yang disediakan, maka kualitas akan semakin baik.

## DAFTAR ACUAN

- [1] Omiya, Isao. (2005). *The Voip World in Japan*. NTT Communications Corp.
- [2] The Group of Experts on IP Telephony/ITU–D. (2003). *The Essential Report in IP Telephony*. International Telecommunication Union.
- [3] Wikipedia. *Voice over IP*. [http://id.wikipedia.org/wiki/Voice\\_over\\_IP](http://id.wikipedia.org/wiki/Voice_over_IP)
- [4] Raharja, Anton. (2006). *VoIP Fundamental*. <http://www.voiprakyat.or.id>
- [5] *Protokol Internet*. [http://id.wikipedia.org/wiki/Internet\\_Protocol](http://id.wikipedia.org/wiki/Internet_Protocol)
- [6] Nugroho, Djunda Afief. (2008). *Rancang Bangun dan Unjuk Kerja Uplink Video Telephony Berbasis SIP Menggunakan H.263 dan H.263v2*.
- [7] *Transmission Control Protocol*.  
[http://id.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](http://id.wikipedia.org/wiki/Transmission_Control_Protocol)
- [8] *User Datagram Protocol* <http://id.wikipedia.org/wiki/Udp>
- [9] *Real-time Transport Protocol*. [http://en.wikipedia.org/wiki/Real-time\\_Transport\\_Protocol](http://en.wikipedia.org/wiki/Real-time_Transport_Protocol)
- [10] Atmono, Widi. (2008). *Rancang Bangun Security pada Sistem VoIP Opensource Trixbox*.
- [11] Chen ,Jyh-Cheng dan Tao Zhang. (2004). *IP-Based Next-Generation Wireless Networks, System, Architecture, and Protocols*. John Wiley & Sons.
- [12]Toncar, Vladimir. *VoIP Basics: Overview of Audio Codecs*.  
[http://toncar.cz/Tutorials/VoIP/VoIP\\_Basics\\_Overview\\_of\\_Audio\\_Codecs.html](http://toncar.cz/Tutorials/VoIP/VoIP_Basics_Overview_of_Audio_Codecs.html)
- [13] Kristo Lehtonen. (2003). *T-61.246 Digital Signal Processing and Filtering GSM Codec*.  
[www.cis.hut.fi/Opinnot/T61.246/Kutri2003/lehtonen\\_gsmdoc.pdf](http://www.cis.hut.fi/Opinnot/T61.246/Kutri2003/lehtonen_gsmdoc.pdf)
- [14] Basuki, Mudji. (2007). *Voice over IP*. <http://i-networking.net/wp-content/uploads/2007/07/voice-over-ip-voip.pdf>
- [15] Prokkola, Jarmo&Mikko Hanski. (2005). *QoS Measurement Methods and Tools*.<http://www.ew.thales.no/1.2%20QoS%20Measurements%20Methods%20and%20Tools.pdf>

## DAFTAR PUSTAKA

- Abadi, Kristoforus. (2006). *Quo Vadis Regulasi VoIP di Indonesia*. <http://kristha.blogspot.com/2006/07/quo-vadis-regulasi-voip-di-indonesia.html>
- Cisco. *Voice Over IP - Per Call Bandwidth Consumption*.  
[http://www.cisco.com/en/US/tech/tk652/tk698/technologies\\_tech\\_note09186a0080094ae2.shtml](http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml)
- Draft Revised ITU-T Recommendation G.114., “*One-way Transmission Time*”.  
ITU-T
- Jim Van Meggelen, et all. (2007). *Asterisk The Future of Telephony*. O’Reilly.
- Ludfy, Akhmad. (2005). *Protocol Signaling SIP (Session Initiation Protocol)*
- Mubarok, Rudi Syahru, et all. (2005). *Perbandingan Kinerja Speech Codec G.711 dan GSM pada Implementasi Softswitch dengan Protokol SIP*. ITB.
- Raharja, Anton & Asoka Wardhana. (2008). *Administration Guide Briker*.  
<http://www.briker.org>