

**RANCANG BANGUN DAN UNJUK KERJA
MOBILE VOIP BERBASIS *SESSION INITIATION*
PROTOCOL DENGAN MENGGUNAKAN *CODEC*
G.711, GSM, DAN iLBC**

SKRIPSI

Oleh

MUHAMMAD ZUHDAN

04 04 03 0636



**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP, 2007/2008**

**RANCANG BANGUN DAN UNJUK KERJA
MOBILE VOIP BERBASIS *SESSION INITIATION*
PROTOCOL DENGAN MENGGUNAKAN *CODEC*
G.711, GSM, DAN iLBC**

Oleh

MUHAMMAD ZUHDAN

04 04 03 0636



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GENAP, 2007/2008**

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

RANCANG BANGUN DAN UNJUK KERJA MOBILE VOIP BERBASIS SESSION INITIATION PROTOCOL DENGAN MENGUNAKAN CODEC G.711, GSM, DAN iLBC

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada pendidikan Sarjana S1 Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan duplikasi dari skripsi yang telah dipublikasikan dan atau pernah dipakai untuk skripsi di lingkungan universitas Indonesia maupun di perguruan tinggi atau instansi manapun, kecuali bagian yang sumber informasinya telah dicantumkan sebagaimana mestinya.

Depok, 24 Maret 2008

Muhammad Zuhdan

NPM 0404030636

LEMBAR PENGESAHAN

Skripsi dengan judul:

**RANCANG BANGUN DAN UNJUK KERJA MOBILE VOIP
BERBASIS SESSION INITIATION PROTOCOL DENGAN
MENGUNAKAN CODEC G.711, GSM, DAN iLBC**

Dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia dan telah disidangkan pada tanggal 25 Maret 2008.

Depok, 24 Maret 2008

Dosen Pembimbing

M. Suryanegara, S.T., M.Sc.

NIP. 040 705 0189

UCAPAN TERIMA KASIH

Puji syukur hanya kepada ALLAH SWT, Yang Maha Kasih, sehingga skripsi ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada :

M. Suryanegara, S.T., M.Sc.

Selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan pengarahan, diskusi dan bimbingan serta persetujuan sehingga seminar ini dapat selesai dengan baik.

Selain itu penulis juga mengucapkan terima kasih kepada:

1. Kedua orang tua serta kakak-kakak saya yang telah memberikan doa dan dukungan moril maupun materi sehingga tugas ini dapat diselesaikan dengan baik.
2. Bapak Agus Awaludin yang telah memberikan izin pemakaian fasilitas PPSI serta telah memberikan sedikit banyak masukan dalam pengerjaan skripsi ini.
3. Rekan-rekan seperjuangan, Jusril A. Hidayat, M. Ginta Mardalin, Ganis Zulfa S., Aji Teguh P, dan Djunda A. Nugraha atas dukungan dan kebersamaan selama ini.
4. Rekan-rekan elektro khususnya angkatan 2004 atas semangat yang diberikan kepada penulis.

Muhammad Zuhdan

NPM 04 04 03 0636

Departemen Teknik Elektro

Dosen Pembimbing

M.Suryanegara, S.T., M.Sc.

**RANCANG BANGUN DAN UNJUK KERJA *MOBILE VOIP*
BERBASIS *SESSION INITIATION PROTOCOL* DENGAN
MENGUNAKAN *CODEC G.711, GSM, DAN iLBC***

ABSTRAK

Teknologi telekomunikasi telah berkembang dengan pesat. Sekarang teknologi seluler juga sudah memanfaatkan sistem *packet-switched* yang dapat dimanfaatkan untuk berbagai keperluan, salah satunya adalah VoIP. Selain dunia telekomunikasi, teknologi *open source* juga telah berkembang dengan pesat sehingga memudahkan pengembang untuk membangun berbagai sistem dan aplikasi yang bermanfaat. Perangkat lunak yang bersifat *open source ini* pada umumnya bisa didapatkan secara gratis.

Kedua teknologi tersebut bisa dimanfaatkan untuk membuat sebuah sistem *mobile VoIP*. *Voice over IP* merupakan teknologi komunikasi suara yang bekerja dengan memecah suara menjadi paket-paket data yang kemudian ditransfer melalui jaringan IP. Pada sebuah sistem VoIP, diperlukan sebuah *signaling protocol* seperti H.323 dan SIP. SIP merupakan protokol masa depan karena merupakan protokol utama pada sistem *IP Multimedia Subsystem*.

Mobile VoIP dibangun dengan memanfaatkan perangkat *open source* sebagai SIP *server* serta *client* berupa ponsel yang mendukung paket java JSR-180. Unjuk kerja dari sistem telah dievaluasi dan dianalisis dengan memanfaatkan *codec* yang berbeda serta dijalankan pada jaringan GPRS dari dua operator yang berbeda.

Codec GSM-FR memberikan performa yang paling optimum dengan *jitter* sebesar 4,39 ms sampai 12,76 ms dengan *bandwidth* 25,56 kbps sampai 28,79 kbps.

Kata kunci : VoIP, SIP, JSR-180, GPRS

Muhammad Zuhdan

Counselor :

NPM 04 04 03 0636

M.Suryanegara, S.T., M.Sc.

Electrical Engineering Department

**SESSION INITIATION PROTOCOL-BASED MOBILE-VOIP
DEVELOPMENT AND PERFORMANCE USING
G.711, GSM, AND ILBC CODECS**

ABSTRACT

Telecommunication technology has been developed rapidly. Nowadays, packet-switched system has been implemented in cellular technology which can be used for many purposes, one of them is VoIP. Open source technology is also developing so fast. Developers are very pleased with it because they can create and build new useful system and application. Today, we could get many open source software freely.

Both technology above are very useful and could be used to build a mobile VoIP system. Voice over IP is a speech communication technology that divides speech into IP packets and then transports it through IP network. VoIP system use signaling protocol such as H.323 and SIP. SIP is a future signaling protocol because it is a main protocol in IP Multimedia Subsystem.

Mobile VoIP system was build by using open source software as SIP server, and cell phones as clients which can support JSR-180 java package. This system has been tested, analyzed, and evaluated by using different CODECS and operated on two different GPRS network provider. Mobile VoIP system itself is build to give optimum performance that is with small jitter, low bandwidth, good voice quality, and low operating cost.

GSM-FR codec gave the best performance with *jitter* number within the interval 4,39 ms and 12,76 ms, and with bandwidth consumption between 25,56 kbps and 28,79 kbps.

Keywords : VoIP, SIP, JSR-180, GPRS

DAFTAR ISI

HALAMAN JUDUL	ii
PERNYATAAN KEASLIAN SKRIPSI.....	iii
LEMBAR PENGESAHAN	iv
UCAPAN TERIMA KASIH.....	v
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
DAFTAR SINGKATAN	xiii
BAB 1 PENDAHULUAN	1
1.1. LATAR BELAKANG	1
1.2. PERUMUSAN MASALAH.....	2
1.3. TUJUAN.....	2
1.4. PEMBatasan MASALAH	2
1.5. METODOLOGI PENELITIAN	2
1.6. SISTEMATIKA PENULISAN	2
BAB 2 VOICE OVER INTERNET PROTOCOL.....	4
2.1. TEKNOLOGI VOICE OVER INTERNET PROTOCOL.....	4
2.1.1. Keuntungan dan Parametar Kualitas VoIP	5
2.2. <i>SESSION INITIATION PROTOCOL</i>	7
2.2.1. Pengertian Session Initiation Protocol.....	7
2.2.2. <i>SIP Server</i>	8
2.2.3. Penamaan dan Pengalamatan SIP.....	10
2.2.4. <i>SIP Message</i>	11
2.2.5. Pembentukan dan Terminasi Sebuah Sesi	13
2.3. PROTOKOL-PROTOKOL LAIN YANG TERKAIT	14
2.3.1. <i>Session Description Protocol</i>	14
2.3.2. <i>Real-time Transport Protocol</i>	17
2.3.3. <i>RTP Control Protocol</i>	18
BAB 3 PERANCANGAN <i>MOBILE</i> VoIP	19
3.1. ARSITEKTUR <i>MOBILE</i> VoIP.....	19

3.1.1. Sisi <i>Client</i>	20
3.1.2. Sisi <i>Server</i>	22
3.2. TOPOLOGI JARINGAN	26
3.2.1. Topologi VoIP.....	26
3.2.2. Topologi GPRS	27
3.3. <i>CODEC</i> YANG DIGUNAKAN.....	28
3.3.1. G.711 (a-law dan u-law)	28
3.3.2. GSM.....	29
3.3.3. iLBC	30
BAB 4 ANALISIS DAN UJICOBA.....	32
4.1. MATRIKS ANALISIS	32
4.2. ANALISIS HASIL UJICOBA DAN KOMPARASI	32
4.2.1. Komparasi Performa Sistem Berdasarkan <i>Codec</i>	33
4.2.2. Komparasi Performa Sistem Berdasarkan Operator	38
4.3. PEMANFAATAN SISTEM.....	44
4.3.1. Solusi Telepon Murah	45
4.3.2. Sambungan Langsung Internasional dengan Harga Miring	45
BAB 5 KESIMPULAN.....	46
DAFTAR ACUAN	47

DAFTAR GAMBAR

Gambar 2.1. Internet Multimedia <i>Protocol stack</i> [1].....	4
Gambar 2.2. Contoh sebuah SIP sederhana.....	8
Gambar 2.3. Contoh panggilan SIP dengan menggunakan <i>proxy server</i>	9
Gambar 2.4. Contoh panggilan SIP menggunakan <i>redirect server</i>	9
Gambar 2.5. Contoh proses registrasi SIP	10
Gambar 2.6. Sesi SIP pada saat <i>proxy mode</i>	14
Gambar 2.7. Sesi SIP pada saat <i>redirect mode</i>	14
Gambar 2.8. RTP <i>packet header</i> [1]	17
Gambar 3.1. Arsitektur umum <i>mobile VoIP</i> yang dibangun.....	19
Gambar 3.2. Diagram alir penggunaan Fring	21
Gambar 3.3. Rancang bangun SIP <i>server</i> 152.118.184.17.....	23
Gambar 3.4. Diagram topologi VoIP yang disederhanakan [5]	26
Gambar 3.5. Arsitektur jaringan GPRS [7]	27
Gambar 3.6. Struktur protokol yang digunakan untuk transfer data pada GPRS [6]	28
Gambar 3.7. Diagram dari <i>codec</i> GSM-FR LPC-RPE [8]	29
Gambar 3.8. Diagram alir encoder ILBC [9].....	30
Gambar 4.1. Komparasi <i>jitter</i> menggunakan iLBC	33
Gambar 4.2. Komparasi <i>bandwidth</i> menggunakan iLBC	33
Gambar 4.3. Grafik komparasi <i>jitter uplink client</i> A dengan <i>codec</i> G.711	34
Gambar 4.4. Grafik komparasi <i>bandwidth uplink client</i> A dengan <i>codec</i> G.711	34
Gambar 4.5. Grafik komparasi <i>jitter uplink client</i> B dengan <i>codec</i> G.711	35
Gambar 4.6. Grafik komparasi <i>bandwidth uplink client</i> B dengan <i>codec</i> G.711	35
Gambar 4.7. Grafik komparasi <i>jitter</i> pada <i>uplink client</i> A dengan <i>codec</i> GSM..	36
Gambar 4.8. Grafik komparasi <i>jitter</i> pada <i>uplink client</i> B dengan <i>codec</i> GSM..	36
Gambar 4.9. Grafik komparasi <i>bandwidth</i> pada <i>uplink client</i> A dengan <i>codec</i> GSM.....	37
Gambar 4.10. Grafik komparasi <i>bandwidth</i> pada <i>uplink client</i> B dengan <i>codec</i> GSM	37

Gambar 4.11. Grafik komparasi <i>jitter</i> pada <i>uplink client</i> A menggunakan Telkomsel	38
Gambar 4.12. Grafik komparasi <i>bandwidth</i> pada <i>uplink client</i> A menggunakan Telkomsel	39
Gambar 4.13. Grafik komparasi <i>jitter</i> pada <i>uplink client</i> B menggunakan Telkomsel	39
Gambar 4.14. Grafik komparasi <i>bandwidth</i> pada <i>uplink client</i> B menggunakan Telkomsel	40
Gambar 4.15. Grafik komparasi <i>jitter</i> pada <i>uplink client</i> A menggunakan Indosat	41
Gambar 4.16. Grafik komparasi <i>bandwidth</i> pada <i>uplink client</i> A menggunakan Indosat	41
Gambar 4.17. Grafik komparasi <i>jitter</i> pada <i>uplink client</i> B menggunakan Indosat	42
Gambar 4.18. Grafik komparasi <i>bandwidth</i> pada <i>uplink client</i> B menggunakan Indosat	42

DAFTAR TABEL

Tabel 2.1. Struktur dari SIP <i>message</i>	11
Tabel 2.2 Metode-metode umum untuk SIP <i>request</i>	12
Tabel 2.3. Status-code pada SIP <i>response</i> [2].....	13
Tabel 2.4. contoh SDP <i>message</i> dengan berbagai <i>field</i>	15
Tabel 2.5. SDP <i>field</i> dan urutannya.....	16
Tabel 2.6. Tipe-tipe paket RTCP [1].....	18
Tabel 3.1. <i>Port-port</i> yang digunakan pada <i>Mobile VoIP</i>	20
Tabel 3.2. asterisk.conf.....	24
Tabel 3.3. rtp.conf.....	24
Tabel 3.4. sip.conf.....	25
Tabel 3.5. sip_additional.conf.....	25
Tabel 3.6. sip_general_additional.conf.....	26
Tabel 4.1. Nilai parameter-parameter <i>mobile VoIP</i> menggunakan simpati-Telkomsel.....	43
Tabel 4.2. Nilai parameter-parameter <i>mobile VoIP</i> menggunakan Indosat-m3.....	43

DAFTAR SINGKATAN

BR	<i>Bit Rate</i>
BW	<i>Bandwidth</i>
CODEC	<i>Coder Decoder</i>
DMZ	<i>De-Militerized Zone</i>
GGSN	<i>Gateway GPRS Support Node</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile</i>
GSM-FR	<i>GSM Full Rate</i>
HTTP	<i>Hyper Text Transport Protocol</i>
IETF	<i>Internet Engineering Task Force</i>
iLBC	<i>Internet Low Bit-rate Codec</i>
IP	<i>Internet Protocol</i>
LPC-RPE	<i>Linear Prediction Coding with Regular Pulse Excitation</i>
NAT	<i>Network Address Transverse</i>
PBX	<i>Private Branch Exchange</i>
PC	<i>Personal Computer</i>
Ponsel	<i>Telepon seluler</i>
PSTN	<i>Public Swithed Telephone Network</i>
QoS	<i>Quality of Service</i>
RFC	<i>Request for Comments</i>
RTCP	<i>Real-time Transport Control Protocol</i>
RTP	<i>Real-time Transport Protocol</i>
SDP	<i>Session Description Protocol</i>
SGSN	<i>Serving GPRS Support Node</i>
SIP	<i>Session Initiation Protocol</i>
SIPS	<i>SIP Secure</i>
SLI	<i>Sambungan Langsung Internasional</i>
TCP	<i>Transmission Control Protocol</i>
UA	<i>User Agent</i>
UAC	<i>User Agent Client</i>

UAS	<i>User Agent Server</i>
UDP	<i>User Datagram Protocol</i>
UE	<i>User Equipment</i>
URI	<i>Uniform Resource Identifier</i>
VoIP	<i>Voice over IP</i>



BAB 1

PENDAHULUAN

1.1. LATAR BELAKANG

Perkembangan teknologi informasi sampai saat ini tergolong sangat pesat khususnya teknologi seluler. Teknologi GPRS dan teknologi generasi ketiga memungkinkan pengiriman data dengan *packet switching* dengan kecepatan sampai 2 Mbps [7]. Teknologi ini memungkinkan konsumen untuk melakukan berbagai aplikasi seperti *video call*, *real-time video sharing*, *content download*, VoIP, dan berbagai aplikasi multimedia lainnya.

Sampai sekarang, pemanfaatan teknologi tersebut masih terbatas pada hal-hal yang bersifat hiburan. Sebenarnya, teknologi ini bisa dimanfaatkan untuk aplikasi yang lebih berguna dan memiliki nilai ekonomis. Salah satu pemanfaatannya adalah untuk aplikasi VoIP. VoIP menawarkan alternatif komunikasi yang beda dengan komunikasi konvensional yang bersifat *circuit switching* [7]. Teknologi VoIP mengadopsi teknologi *packet switching* dimana suara terlebih dahulu dipecah menjadi paket-paket IP dan kemudian mengirimnya melalui jaringan publik sehingga VoIP tidak mengenal batas wilayah [4].

Selain dunia seluler, perkembangan yang pesat juga ditunjukkan oleh teknologi *open source*. Teknologi ini memungkinkan pengembang untuk membangun sebuah sistem yang bersifat *free* dan mudah untuk dikembangkan. Selain perangkat lunak yang bersifat *open source*, program berbasis java banyak ditawarkan secara gratis. Asterisk merupakan salah satu perangkat lunak *open source* yang dapat digunakan sebagai *server* VoIP, sedangkan Fring merupakan program gratis yang dapat dimanfaatkan sebagai *interface* VoIP.

Dua teknologi ini dapat diintegrasikan untuk membuat sebuah teknologi *mobile* VoIP, yaitu VoIP yang dapat digunakan melalui sebuah ponsel. Prinsip dari sistem *mobile* VoIP ini sebenarnya sederhana yaitu dengan membangun sebuah *server* yang mempunyai IP publik sehingga dapat diakses oleh ponsel melalui jaringan GPRS.

1.2. PERUMUSAN MASALAH

Konsep komunikasi VoIP diimplementasikan pada teknologi seluler sehingga dapat diakses dari mana saja. Untuk mengakses sistem ini digunakan jaringan GPRS.

1.3. TUJUAN

Tujuan dari penyusunan skripsi ini adalah untuk merancang dan membangun sebuah sistem *mobile* VoIP, serta menganalisis unjuk kerja sistem yang dibangun berdasarkan *codec* dan operator

1.4. PEMBATAAN MASALAH

Masalah yang dibahas pada skripsi ini dibatasi pada rancang bangun sebuah sistem *mobile* VoIP dengan sebuah *standalone server* yang hanya dapat diakses menggunakan jaringan internet dan jaringan GPRS. *Server* dibangun dengan menggunakan perangkat lunak yang bersifat *open source* dan *free*.

1.5. METODOLOGI PENELITIAN

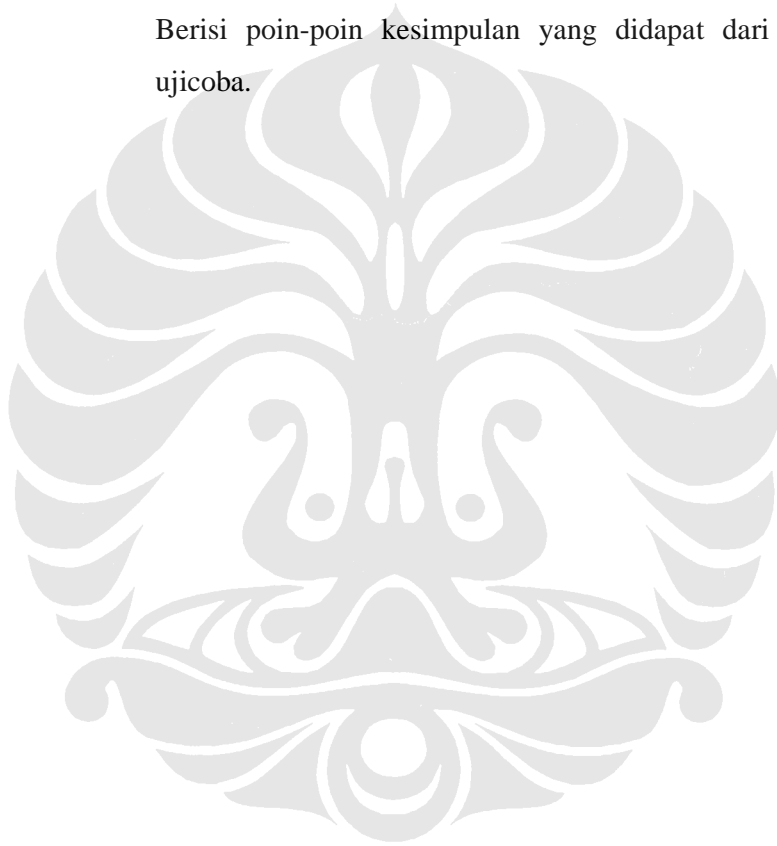
Penelitian dilakukan dengan melakukan studi literatur, kemudian melakukan implementasi dengan membangun sebuah *server* VoIP, lalu kemudian melakukan evaluasi terhadap hasil unjuk kerja dari sistem *mobile* VoIP yang telah dibangun.

1.6. SISTEMATIKA PENULISAN

Sistematika penulisan pada laporan kerja praktek ini ialah sebagai berikut :

- Bab 1 *Pendahuluan*
Berisikan tentang latar belakang, tujuan, pembatasan masalah, dan sistematika penulisan.
- Bab 2 *Voice Over Internet Protocol*
Membahas mengenai teori VoIP secara umum, *Session Initiation Protocol* sebagai salah satu *signaling* pada VoIP, serta protokol-protokol lain yang terkait dengan sistem *mobile* VoIP.

- Bab 3 Perancangan *Mobile VoIP*
Bab ini membahas tentang perancangan sistem *mobile VoIP* mulai dari arsitekturnya, konfigurasi *SIP server*, ponsel sebagai *client*, topologi jaringan GPRS serta *codec* yang digunakan.
- Bab 4 Analisis dan Ujicoba
Bab ini membahas mengenai hasil pengujian dan evaluasi unjuk kerja dari sistem *mobile VoIP* yang dijalankan pada jaringan GPRS.
- Bab 5 Kesimpulan
Berisi poin-poin kesimpulan yang didapat dari hasil analisis dan ujicoba.



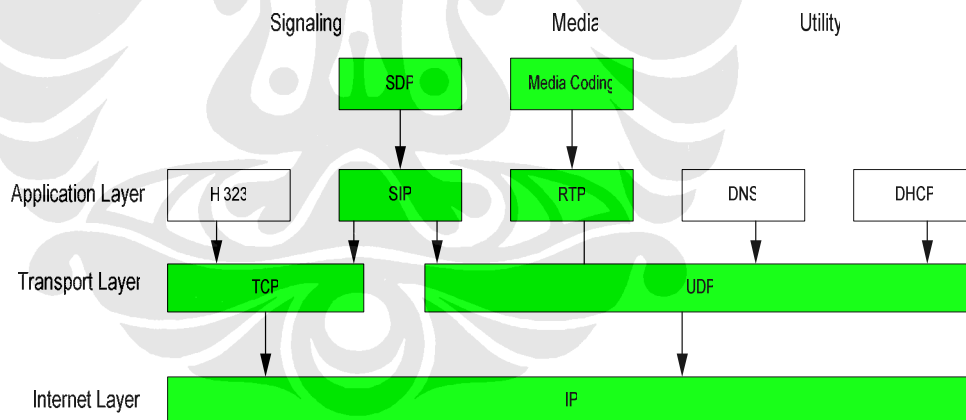
BAB 2

VOICE OVER INTERNET PROTOCOL

2.1. TEKNOLOGI VOICE OVER INTERNET PROTOCOL

Voice over IP merupakan layanan suara pada sebuah jaringan IP. Jika dilihat dari sisi *end-user*, VoIP memiliki sedikit perbedaan dibandingkan dengan layanan suara biasanya. Akan tetapi dari sisi operator, VoIP dapat memberikan penurunan biaya yang cukup signifikan. Biaya yang rendah ini akan mendorong terbentuknya *provider-provider* baru sehingga pada akhirnya dapat mempengaruhi biaya yang harus dibayarkan oleh *user* [4].

VoIP bukan hanya sebuah layanan komunikasi suara, akan tetapi teknologi ini pun bisa menyediakan layanan komunikasi video. Prinsip kerja dari VoIP adalah dengan mengkonversi data analog ke *digital* kemudian memecahnya menjadi paket-paket IP untuk kemudian dikirimkan melalui jaringan IP. **Gambar 2.1** memperlihatkan susunan *stack* pada internet media *protocol*.



Gambar 2.1. Internet Multimedia *Protocol stack* [1]

Pada **Gambar 2.1**, protokol yang berwarna hijau merupakan protokol yang digunakan pada VoIP berbasis SIP. Dari gambar tersebut dapat dilihat bahwa VoIP menggunakan TCP dan UDP sebagai *transport layer*-nya.

- TCP

TCP berfungsi untuk menjaga reliabilitas dari *connection-oriented transport* over IP. Konsep dasar kerja TCP adalah mengirim dan menerima informasi yang berbentuk semen-segmen informasi dengan panjang data bervariasi pada suatu datagram internet. TCP menjamin reliabilitas hubungan komunikasi dengan 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. Jika sinyal ACK ini tidak diterima pada interval pada waktu tertentu, maka data akan dikirimkan kembali. Pada sisi penerima, nomor urut tadi berguna untuk mencegah kesalahan urutan data dan duplikasi data.

- UDP

UDP menyediakan sebuah *unreliable transport* dan bersifat *connection-less*. Kompleksitas pada TCP sebagian besar tidak terdapat pada UDP. Seperti *sequence number*, ACK, dan *window sizes*. UDP pada VoIP digunakan untuk mengirimkan audio *stream* karena pada pengiriman audio *streaming* yang berlangsung terus menerus lebih mementingkan kecepatan pengiriman data agar tiba di tujuan tanpa memperhatikan adanya paket yang hilang walaupun mencapai 50% dari jumlah paket yang dikirimkan [3].

2.1.1. Keuntungan dan Parametar Kualitas VoIP

Banyak keuntungan yang akan kita dapatkan dalam penggunaan VoIP. Sudut pandangnya bisa dilihat sebagai teknologi pengganti PBX atau PSTN yang saat ini masih digunakan. Selain itu, kita juga bisa melihat dari potensi pengembangannya sebagai teknologi komunikasi yang lebih dinamis. Berikut ini adalah beberapa keuntungan yang kita dapatkan dari penggunaan VoIP:

- Harga Peralatan yang lebih murah
- Efisiensi *bandwidth*
- Biaya perawatan yang murah
- Munculnya aplikasi baru (*Value Added Services*)

Selain beberapa keuntungan yang dapat kita dapatkan dari adanya VoIP, terdapat pula beberapa permasalahan yang muncul dan masih terus dicari solusi yang terbaik. Permasalahan tersebut biasanya timbul dalam bingkai *Quality of Service* (QoS). Permasalahan tersebutlah yang kemudian menjadi kelemahan teknologi VoIP saat ini. Agar jaringan IP/internet dapat dengan sukses melewati paket data berupa suara ataupun video, jaringan ini harus dimodifikasi sedemikian rupa sehingga mampu memberikan *Quality of Service* sesuai permintaan aplikasi VoIP ini. Paramater tersebut ialah:

- *CODEC*

Codec merupakan faktor pertama yang menentukan kualitas dari sebuah panggilan. Pemilihan *CODEC* harus disesuaikan dengan jaringan dan peralatan yang tersedia agar dapat menghasilkan kualitas yang optimum.

- Jaringan

Kemampuan jaringan merupakan faktor yang mempengaruhi VoIP semenjak dari perencanaan disain, implementasi dan penggunaannya. Jaringan dapat mempengaruhi kualitas daalam berbagai cara antara lain *bandwidth*, *delay*, *jitter*, dan *packet loss*.

- *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.

- *Delay*

router dan jaringan IP yang memiliki karakteristik khusus yang menyulitkan pengontrolan *delay* dan variasinya (*jitter*).

- *Jitter*

Jitter pada intinya adalah variasi dalam *delay*, terjadi karena adanya perubahan terhadap karakteristik dari suatu sinyal sehingga menyebabkan terjadinya masalah terhadap data yang dibawa oleh sinyal tersebut [3].

- *Packet Loss*

Paket *loss* artinya hilangnya paket data yang sedang dikirimkan. Hilangnya data ini bisa disebabkan karena *Jitter* atau karena adanya

permasalahan di perangkat-perangkat jaringan seperti router yang terlalu sibuk, jalur komunikasi yang terlalu padat penggunaanya [3].

2.2. SESSION INITIATION PROTOCOL

2.2.1. Pengertian Session Initiation Protocol

Session Initiation Protocol (SIP) merupakan salah satu standar pensinyalan dan pengontrolan sesi dari *packet telephony* yang dikembangkan oleh IETF sebagai bagian dari *Internet Multimedia Conferencing Architecture* [1]. SIP merupakan sebuah *application-layer protocol* untuk membentuk, memodifikasi, dan menterminasi sebuah sesi multimedia [2]. Adapun dalam mengatur sebuah sesi komunikasi multimedia, SIP mempunyai beberapa fungsi penting yang antara lain [2]:

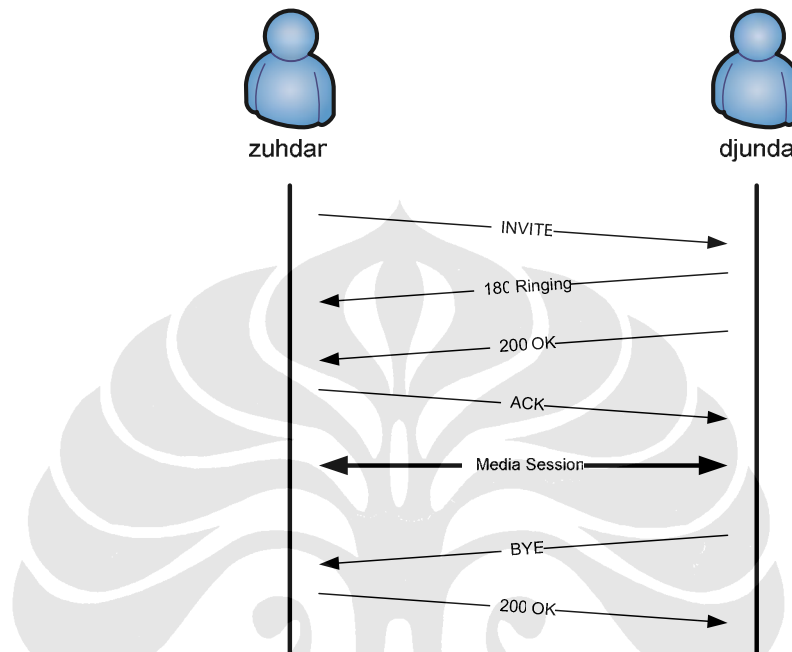
- Dapat menentukan lokasi *user*.
- Dapat menentukan apakah seorang *user* ingin berpartisipasi dalam sebuah sesi multimedia.
- Dapat mengetahui kemampuan terminal dari si *user*.
- Mempersiapkan sebuah sesi.
- Mengontrol sebuah sesi termasuk memodifikasi parameter-parameter ataupun fungsi-fungsi untuk memberikan servis pada sebuah sesi.

Seperti layaknya HTTP, SIP merupakan *client-server protocol* yang menggunakan *model* transaksi *request* dan *response*. Komponen-komponen utama dalam arsitektur SIP antara lain [2]:

- SIP *user agent*
UA merupakan sebuah *endpoint* seperti IP *phone*, PC, dan lain sebagainya. Sebuah UA dapat berfungsi sebagai *client* (UAC) dan *server* (UAS).
- SIP *redirect server*
Sebuah *redirect server* merupakan UAS yang menghasilkan *response* untuk mengalihkan sebuah *request* ke lokasi lainnya.
- SIP *proxy server*
Sebuah *proxy server* bertindak sebagai perantara diantara UA untuk mengirim SIP *message* ke *user* yang dituju

- SIP registrar

Sebuah registrar merupakan sebuah UAS yang memproses SIP *REGISTER request*. Registrar menyimpan pemetaan dari SIP *user name* ke *address* dan merupakan ujung awal dari *location service*.



Gambar 2.2. Contoh sebuah SIP sederhana

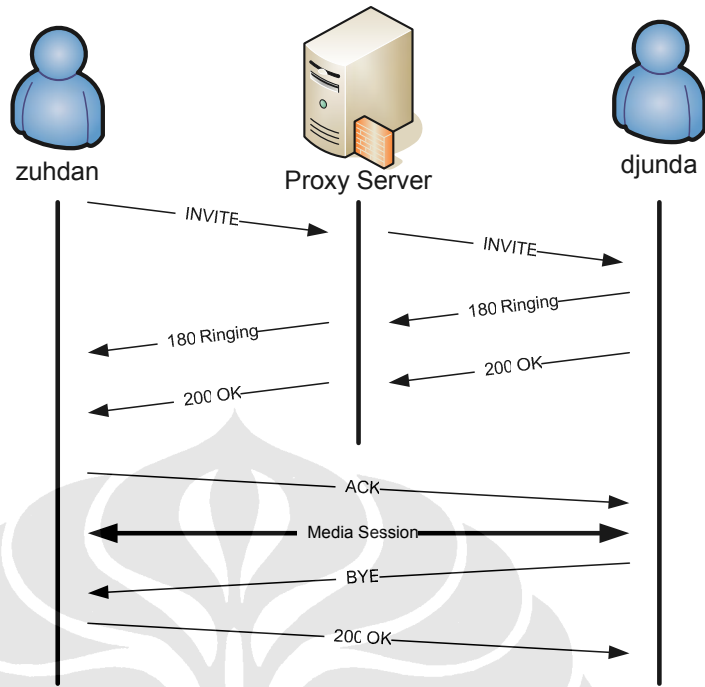
Pada **Gambar 2.2** dapat kita lihat gambaran sederhana dari SIP *request* seperti INVITE ACK, dan BYE, serta SIP *response* seperti 180 Ringing dan 200 OK. Penjelasan lebih lanjut mengenai SIP *message* dapat dilihat pada bagian 2.2.4.

2.2.2. SIP Server

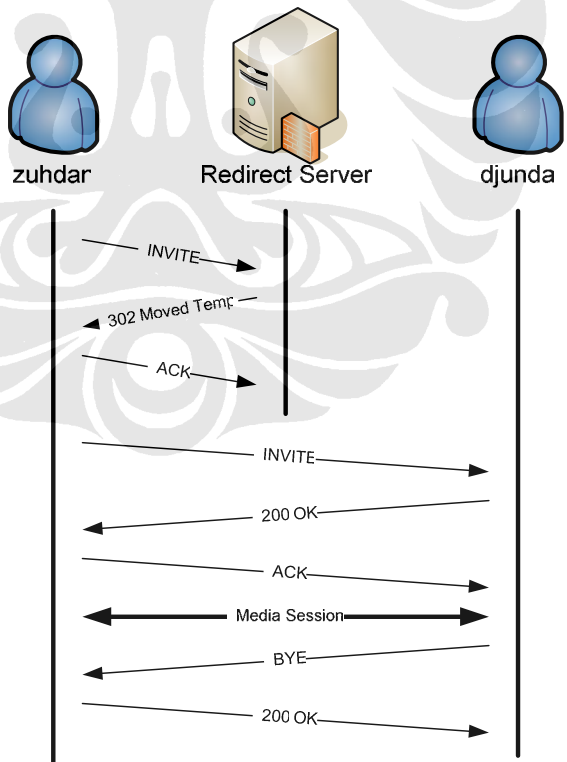
Seperti yang telah dijelaskan sebelumnya, SIP *server* merupakan sebuah aplikasi yang menerima SIP *request* dan memberikan respon terhadapnya.

- Proxy Server

Proxy Server biasanya mempunyai akses terhadap sebuah *database* atau *location service* dalam memproses *request*. *Database* yang digunakan oleh sebuah *proxy server* dapat terdiri dari berbagai tipe *database* [1]. Gambar 2.3 memperlihatkan sebuah *proxy server* bertindak sebagai fasilitator dari pertukaran SIP *message* diantara dua *user*.



Gambar 2.3. Contoh panggilan SIP dengan menggunakan *proxy server*



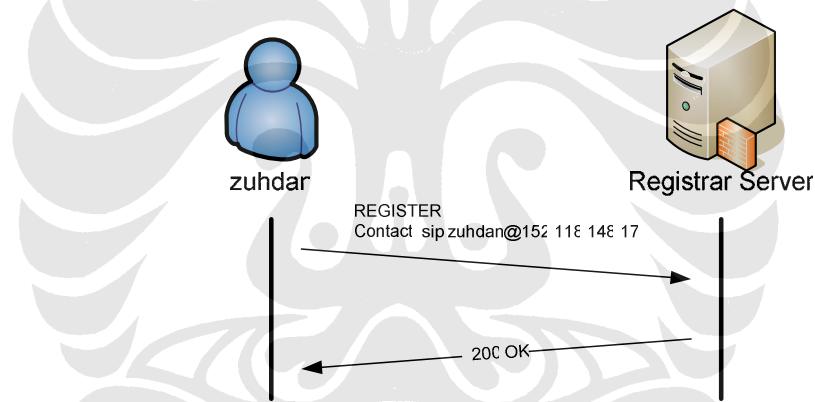
Gambar 2.4. Contoh panggilan SIP menggunakan *redirect server*

- *Redirect Server*

Server jenis ini merupakan SIP *server* yang memberikan respon terhadap *request* akan tetapi tidak meneruskannya. *Server redirect* mempunyai *database* sama halnya dengan *proxy server* yang berisi informasi penting untuk memproses terjadinya sesi komunikasi [1]. Perbedaan mendasar *redirect server* dengan *proxy server* dapat dilihat dengan membandingkan **Gambar 2.3** dan **Gambar 2.4**.

- *Registration Server (Registrar)*

Seperti yang telah dijelaskan di bagian 2.2.1, *server* ini hanya menerima SIP *REGISTER request* dari *user*. Informasi dari *user* kemudian dapat diakses oleh *server* lainnya dalam sistem administrasi yang sama. Proses registrasi *user* SIP dapat dilihat pada **Gambar 2.5**.



Gambar 2.5. Contoh proses registrasi SIP

2.2.3. Penamaan dan Pengalamatan SIP

Setiap *user* SIP dapat dibedakan secara unik menggunakan SIP *Uniform Resource Identifier (URI)* yang mirip dengan alamat email dimana terdapat *user name* dan *host name*. Bentuk umum dari URI adalah sebagai berikut.

`sip:user:password@host:port;uri-parameters?headers`

Sebagai contoh, alamat URI dari zuhdan adalah `sip:zuhdan@152.118.148.17:5060` dimana 152.118.148.17 merupakan *host* dari SIP *provider*. Sebuah *host* dapat diidentifikasi menggunakan baik IPv4/IPv6 atau sebuah FQDN (*Fully Qualified Domain Name*). Sebuah *secure* SIP URI

disebut SIPS URI menandakan sebuah cara yang aman dan terenkripsi untuk mengirimkan SIP *message*. Format dari SIPS URI sama dengan SIP URI akan tetapi bagian sip diganti dengan sips.

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 *case sensitive* kecuali ada pengaturan lain [2].

Bagian *port* merupakan nomor *port* dimana *message* akan dikirim. Sedangkan *uri-parameters* menentukan parameter-parameter URI seperti *transport*, *maddr*, *ttl*, *user*, *method*, dan *lr*. Jenis-jenis parameter ini tidak akan dijelaskan lebih lanjut dalam tulisan ini. Contoh dari SIP URI dengan parameter URI dapat dilihat sebagai berikut.

```
sip:zuhdan@152.118.148.17:5060;transport=udp;user=ip;method=INVITE;ttl=10;maddr=152.118.101.20?Subject=tugasakhir
```

Sebagai catatan, pada sebuah SIP URI, hanya bagian *host* yang bersifat *mandatory* [2], sehingga SIP URI yang simple dapat dilihat seperti contoh berikut.

```
sip:zuhdan@152.118.148.17
```

2.2.4. SIP Message

Seperti yang telah diketahui sebelumnya, SIP *message* ditulis dalam *cleartext* dengan kata lain SIP merupakan *text-based protocol*. Struktur SIP *message* dapat dilihat pada **Tabel 2.1**.

Tabel 2.1.

Struktur dari SIP *message*

Start-line	Header Field(s)
INVITE	INVITE sip:2002@152.118.148.17 SIP/2.0
From:	<sip:2000@152.118.148.17>;tag=4f7cc7f012fb1810999dc48bb2a699b9
To:	sip:2002@152.118.148.17
Via:	SIP/2.0/UDP 63.215.199.74:52752;iid=2;branch=z9hG4bK4f7cc7f012fb1810999cc48bb2a699b9;rport
CSeq:	4711 INVITE
Call-ID:	3476c7f0-12fb-1810-8bc8-c48bb2a699b9
Contact:	<sip:2000@63.215.199.74:52752>

```
User-Agent: fring
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: 328
```

Empty Line

Message Body
(optional)

```
v=0
o=- 1205468486 1205468486 IN IP4 63.215.199.74
s=OSS RTP Session
c=IN IP4 63.215.199.74
t=0 0
m=audio 53552 RTP/AVP 96 3 107 8 0 101
.....
```

Berdasarkan **Tabel 2.1** terlihat bahwa setiap SIP *message* terdiri dari:

- Sebuah *start-line*
- Satu atau lebih *header*
- Baris kosong
- Opsional *message body*

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. Sedangkan *response message* ditandai oleh status line yang berisi versi protokol SIP diikuti oleh *status-code* numerik dan penjelasan tekstualnya. Metode-metode umum yang sering digunakan pada SIP antara lain dapat dilihat pada **Tabel 2.2**.

Sebuah status-code merupakan 3 digit integer yang berfungsi untuk mengindikasikan respon dari sebuah *request*. Terdapat enam kelas SIP *response* dimana digit pertama dari *status-code* mengindikasikan kelas dari SIP *response* tersebut. Kelas-kelas dari SIP *response* dapat dilihat pada **Tabel 2.3**.

Tabel 2.2

Metode-metode umum untuk SIP *request*

INVITE	:	Digunakan <i>user</i> untuk mengundang <i>user</i> lainnya untuk melakukan sebuah sesi SIP
REGISTER	:	Digunakan untuk mendaftarkan informasi ke <i>server</i>
BYE	:	Digunakan untuk mengakhiri sesi
ACK	:	Digunakan untuk menkonfirmasi final <i>response</i>
CANCEL	:	Digunakan untuk membatalkan SIP <i>request</i>
OPTION	:	Digunakan untuk menanyakan kemampuan <i>server</i>

Tabel 2.3.*Status-code pada SIP response [2]*

Kelas	Nama kelas	Deskripsi
1xx	Informational	Mengindikasikan sebuah <i>request</i> sedang dalam proses
2xx	Success	Mengindikasikan metode pada sebuah <i>request</i> telah diterima
3xx	Redirection	Tindakan lebih jauh perlu dilakukan oleh pengirim untuk melengkapi <i>requestnya</i>
4xx	Client error	<i>Request</i> memiliki kesalahan sintak sehingga tidak dapat dipenuhi
5xx	Server error	<i>Server</i> gagal memenuhi <i>request</i> yang valid
6xx	Global error	<i>Request</i> tidak dapat dipenuhi di semua <i>server</i>

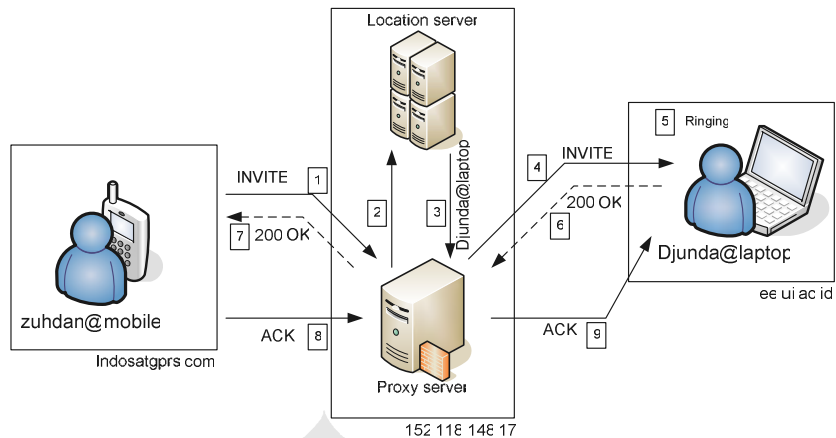
Header field digunakan untuk membawa informasi yang dibutuhkan untuk mengatur sebuah sesi SIP. Sebuah *header field* berisi nama *field* lalu diikuti titik dua (":") dan isi dari *field*. Beberapa *field* yang umum dipakai SIP antara lain seperti *To:*, *From:*, *Subject:*, *Via:*, *Contact:*, *Max-forward:*, *Cseq:*, *Call-ID:*, *User-Agent:*, dan lain sebagainya. Beberapa dari *header field* hanya digunakan pada *request message* dan beberapa hanya digunakan untuk *response message*.

Setelah *header field* baik *request* ataupun *response message*, terdapat sebuah *message body* dengan berbagai tipe dimana tipenya ditandai oleh *header field* yang berbeda. Sebagai contoh, *Content-Type:application/sdp* menandakan bawa *message body* merupakan deskripsi dari sebuah sesi dalam format SDP. Penjelasan lebih lanjut mengenai SDP dapat dilihat pada bagian 2.3.1.

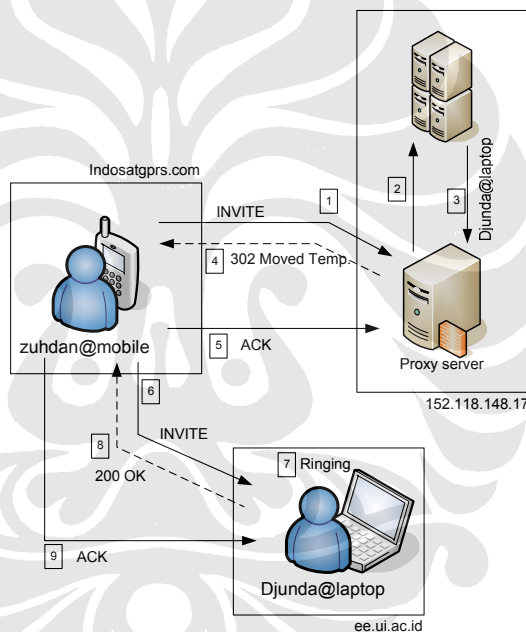
2.2.5. Pembentukan dan Terminasi Sebuah Sesi

Sebuah sesi SIP dapat dibentuk menggunakan *peer-to-peer mode* ataupun *server mode*. Pada model *peer-to-peer*, si pemanggil membuat panggilan langsung tanpa melalui *server*. Untuk hal ini si pemanggil harus mengetahui lokasi dari *user* yang akan dipanggilnya. Sedangkan untuk *server mode*, terdapat dua sistem yang digunakan, yaitu *SIP proxy server* dan *SIP redirect server*. Kedua jenis *server* ini sudah dijelaskan pada bagian 2.2.2.

Untuk mengetahui lebih jelas mengenai pembentukan sesi SIP menggunakan *server mode*, dapat dilihat pada **Gambar 2.6** dan **Gambar 2.7**.



Gambar 2.6. Sesi SIP pada saat *proxy mode*



Gambar 2.7. Sesi SIP pada saat *redirect mode*

2.3. PROTOKOL-PROTOKOL LAIN YANG TERKAIT

2.3.1. *Session Description Protocol*

SDP yang didefinisikan dengan RFC 2327 [1] dikembangkan oleh IETF untuk mendiskripsikan sesi multimedia. SDP lebih ke arah sintak deskripsi daripada sebuah protokol karena tidak mempunyai kemampuan negosiasi media secara menyeluruh [1]. Penggunaan SDP adalah dengan memasukkan SDP ke

dalam *message body* protokol lain. Seperti dijelaskan pada bagian 2.2.4. , SIP *message* bisa berisi SDP dengan penanda `Content-Type:application/sdp`.

Seperti yang dispesifikasikan oleh RFC 2327 [2], SDP terdiri atas:

- Nama dan tujuan dari sesi.
- Waktu aktivasi dan terminasi sesi.
- Media yang digunakan (tipe dan *encoding*).
- Informasi lainnya seperti alamat, nomor *port*, format media, dan sebagainya.

Mirip dengan SIP, SDP menggunakan *text coding* yang terdiri atas baris-baris yang disebut *fields*.

Contoh dari SDP *message* dapat dilihat pada **Tabel 2.4**.

Tabel 2.4.

Contoh SDP *message* dengan berbagai *field*

```
v=0
o=- 1205468486 1205468486 IN IP4 63.215.199.74
s=OSS RTP Session
c=IN IP4 63.215.199.74
t=0 0
m=audio 53552 RTP/AVP 96 3 107 8 0 101
a=rtpmap:96 AMR/8000
a=fmtp:96 mode-set=0
a=rtpmap:3 GSM/8000
a=rtpmap:107 iLBC/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Dari **Tabel 2.4** dapat dilihat bahwa format teks pada SDP dapat dirumuskan sebagai berikut [2]:

`<type>=<value>`

dimana `<type>` harus merupakan satu karakter dan bersifat *case sensitive*, sedangkan `<value>` berisi deskripsinya dan bersifat *case sensitive* juga. Bentuk umum dari SDP *message* dapat dirumuskan sebagai berikut [1]:

x=parameter1 parameter2 ... parameterN

Kumpulan *mandatory* dari SDP *field* dapat dilihat pada **Tabel 2.5.**

Tabel 2.5.

SDP *field* dan urutannya

<i>Field</i>	Nama	Keterangan
v=	Version	Nomor versi protokol
o=	Origin	Pemilik dan pengenalan sesi
s=	Subject	Nama sesi
i=	Information	Informasi mengenai sesi
u=	uri	URI
e=	Email	Alamat email
p=	Phone	Nomor telephone
c=	Connection	Informasi mengenai koneksi
b=	Bandwidth	Informasi <i>bandwidth</i>
t=	Time	Waktu mulai dan waktu berhenti sesi
r=	Repeat times	Jumlah pengulangan
z=	Time zones	Koreksi zona waktu
k=	Encryption key	Enkripsi
a=	Attributes	Baris atribut
m=	Media announcements	Informasi media
a=	Attributes	Atribut media

Menurut Alan B. Johnston di bukunya yang berjudul *SIP: Understanding the Session Initiation Protocol*, berkata bahwa pada umumnya, SIP menggunakan SDP *field* seperti *version*, *origin*, *subject*, *time*, *connection*, dan satu atau lebih *media* dan *attribute*. *Origin*, *subject*, dan *time* tidak digunakan oleh SIP tetapi digunakan untuk kepentingan kompatibilitas. Pada standar SDP, *subject field* merupakan *field* yang harus ada dan harus berisi setidaknya satu karakter, jika tak ada *subject*, dianjurkan menjadi *s=-*.

Dibuku yang sama Alan B. Johnston juga mengatakan bahwa SIP menggunakan *field-field connection*, *media*, dan *attribute* untuk mengatur terjadinya sebuah sesi diantara UA. Karena jenis media dan *codec* yang digunakan

merupakan bagian dari negosiasi koneksi, SIP dapat menggunakan SDP untuk menspesifikasi berbagai alternatif jenis media untuk menerima atau menolak jenis media tersebut secara selektif. Hal ini dapat dilihat pada **Tabel 2.4**.

2.3.2. Real-time Transport Protocol

RTP dikembangkan untuk memungkinkan terjadinya pengiriman paket secara *real-time* seperti video, *voice*, dan informasi lainnya melalui IP. RTP didefinisikan oleh IETF RFC 1889 [1]. RTP tidak menyediakan QoS akan tetapi RTP memungkinkan dilakukannya deteksi terhadap beberapa kelemahan seperti [1]:

- *Packet loss.*
- *Variable transport delay.*
- *Out of sequence packet arrival.*
- *Asymmetric routing.*

RTP merupakan protokol *application layer* yang menggunakan UDP sebagai *transport*-nya melalui IP. RTP menggunakan *bit-oriented header* mirip dengan UDP dan IP. *Packet header* versi kedua dari RTP dapat dilihat pada **Gambar 2.8**.

V	P	X	CC	M	PT	Sequence Number	Timestamp	SSRC
---	---	---	----	---	----	-----------------	-----------	------

Gambar 2.8. RTP packet header [1]

- *Version (V)*. Merupakan versi dari RTP.
- *Padding (P)*. Pada umumnya digunakan jika media *stream* dienkripsi.
- *Extension (X)*. Merupakan ekstensi tambahan yang mengikuti *header* yang dibuat oleh tipe *payload* tertentu.
- *CSCR count (CC)*.
- *Marker (M)* berfungsi menandai awal dari frame baru pada video.
- *Payload Type (PT)* merupakan 7-bit *field* yang menandai *codec* yang digunakan.
- *Sequence number* berfungsi untuk mendeteksi hilangnya *packet* dimana jumlah *sequence number* akan bertambah untuk setiap paket RTP yang dikirim.

- *Timestamps* mengindikasikan waktu relatif ketika *payload* di-*sample*.
- *Synchronization Source Identifier* (SSRCI) mengidentifikasi *sender* dari paket RTP.
- *CSCR Contributing Source Identifier*.

Packet loss dapat dideteksi menggunakan jarak antar *sequence number*, sedangkan *variable delay* atau *jitter* dapat dideteksi menggunakan *timestamps*.

2.3.3. RTP Control Protocol

RTCP merupakan protokol yang memungkinkan partisipan dalam sebuah sesi RTP saling mengirimkan laporan kualitas dan statistik serta saling bertukar informasi. Tipe-tipe dari paket RTCP dapat dilihat pada **Tabel 2.6**. Kegunaan dari laporan tersebut adalah untuk mengetahui kualitas dari koneksi yang dibuat termasuk informasi seperti [1]:

- Jumlah paket yang dikirim dan diterima
- Jumlah paket yang hilang
- *Jitter* dari paket

Tabel 2.6.

Tipe-tipe paket RTCP [1]

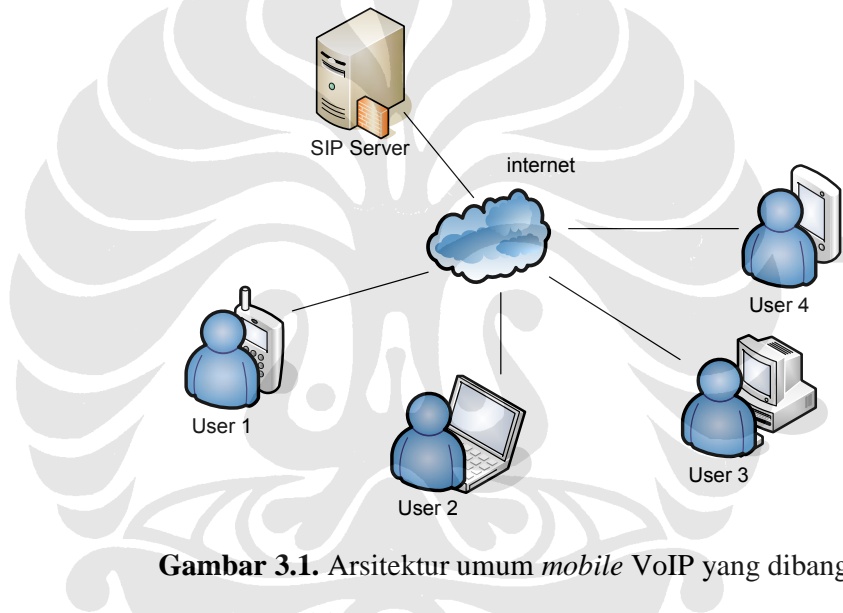
Tipe paket	Nama paket	Deskripsi
SR	<i>Sender report</i>	Dikirim oleh partisipan yang mengirim dan menerima paket RTP
RR	<i>Receiver report</i>	Dikirim oleh partisipan yang hanya menerima paket RTP
SDES	<i>Source description</i>	Berisi informasi mengenai partisipan pada suatu sesi
BYE	<i>Bye</i>	Dikirim untuk terminasi sesi RTP
APP	<i>Application specific</i>	Berisi profil tertentu

BAB 3

PERANCANGAN *MOBILE* VoIP

3.1. ARSITEKTUR *MOBILE* VoIP

Mobile VoIP yang akan dibangun didasarkan pada teknologi IMS. *Signaling protocol* yang digunakan adalah SIP, sedangkan pada lapisan *transport* menggunakan UDP. *Voice* akan dipecah menjadi paket-paket IP yang kemudian akan dikirimkan pada jaringan internet menggunakan protokol RTP. Gambaran umum arsitekturnya dapat dilihat pada **Gambar 3.1**.



Gambar 3.1. Arsitektur umum *mobile* VoIP yang dibangun

Pada arsitektur yang dibangun, tidak terdapat adanya *gateway*, sehingga setiap *user agent* harus dapat menerjemahkan paket-paket IP serta dapat *men-generate* dan menerjemahkan SIP *message* tersebut. Tidak semua ponsel memiliki kemampuan tersebut diatas. Sehingga ponsel yang dapat digunakan harus memiliki syarat-syarat tertentu. Syarat-syarat tersebut antara lain adalah mampu mengakses jaringan GPRS, mampu menerjemahkan paket-paket IP, memiliki format media dan *codec* tertentu, serta mampu *men-generate* SIP *message* (JSR-180).

Untuk syarat yang terakhir setiap vendor ponsel memiliki gaya tertentu dalam penyediaannya. Untuk Nokia™ contohnya, fungsi JSR-180 hanya

didukung oleh ponsel yang menggunakan platform Symbian s60 3rd edition, sedangkan untuk Sonyericsson™ hanya ponsel yang memiliki Java Platform-8 yang memiliki fasilitas JSR-180.

Pengaturan *port-port* yang digunakan oleh *server* pada arsitektur *mobile* VoIP ini dapat dilihat pada **Tabel 3.1**.

Tabel 3.1.

Port-port yang digunakan pada Mobile VoIP

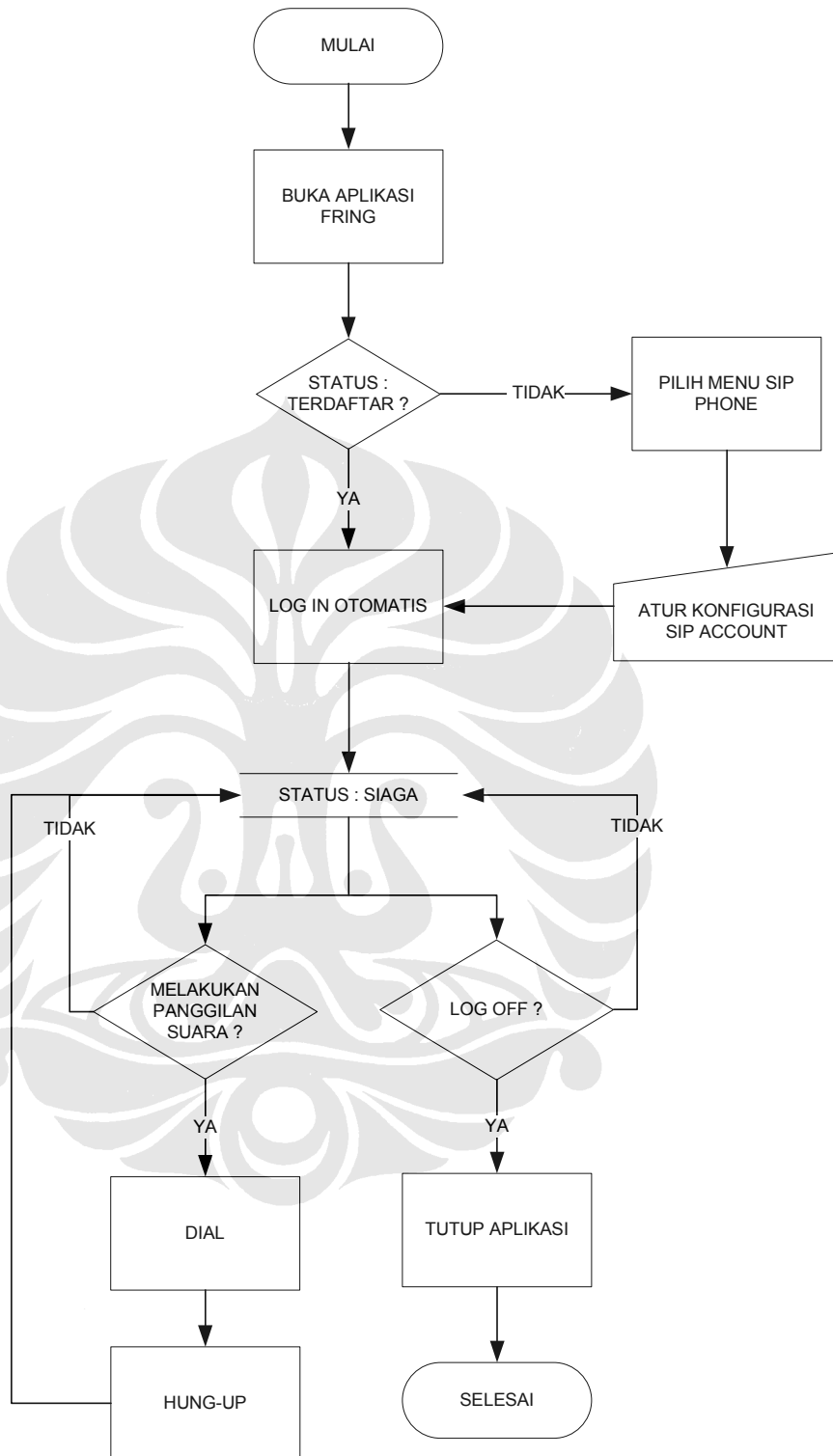
Nomor port	Jenis port	Fungsi
8080	TCP	Pengaturan transmisi melalui HTTP
5060	UDP	Pengaturan transmisi SIP/SDP
12300-12400	UDP	Kanal transmisi RTP dan RTCP

3.1.1. Sisi Client

Client yang digunakan terdiri dari dua buah ponsel 3G keluaran NOKIA™ dengan seri 6120 classic dan N73. Pemilihan kedua ponsel tersebut didasarkan pada kemampuan ponsel yang mendukung JSR-180 serta memiliki berbagai macam *codec* yang dibutuhkan.

Sedangkan pada kedua ponsel, digunakan perangkat lunak Fring sebagai antarmukanya. Fring merupakan perangkat lunak berbasis J2ME yang memiliki kemampuan untuk menghubungkan ponsel dengan berbagai VoIP *server* dengan berbagai protokol seperti gizmo, skype, gtalk, dan lain sebagainya. Tetapi yang paling penting adalah Fring dapat terhubung dengan SIP *server* dimana saja.

Alasan pemilihan Fring sebagai antarmuka pada ponsel didasarkan pada kemampuan Fring yang dapat terhubung dengan SIP *server* apapun dan bisa didapatkan secara gratis dari situs resminya. Cara penggunaan Fring dapat dilihat pada **Gambar 3.2**.



Gambar 3.2. Diagram alir penggunaan Fring

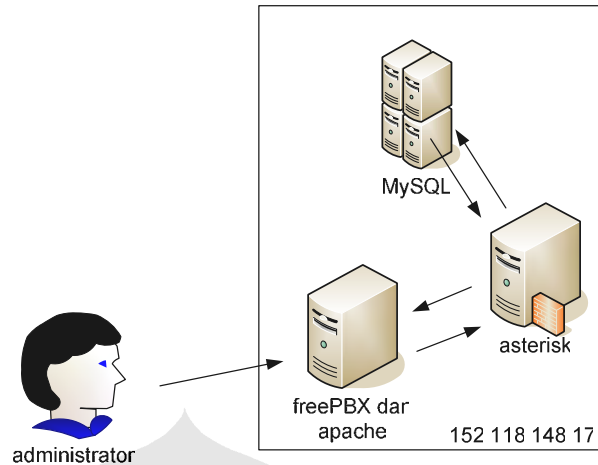
3.1.2. Sisi Server

Server SIP yang digunakan dibangun diatas *platform opensource* Linux Ubuntu. Tipe *server* yang dibentuk adalah *proxy mode server* yang memiliki fungsi seperti yang telah dijelaskan pada bagian 2.2.2. Dari **Gambar 2.3** dan **Gambar 2.6**, terlihat bahwa sebuah *proxy server* bertindak sebagai perantara dalam pengiriman *SIP message*. Sedangkan untuk transfer media, tidak memanfaatkan *SIP server* sama sekali. Akan tetapi, pada sistem yang menggunakan NAT, *proxy server* juga berperan sebagai *media gateway* sehingga topologi jaringan berbasis SIP tidak dapat bersifat P2P untuk transfer media. Hal ini mengakibatkan *SIP server* selalu meletakkan posisinya diantara para *user* pada saat transfer media dalam sebuah sesi sedang berlangsung. Sehingga kapasitas *bandwidth* yang dibutuhkan lebih besar.

Proxy server yang dibangun, dibuat dengan memanfaatkan program asterisk versi 1.4.18 yang juga bersifat *opensource* dan dapat diinstall pada platform Linux apapun. Asterisk merupakan perangkat lunak yang telah digunakan secara luas oleh para pengembang VoIP. Selain memiliki kemampuan untuk membuat sebuah sesi komunikasi *voice*, asterisk juga dapat melakukan komunikasi *video call* dengan memanfaatkan SDP. Asterisk mendukung berbagai jenis protokol pensinyalan seperti H323, iAX, dan SIP. Pada skripsi ini penulis hanya fokus pada SIP.

Selain asterisk, agar *proxy server* dapat berfungsi diperlukan berbagai perangkat lunak lainnya, yaitu freePBX, apache, MySQL, dan PHP. MySQL berfungsi sebagai penyimpan *database* yang diperlukan, freePBX sebagai *user interface*, dan apache sebagai *web server* sehingga *server* dapat dikonfigur secara *web-based*. IP publik yang digunakan pada *server* ini adalah 152.118.148.17. **Gambar 3.3** memperlihatkan rancang bangun *SIP server* serta *interfacing*-nya dengan *administrator*.

Hal lain yang harus diperhatikan adalah letak letak dari *server* pada jaringan. *SIP server* sebaiknya diletakkan pada DMZ sehingga jaringan utama tetap aman dan dapat menggunakan koneksi yang optimal.



Gambar 3.3. Rancang bangun SIP server 152.118.184.17

Asterisk memiliki kemampuan yang luar biasa sebagai VoIP server, akan tetapi pada skripsi ini, penulis tidak menggunakan asterisk terlalu mendalam. Konfigurasi asterisk yang penulis lakukan hanya bertujuan agar asterisk dapat melakukan fungsi dasarnya sebagai SIP server yang dapat mendukung komunikasi VoIP serta dapat mengirimkan video secara *real-time*. Adapun fungsi lain asterisk seperti *trunking* ke SIP server lainnya, *voicemail*, *conferencing*, dan lain sebagainya tidak penulis bahas dalam skripsi ini.

Berikut akan dibahas konfigurasi yang harus dilakukan pada asterisk agar dapat berfungsi sebagai SIP server dengan dukungan video. Cara konfigurasi asterisk adalah dengan mengubah isi dari file-file berekstensi *.conf*. Hal ini dapat dilakukan dengan menggunakan freePBX sebagai UI, akan tetapi untuk beberapa file harus dilakukan secara manual. Beberapa pengaturan mendasar pada server yang dibangun adalah pada file-file berikut:

- asterisk.conf

File ini berisi lokasi untuk file-file konfigurasi yang dibutuhkan oleh asterisk. Pengaturan default dari asterisk.conf dihasilkan pada saat melakukan *command make samples* sewaktu penginstalan. Isi dari file ini dapat dilihat pada **Tabel 3.2**.

Tabel 3.2. asterisk.conf

```
[directories]
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astdatadir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run/asterisk
astlogdir => /var/log/asterisk
```

- rtp.conf

File ini berfungsi untuk mengontrol *port* RTP yang digunakan oleh asterisk untuk mengirim dan menerima aliran RTP. RTP sendiri digunakan oleh SIP untuk membawa media. Pada setiap panggilan dua arah menggunakan SIP, jumlah *port* yang biasa digunakan berjumlah 5 *port*, antara lain 5060 untuk SIP, satu *port* untuk aliran data RTP dan satu *port* untuk RTCP pada satu arah, serta satu *port* aliran data RTP dan satu *port* RTCP pada arah yang berlawanan. *Port* RTP dapat dibatasi dengan mengatur *port* awal dan *port* akhirnya. Hal ini dapat dilihat pada

Tabel 3.3. rtp.conf

```
;
; RTP Configuration
;
[general]
;
; RTP start and RTP end configure start and end addresses
;
rtpstart=12300
rtpend=12400
```

- sip.conf

File ini berfungsi untuk mengatur opsi-opsi yang digunakan oleh protokol SIP pada asterisk. File ini berisi semua konfigurasi yang mempengaruhi kerja SIP pada asterisk, seperti berisi tentang konfigurasi ekstensi *user*, *codec* yang digunakan, dukungan video, dan pengaturan

lainnya. File sip.conf biasanya terintegrasi dengan file-file tambahan lain seperti. Sebagian konfigurasi SIP pada asterisk dapat dilihat pada

Tabel 3.4. sip.conf

```
[general]
videosupport=yes
;
; enable and force the sip jitterbuffer. If these settings are
desired
; they should be set in the sip_general_custom.conf file as
this file
; will get overwritten during reloads and upgrades.
;
; jbenable=yes
; jbforce=yes

; These will all be included in the [general] context
;
#include sip_general_additional.conf
#include sip_general_custom.conf
#include sip_nat.conf
#include sip_registrations_custom.conf
#include sip_registrations.conf

; These should all be expected to come after the [general]
context
;
#include sip_custom.conf
#include sip_additional.conf
#include sip_custom_post.conf
```

Tabel 3.5. sip_additional.conf

```
[2000]
type=friend
secret=abc0
record_out=Adhoc
record_in=Adhoc
qualify=yes
port=5060
pickupgroup=
nat=yes
mailbox=2000@device
host=dynamic
dtmfmode=rfc2833
dial=SIP/2000
context=from-internal
canreinvite=yes
callgroup=
callerid=device <2000>
allow=h263+
accountcode=
```

```

call-limit=50

[2001]
type=friend
secret=abc1
record_out=Adhoc
record_in=Adhoc
qualify=yes
.....

```

Tabel 3.6. sip_general_additional.conf

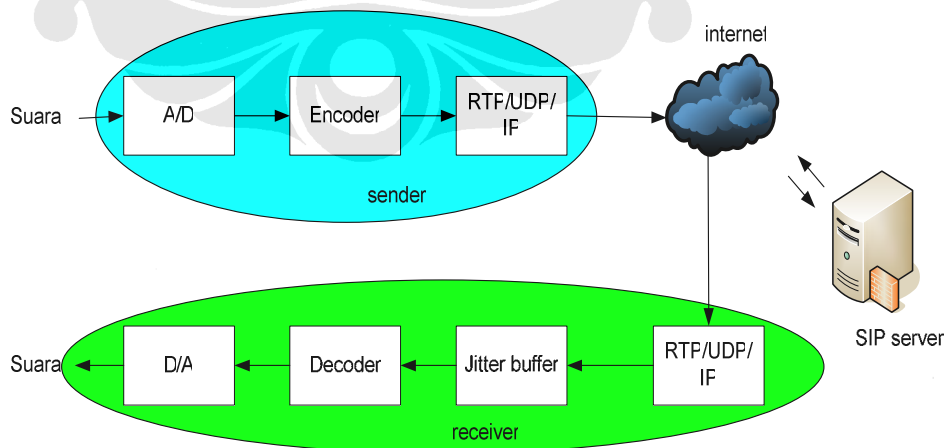
```

vmexten=*97
bindport=5060
bindaddr=0.0.0.0
disallow=all
allow=ulaw
allow=alaw
context=from-sip-external
callerid=Unknown
notifyringing=yes
notifyhold=yes
limitonpeers=yes
tos_sip=cs3
tos_audio=ef
tos_video=af41

```

3.2. TOPOLOGI JARINGAN

3.2.1. Topologi VoIP



Gambar 3.4. Diagram topologi VoIP yang disederhanakan [5]

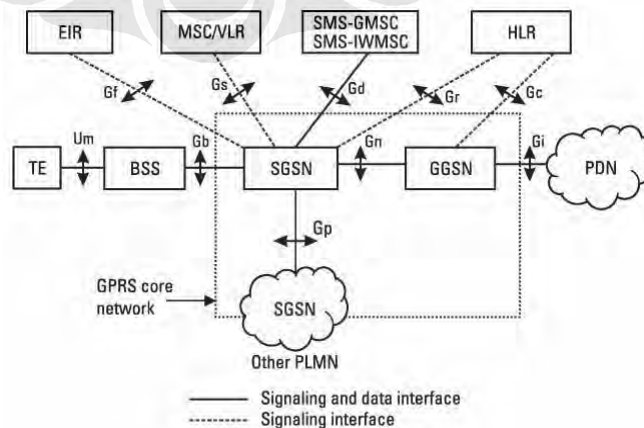
Topologi sederhana sebuah jaringan VoIP dapat dilihat pada **Gambar 3.4**. Blok diagram yang dilingkar dengan warna biru merupakan sisi pengirim, sedangkan yang berwarna hijau merupakan sisi penerima. Seperti yang telah dijelaskan pada bagian 2.1.1. salah satu parameter kualitas VoIP adalah *jitter*. Untuk mengkompensasi pengaruh dari *jitter* ini dibutuhkan *jitter buffer*. *Jitter buffer* dipasang pada sisi penerima.

Asterisk sebenarnya juga mempunyai *jitter buffer*. Akan tetapi penggunaannya diperuntukkan hanya jika sisi penerima tidak memiliki *jitter buffer*. *Jitter buffer* pada umumnya diatur sebesar 20 ms sampai 200 ms. Cara mengaktifkan *jitter buffer* pada asterisk adalah dengan mengaktifkan perintah `jbenable` dan `jbforce` pada file `sip.conf`.

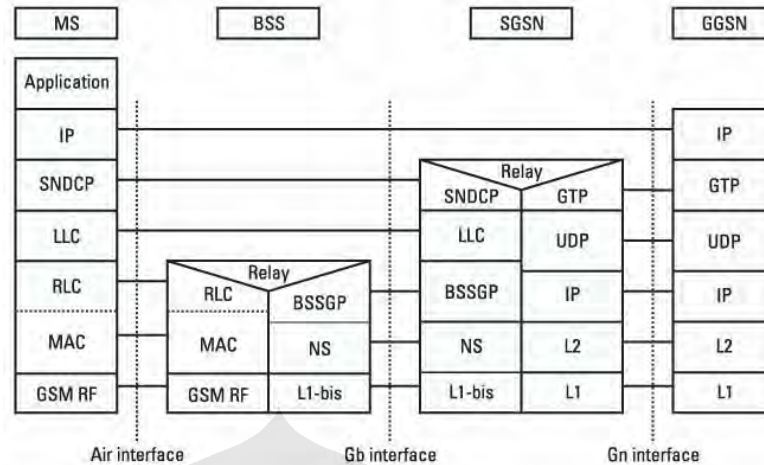
3.2.2. Topologi GPRS

Pada tahun 2001, GPRS ditambahkan ke dalam GSM untuk meningkatkan *data rate* sehingga memberikan transmisi data dalam bentuk paket-paket, serta memberikan *throughput* yang tinggi. GPRS menggunakan jaringan GSM ditambah dengan *support node* seperti GGSN dan SGSN.

Secara teori, *throughput* dari GPRS adalah sebesar 160 kbps dimana GPRS menggunakan delapan *timeslot* dengan *throughput* maksimal tiap slot adalah 20 kbps. Akan tetapi pada kenyataannya *bit rate* maksimum hanya 80 kbps karena hanya 4 slot yang bisa digunakan [6]. **Gambar 3.5** menunjukkan arsitektur jaringan GPRS, sedangkan **Gambar 3.6** memperlihatkan struktur protokol yang digunakan untuk transfer data pada GPRS.



Gambar 3.5. Arsitektur jaringan GPRS [7]



Gambar 3.6.

Struktur protokol yang digunakan untuk transfer data pada GPRS [6]

3.3. CODEC YANG DIGUNAKAN

Seperti yang telah dijelaskan pada bagian 2.1.1. *CODEC* merupakan faktor pertama yang mempengaruhi kualitas dari VoIP. Pemilihan *CODEC* akan berpengaruh terhadap BR dan BW yang digunakan. Pada skripsi ini digunakan tiga jenis *CODEC*, yaitu G.711, GSM, dan iLBC.

3.3.1. G.711 (a-law dan u-law)

G.711 merupakan standard format *encoding* yang dibuat oleh ITU-T. *Codec* ini mempunyai dua versi yaitu u-law dan A-law. Pada *codec* ini, suara di-*sampling* pada 8 KHz dengan *output rate* sebesar 64 kbps.

Perhitungan pemakaian *bandwidth* untuk *codec* ini dapat dijabarkan sebagai berikut.

$$Bit_Rate = 64kbps$$

$$packet_length = 20ms$$

$$packet/s = \frac{1}{20ms} = 50$$

$$payload = \frac{Bit_Rate}{packet/s} = \frac{64000bps}{50/s} = 1280bit = 160bytes$$

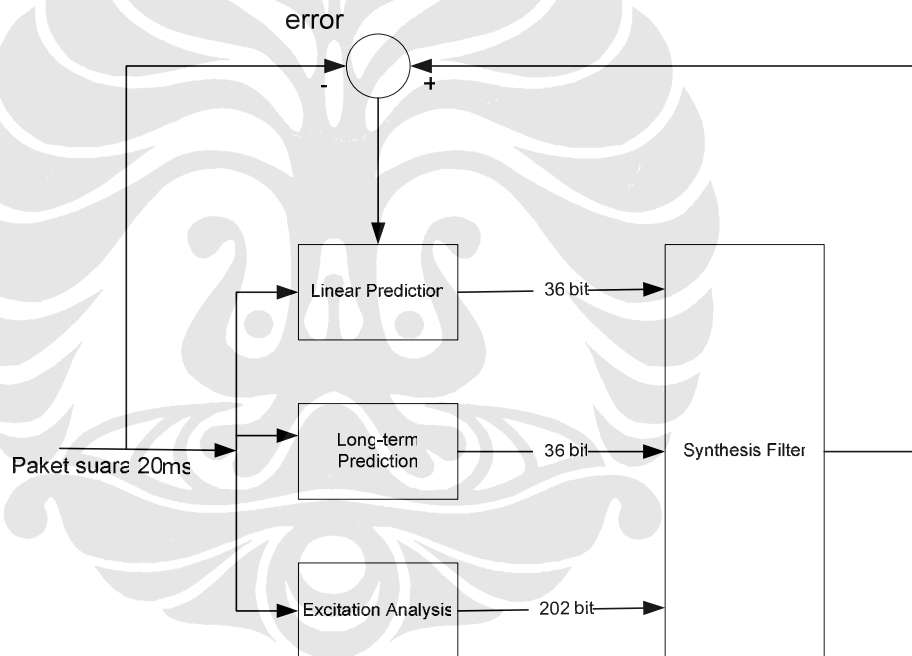
$$IP_header(IP + UDP + RTP) = 40\text{byte}$$

$$bandwidth = (payload + IP_header) \times packet / s \times 8\text{bit} / \text{byte}$$

$$bandwidth = (160 + 40) \times 50 \times 8 = 80\text{kbps}$$

3.3.2. GSM

GSM yang digunakan pada asterisk merupakan GSM-FR yang biasa disebut juga GSM 06.10. *Codec* ini menggunakan *Linear Predictive Coding with Regular Pulse Excitation (LPC-RPE)*. GSM-FR merupakan *codec* suara yang beroperasi pada 13.2 kbps dengan lebar paket 20ms. Setiap paket berisi *payload* sebesar 264 bit atau sebesar 33 byte. Encoder dari GSM-FR LPC-RPE dapat dilihat pada **Gambar 3.7**.



Gambar 3.7. Diagram dari *codec* GSM-FR LPC-RPE [8]

Perhitungan pemakaian *bandwidth* untuk *codec* ini dapat dijabarkan sebagai berikut.

$$\text{Bit_Rate} = 13,2\text{kbps}$$

$$\text{packet_length} = 20\text{ms}$$

$$\text{packet / s} = \frac{1}{20\text{ms}} = 50$$

$$\text{payload} = \frac{\text{Bit_Rate}}{\text{packet / s}} = \frac{13200\text{bps}}{50 / \text{s}} = 264\text{bit} = 33\text{bytes}$$

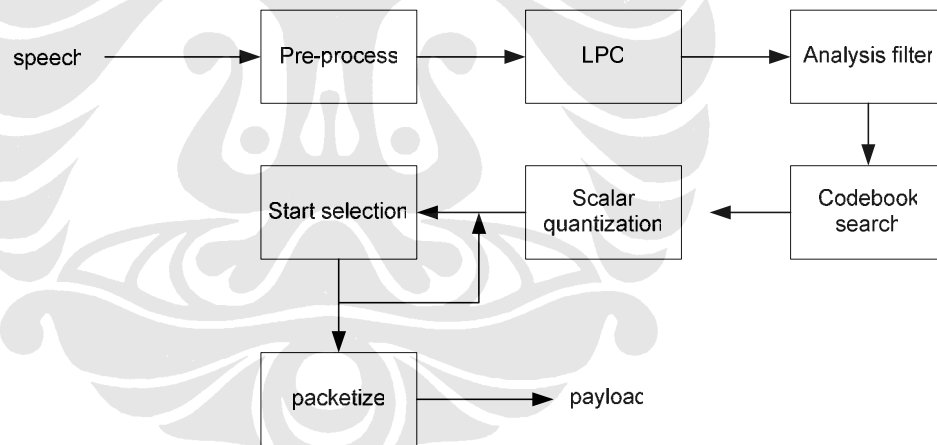
$$\text{IP_header}(\text{IP} + \text{UDP} + \text{RTP}) = 40\text{byte}$$

$$\text{bandwidth} = (\text{payload} + \text{IP_header}) \times \text{packet / s} \times 8\text{bit / byte}$$

$$\text{bandwidth} = (33 + 40) \times 50 \times 8 = 29,2\text{kbps}$$

3.3.3. iLBC

Codec iLBC yang digunakan pada skripsi ini mempunyai *bit rate* sebesar 13.3kbps dengan lebar paket 30ms. Setiap paket berisi *payload* sebesar 400 bit atau 50 byte. Diagram alir dari encoder iLBC dapat dilihat pada **Gambar 3.8**.



Gambar 3.8. Diagram alir encoder iLBC [9]

Sedangkan perhitungan besar *bandwidth* yang digunakan dapat dijelaskan sebagai berikut.

$$\text{Bit_Rate} = 13,3\text{kbps}$$

$$\text{packet_length} = 30\text{ms}$$

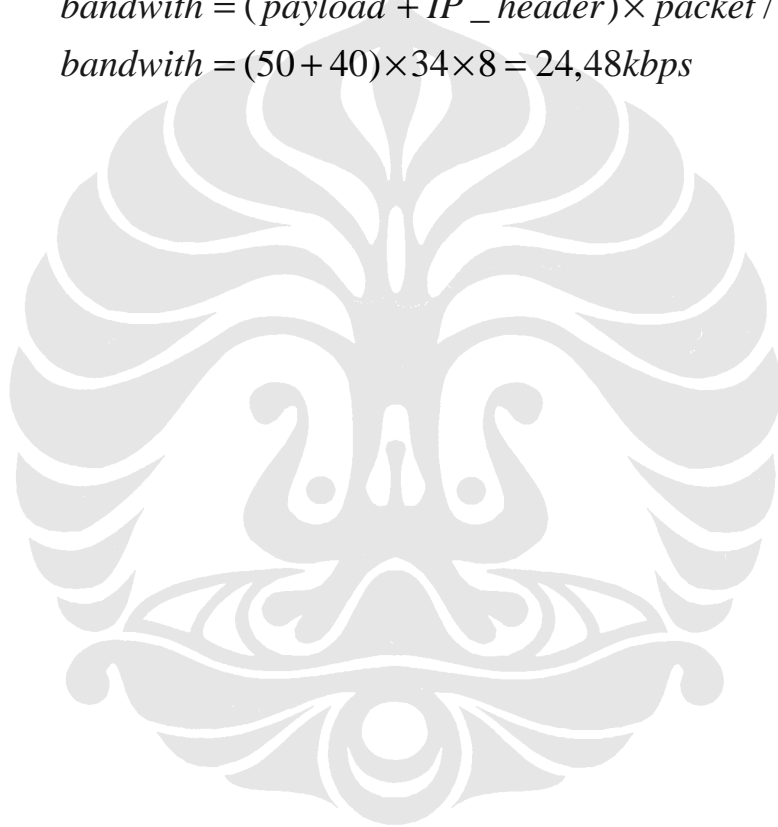
$$\text{packet / s} = \frac{1}{30\text{ms}} = 33,33 \approx 34$$

$$\text{payload} = \frac{\text{Bit_Rate}}{\text{packet / s}} = \frac{13200\text{bps}}{33,33 / \text{s}} = 400\text{bit} = 50\text{bytes}$$

$$\text{IP_header}(\text{IP} + \text{UDP} + \text{RTP}) = 40\text{byte}$$

$$\text{bandwith} = (\text{payload} + \text{IP_header}) \times \text{packet / s} \times 8\text{bit / byte}$$

$$\text{bandwith} = (50 + 40) \times 34 \times 8 = 24,48\text{kbps}$$



BAB 4

ANALISIS DAN UJICOBA

4.1. MATRIKS ANALISIS

Sistem *mobile VoIP* yang dibangun akan dianalisis berdasarkan beberapa parameter. Parameter-parameter tersebut antara lain adalah tiga jenis *codec*, dua operator, dan dua variable, *jitter* dan *bandwidth*. *Codec* yang digunakan antara lain G.711, GSM, dan ILBC. Ketiga *codec* tersebut akan dilihat performanya berdasarkan konsumsi *bandwidth*, *jitter*, serta harganya berdasarkan operator yang digunakan. Adapun jaringan yang digunakan pada pengambilan data merupakan jaringan GPRS yang telah dijelaskan pada bagian 3.2.

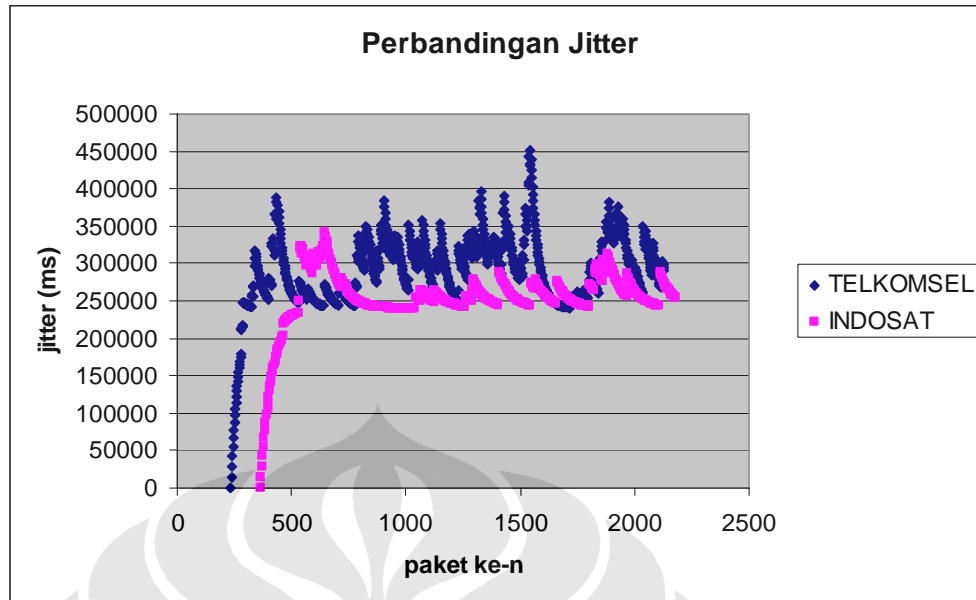
Dari semua parameter diatas, akan dicari berbagai konfigurasi yang dapat memberikan performa yang diinginkan sesuai dengan kebutuhan.

4.2. ANALISIS HASIL UJICOBA DAN KOMPARASI

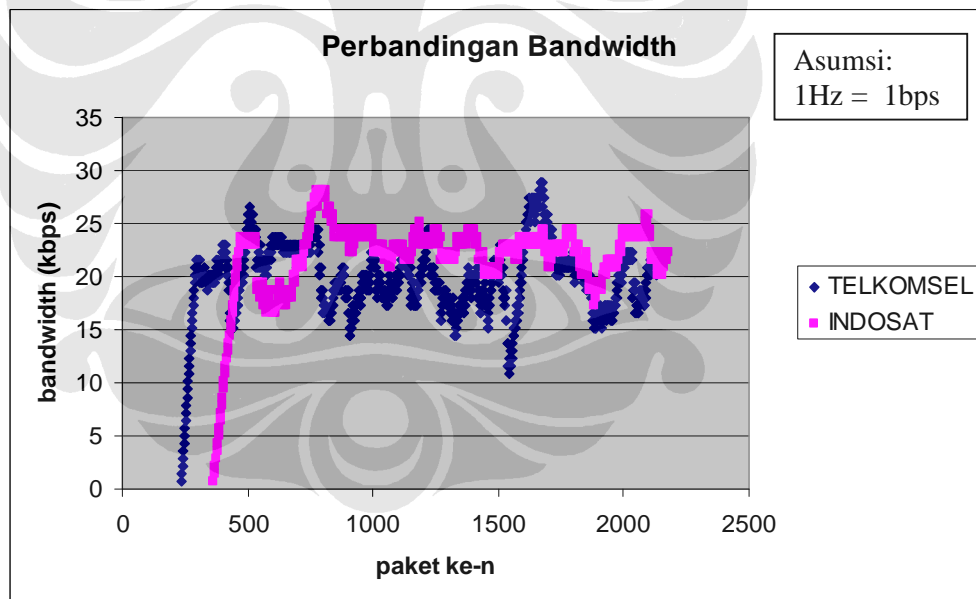
Ujicoba dilakukan dengan menggunakan jaringan GPRS dua operator berbeda, yaitu PT. Telkomsel dan PT. Indosat. UE yang digunakan adalah satu buah ponsel Nokia N73 dan satu buah Nokia 6120classic yang memiliki kemampuan multimedia yang identik.

Analisis dikhususkan untuk membahas performa *uplink* sistem pada jaringan GPRS yang digunakan. *Jitter* dan *bandwidth* yang didapat akan dibandingkan berdasarkan jenis *codec* dan operator yang digunakan.

Data yang didapat menunjukkan bahwa ketika sistem menggunakan jenis *codec* iLBC, paket-paket suara tidak terkirim sebagaimana mestinya. Hal ini disebabkan oleh karena *jitter* yang terjadi sangat besar seperti yang dapat dilihat pada **Gambar 4.1**. Meskipun *bandwidth* yang dibutuhkan kecil seperti yang ditunjukkan pada **Gambar 4.2**, *jitter* yang besar menunjukkan bahwa komunikasi tidak dapat berlangsung secara *real-time*. Oleh karena itu iLBC tidak dapat digunakan pada sistem ini.



Gambar 4.1. Komparasi *jitter* menggunakan iLBC



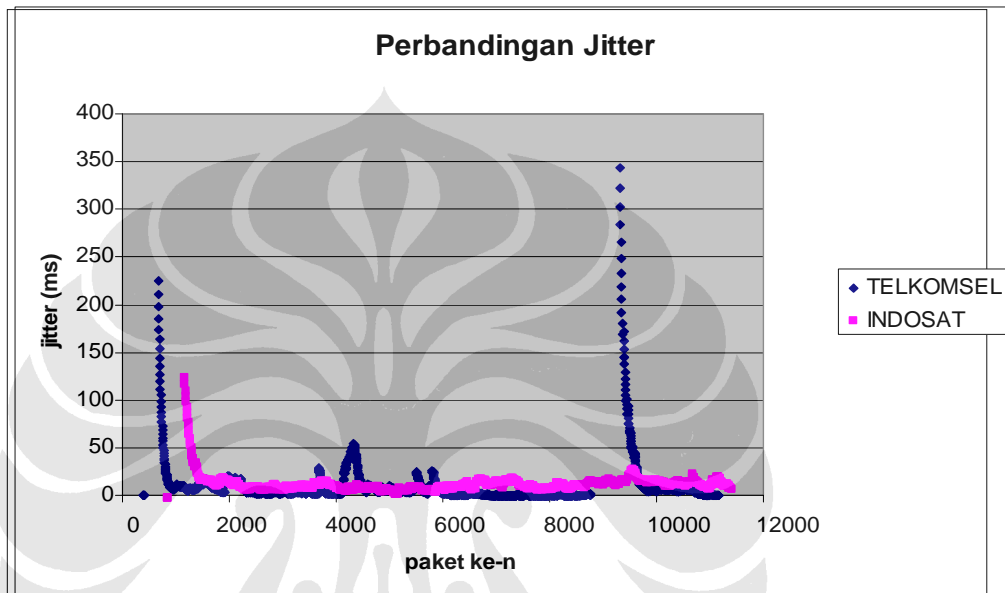
Gambar 4.2. Komparasi *bandwidth* menggunakan iLBC

4.2.1. Komparasi Performa Sistem Berdasarkan *Codec*

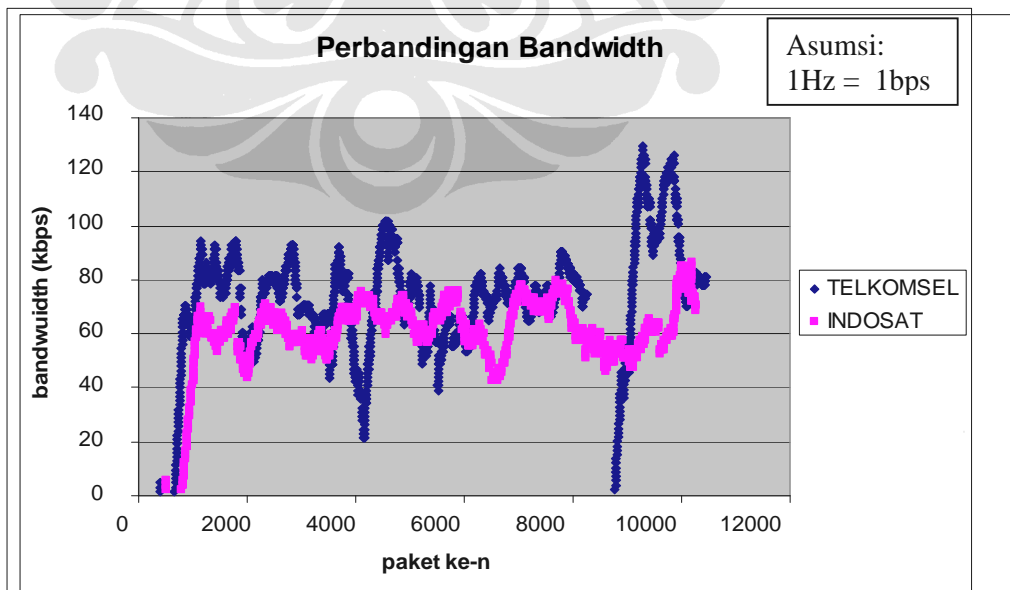
Codec yang akan dibandingkan hanya dua yaitu G.711 dan GSM, mengingat iLBC sudah tidak memenuhi kriteria untuk memberikan komunikasi yang *real-time*.

G.711

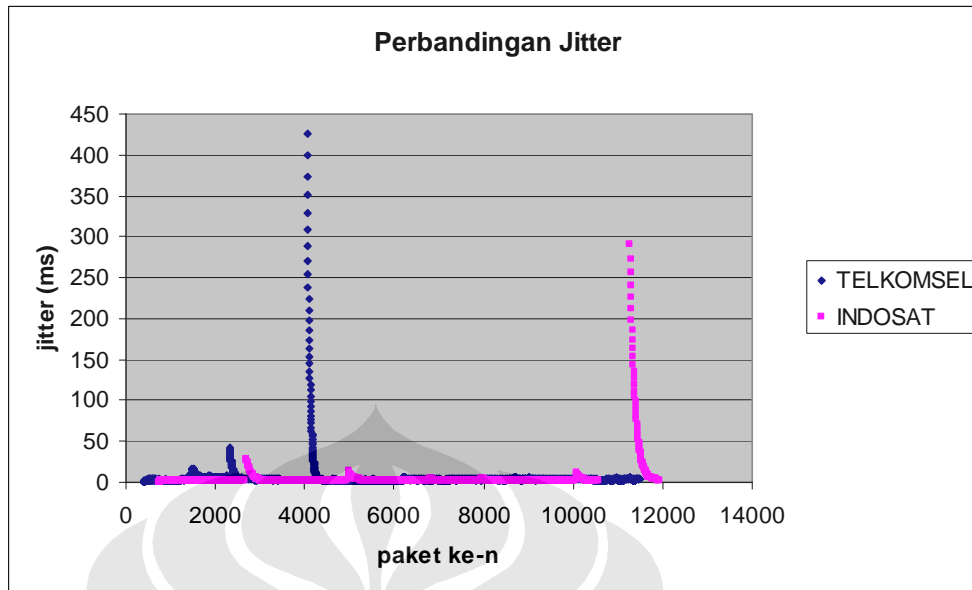
Komunikasi berlangsung antara dua buah *client* berubah ponsel. Masing-masing *client* mengirimkan paket-paket suara sehingga terdapat dua buah *uplink* pada komunikasi ini. **Gambar 4.3** dan **Gambar 4.4** memperlihatkan *jitter* yang terjadi serta *bandwidth* yang terpakai pada *uplink client A*, sedangkan **Gambar 4.5** dan **Gambar 4.6** memperlihatkan *uplink client B*.



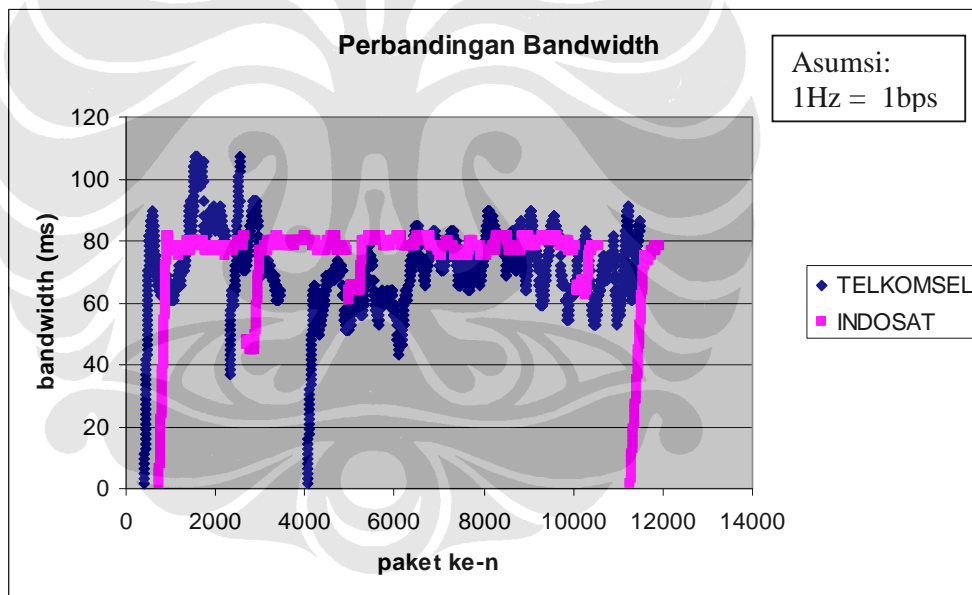
Gambar 4.3. Grafik komparasi *jitter uplink client A* dengan *codec G.711*



Gambar 4.4. Grafik komparasi *bandwidth uplink client A* dengan *codec G.711*



Gambar 4.5. Grafik komparasi *jitter uplink client B* dengan *codec G.711*

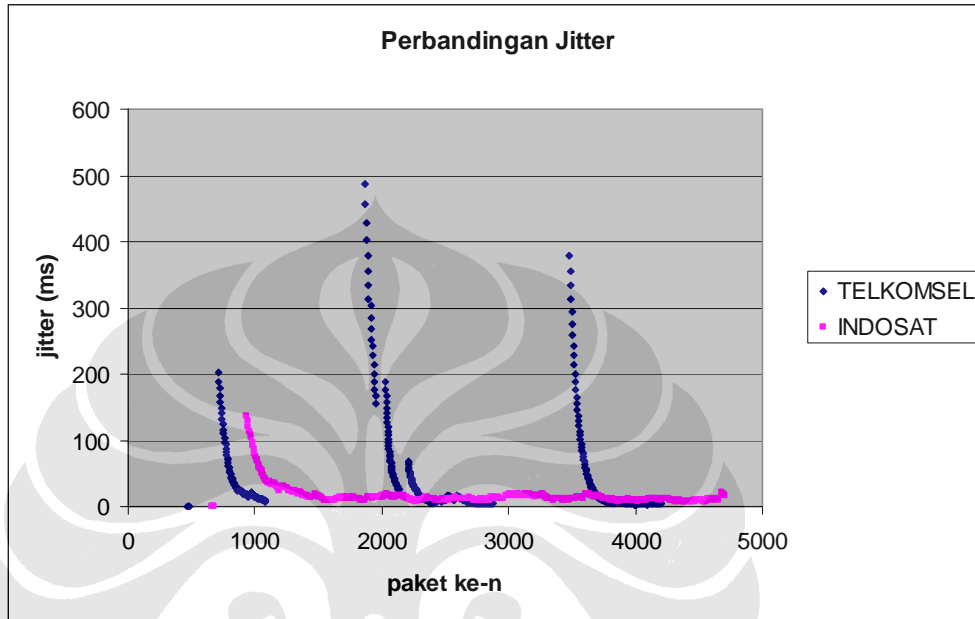


Gambar 4.6. Grafik komparasi *bandwidth uplink client B* dengan *codec G.711*

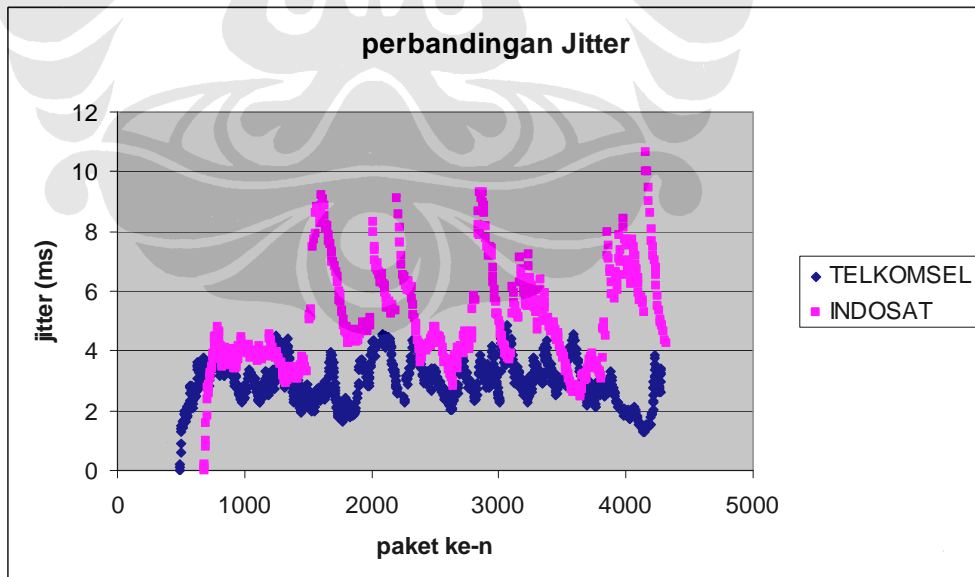
Jitter yang dihasilkan pada saat penggunaan operator Indosat memiliki nilai yang lebih stabil daripada saat menggunakan Telkomsel seperti yang diperlihatkan pada **Gambar 4.3** dan **Gambar 4.5**. Begitu pula dengan penggunaan *bandwidth*, seperti yang diperlihatkan **Gambar 4.4** dan **Gambar 4.6**, *bandwidth* Indosat lebih stabil daripada Telkomsel.

GSM

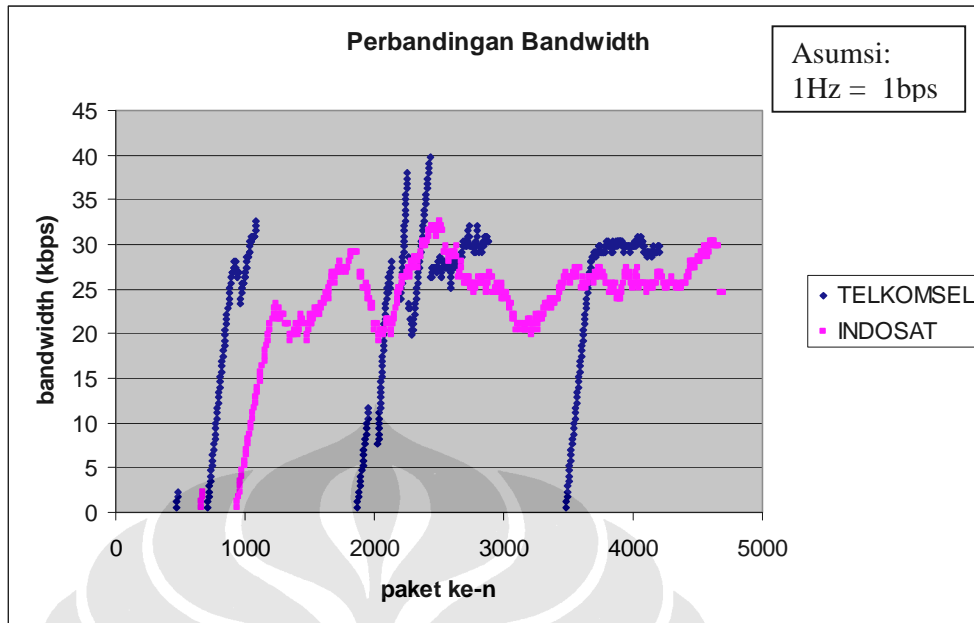
Hal yang sama juga dilakukan dengan menggunakan *codec* GSM. Dari data yang didapat akan kita lihat *jitter* yang terjadi ataupun *bandwidth* yang dibutuhkan.



Gambar 4.7. Grafik komparasi *jitter* pada *uplink client* A dengan *codec* GSM

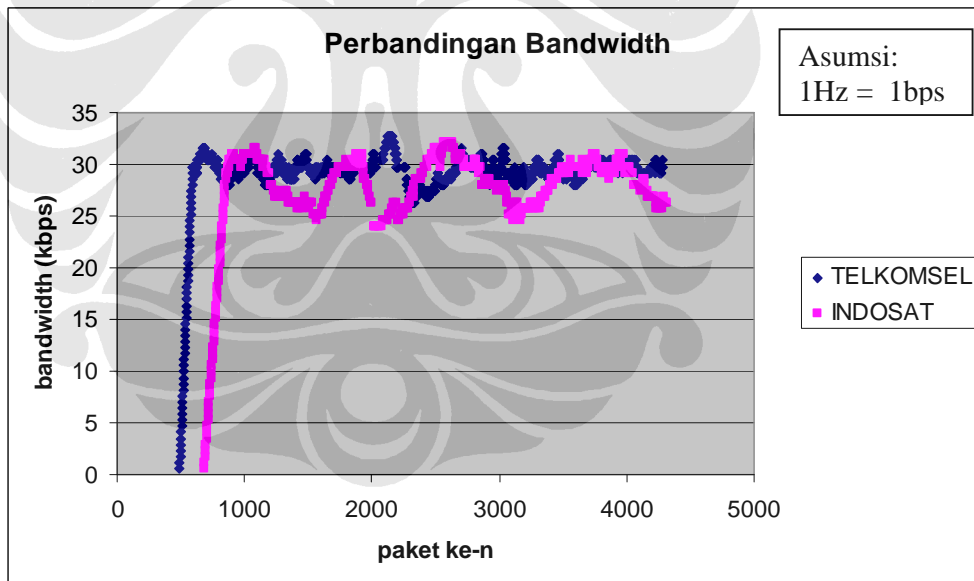


Gambar 4.8. Grafik komparasi *jitter* pada *uplink client* B dengan *codec* GSM



Gambar 4.9.

Grafik komparasi *bandwidth* pada *uplink client A* dengan *codec GSM*



Gambar 4.10.

Grafik komparasi *bandwidth* pada *uplink client B* dengan *codec GSM*

Gambar 4.7 sampai **Gambar 4.10** memperlihatkan perbandingan performa sistem dengan menggunakan dua operator yang berbeda. Dari semua

grafik tersebut terlihat bahwa *jitter* dan *bandwidth* pada satu *uplink* lebih stabil daripada *uplink* yang lain.

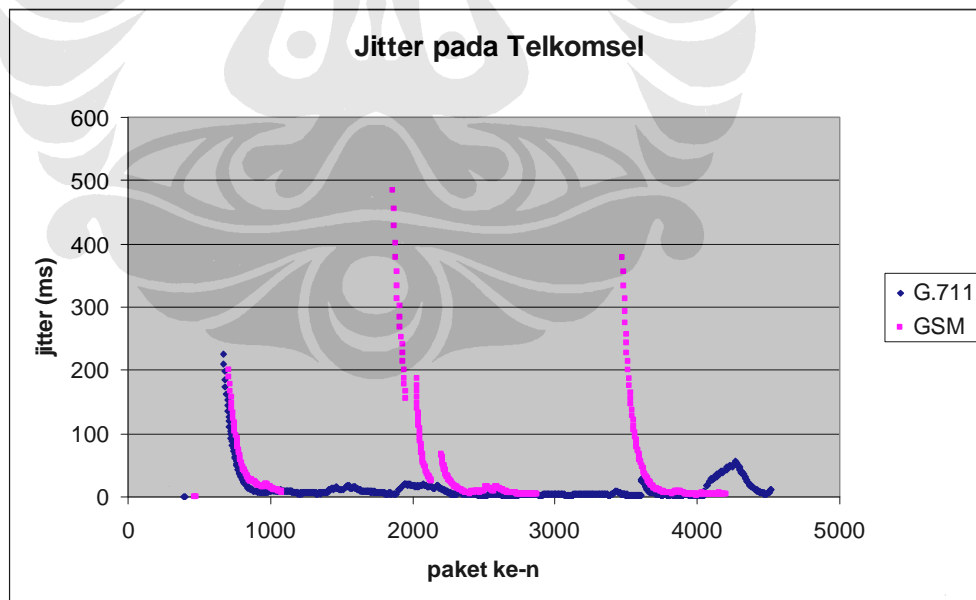
Grafik yang putus-putus memperlihatkan bahwa terjadinya interval yang besar diantara paket-paket yang dikirim. Hal ini dapat kita lihat dengan membandingkan **Gambar 4.7** dan **Gambar 4.9** dimana kedua grafik tersebut sejalan. Terlihat bahwa ketika pkt tidak terkirim, *bandwidth* yang dipakai juga bernilai nol.

4.2.2. Komparasi Performa Sistem Berdasarkan Operator

Pembandingan dilakukan terhadap performa GPRS dari dua operator berbeda dalam menunjang sistem *mobile* VoIP yang dibangun. parameter yang dibandingkan merupakan dua *codec* berbeda, G.711 dan GSM.

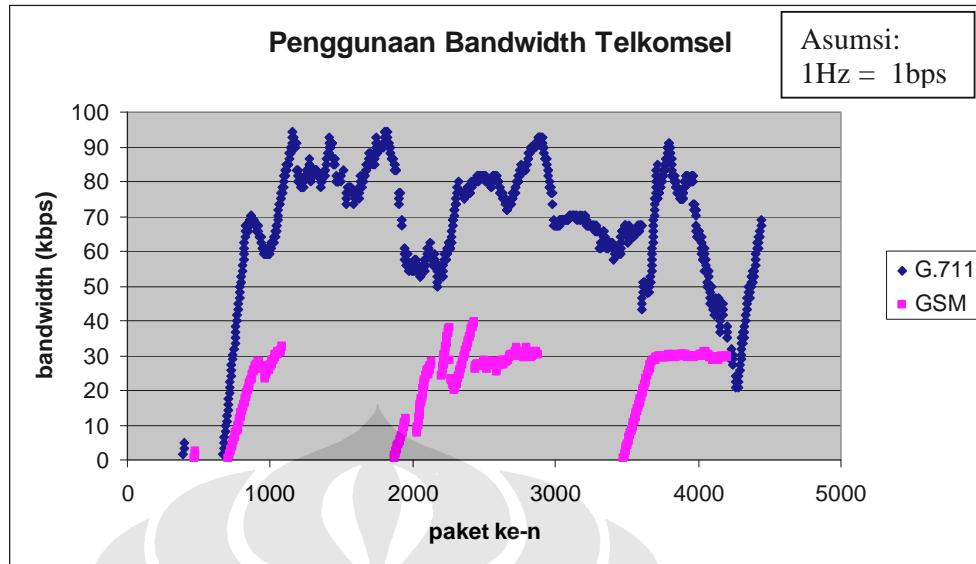
TELKOMSEL

Komparasi ini bertujuan untuk melihat besar *jitter* dan *bandwidth* yang diperlukan oleh operator untuk menunjang sistem berdasarkan *codec* yang digunakan. Analisis dilakukan terhadap data *uplink* dari kedua *client*.



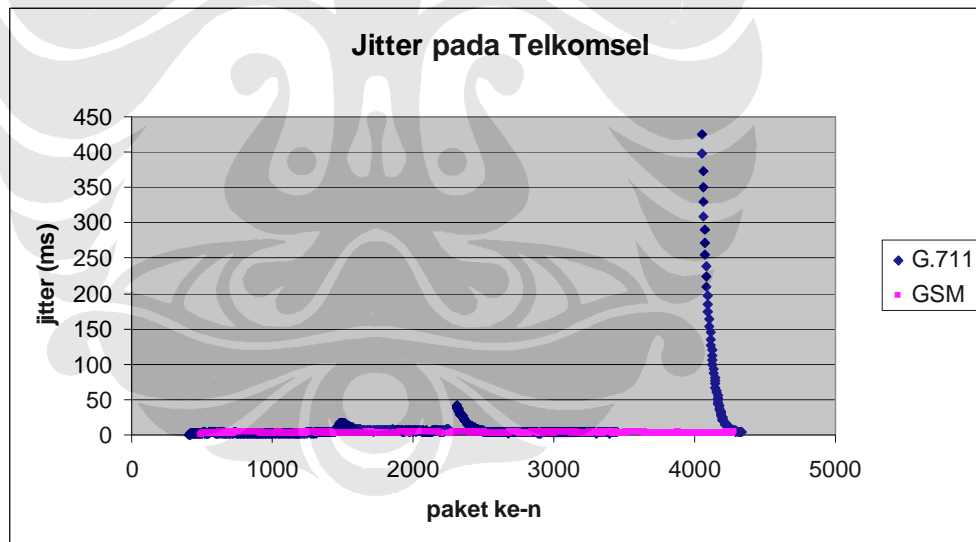
Gambar 4.11.

Grafik komparasi *jitter* pada *uplink client* A menggunakan Telkomsel



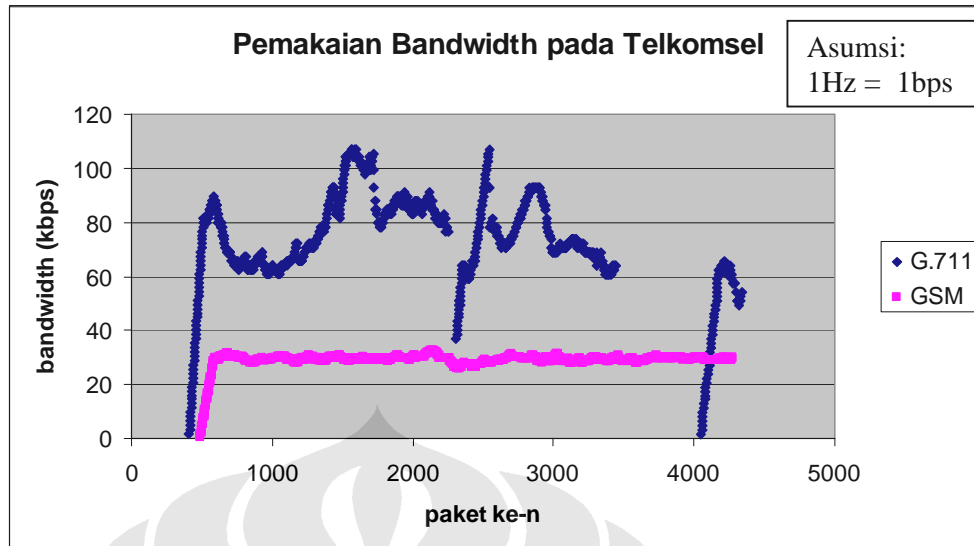
Gambar 4.12.

Grafik komparasi *bandwidth* pada *uplink client A* menggunakan Telkomsel



Gambar 4.13.

Grafik komparasi *jitter* pada *uplink client B* menggunakan Telkomsel



Gambar 4.14.

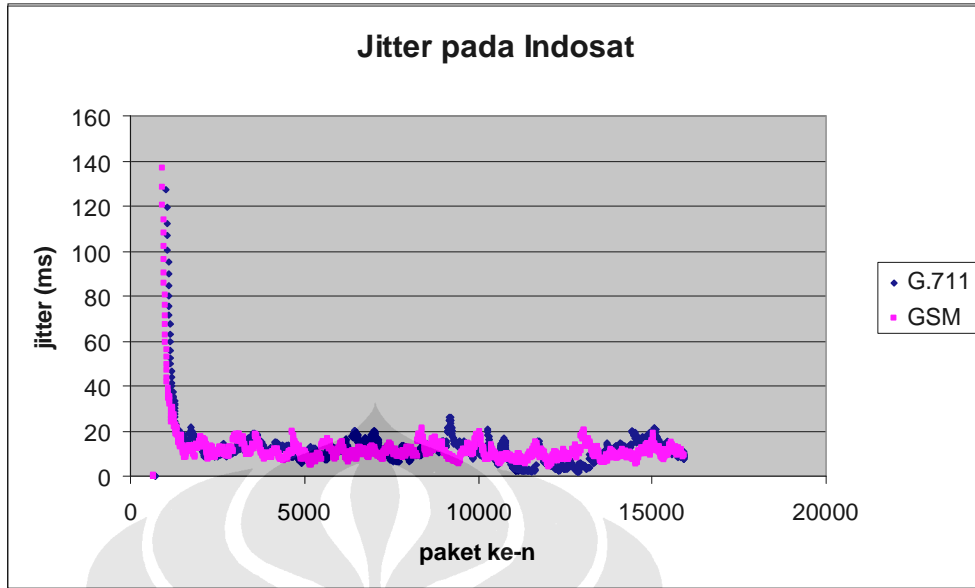
Grafik komparasi *bandwidth* pada *uplink client B* menggunakan Telkomsel

Jika dibandingkan antara *uplink client A* dan *client B*, terdapat perbedaan yang jelas. Pada *uplink client A*, *jitter* yang diberikan oleh *codec* G.711 lebih kecil daripada GSM. Sebaliknya, GSM memberikan *jitter* yang lebih kecil pada *uplink client B* seperti yang terlihat pada **Gambar 4.11** dan **Gambar 4.13**. Hal ini disebabkan karena komunikasi dilakukan melalui jaringan IP publik yang tidak stabil.

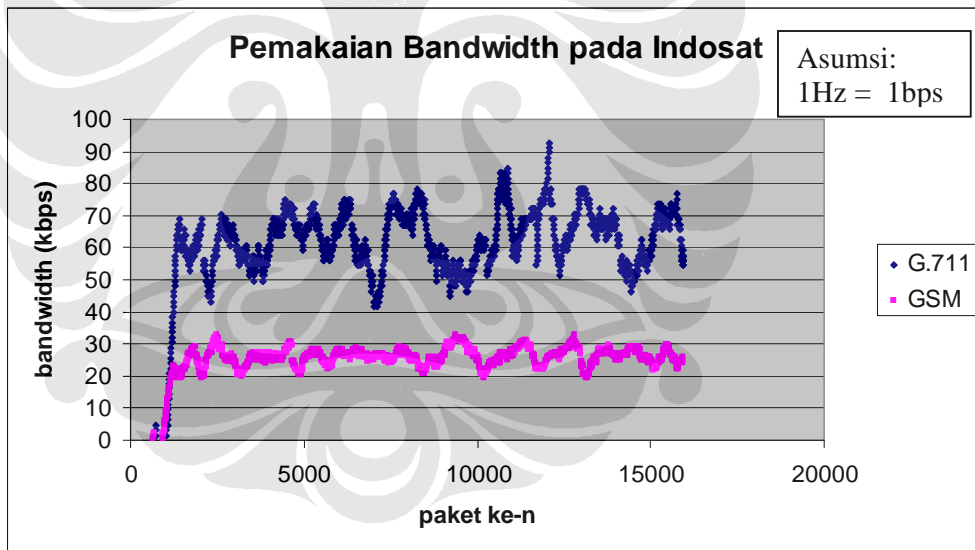
Sedangkan untuk pemakaian *bandwidth*, GSM jauh lebih kecil dibandingkan dengan G.711, hal ini dapat dilihat pada **Gambar 4.12** dan **Gambar 4.14**.

INDOSAT

Komparasi ini bertujuan untuk melihat besar *jitter* dan *bandwidth* yang diperlukan oleh operator untuk menunjang sistem berdasarkan *codec* yang digunakan. Analisis dilakukan terhadap data *uplink* dari kedua *client*.

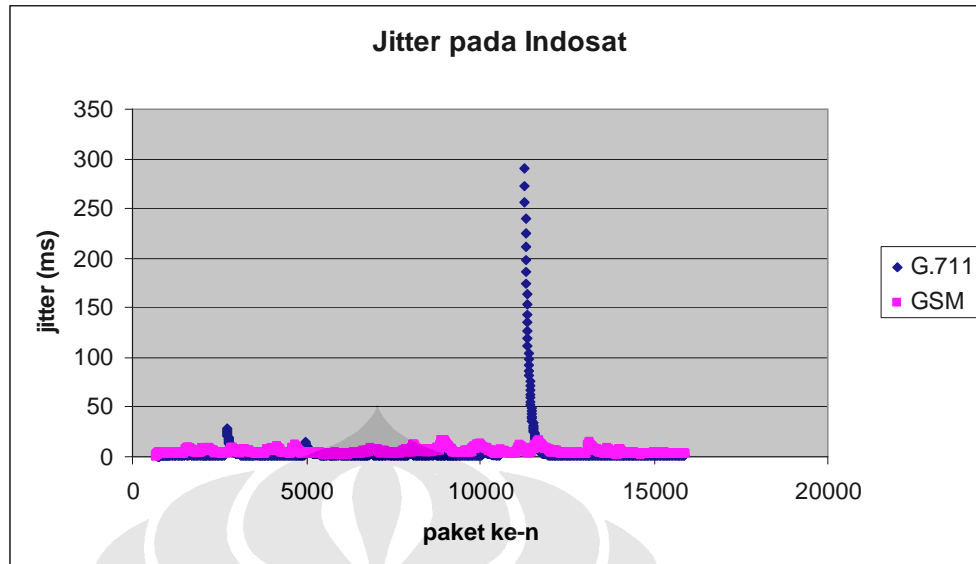


Gambar 4.15. Grafik komparasi *jitter* pada *uplink client* A menggunakan Indosat

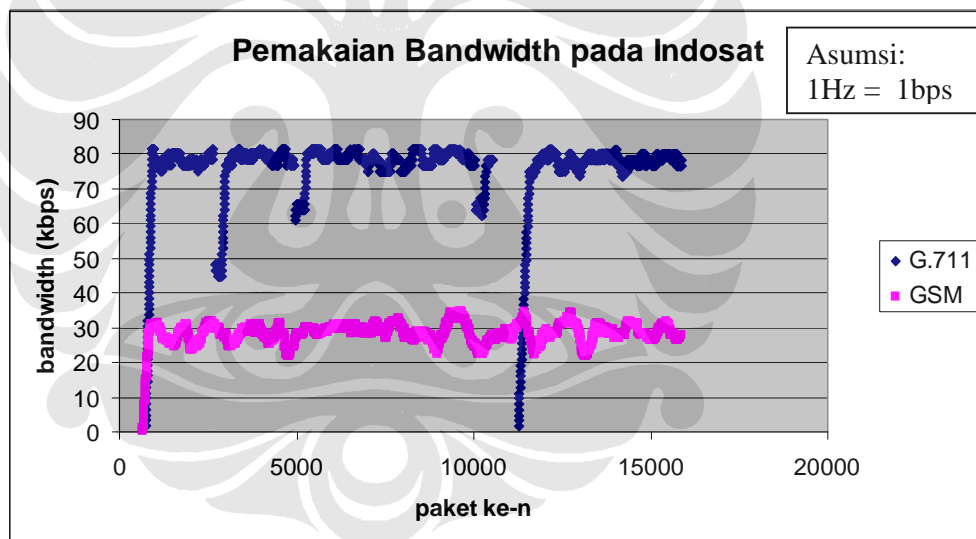


Gambar 4.16.

Grafik komparasi *bandwidth* pada *uplink client* A menggunakan Indosat



Gambar 4.17. Grafik komparasi *jitter* pada *uplink client B* menggunakan Indosat



Gambar 4.18.

Grafik komparasi *bandwidth* pada *uplink client B* menggunakan Indosat

Pada penggunaan GPRS Indosat, seperti yang ditunjukkan oleh **Gambar 4.15** dan **Gambar 4.17**, *jitter* yang terjadi relatif stabil dan relatif sama untuk *codec* G7.11 dan GSM. Akan tetapi, untuk pemakaian *bandwidth*, *codec* GSM jauh lebih kecil daripada G.711.

Dari data-data dan grafik yang diperoleh, dapat dibuat sebuah rangkuman komparasi seperti yang terlihat pada **Tabel 4.1** dan **Tabel 4.2**.

Tabel 4.1.

Nilai parameter-parameter *mobile* VoIP menggunakan simpati-Telkomsel

SIMPATI-TELKOMSEL						Asumsi: 1Hz = 1bps
Codec	Paket Dikirim	Paket Diterima	Loss	Jitter Rata2(Ms)	BW Rata2 (Kbps)	Ket
G.711	2927	2582	388 (13,1%)	11,525	74,57	OK
	2879	2879	411 (12,5%)	6,29	71,41	OK
GSM	906	516	2 (0,4%)	38,74	25,225	OK
	1332	1385	0 (0%)	4,89	28,96	OK
ILBC	177	0	0 (0%)	238614	22,17	GAGAL
	930	930	214 (18,7%)	294930	19,78	GAGAL

Tabel 4.2.

Nilai parameter-parameter *mobile* VoIP menggunakan Indosat-m3

INDOSAT-M3						Asumsi: 1Hz = 1bps
Codec	Paket Dikirim	Paket Diterima	Loss	Jitter Rata2(Ms)	BW Rata2 (Kbps)	Ket
G.711	2267	2029	352 (14,8%)	12,535	63,75	OK
	2312	2312	59 (2,5%)	4,24	75,91	OK
GSM	3244	3039	2 (0,1%)	12,76	25,56	OK
	3451	3451	1 (0%)	4,39	28,79	OK
ILBC	127	0	0 (0%)	239899	21,3	GAGAL
	1130	1130	123 (9,8%)	356168	21,92	GAGAL

Pada semua grafik yang telah ditampilkan pada bagian 4.2. terlihat bahwa pemakaian *bandwidth* pada setiap jenis komunikasi yang dilakukan tidak stabil. Hal ini disebabkan karena komunikasi dilakukan menggunakan jaringan IP publik via GPRS. Seperti yang telah diketahui bersama bahwa keadaan jaringan IP publik via GPRS tidak pernah stabil dikarenakan tidak adanya QOS yang disediakan oleh pihak operator. Akibat ketidakstabilan jaringan ini mengakibatkan *jitter* yang terjadi juga bervariasi.

Dari **Tabel 4.1** dan **Tabel 4.2** terlihat bahwa, baik dengan menggunakan simpati-Telkomsel ataupun Indosat-m3, komunikasi yang dilakukan dengan *codec* ILBC tidak berfungsi dengan baik. *Jitter* yang dihasilkan sangat besar dan aliran

paket data dari *client* A tidak sampai ke *client* B. Grafik *jitter* pada komunikasi menggunakan *codec* ILBC seperti yang terlihat pada bagian 4.2. menunjukkan besar *jitter* yang bertambah secara drastis lalu bersaturasi pada besar tertentu. Angka *jitter* yang mencapai ribuan detik, menunjukkan bahwa komunikasi tidak dapat berjalan secara *real-time*. Oleh karena itu dapat disimpulkan bahwa *codec* ILBC tidak dapat mendukung *mobile* VoIP.

Jika kita bandingkan data-data yang didapat berdasarkan *jitter*, *loss*, dan penggunaan *bandwidth* seperti yang ditunjukkan pada **Tabel 4.1** dan **Tabel 4.2**, dapat kita lihat bahwa, untuk setiap *codec* yang sama, besar *packet loss*, *jitter*, dan konsumsi *bandwidth* dengan menggunakan simpati-Telkomsel ataupun Indosat-m3 relatif bernilai sama. Hanya sedikit perbedaan yang terjadi diantara kedua operator tersebut.

Lebih jauh lagi, jika kita bandingkan *codec* yang berbeda pada setiap operator, terlihat bahwa *jitter* yang terjadi untuk *codec* G.711 dan *codec* GSM tidak terlalu berbeda jauh. Akan tetapi untuk *packet loss*, GSM lebih sedikit dibandingkan dengan G.711. GSM kembali menunjukkan keunggulannya dengan penggunaan *bandwidth* yang jauh lebih kecil daripada yang dibutuhkan oleh G.711. Selain itu kualitas suara yang disuguhkan oleh GSM tidak kalah dengan G.711.

Parameter lain yang harus dibandingkan adalah harga yang harus dikeluarkan. PT. Telkomsel mematok harga akses GPRS sebesar Rp.10,00 per kb, sedangkan PT. Indosat menarifikasi akses GPRS sebesar Rp.1,00 per kb. Sehingga untuk kualitas yang sama berdasarkan ujicoba, PT. Indosat menawarkan harga yang jauh lebih murah daripada PT. Telkomsel. Akan tetapi saat ujicoba, koneksi GPRS Telkomsel lebih cepat tersambung dibandingkan dengan Indosat yang koneksinya sering putus.

4.3. PEMANFAATAN SISTEM

Sistem ini dibangun sebagai salah satu alternatif komunikasi suara dengan harga miring. Sistem pembayaran yang berdasarkan data yang diakses menyebabkan harga dapat ditekan dengan memanfaatkan teknologi kompresi yang bagus seperti G.729 dengan *bit rate* sebesar 8kbps. Sistem ini dapat dimanfaatkan untuk banyak hal.

4.3.1. Solusi Telepon Murah

Sistem *mobile* VoIP yang dibangun menawarkan harga komunikasi yang jauh lebih murah dan tidak mengenal batas wilayah. Asalkan ponsel dapat mengakses jaringan GPRS operator yang digunakannya, maka komunikasi dapat berlangsung. Lebih jauh lagi sistem ini dapat diintegrasikan dengan PSTN menggunakan semacam *gateway* sehingga pengguna *mobile* VoIP juga bisa berkomunikasi dengan pengguna PSTN.

4.3.2. Sambungan Langsung Internasional dengan Harga Miring

Pemanfaatan teknologi VoIP untuk melakukan komunikasi SLI telah diterapkan oleh TELKOM 017, akan tetapi sistem ini belum mendukung penggunaan yang bersifat *mobile*. Sistem *mobile* VoIP yang dibangun dapat digunakan untuk SLI, hal ini disebabkan karena sistem ini terhubung dengan jaringan IP publik sehingga dapat diakses dari mana saja termasuk dari luar negeri.

Akan tetapi, untuk penggunaan yang lebih efisien, VoIP *server* yang dibangun dapat diintegrasikan dengan VoIP *server* lainnya termasuk di luar negeri dengan pengaturan *trunking* sehingga untuk melakukan komunikasi, *user* tidak harus mendaftar pada *server* yang sama.

BAB 5

KESIMPULAN

Terdapat beberapa kesimpulan yang dapat diambil dari perancangan, pembangunan, sampai ujicoba sistem *mobile* VoIP ini, antara lain:

1. Sistem dapat berjalan baik dan dapat dimanfaatkan sebagai alternatif teknologi telekomunikasi.
2. *Codec* iLBC tidak bisa dimanfaatkan pada sistem *mobile* VoIP karena *jitter* yang dihasilkan diatas 200000 ms.
3. Saat menggunakan operator Telkomsel, *codec* yang menghasilkan *jitter* dan *bandwidth* yang paling optimum adalah GSM dengan *jitter* rata-rata sebesar 4,89 ms sampai 38,74 ms dengan *bandwidth* 25,225 kbps sampai 28,96 kbps.
4. Ketika menggunakan operator Indosat, GSM juga memberikan hasil yang optimum dengan *jitter* sebesar 4,39 ms sampai 12,76 ms dengan *bandwidth* 25,56 kbps sampai 28,79 kbps.
5. Pemanfaatan GPRS Telkomsel dan Indosat menunjukkan kualitas yang relatif sama dengan besar *jitter* rata-rata serta *bandwidth* yang tidak berbeda jauh.
6. GPRS Indosat menghasilkan *jitter* dan *bandwidth* yang lebih stabil dibandingkan dengan GPRS Telkomsel dalam pemanfaatannya pada sistem *mobile* VoIP.
7. Dengan kualitas yang relatif sama untuk *mobile* VoIP, PT Indosat menawarkan akses GPRS yang jauh lebih murah.

DAFTAR ACUAN

- [1] Alan B. Johnston, "SIP: Understanding the Session Initiation Protocol", Artech House, 2001
- [2] Jyh-Cheng Chen dan Tao Zhang, "IP-Based Next-Generation Wireless Networks, System, Architecture, and Protocols", John Wiley & Sons, 2004
- [3] Mudji Basuki, "Voice over IP", Lintasarta, 2007
- [4] Apridian, "VoIP, Sejarah dan Perkembangannya", channel-11, 2007
- [5] Sen M. Kuo et al, "*Real-time* Digital Signal Processing, Implementations and Applications 2nd edition", John Wiley & Sons, 2006
- [6] Juliet Bates et al, "Converger Multimedia Networks", John Wiley & Sons, 2006
- [7] Emmanuel Seurre et al, "GPRS for *Mobile* Internet", Artech House, 2003
- [8] Kristo Lehtonen, "GSM Codec", T-61.264 Digital Signal Processing and Filtering. Diakses dari http://www.cis.hut.fi/Opinnot/T-61.246/Kutri2003/lehtonen_gsmdoc.pdf pada bulan Maret 2008
- [9] S. Andersen et al, "Internet Low Bit Rate Codec (iLBC)", Global IP Sound, Desember 2004. Diakses pada bulan Maret 2008 dari <http://www.ietf.org/rfc/rfc3951.txt>
- [10] Ajay R. Mishra, "Fundamentals of Cellular Network Planning and Optimisation" (Chicester: John Wiley & Sons, 2004), hal.109-114
- [11] Jim Van Meggelem et al, "Asterisk, The Future of Telephony", (USA: O'Reilly Media Inc., 2007)
- [12] "VoIP *Bandwidth* Calculation", Newport-Networks, 2005, hal. 3-5. Diakses pada bulan Maret 2008 dari <http://www.newport-networks.com/cust-docs/52-VoIP-Bandwidth.pdf>