

**RANCANG BANGUN DAN UNJUK KERJA
UPLINK VIDEO TELEPHONY BERBASIS SIP
MENGUNAKAN H.263 DAN H.263v2**

SKRIPSI

oleh:

DJUNDA AFIEF NUGROHO

04 04 03 030X



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

**RANCANG BANGUN DAN UNJUK KERJA
UPLINK VIDEO TELEPHONY BERBASIS SIP
MENGUNAKAN H.263 DAN H.263v2**

SKRIPSI

oleh:

DJUNDA AFIEF NUGROHO

04 04 03 030X



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

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

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

RANCANG BANGUN DAN UNJUK KERJA UPLINK VIDEO TELEPHONY BERBASIS SIP MENGUNAKAN H.263 DAN H.263v2

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 27 Maret 2008

Djunda Afief Nugroho

NPM 040403030X

PENGESAHAN

Skripsi dengan judul:

**RANCANG BANGUN DAN UNJUK KERJA
UPLINK VIDEO TELEPHONY BERBASIS SIP
MENGUNAKAN H.263 DAN H.263v2**

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang ujian skripsi.

Dosen Pembimbing 1,

Fitri Yuli Zulkifli, S.T., M.Sc.
NIP. 132 206 671

Depok, 27 Maret 2008

Dosen Pembimbing 2,

M. Suryanegara, S.T., M.Sc.
NIK. 040 705 0189

UCAPAN TERIMA KASIH

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

Fitri Yuli Zulkifli, S.T., M.Sc. dan 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 keluarga saya yang telah memberikan doa dan dukungan moril maupun materi sehingga skripsi ini dapat diselesaikan dengan baik.
2. Agus Awaludin selaku administrator PPSI yang telah memberikan pengarahan dalam mengerjakan skripsi ini.
3. Muhammad Zuhdan sebagai sahabat seperjuangan yang banyak membantu, mengarahkan, dan memotivasi untuk menyelesaikan skripsi ini.
4. Rekan-rekan elektro khususnya angkatan 2004 atas semangat yang diberikan kepada penulis.

Djunda Afief Nugroho
NPM 04 04 03 030X
Departemen Teknik Elektro

Dosen Pembimbing:
Fitri Yuli Zulkifli, S.T., M.Sc.
M.Suryanegara, S.T., M.Sc.

**RANCANG BANGUN DAN UNJUK KERJA *UPLINK* VIDEO
TELEPHONY BERBASIS SIP MENGGUNAKAN H.263 DAN H.263v2**

ABSTRAK

Teknologi telekomunikasi dan komputer berkembang sangat cepat. Perkembangan internet yang pesat merupakan hal penting yang mendukung perkembangan teknologi telekomunikasi. Teknologi internet dimanfaatkan menjadi media komunikasi untuk data dan suara. Dampaknya adalah migrasi semua *transport network* telekomunikasi secara bertahap menuju IP. Hal inilah yang menyebabkan lahirnya teknologi VoIP (*Voice over Internet Protocol*). Sistem VoIP memerlukan sebuah *signaling protocol* seperti H.323 dan SIP.

Video *telephony* dibangun dengan memanfaatkan SIP sebagai *signaling protocol* dan perangkat lunak yang bersifat *open source* dan *freeware*. Perangkat lunak yang digunakan terdiri dari X-lite v3.0 sebagai *client interface*, Linux Ubuntu v7.04 sebagai *server operating system*, dan beberapa perangkat lunak yang digunakan untuk membuat dan mengkonfigurasi *server* menjadi SIP *server* seperti asterisk versi 1.4.18, freePBX, apache, MySQL, dan PHP 5.

Uji coba video *telephony* dilakukan dengan mengkonfigurasi video *codec* H.263 dan H.263v2 dan diaplikasikan pada jaringan internal UI yang menyediakan *bandwidth* sebesar 100 Mbps, jaringan fastnet yang menyediakan *bandwidth* sebesar 746 Kbps, dan jaringan speedy yang menyediakan *bandwidth* sebesar 144.10 Kbps. Unjuk kerja dievaluasi dan dianalisis dengan melakukan pengamatan terhadap *uplink jitter* yang terjadi dan *uplink bandwidth* yang digunakan pada setiap konfigurasi.

Hasil uji coba terhadap unjuk kerja *uplink* video *telephony* berbasis SIP menunjukkan aplikasi ini dapat bekerja dengan baik pada *Local Area Network* Universitas Indonesia (LAN UI) dan menggunakan H.263v2 dengan *jitter* rata-rata yang dihasilkan berkisar antara 9.29 ms sampai 25.56 ms dengan pemakaian *bandwidth* berkisar antara 55.72 Kbps sampai 161.55 Kbps. Selain itu, uji coba ini menunjukkan H.263v2 sebagai *codec* yang memiliki skalabilitas yang lebih baik dalam penggunaan *bandwidth* dibandingkan H.263.

Kata kunci : VoIP, SIP, Video *Telephony*, H.263, H.263v2

Djunda Afief Nugroho
NPM 04 04 03 030X
Electrical Engineering Departement

Counselor :
Fitri Yuli Zulkifli, S.T., M.Sc
M.Suryanegara, S.T., M.Sc.

**DEVELOPMENT AND UPLINK PERFORMANCE SIP-BASED VIDEO
TELEPHONY USING H.263 DAN H.263v2**

ABSTRACT

Telecommunication and computer technology have been developed rapidly. Internet which has been developed rapidly is important to support the development of telecommunication technology. Internet technology has been used as communication media for data and voice. The result is migration of all telecommunication transport network in phases to IP (Internet Protocol) based and development of VoIP technology. VoIP system uses signaling protocol such as H.323 and SIP.

Video telephony was built by using SIP as signaling protocol with characters open source and free software. The software which has been used on video telephony application consists of X-lite v3.0 as client interface, Linux Ubuntu v7.04 as server operating system, and some software that are needed to develop and configure server as SIP server that consists of asterisk v1.4.18, freePBX, apache, MySQL, and PHP 5.

Test of video telephony application has been done by configuring video codec H.263 and H.263v2. This configuration is applied on UI network, fastnet network, and speedy network. Performance is evaluated and analyzed by observing uplink of jitter and bandwidth for each configuration.

The test result from uplink video telephony performance SIP based shows this configuration can work optimal on UI network which used H.263v2 with mean jitter within interval from 9.29 ms to 25.56 ms and with bandwidth consumption between 55.72 Kbps and 161.55 Kbps. Moreover, the test showed H.263v2 as a codec has better performance on Bandwidth consumption than H.263.

Keywords: VoIP, SIP, Video Telephony, H.263, H.263v2

DAFTAR ISI

HALAMAN JUDUL	ii
PERNYATAAN KEASLIAN SKRIPSI.....	iii
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. TUJUAN.....	2
1.3. PEMBatasan MASALAH	2
1.4. SISTEMATIKA PENULISAN	2
BAB 2 VOIP DAN SIP.....	4
2.1. <i>VOICE OVER INTERNET PROTOCOL (VOIP)</i>	4
2.1.1. Pengantar VoIP	4
2.1.2. Protokol Penunjang Jaringan VoIP	6
2.1.3. Keuntungan dan Kelemahan Penggunaan VoIP	9
2.2. <i>SESSION INITIATION PROTOCOL (SIP)</i>	11
2.2.1. Pengertian, karakteristik, dan Fungsi SIP	11
2.2.2. Arsitektur SIP.....	12
2.2.3. Nama dan Pengalamatan.....	15
2.2.4. <i>SIP Message</i>	15
2.2.5. <i>Session Description Protocol</i>	18
2.2.6. Pembentukan dan Terminasi Sesi	20
BAB 3 PERANCANGAN VIDEO <i>TELEPHONY</i> BERBASIS SIP.....	22
3.1. ARSITEKTUR VIDEO <i>TELEPHONY</i> BERBASIS SIP.....	22

3.1.1. Arsitektur <i>Server</i>	23
3.1.2. Arsitektur <i>Client</i>	27
3.2. VIDEO <i>CODEC</i>	29
3.2.1. H.263	29
3.2.2. H.263v2	29
BAB 4 ANALISIS DAN UJI COBA	30
4.1. SKENARIO UJI COBA	30
4.2. UJI COBA JARINGAN INTERNAL UI	31
4.2.1. <i>Uplink Client</i> 152.118.101.146 ke <i>Server</i>	31
4.2.2. <i>Uplink Client</i> 152.118.101.206 ke <i>Server</i>	33
4.2.3. Perbandingan Unjuk Kerja Kedua <i>Client</i>	35
4.3. UJI COBA ANTARA LAN UI DAN FASTNET	36
4.3.1. <i>Uplink Client</i> 118.137.23.94 ke <i>Server</i>	36
4.3.2. <i>Uplink Client</i> 152.118.101.206 ke <i>Server</i>	38
4.3.3. Perbandingan Unjuk Kerja Kedua <i>Client</i>	40
4.4. UJI COBA ANTARA LAN UI DAN SPEEDY	41
4.4.1. <i>Uplink Client</i> 152.118.101.146 ke <i>Server</i>	41
4.4.2. <i>Uplink Client</i> 125.161.136.172 ke <i>Server</i>	43
4.4.3. Perbandingan Unjuk Kerja Kedua <i>Client</i>	45
4.5. PERBANDINGAN UNJUK KERJA <i>UPLINK</i> UI NETWORK, FASTNET, DAN SPEEDY	46
BAB 5 KESIMPULAN	48
DAFTAR ACUAN	49

DAFTAR GAMBAR

Gambar 2.1. Arsitektur Jaringan VoIP [4]	5
Gambar 2.2 Layer VoIP	5
Gambar 2.3 Header RTP [7]	6
Gambar 2.4 Struktur paket TCP [7]	8
Gambar 2.5 Struktur paket UDP [7]	9
Gambar 2.6 Contoh panggilan SIP menggunakan <i>redirect server</i>	13
Gambar 2.7 Contoh panggilan SIP dengan menggunakan <i>Proxy server</i>	14
Gambar 2.8 Contoh proses registrasi SIP	14
Gambar 2.9 SIP pada <i>Redirect mode</i> [11]	20
Gambar 2.10 SIP pada <i>Proxy mode</i> [11]	21
Gambar 3.1. Arsitektur umum video <i>telephony</i>	22
Gambar 3.2 Tampilan X-Lite 3.0	27
Gambar 3.3 Diagram alir penggunaan X-Lite	28
Gambar 4.1 Topologi Jaringan Uji coba	30
Gambar 4.2 Grafik nomor paket vs <i>jitter Uplink client 152.118.101.146 ke server</i>	32
Gambar 4.3 Grafik nomor paket vs <i>bandwidth Uplink client 152.118.101.146 ke server</i>	33
Gambar 4.4 Grafik nomor paket vs <i>Jitter Uplink client 152.118.101.206 ke server</i>	34
Gambar 4.5 Grafik nomor paket vs <i>bandwidth Uplink client 152.118.101.206 ke server</i>	35
Gambar 4.6 Grafik nomor paket vs <i>Jitter Uplink client 118.137.23.94 ke server</i>	36
Gambar 4.7 Grafik nomor paket vs <i>bandwidth Uplink client 118.137.23.94 ke server</i>	37
Gambar 4.8 Grafik nomor paket vs <i>Jitter Uplink client 152.118.101.206 ke server</i>	39
Gambar 4.9 Grafik nomor paket vs <i>bandwidth uplink client 152.118.101.206 ke server</i>	40

Gambar 4.10 Grafik nomor paket vs <i>Jitter Uplink client</i> 152.118.101.146 ke <i>server</i>	42
Gambar 4.11 Grafik nomor paket vs <i>bandwidth Uplink client</i> 152.118.101.146 ke <i>server</i>	43
Gambar 4.12 Grafik Paket vs <i>Jitter Uplink client</i> 125.161.136.172 ke <i>server</i>	44
Gambar 4.13 Grafik nomor paket vs <i>bandwidth Uplink client</i> 125.161.136.172 ke <i>server</i>	45



DAFTAR TABEL

Tabel 2.1 Tipe-tipe paket RTCP [6].....	7
Tabel 2.2 SDP <i>field</i> dan urutannya	19
Tabel 4.1 Perbandingan <i>Jitter uplink</i> jaringan internal UI.....	35
Tabel 4.2 Perbandingan <i>bandwidth uplink</i> jaringan internal UI	35
Tabel 4.3 Perbandingan <i>jitter uplink</i> jaringan internal UI dan fastnet.....	40
Tabel 4.4 Perbandingan <i>bandwidth uplink</i> jaringan internal UI dan fastnet.....	40
Tabel 4.5 Perbandingan <i>Jitter uplink</i> jaringan internal UI dan speedy.....	45
Tabel 4.6 Perbandingan <i>bandwidth uplink</i> jaringan internal UI dan speedy	45
Tabel 4.7 Perbandingan <i>jitter uplink</i> UI, Fastnet, dan Speedy.....	46
Tabel 4.8 Perbandingan <i>bandwidth uplink</i> UI, Fastnet, dan Speedy	46

DAFTAR SINGKATAN

<i>CODEC</i>	<i>Coder Decoder</i>
<i>DMZ</i>	<i>De-Militarized Zone</i>
<i>IETF</i>	<i>Internet Engineering Task Force</i>
<i>IMS</i>	<i>IP Multimedia System</i>
<i>IP</i>	<i>Internet Protocol</i>
<i>IPTV</i>	<i>Internet Protocol Television</i>
<i>IPv4</i>	<i>Internet Protocol version 4</i>
<i>IPv6</i>	<i>Internet Protocol version 6</i>
<i>ISP</i>	<i>Internet Service Provider</i>
<i>ITU-T</i>	<i>International Telecommunication Union</i>
<i>NAT</i>	<i>Network Address Transverse</i>
<i>LAN</i>	<i>Local Area Network</i>
<i>MGW</i>	<i>Media Gateway</i>
<i>PC</i>	<i>Personal Computer</i>
<i>PSTN</i>	<i>Public Switch Telephone Network</i>
<i>QoS</i>	<i>Quality of Service</i>
<i>RTP</i>	<i>Real Time Protocol</i>
<i>RTCP</i>	<i>Real Time Control Protocol</i>
<i>SSRCI</i>	<i>Synchronization Source Identifier</i>
<i>SIP</i>	<i>Session Initiation Protocol</i>
<i>TCP</i>	<i>Transmission Control Protocol</i>
<i>UA</i>	<i>User Agent Client</i>
<i>UAC</i>	<i>User Agent Client</i>
<i>UAS</i>	<i>User Agent Server</i>
<i>UDP</i>	<i>User Datagram Protocol</i>
<i>UE</i>	<i>User Equipment</i>
<i>URI</i>	<i>Uniform Resource Identifier</i>
<i>VoIP</i>	<i>Voice over IP</i>

BAB 1

PENDAHULUAN

1.1. LATAR BELAKANG

Perkembangan teknologi telekomunikasi dan komputer belakangan ini sangat cepat. Perkembangan teknologi komputer ditandai dengan adanya konversi data *analog* menjadi *digital* yang mengakibatkan manipulasi data menjadi lebih mudah dan cepat. Hal ini menyebabkan tren teknologi telekomunikasi menjadi telekomunikasi *digital* dan kecepatan telekomunikasi sudah mendukung untuk meluncurkan berbagai aplikasi [1]. Selain itu, perkembangan teknologi telekomunikasi dan komputer juga dipengaruhi adanya perubahan metode transmisi data dan informasi dari *circuit switching* menjadi *packet switching*. Telekomunikasi tidak hanya dapat dilakukan dengan bentuk suara tetapi juga dalam bentuk gambar ataupun video. Perkembangan ini akan terus berlanjut dan teknologi telekomunikasi akan lebih kompleks di masa mendatang. Hal tersebut menumbuhkan berbagai aplikasi baru dengan teknologi yang lebih kompleks juga [1].

Perkembangan internet yang pesat merupakan hal penting yang mendukung perkembangan teknologi telekomunikasi. Teknologi internet dimanfaatkan menjadi media *transport* untuk komunikasi data dan suara. Dampaknya adalah migrasi semua *transport network* telekomunikasi secara bertahap menuju IP. Hal inilah yang menyebabkan lahirnya teknologi VoIP (*Voice over Internet Protocol*).

Quality of Service (QoS) [2,3] adalah permasalahan mendasar VoIP. *Jitter* dan *bandwidth* merupakan dua hal yang termasuk dalam kajian QoS. Berbagai *provider* internet menjamur dan menawarkan berbagai kualitas layanan *downlink* yang tinggi. Namun mereka sering melupakan *uplink* dari layanan yang mereka tawarkan.

Anton Raharja [4] menyatakan bahwa terdapat solusi dalam pengembangan VoIP yaitu pengembangan SIP sebagai basis VoIP. Adanya penambahan fitur video pada berbagai aplikasi VoIP menuntut kualitas jaringan

VoIP yang baik. Video *telephony* merupakan salah satu aplikasinya. Berbagai cara dicoba agar teknologi ini dapat dimanfaatkan secara maksimal. Mulai dari pemilihan *codec* sampai dengan penyediaan *bandwidth* yang lebih besar. Untuk itu, diperlukan konfigurasi aplikasi yang berbasis IP yang dapat beradaptasi dengan kondisi teknologi sekarang.

Penerapan sebuah teknologi video *telephony* berbasis SIP merupakan sebuah solusi dari permasalahan yang terjadi pada pemanfaatan video *telephony*. Konfigurasi yang dijadikan parameter unjuk kerja video *telephony* berbasis SIP ini adalah H.263 dan H.263v2 yang mampu digunakan dengan *bit rate* rendah serta jaringan LAN UI, fastnet, dan speedy yang memiliki perbedaan *available bandwidth*.

1.2. TUJUAN

1. Merancang dan membangun arsitektur video *telephony* berbasis SIP
2. Membandingkan dan menganalisis unjuk kerja video *codec* H.263 dan H.263v2 pada jaringan speedy, fastnet, dan jaringan UI.

1.3. PEMBATAAN MASALAH

1. Skripsi ini menganalisis unjuk kerja *uplink* (dari *client* ke *server*) ketika sebuah sesi video *telephony* berbasis SIP berlangsung
2. Pembahasan dibatasi pada kondisi paket video ketika transmisi dengan *codec* H.263 dan H.263v2
3. Analisis unjuk kerja difokuskan pada *jitter* dan *bandwidth* pada pengiriman paket video

1.4. SISTEMATIKA PENULISAN

Skripsi ini terdiri dari 5 bab dengan sistematika penulisan yang diterapkan dalam skripsi ini menggunakan urutan sebagai berikut :

BAB 1 PENDAHULUAN

Bagian ini membahas tentang latar belakang, tujuan, pembatasan masalah, dan sistematika penulisan.

BAB 2 VOIP DAN SIP

Membahas tentang teori tentang teknologi dasar pendukung video *telephony* berbasis SIP. Mulai dari pengantar VoIP, *transport protocol*, *codec* , serta SIP secara menyeluruh.

BAB 3 PERANCANGAN VIDEO *TELEPHONY* BERBASIS SIP

Membahas tentang arsitektur perancangan video *telephony* yang terdiri dari *operating system server*, konfigurasi *server*, serta *hardware* dan *software client*.

BAB 4 ANALISIS DAN UJI COBA

Membahas mengenai hasil pengujian dan evaluasi unjuk kerja dari *uplink* video *telephony* berbasis SIP yang dijalankan pada jaringan LAN UI, fastnet, dan speedy.

BAB 5 KESIMPULAN

Merupakan bagian akhir dari laporan ini yang memuat pernyataan singkat dan tepat yang dijabarkan dari hasil studi literatur dan dasar teori.

BAB 2

VOIP DAN SIP

2.1. VOICE OVER INTERNET PROTOCOL (VOIP)

2.1.1. Pengantar VoIP

Voice over Internet Protocol (VoIP) merupakan sebuah integrasi teknologi *hardware* dan *software* yang memungkinkan adanya percakapan melalui media internet. VoIP yang biasa disebut *internet telephony* mampu melewati trafik suara, video, dan data dalam bentuk paket-paket melalui jaringan IP (*packet-switched network*). Data, suara, dan video yang akan ditransmisikan terlebih dahulu diubah dari kode analog ke *digital*. Data, suara, dan video yang telah diubah ke dalam kode *digital* dikompresi dan kemudian dipisahkan menjadi paket-paket. Kemudian paket-paket tersebut dialamatkan dan dikirim secara terpisah melalui jaringan internet menuju tujuan sama yang telah ditentukan. Selama transmisi, paket-paket tersebut kemungkinan mengalami *delay*, *lost*, ataupun *error*. Setelah paket-paket ditransmisikan dan tiba di tujuan, paket-paket tersebut disatukan kembali dan di-dekompresi untuk membentuk data ke bentuk aslinya.

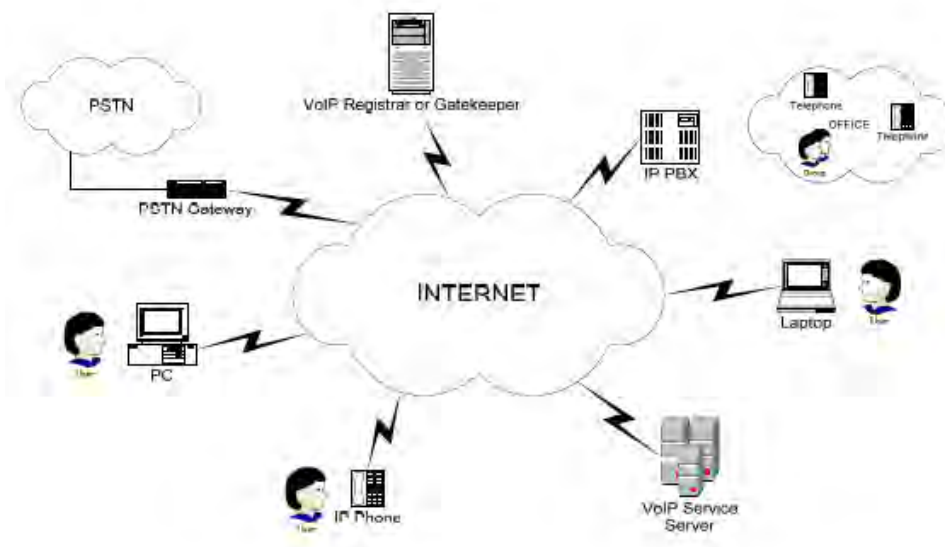
Pada dasarnya, VoIP merupakan revolusi telepon konvensional. Oleh karena itu, semua layanan telepon konvensional dapat pula dilakukan dengan teknologi VoIP. Layanan yang merupakan aplikasi tambahan dengan adanya VoIP diantaranya adalah *voicemail*, *instant messaging*, dan *video telephony*. *End-point* VoIP dapat dilakukan dengan berbagai media yaitu:

- PC ke PC
- PC ke telepon
- Telepon ke PC
- Telepon ke telepon

Gambar 2.1 memperlihatkan arsitektur jaringan VoIP yang terdiri dari *server* dan *client* yang dapat menggunakan berbagai jenis *end-point*.

Protokol VoIP secara umum dibagi menjadi dua bagian, yaitu kontrol (*signaling*) dan data *voice* [5]:

- 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*. Protokol *signaling* yang biasa digunakan adalah H.323 dan SIP.
- Data *voice* adalah trafik *user* berupa informasi yang disampaikan *end-to-end* yang dikenal juga sebagai *Packet Voice*.



Gambar 2.1. Arsitektur Jaringan VoIP [4]

VoIP menggunakan IP sebagai *basic transport*. Sedangkan pada *transport layer*, VoIP menggunakan TCP dan UDP. Gambar 2.2 menunjukkan susunan *protocol VoIP* di jaringan.

VoIP		
TCP		UDP
IP		
HDLC	ATM	ETHERNET
PHYSICAL		

Gambar 2.2 Layer VoIP

2.1.2. Protokol Penunjang Jaringan VoIP

a. *Real Time Protocol*

Real Time Protocol (RTP) merupakan format standardisasi paket untuk pengiriman suara dan video melalui jaringan internet. 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*. RTP memungkinkan dilakukannya deteksi terhadap beberapa kelemahan seperti [6]:

- *Packet loss*
- *Transport Delay.*
- Paket tiba secara tidak berurutan

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
V=2		P	X	CC		M	PT				sequence number																				
timestamp																															
synchronization source (SSRC) identifier																															
... contributing source (CSRC) identifiers (optional) ...																															
header extension (optional)																															

Gambar 2.3 Header RTP [7]

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*: berfungsi untuk mendeteksi hilangnya paket 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*: *field* ini hanya ada jika paket RTP telah dikirim oleh *mixer*.

b. Real Time Control Protocol

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 [6]:

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

Tabel 2.1 menunjukkan tipe-tipe paket RTCP

Tabel 2.1 Tipe-tipe paket RTCP [6]

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

c. Transmission Control Protocol

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*. Konsep dasar kerja TCP adalah mengirim dan menerima informasi yang berbentuk semen-segmen informasi dengan panjang data bervariasi pada suatu datagram internet. Selama transmisi, TCP membangun koneksi yang kuat antara *host* pengirim dan *host* penerima [7]. 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. Dalam hubungannya dengan VoIP, TCP digunakan pada saat *signaling*. TCP digunakan untuk menjamin *setup* suatu *Call* pada sesi *signaling* [8]. Gambar 2.4 menunjukkan struktur paket TCP.

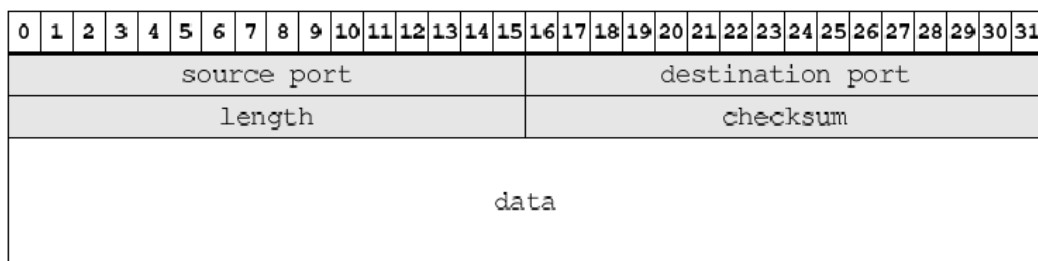
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
source port																destination port															
sequence number																															
acknowledgment number																															
offset				rsvd.				ECN				flags				window															
checksum																urgent pointer															
options (optional)																															
data																															

Gambar 2.4 Struktur paket TCP [7]

d. User Datagram Protocol

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. UDP pada VoIP digunakan untuk mengirimkan *audio stream* yang dikirimkan secara terus-menerus. VoIP menggunakan UDP 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 hanya mencapai 50% dari jumlah paket yang dikirimkan [8]. Data dikirimkan dari *host* pengirim ke *host* penerima menggunakan *fixed rate* tanpa memandang hambatan dan kemampuan penerima [7]. 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. Untuk aplikasi *streaming multimedia*, pengiriman ulang paket tidaklah bermanfaat. Alasannya adalah adanya pembatasan waktu yang diharuskan oleh aplikasi *multimedia*. Paket-paket yang memuat data *multimedia* dan telah dikirimkan dua kali tidak akan tiba pada waktu yang sama [7]. Pengiriman ulang bukan hanya tidak berguna tetapi juga dapat mengakibatkan hal yang berbahaya bagi unjuk kerja aplikasi *streaming*. Gambar 2.5 menunjukkan struktur paket UDP.



Gambar 2.5 Struktur paket UDP [7]

2.1.3. Keuntungan dan Kelemahan Penggunaan VoIP

Banyak keuntungan yang akan kita dapatkan dalam penggunaan VoIP. Sudut pandangnya bisa dilihat sebagai teknologi pengganti PBX atau PSTN yang sampai 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 [9]:

- **Harga Peralatan yang lebih murah:** teknologi voice over IP dibangun berbasis teknologi internet yang perkembangannya sangat cepat.

Perkembangan yang sangat cepat ini menyebabkan harga peralatan akan turun dengan sangat cepat. Berbeda dengan teknologi PBX dan PSTN yang cenderung statis.

- **Efisiensi *bandwidth***: sistem VoIP *scalable* dalam menangani jumlah panggilan yang banyak (*large Call volume*) dan *traffic prioritization* yang akan menjamin bahwa paket dapat dengan cepat diproses. Besar kanal yang saat ini digunakan oleh satu kanal suara dengan teknologi VoIP dapat digunakan oleh delapan kanal suara, sehingga otomatis tarif telekomunikasi pun akan menurun drastis. Penurunan tarif akan semakin terasa jika sistem ini digunakan untuk SLJJ.
- **Biaya perawatan yang murah**: karena VoIP berbasis pada teknologi internet, jaringan internet yang sudah ada dapat diperluas sehingga mendukung teknologi ini dengan harga yang tidak mahal. IP *phone* dapat dengan mudah dipindah, ditambah dan diubah. Hal ini disebabkan karena VOIP dapat dipasang di sembarang *ethernet jack* dan IP *address*, tidak seperti sistem telepon tradisional yang harus mempunyai *port* yang khusus di PBX. Selain itu, jaringan dapat dikelola dengan cara dan pengetahuan yang tidak jauh berbeda dengan mengelola jaringan internet. Pengelola internet di jaringan yang bersangkutan cukup mengatur satu jaringan saja. Tidak diperlukan training tambahan karena orang IT sudah terbiasa dengan peralatan jaringan yang digunakan.
- **Munculnya aplikasi baru**: karena VoIP dibangun diatas jaringan internet, yang dikenal dengan timbulnya berbagai jenis aplikasi baru yang menarik, maka dimungkinkan adanya pengembangan dan penambahan *service* baru untuk *value added service* (VAS) servis yang sudah ada. Contohnya *user* dapat menelepon tanpa harus memutuskan sambungan internetnya. *User* dapat dihubungi walaupun sedang berinternet dan lain-lain.

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) dan inilah yang kemudian menjadi kelemahan teknologi VoIP saat ini. Agar jaringan internet berhasil melewatkan paket data berupa suara ataupun

video, jaringan ini harus dimodifikasi sedemikian rupa sehingga mampu memberikan *Quality of Service* sesuai permintaan aplikasi VoIP ini. Parameter tersebut yaitu [8]:

- **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* merupakan variasi dalam *delay* dan terjadi karena adanya perubahan terhadap karakteristik dari suatu sinyal sehingga menyebabkan terjadinya masalah terhadap data yang dibawa oleh sinyal tersebut.
- **Packet loss:** *Paket loss* merupakan 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 penggunaannya.

2.2. SESSION INITIATION PROTOCOL (SIP)

Terdapat dua standardisasi utama dalam *signaling* VoIP yaitu H.323 yang dibuat oleh ITU-T (*International Telecommunications Union–Technology*) dan SIP yang dikembangkan oleh IETF (*Internet Engineering Task Force*).

2.2.1. Pengertian, karakteristik, dan Fungsi SIP

Session Initiation Protocol (SIP) merupakan salah satu standardisasi pensinyalan dan pengontrolan sesi dari *packet telephony* yang dikembangkan oleh IETF sebagai bagian dari *Internet Multimedia Conferencing Architecture* [6]. SIP merupakan protokol kontrol lapisan aplikasi yang berfungsi untuk membuat, memodifikasi, dan terminasi sebuah sesi dengan satu atau lebih *participant*. Karena SIP termasuk salah satu *signaling protocol* VoIP, maka SIP bukanlah protokol *transport* yang membawa paket data, suara ataupun video. Protokol yang melakukan tugas tersebut biasanya adalah TCP atau UDP yang menggunakan *port* 5060. Berdasarkan definisi tersebut, maka SIP memiliki beberapa fungsi umum yaitu [10]:

- *Call initiation*
 - Membangun sebuah sesi komunikasi
 - Negosiasi media transfer protokol
 - Mengundang *user agent* lain untuk bergabung di dalam sesi komunikasi
- *Call modification*
 - Bila perlu, SIP dapat memodifikasi sesi komunikasi
- *Call termination*
 - Menutup sesi komunikasi

Sesi multimedia merupakan proses mengalirnya data antara pengirim dan penerima selama hubungan anatara keduanya terjadi. Dalam hal pengaturan komunikasi multimedia, maka SIP memiliki kemampuan untuk [11]:

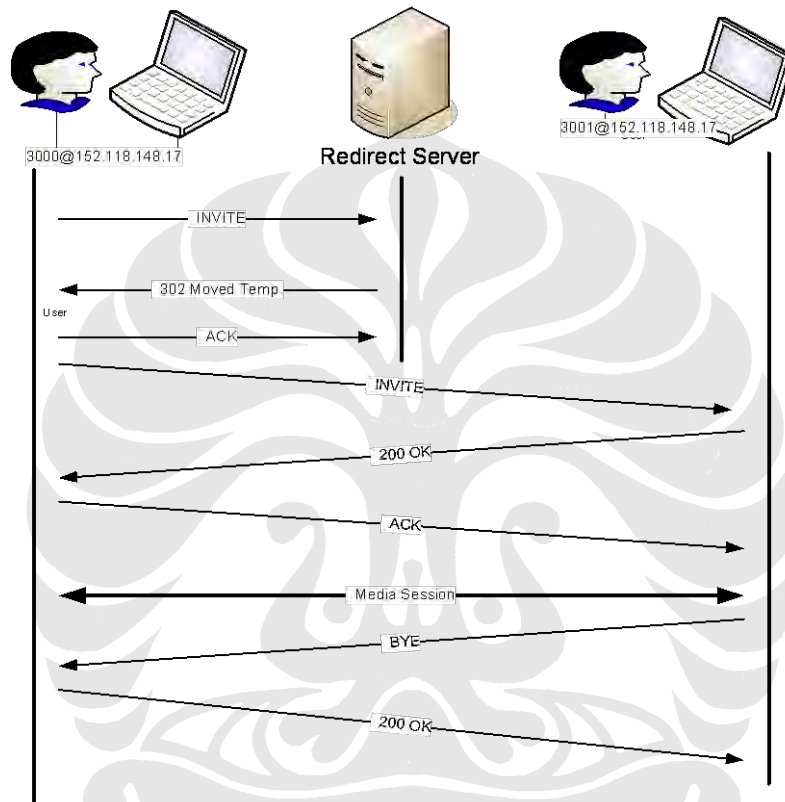
- Menentukan tujuan lokasi terakhir pengguna
- Menentukan apakah pengguna akan berpartisipasi pada sebuah sesi
- Menentukan kemampuan terminal pengguna
- Membangun sebuah sesi
- Mengatur sesi

2.2.2. Arsitektur SIP

Seperti halnya HTTP (*Hyper Text Transfer Protocol*), SIP adalah protokol komunikasi dan terdiri dari *client* dan *server*. SIP *client* adalah elemen jaringan yang membangkitkan SIP *request* dan menerima SIP *response* sedangkan SIP *server* adalah elemen jaringan yang menerima SIP *request* untuk menanggapi *request* tersebut dan mengirimkan kembali *response* atas *request* tersebut [11]. SIP memiliki 4 komponen utama dalam arsitekturnya yaitu:

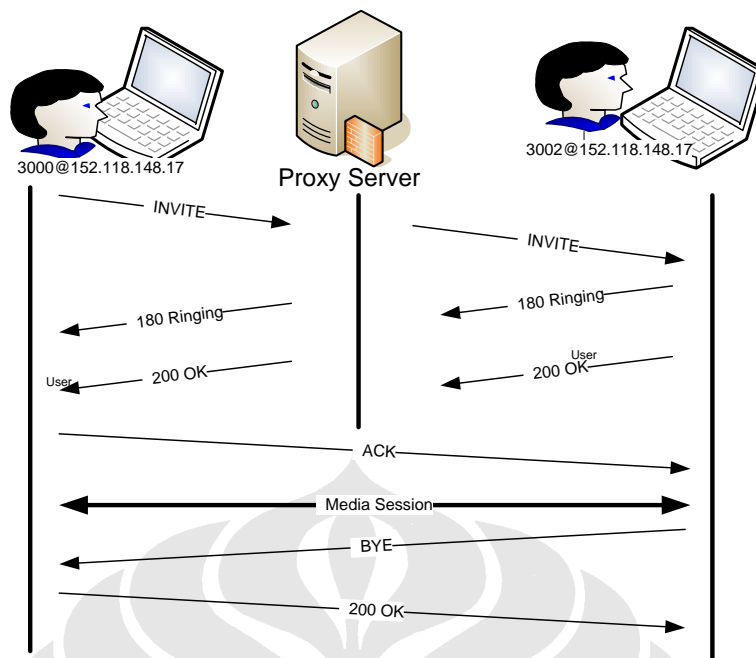
- **SIP User Agent:** *User Agent* (UA) adalah sebuah *internet endpoint*, seperti telepon atau PC, yang digunakan untuk membangun, memodifikasi, dan terminasi sebuah sesi [11]. *User Agent* terdiri dari *User Agent Server* (UAS) dan *User Agent Client* (UAC). UAC adalah aplikasi pemanggil yang memulai dan mengirimkan SIP *request* sedangkan UAS berperan dalam menerima dan menjawab SIP *request* [8].
- **SIP redirect server:** *redirect server* merupakan UAS yang menghasilkan respon untuk mengalihkan sebuah *request* ke lokasi lain [11]. Oleh karena itu,

redirect server bertugas melakukan *mapping address* ke *address* baru dan mengembalikan *address* baru tersebut ke *client*. Gambar 2.6 menunjukkan contoh panggilan SIP oleh *user* dengan URI sip:3000@152.118.148.17 ke sip:3001@152.118.148.17 menggunakan *redirect server*.



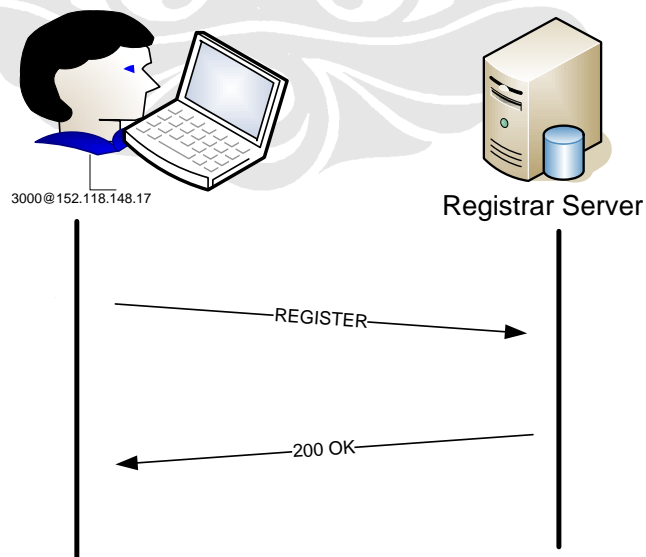
Gambar 2.6 Contoh panggilan SIP menggunakan *redirect server*

- **SIP Proxy Server:** *Proxy server* memuat UAC dan UAS serta berperan menghubungkan satu atau lebih *client* atau *server* lain dan menyampaikan *Call request* atau sebagai penengah antar-*user agent* untuk melakukan *routing SIP message* ke *user* tujuan. Gambar 2.7 menunjukkan contoh panggilan SIP oleh *user* dengan URI sip:3000@152.118.148.17 ke sip:3001@152.118.148.17 menggunakan *proxy server*.



Gambar 2.7 Contoh panggilan SIP dengan menggunakan *Proxy server*

- **SIP registrar:** Sebuah *registrar* merupakan sebuah UAS yang memproses SIP *REGISTER request*. *Registrar* menyimpan pemetaan dari SIP *username* ke *address* dan merupakan ujung awal dari *location service*. Gambar 2.8 menunjukkan proses registrasi *user* SIP



Gambar 2.8 Contoh proses registrasi SIP

2.2.3. Nama dan Pengalamatan

Setiap komponen SIP memiliki pengalamatan masing-masing. Sebagai contoh, setiap pengguna SIP teridentifikasi oleh sebuah SIP *Uniform Resource Identifier* (URI) [11]. 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 Djunda yang terdaftar di host 152.118.148.17, maka Djunda memiliki SIP URI:

```
sip:djunda@152.118.148.17
```

dengan 152.118.148.17 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@host:port;uri-parameters?headers
```

Bagian pertama bisa berupa sip: atau sips:. Berikutnya adalah *userinfo* yang terdiri atas *username* dan *password* dengan format *user:password@host*. 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 [11].

2.2.4. SIP Message

Setiap SIP *message* dapat berupa *request message* atau *response message*. Sebuah *request message* dikirim dari *client* (UAC) ke *server* (UAS) [11]. SIP *message* ditulis dalam *cleartext*. Oleh karena itu, SIP merupakan *text-based protocol*. Berikut ini adalah metode *message* yang didefinisikan pada SIP:

- INVITE: digunakan oleh *user* untuk membangun sebuah sesi SIP [12].
- ACK: digunakan untuk konfirmasi respon akhir [13].
- CANCEL: digunakan untuk membatalkan sebuah SIP *request* [13].
- OPTIONS: digunakan untuk menanyakan kemampuan *server* [13].

- *REGISTER*: digunakan oleh *user* untuk *register* informasi (contoh: lokasi terakhir *user*) dengan *server* [13].
- *INFO*: digunakan untuk membawa sesi yang berhubungan dengan kontrol informasi [12].
- *SUBSCRIBE*: digunakan untuk *request* kondisi terakhir dan *update* kondisi dari sebuah titik *remote* [14].
- *NOTIFY*: digunakan untuk notifikasi SIP *node* bahwa sebuah kejadian yang telah diminta oleh metode *SUBSCRIBE* awal telah terjadi [14].
- *PRACK*: digunakan untuk menyediakan *Provisional Response ACKnowledgement* (*PRACK*) yang dapat diandalkan [15].
- *UPDATE*: digunakan untuk meng-*update* parameter sesi seperti kumpulan media stream dan *codec* -nya [16].
- *MESSAGE*: digunakan untuk melakukan transfer *Instant Message* (*IM*) [7].
- *REFER*: digunakan untuk mengarahkan langsung penerima ke sumber lain dengan menggunakan informasi kontak yang tersedia pada permintaan *REFER*. Ini dapat digunakan untuk transfer panggilan [17].

SIP *message* ditulis dalam cleartext dengan kata lain SIP merupakan text-based *protocol*. setiap SIP *message* terdiri dari [11]:

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

Berikut ini adalah contoh bentuk sebuah SIP *message*:

<i>Start-line</i>	INVITE sip:3000@152.118.148.17 SIP/2.0
<i>Header</i>	From: <sip:3000@152.118.148.17>;tag=4f7cc7f012fb1810999dc48bb2a699b9
	To: sip:3001@152.118.148.17
<i>Field(s)</i>	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:3001@152.118.101.206>
	User-Agent: X-Lite release 1002tx stamp 29712
	Max-Forwards: 70

Content-Type: application/sdp
Content-Length: 328

Empty Line

Message Body v=0
(optional) 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
.....

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

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 [11]. Beberapa dari *Header field* hanya digunakan pada *request message* dan beberapa hanya digunakan untuk *response message*.

Status-code merupakan 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* [11]:

- 1xx: *Informational* - Mengindikasikan sebuah *request* sedang dalam proses
- 2xx: *Success* - Mengindikasikan metode pada sebuah *request* telah diterima
- 3xx: *Redirection* - Tindakan lebih jauh perlu dilakukan oleh pengirim untuk melengkapi *request*-nya
- 4xx: *Client error* - *Request* memiliki kesalahan *syntax* sehingga tidak dapat dipenuhi
- 5xx: *Server error* - *Server* gagal memenuhi *request* yang *valid*
- 6xx: *Global error*- *Request* tidak dapat dipenuhi di semua *server*

2.2.5. Session Description Protocol

Session Description Protocol (SDP) dikembangkan oleh IETF untuk mendiskripsikan sesi multimedia [11]. SDP mengarah ke sintaks deskripsi dan tidak mengarah sebagai sebuah protokol karena tidak mempunyai kemampuan negosiasi media secara menyeluruh [6]. SDP digunakan dengan memasukkan SDP ke dalam *message body* protokol lain., SIP *message* bisa berisi SDP dengan penanda `Content-Type:application/sdp`. Seperti yang telah dispesifikasikan oleh RFC 2327 [11], 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.

Sama halnya dengan SIP, SDP menggunakan *text coding* yang terdiri atas baris-baris yang disebut *fields*. Tabel 2.2 menunjukkan SDP *field* dan urutannya dan berikut ini adalah contoh dari SDP *message*:

```
v=0
o=- 3 2 IN IP4 152.118.101.206
s=<CounterPath eyeBeam 1.5>
c=IN IP4 152.118.101.206
t=0 0
m=audio 52692 RTP/AVP 107 119 0 98 8 3 101
a=alt:1 1 : JAhJ+3Dd V2+DECqv 152.118.101.206 52692
a=fmtp:101 0-15
a=rtpmap:107 BV32/16000
a=rtpmap:119 BV32-FEC/16000
a=rtpmap:98 iLBC/8000
a=rtpmap:101 telephone-event/8000
a=sendrecv
a=x-rtp-session-id:9AA6C678DEEB44DBBB99BD3D91A8B87F
m=Video 10950 RTP/AVP 115 34
a=alt:1 1 : +HGpILyg c+CF12Kt 152.118.101.206 10950
a=fmtp:115 QCIF=2 CIF=3 MAXBR=1960
a=fmtp:34 QCIF=2 MAXBR=1960
a=rtpmap:115 H263-1998/90000
a=rtpmap:34 H263/90000
a=sendrecv
a=x-rtp-session-id:9E822D03FE504800ABAA3C6B2153B645
```

Dari SDP *message* di atas, kita dapat melihat bahwa format teks pada SDP dapat dirumuskan sebagai berikut [5]:

<type>=<value>

dengan <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 [11]:

x=parameter1 parameter2 ... parameterN

Tabel 2.2 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=	<i>Phone</i>	Nomor <i>telephone</i>
c=	<i>Connection</i>	Informasi mengenai koneksi
b=	<i>Bandwidth</i>	Informasi <i>bandwidth</i>
t=	<i>Time</i>	Waktu mulai dan waktu berhenti sesi
r=	Repeat <i>times</i>	Jumlah pengulangan
z=	<i>Time zones</i>	Koreksi zona waktu
k=	<i>Encryption key</i>	Enkripsi
a=	<i>Attributes</i>	Baris atribut
m=	Media announcements	Informasi media
a=	<i>Attributes</i>	Atribut media

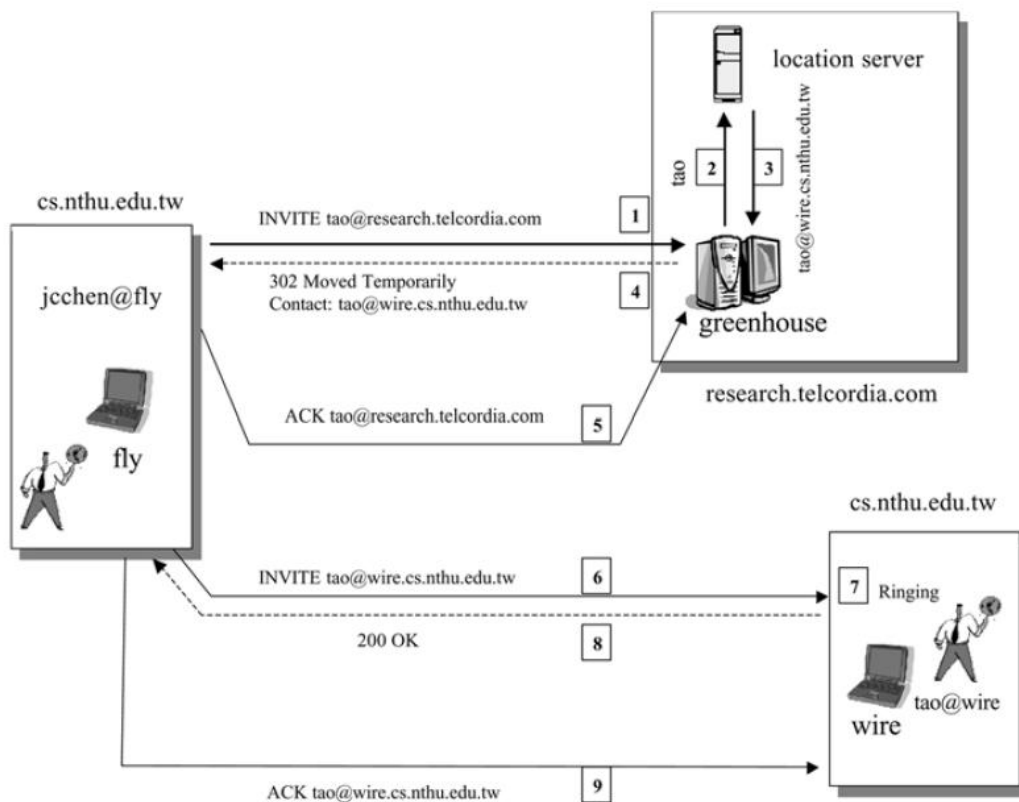
SIP menggunakan SDP *field* seperti *version*, *origin*, *subject*, *time*, *connection*, dan satu atau lebih media dan *attribute*. *Origin*, *subject*, dan *time* tidak digunakan oleh SIP tetapi digunakan untuk kepentingan kompatibilitas. Pada standar SDP, *subject field* merupakan *field* yang harus ada dan harus berisi setidaknya satu karakter, jika tak ada *subject*, dianjurkan menjadi *s=-* [6].

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 melakukan spesifikasi berbagai alternatif jenis media untuk menerima atau menolak jenis media tersebut secara selektif [6].

2.2.6. Pembentukan dan Terminasi Sesi

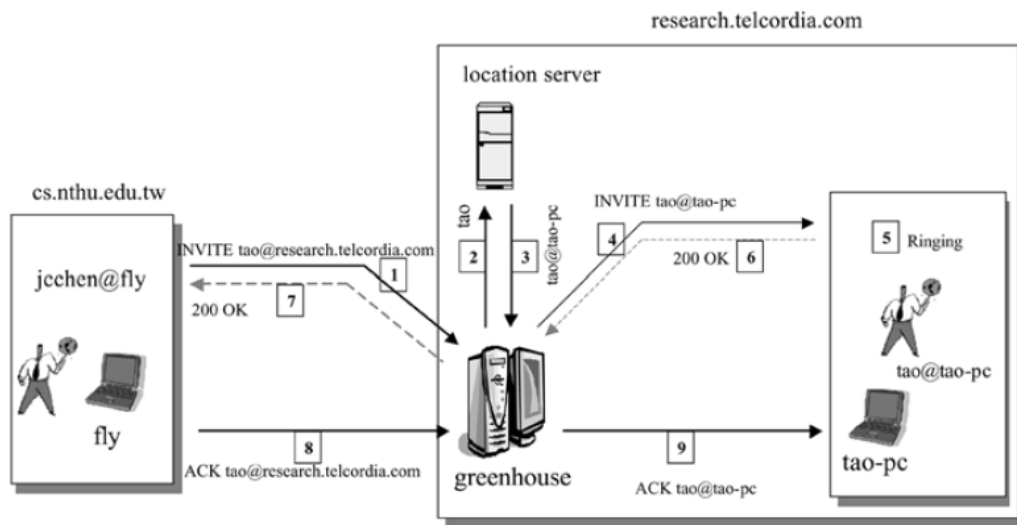
Sebuah sesi SIP dapat dibentuk menggunakan *mode peer-to-peer mode* ataupun *server mode*. Pada *peer-to-peer mode*, pemanggil membuat panggilan langsung tanpa melalui SIP server. Dalam hal ini, pemanggil harus mengetahui lokasi *user* yang akan dipanggilnya. Sedangkan untuk *server mode*, terdapat dua sistem yang digunakan, yaitu SIP *Proxy server* dan SIP *redirect server* [11].



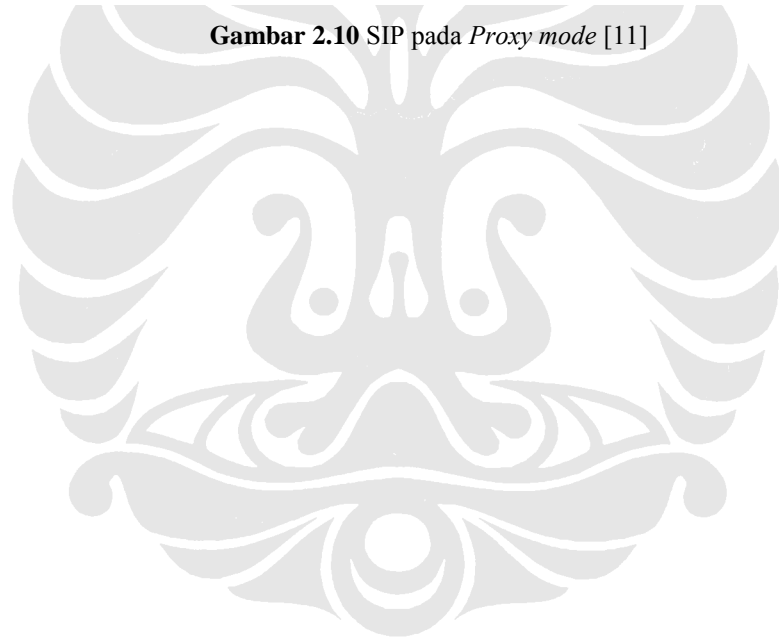
Gambar 2.9 SIP pada *Redirect mode* [11]

Redirect server yang memberikan respon terhadap *request* akan tetapi tidak meneruskannya. *Redirect server* mempunyai database sama halnya dengan *proxy server* yang berisi informasi penting untuk memproses terjadinya sesi komunikasi [11]. Gambar 2.9 menunjukkan proses pembentukan dan terminasi sesi SIP pada saat *redirect mode*. *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 [11]. Gambar 2.10 menunjukkan proses pembentukan dan terminasi sesi SIP pada saat *proxy mode*.



Gambar 2.10 SIP pada *Proxy mode* [11]

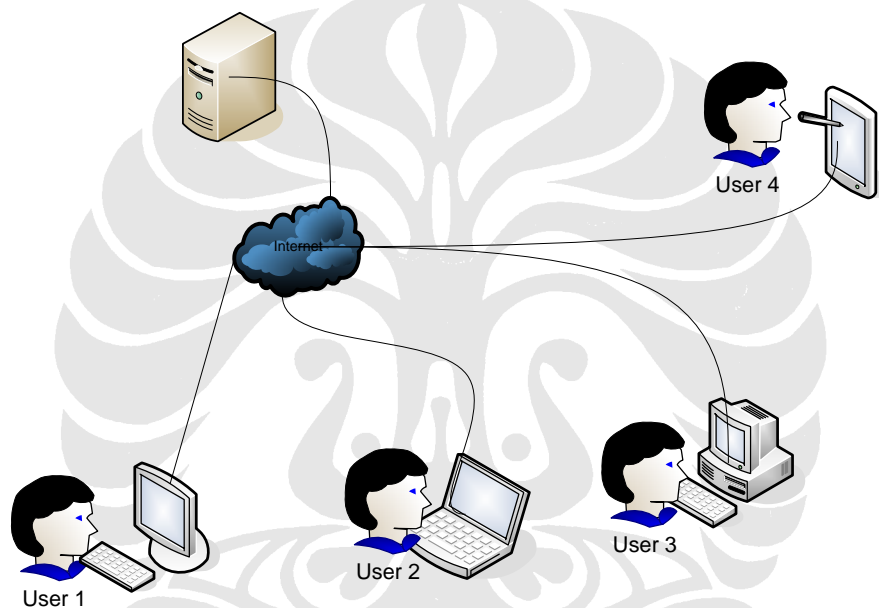


BAB 3

PERANCANGAN VIDEO *TELEPHONY* BERBASIS SIP

3.1. ARSITEKTUR VIDEO *TELEPHONY* BERBASIS SIP

Arsitektur perancangan video *telephony* memiliki basis SIP. SIP merupakan salah satu *signaling* VoIP. *Transport layer* aplikasi ini adalah UDP. Data video yang telah diubah menjadi kode *digital* akan dipecah menjadi paket-paket IP yang kemudian akan dikirimkan melalui jaringan internet menggunakan protokol RTP. Gambaran umum arsitekturnya dapat dilihat pada Gambar 3.1.



Gambar 3.1. Arsitektur umum video *telephony*

Tidak terdapat *gateway* pada arsitektur yang dibangun, sehingga setiap *user agent* harus dapat menerjemahkan paket-paket IP serta dapat men-*generate* dan menerjemahkan SIP *message* tersebut. Pengaturan *port-port* yang digunakan oleh *server* pada arsitektur video *telephony* adalah sebagai berikut:

- **8080:** merupakan port TCP yang berfungsi melakukan pengaturan transmisi melalui HTTP
- **5060:** merupakan port SIP yang berfungsi melakukan pengaturan transmisi SIP/SDP

- **12300-12400**: merupakan port UDP yang bertindak sebagai kanal transmisi RTP dan RTCP

3.1.1. Arsitektur Server

Operating system yang digunakan SIP server adalah Linux Ubuntu versi 7.04. Tipe server yang dibentuk adalah *Proxy mode server*. *Proxy server* bertindak sebagai perantara dalam pengiriman SIP message. SIP server tidak berperan sebagai transfer media. Walaupun demikian, *Proxy server* berperan sebagai media gateway pada sistem yang menggunakan NAT (*Network Address Transverse*), sehingga topologi jaringan berbasis SIP tidak dapat bersifat *peer-to-peer* untuk transfer media. Oleh karena itu, SIP server selalu berada pada posisi antara user dengan user lainnya ketika transfer media dalam sebuah sesi sedang berlangsung. Sehingga kapasitas *bandwidth* yang dibutuhkan lebih besar.

Proxy server memanfaatkan program asterisk versi 1.4.18 yang bersifat *open source* dan dapat di-install pada berbagai versi Linux. 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.

Selain asterisk, *Proxy server* memerlukan berbagai perangkat lunak yang terdiri dari freePBX, apache, MySQL, dan PHP agar dapat berfungsi dengan baik. MySQL berfungsi sebagai penyimpan database yang diperlukan, freePBX sebagai user interface, dan apache sebagai web server sehingga konfigurasi dapat dilakukan pada server secara web-based. IP publik yang digunakan pada server ini adalah 152.118.148.17.

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

Konfigurasi asterisk bertujuan agar asterisk dapat bertindak sebagai SIP server yang dapat mendukung komunikasi VoIP serta dapat mengirimkan video secara real-time. Konfigurasi asterisk dilakukan dengan mengubah isi file berekstensi .conf. Berikut ini adalah konfigurasi file-file berekstensi .conf asterisk agar dapat berfungsi sebagai SIP server dengan dukungan video.

- **asterisk.conf:** *File* ini berisi lokasi untuk konfigurasi *file-file* yang dibutuhkan oleh asterisk. Pengaturan default dari *asterisk.conf* dilakukan pada saat melakukan *command make samples* ketika instalasi *operating system*. Berikut ini adalah isi dari *file* ini.

```
[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 mengontrol *port* RTP yang digunakan oleh asterisk untuk mengirim dan menerima aliran RTP. RTP 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* akhir. Berikut ini adalah konfigurasi *rtp.conf*:

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

- **sip.conf:** *File* ini berisi semua konfigurasi yang mempengaruhi kerja SIP pada asterisk, seperti konfigurasi ekstensi *user*, *codec* yang digunakan,

dukungan video, dan pengaturan lainnya. Berikut ini adalah sebagian konfigurasi sip.conf:

```
[general]
Video support=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
```

Sedangkan konfigurasi sip_additional.conf adalah sebagai berikut:

```
[3000]
type=friend
secret=aaa0
record_out=Adhoc
record_in=Adhoc
qualify=yes
```

```
port=5060
pickupgroup=
nat=yes
mailbox=3000@device
host=dynamic
dtmfmode=rfc2833
dial=SIP/2000
context=from-internal
canreinvite=yes
Callgroup=
Callerid=device <2000>
allow=h263+
accountcode=
Call-limit=50

[3001]
type=friend
secret=aaal
record_out=Adhoc
record_in=Adhoc
qualify=yes
.....
```

Dan berikut ini adalah konfigurasi 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
```

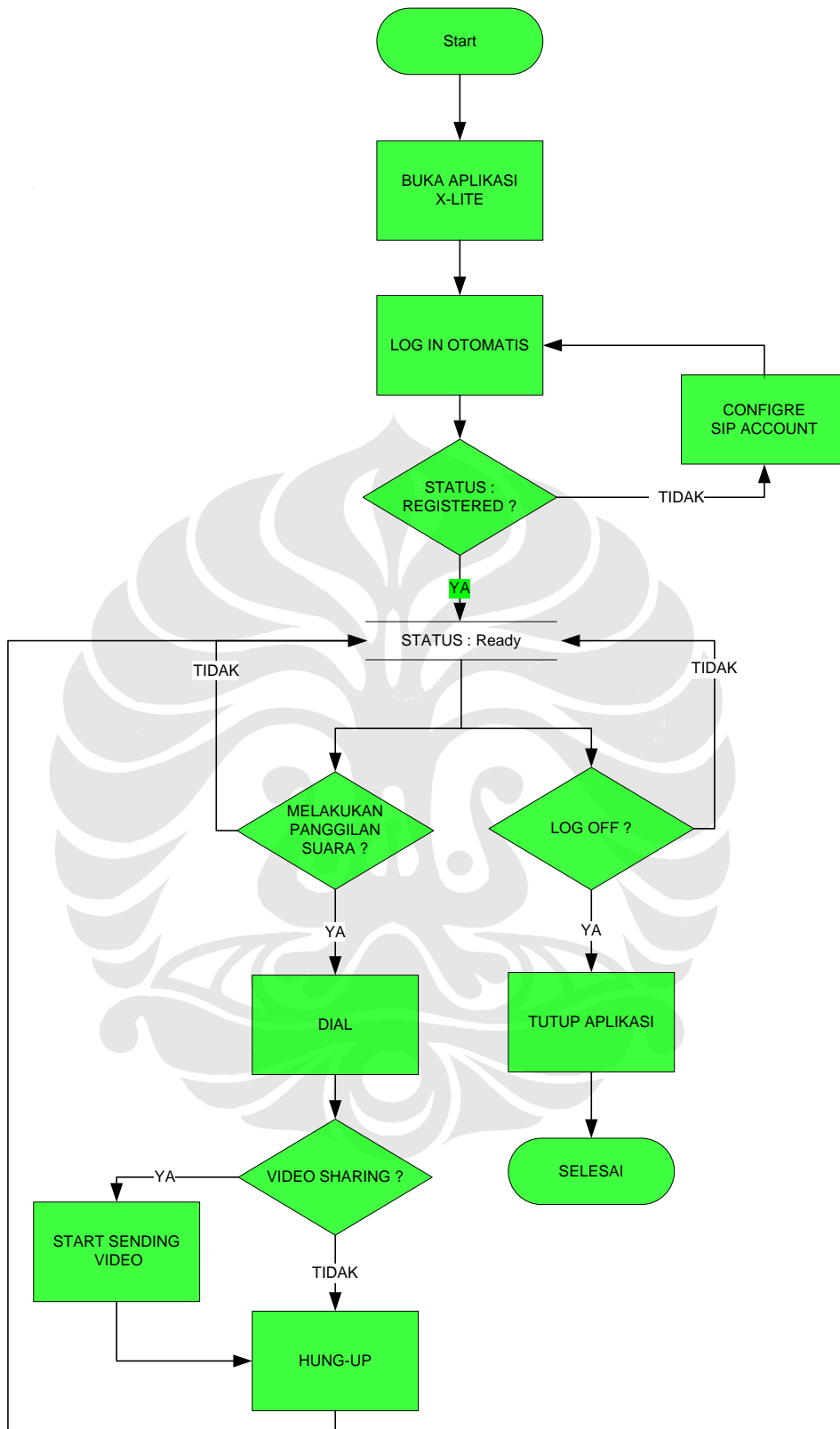
3.1.2. Arsitektur *Client*

Arsitektur *client* terdiri atas satu buah PC atau Laptop yang memiliki webcam dan telah terinstalasi perangkat lunak berupa X-lite 3.0 keluaran counterpath sebagai SIP-*phone*. Alasan pemilihan X-lite disebabkan X-lite merupakan free-software yang mendukung video *call over* IP karena terdapat *codec* H.263 dan H.263v2

Gambar 3.2 adalah tampilan X-lite 3.0 dan gambar 3.3 adalah diagram alir penggunaan X-Lite 3.0. yang digunakan dalam perancangan arsitektur video *telephony* ini



Gambar 3.2 Tampilan X-Lite 3.0



Gambar 3.3 Diagram alir penggunaan X-Lite

3.2. VIDEO CODEC

3.2.1. H.263

H.263 adalah sebuah video *codec* yang distandardisasi oleh ITU-T pada proyek akhir 1995/1996 dan digunakan sebagai format Video *conferencing* dengan kompresi *bit-rate* rendah. *Bit rate* komunikasi maksimum yang direkomendasikan untuk *codec* ini kurang dari 64 Kbps [18]. Alasan penggunaan *codec* ini sebagai rekomendasi video *telephony* adalah *codec* ini tidak memerlukan *bandwidth* yang tinggi. *Codec* ini direkomendasikan untuk penggunaan dengan *bit-rate* rendah dan *low motion*. Setengah *pixel precision* dari *codec* ini digunakan untuk kompensasi gerak. Hal ini sesuai dengan kebutuhan Video *telephony* yang juga tidak memerlukan penggunaan *bit-rate* yang tinggi.

Codec ini dapat dikonfigurasi untuk menurunkan *data-rate* atau memperbaiki *error recovery*. Format video yang didukung *codec* ini adalah SQCIF, QCIF, CIF, 4CIF dan 16CIF. Resolusi SQCIF kira-kira setengah resolusi QCIF, resolusi 4CIF adalah empat kali resolusi CIF, dan resolusi 16CIF adalah 16 kali resolusi CIF [18]. Karena H.263 mampu mendukung format 4CIF dan 16CIF, maka *codec* ini melebihi kualitas standar video *coding* dengan *bit-rate* yang lebih tinggi seperti MPEG 1 dan 2 [17].

3.2.2. H.263v2

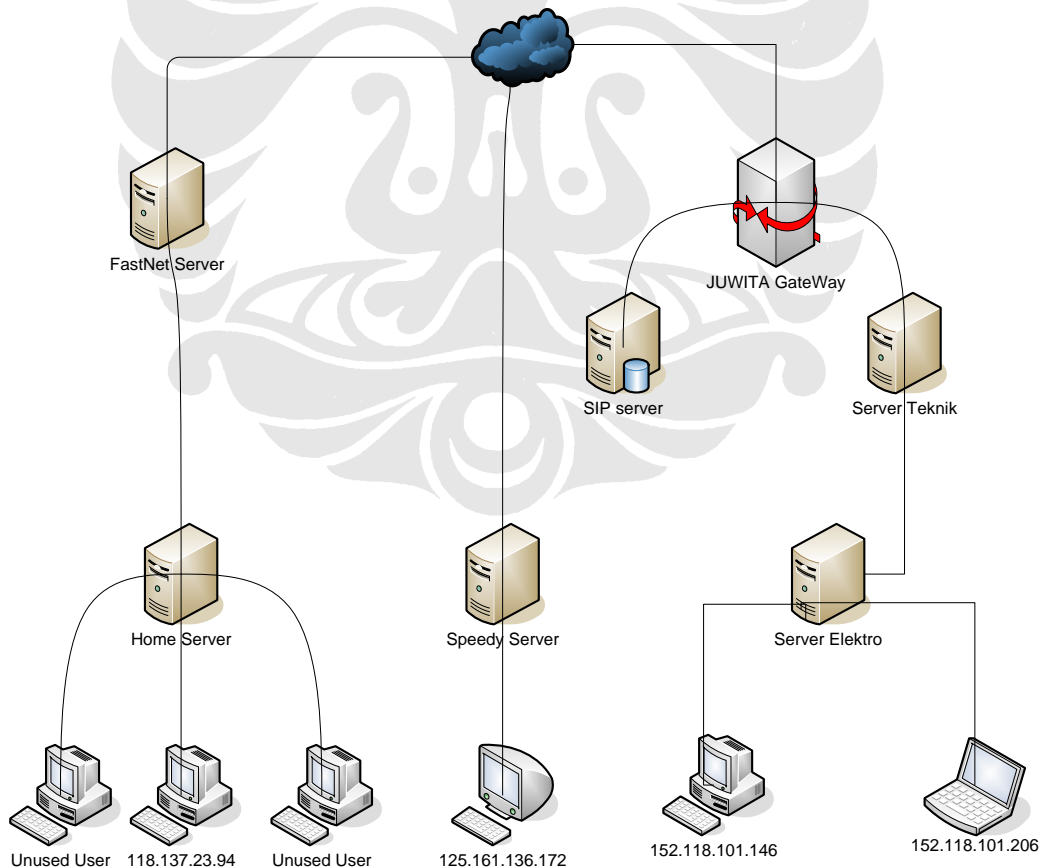
H.263v2 atau yang biasa disebut H.263+ atau H.263-1998 ini merupakan pengembangan dari H.263. beberapa *feature* ditambahkan dengan tujuan meningkatkan unjuk kerja kompresi, meningkatkan kemampuan *error resilience*, menawarkan pilihan skalabilitas *bitstream*, untuk membuat *bitstream* lebih berpotensi digunakan dengan sebuah protokol *transport* dasar seperti RTP, lebih fleksibel dalam paketisasi, dan untuk meminimalisasi *delay* Video [17].

BAB 4

ANALISIS DAN UJI COBA

4.1. SKENARIO UJI COBA

Uji coba unjuk kerja *uplink* video *telephony* dilakukan dengan menjalankan perangkat lunak X-lite versi 3.0 sebagai SIP-*phone* dan memanfaatkan kondisi jaringan internet UI serta berbagai jaringan *provider* internet seperti fastnet dan speedy yang merupakan produk PT TELKOM. SIP *Server* berkedudukan di PPSI UI Depok, *Main client* berada pada jaringan UI sedangkan *second client* menggunakan jaringan speedy yang berada di Mercator dan jaringan fastnet yang berada di Pamulang. Selain itu, uji coba juga dilakukan dengan memanfaatkan jaringan internal UI sebagai *second client*. Gambar 4.1 menunjukkan topologi jaringan uji coba.



Gambar 4.1 Topologi Jaringan Uji coba

Adapun parameter pengujian yang akan dianalisis pada uji coba ini adalah dua parameter QoS yaitu *jitter* yang terjadi dan *bandwidth* yang digunakan untuk setiap pengiriman paket dari *client* ke *server* (*uplink*) dengan variasi *codec* video H.263 dan H.263v2. Selain itu, uji coba juga dilakukan dengan variasi ISP antara LAN UI, fastnet, dan speedy. *Bandwidth* yang disediakan UI adalah 100 Mbps, fastnet menyediakan 746 Kbps, dan speedy menyediakan 144.10 Kbps. Hasil uji coba setiap percobaan dideskripsikan melalui grafik antara nomor paket dan *jitter* atau *bandwidth*. Nomor paket merupakan penomoran setiap paket yang menggambarkan urutan paket.

4.2. UJI COBA JARINGAN INTERNAL UI

Uji coba ini melibatkan dua *client* yang memiliki IP *address* dan SIP URI masing-masing adalah 152.118.101.206 (sip:3000@152.118.148.17) dan 152.118.101.146 (sip:3002@152.118.148.17).

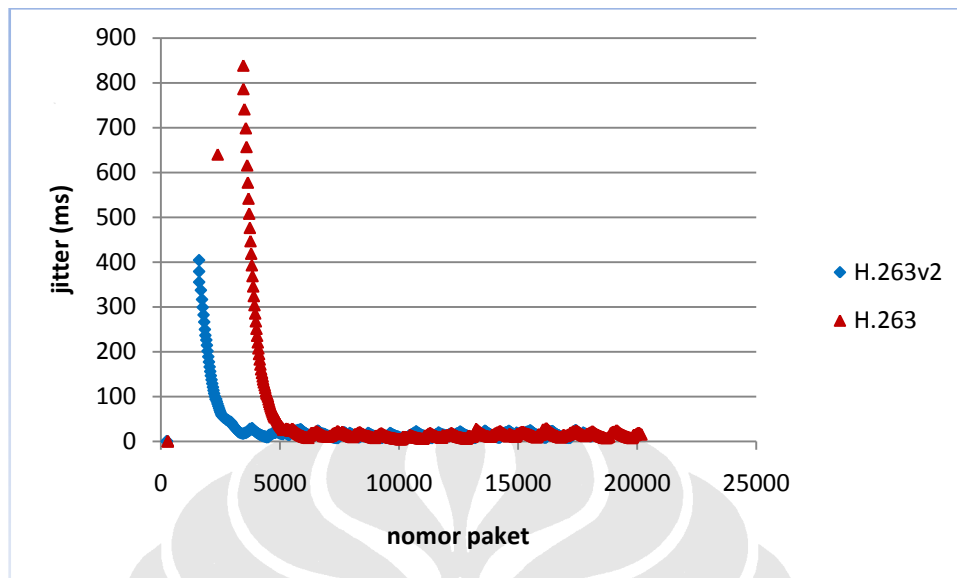
4.2.1. Uplink Client 152.118.101.146 ke Server

a. Paket vs Jitter

Hasil uji coba menunjukkan bahwa *jitter* rata-rata ketika menggunakan *codec* H.263 adalah 38.35 ms sedangkan *jitter* maksimum terjadi ketika pengiriman paket no. 3457 dan besarnya adalah 838.72 ms. Sedangkan *jitter* rata-rata ketika menggunakan *codec* H.263v2 adalah 25.56 ms dan *jitter* maksimum terjadi ketika pengiriman paket no.1604 dan besarnya adalah 404.01 ms.

Grafik pada gambar 4.2 menunjukkan bahwa *jitter* rata-rata *uplink* ketika menggunakan *codec* H.263v2 lebih kecil daripada *jitter* rata-rata ketika menggunakan *codec* H.263. Hal ini disebabkan H.263v2 memiliki skalabilitas *bitstream* dan memungkinkan paketisasi yang lebih baik dibandingkan H.263. Pada pengiriman beberapa paket di awal, *jitter* yang terjadi cukup besar dan terdapat beberapa paket yang tidak terkirimkan pada penggunaan kedua *codec*. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik. Penggunaan IP publik menyebabkan paket harus menunggu ketersediaan *bandwidth* yang memungkinkan paket dapat terkirimkan. *Jitter* yang terjadi pada paket yang

pertama kali dikirimkan bernilai nol. Hal ini disebabkan paket yang pertama kali dikirim merupakan paket acuan penilaian *jitter* paket setelahnya.



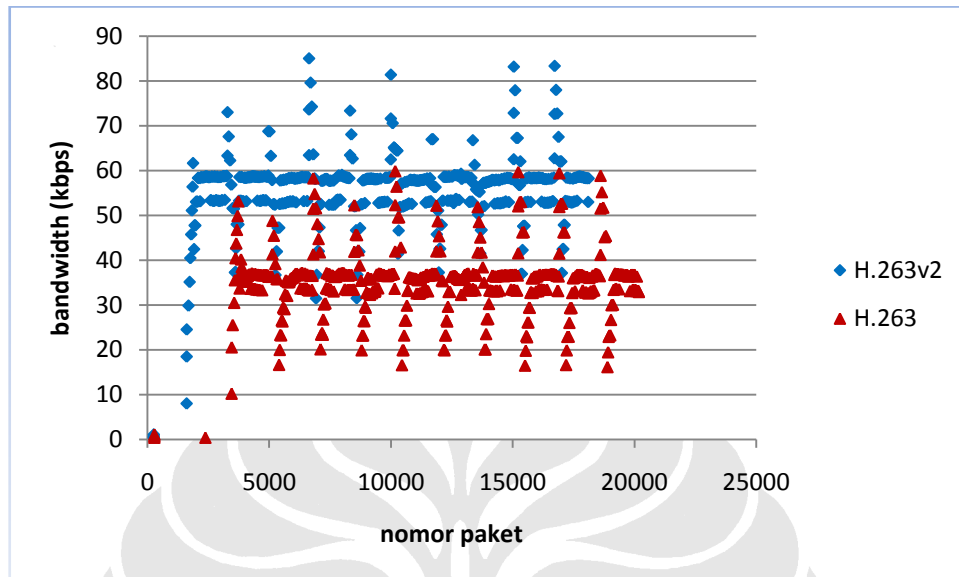
Gambar 4.2 Grafik nomor paket vs *jitter* Uplink client 152.118.101.146 ke server

Terdapat plot grafik yang terputus pada gambar 4.2. Hal ini menunjukkan adanya paket yang tidak terkirimkan. Fenomena ini disebabkan *bandwidth* yang dapat terpakai cukup rendah. Kondisi *bandwidth* yang rendah pada awal pengiriman paket ini dapat dilihat pada gambar 4.3.

b. Paket vs *Bandwidth*

Hasil uji coba menunjukkan bahwa *bandwidth* maksimum yang dipakai ketika menggunakan *codec* H.263 adalah 59.87 Kbps dan *bandwidth* rata-ratanya adalah 35.23 Kbps. Sedangkan *bandwidth* maksimum ketika menggunakan *codec* H.263v2 adalah 84.98 Kbps dan *bandwidth* rata-ratanya adalah 55.72 Kbps. *Bandwidth uplink* yang terpakai ketika menggunakan *codec* H.263v2 lebih besar daripada *bandwidth* terpakai ketika menggunakan *codec* H.263. Hal ini disebabkan H263v2 memiliki *range bit rate* yang lebih besar dibandingkan H.263. Hal ini menyebabkan pemakaian *bandwidth* H.263v2 lebih besar jika dibandingkan H.263. Gambar 4.3 menunjukkan grafik yang mendeskripsikan *bandwidth* yang digunakan selama pengiriman paket. Pada pengiriman beberapa paket di awal, *bandwidth* yang dapat digunakan relatif kecil. Kecilnya *bandwidth* menyebabkan *jitter* bernilai besar pada pengiriman beberapa paket di awal dan

paket. Fenomena ini dapat dilihat dengan membandingkan gambar 4.2 dan gambar 4.3. *Bandwidth* yang terpakai selama pengiriman paket terlihat kurang stabil. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik.



Gambar 4.3 Grafik nomor paket vs *bandwidth* Uplink client 152.118.101.146 ke server

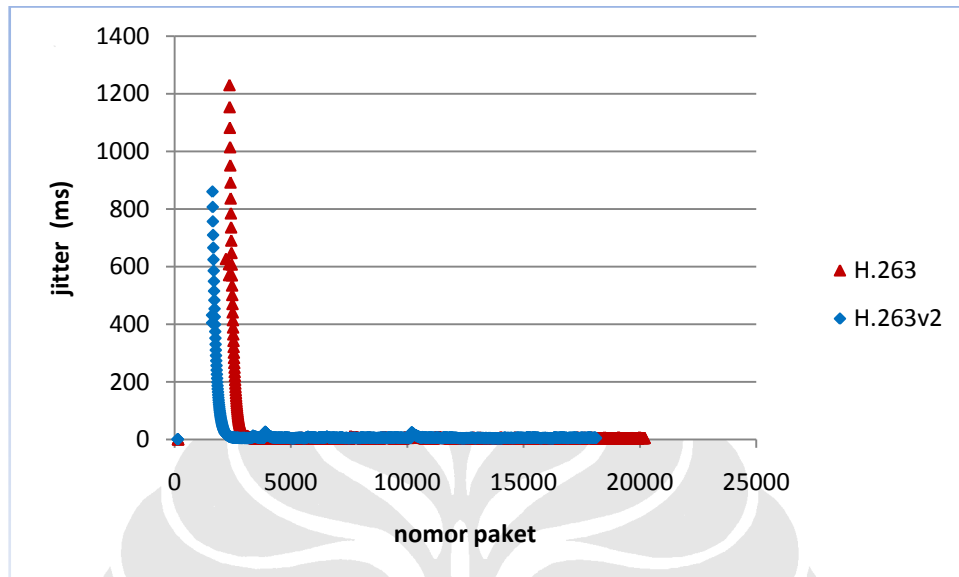
4.2.2. Uplink Client 152.118.101.206 ke Server

a. Paket vs Jitter

Hasil uji coba menunjukkan bahwa *jitter* rata-rata ketika menggunakan *codec* H.263 adalah 16.11 ms sedangkan *jitter* maksimum terjadi ketika pengiriman paket no. 2358 dan besarnya adalah 1229.66 ms. Sedangkan *jitter* rata-rata ketika menggunakan *codec* H.263v2 adalah 13.5 ms dan *jitter* maksimum terjadi ketika pengiriman paket no.1632 dan besarnya adalah 859.67 ms.

Grafik pada gambar 4.4 menunjukkan bahwa *jitter uplink* yang terjadi ketika menggunakan *codec* H.263v2 lebih kecil daripada *jitter* yang terjadi ketika menggunakan *codec* H.263. Hal ini disebabkan H263v2 memiliki skalabilitas *bitstream* dan memungkinkan paketisasi yang lebih baik dibandingkan H.263. Pada pengiriman beberapa paket di awal, *jitter* yang terjadi cukup besar dan terdapat beberapa paket yang tidak terkirimkan pada penggunaan kedua *codec*. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik. Penggunaan IP publik menyebabkan paket harus menunggu ketersediaan *bandwidth* yang memungkinkan paket dapat terkirimkan. *Jitter* yang dihasilkan dari paket yang

pertama kali dikirimkan bernilai nol. Hal ini disebabkan paket yang pertama kali dikirim merupakan paket acuan penilaian *jitter* paket setelahnya.

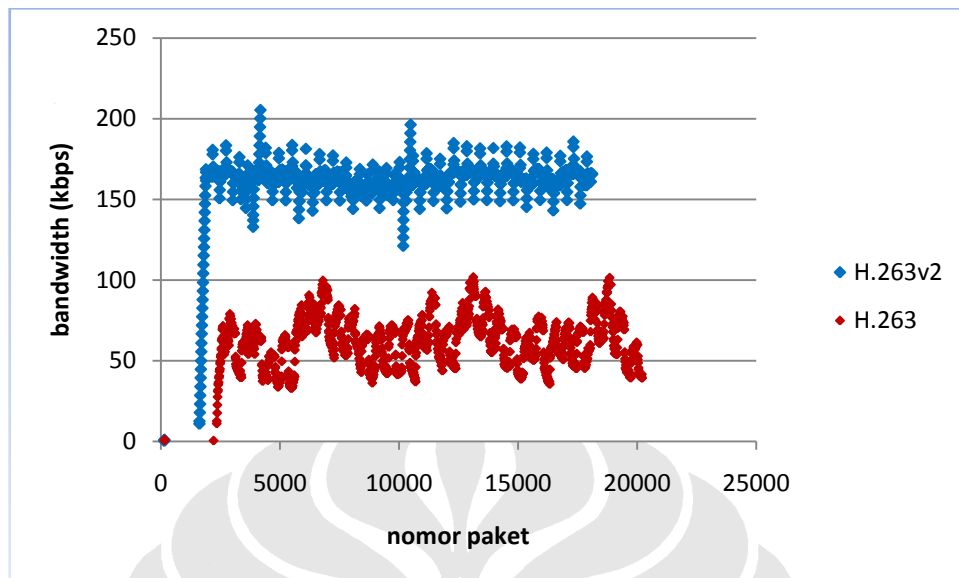


Gambar 4.4 Grafik nomor paket vs *Jitter Uplink client 152.118.101.206 ke server*

b. Paket vs *Bandwidth*

Hasil uji coba menunjukkan bahwa *bandwidth* maksimum yang dipakai ketika menggunakan *codec* H.263 adalah 101.85 Kbps dan *bandwidth* rata-rata adalah 62.22 Kbps. Sedangkan *bandwidth* maksimum ketika menggunakan *codec* H.263v2 adalah 205.3 Kbps dan *bandwidth* rata-ratanya adalah 161.55 Kbps. *Bandwidth uplink codec* H.263v2 lebih besar daripada *bandwidth uplink codec* H.263. Hal ini disebabkan H263v2 memiliki *range bit rate* yang lebih besar dibandingkan H.263.

Pemakaian *bandwidth* H.263v2 lebih besar jika dibandingkan H.263. Gambar 4.5 menunjukkan grafik yang mendeskripsikan *bandwidth* yang digunakan selama pengiriman paket. Pada pengiriman beberapa paket di awal, *bandwidth* yang dapat digunakan relatif kecil. Kecilnya *bandwidth* menyebabkan besarnya rata-rata *jitter* yang terjadi pada pengiriman paket. Fenomena ini dapat dilihat pada gambar 4.4 dan gambar 4.5. *Bandwidth* yang terpakai selama pengiriman paket terlihat kurang stabil. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik.



Gambar 4.5 Grafik nomor paket vs *bandwidth Uplink client* 152.118.101.206 ke *server*

4.2.3. Perbandingan Unjuk Kerja Kedua *Client*

Tabel 4.1 Perbandingan *Jitter uplink* jaringan internal UI

IP <i>Client</i>	<i>Mean Jitter</i>	<i>Mean Jitter</i>	<i>Max. Jitter</i>	<i>Max. Jitter</i>
	H.263	H.263v2	H.263	H.263v2
152.118.101.146	38.35 ms	25.56 ms	838.72 ms	404.01 ms
152.118.101.206	16.11 ms	13.5 ms	1229.66 ms	859.67 ms

Tabel 4.2 Perbandingan *bandwidth uplink* jaringan internal UI

IP <i>Client</i>	<i>Mean Bandwidth</i>	<i>Mean Bandwidth</i>	<i>Max. Bandwidth</i>	<i>Max. Bandwidth</i>
	H.263	H.263v2	H.263	H.263v2
152.118.101.146	35.23 Kbps	55.72 Kbps	59.87 Kbps	84.98 Kbps
152.118.101.206	62.22 Kbps	161.55 Kbps	101.85 Kbps	205.3 Kbps

Tabel 4.1 dan 4.2 menunjukkan perbandingan unjuk kerja *client* dengan IP *address* 152.118.101.146 dan *client* 152.118.101.206. *Client* 152.118.101.146 menghasilkan *jitter* rata-rata lebih besar dan *jitter* maksimum lebih kecil daripada *client* 152.118.101.206. Walaupun kedua *client* berada pada satu LAN, tetapi

bandwidth dan *jitter* yang terjadi tidaklah sama karena jaringan menggunakan IP publik yang tidak stabil.

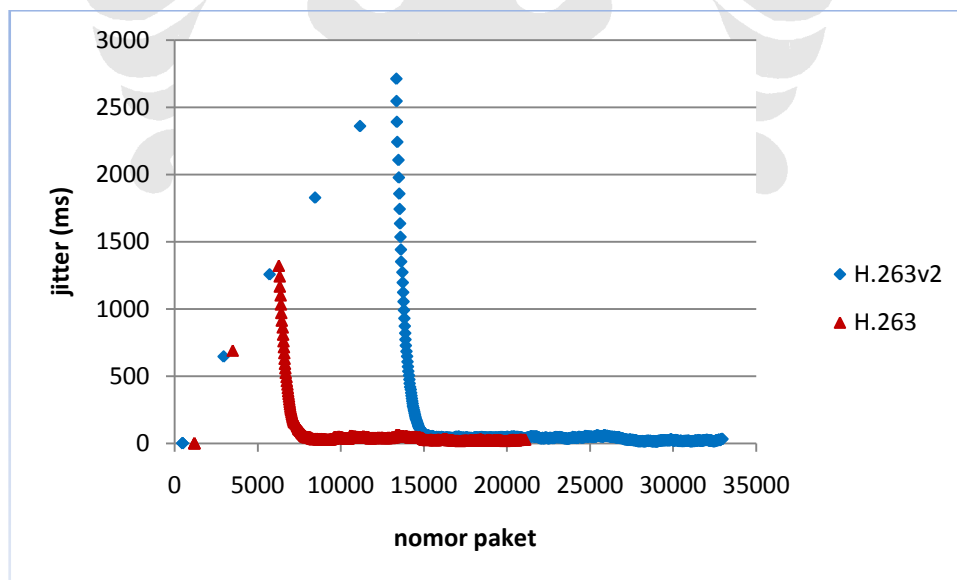
4.3. UJI COBA ANTARA LAN UI DAN FASTNET

Uji coba ini melibatkan dua *client* yang memiliki IP *address* dan SIP URI masing-masing adalah 152.118.101.206 (sip:3000@152.118.148.17) yang merupakan *client* yang berada pada jaringan UI dan 118.137.23.94 (sip:3001@152.118.148.17) yang merupakan *client* yang menggunakan fastnet.

4.3.1. Uplink Client 118.137.23.94 ke Server

a. Paket vs Jitter

Hasil uji coba menunjukkan bahwa *jitter* rata-rata ketika menggunakan *codec* H.263 adalah 74.82 ms sedangkan *jitter* maksimum terjadi ketika pengiriman paket no. 6268 dan besarnya adalah 1320.85 ms. Sedangkan *jitter* rata-rata ketika menggunakan *codec* H.263v2 adalah 90.99 ms dan *jitter* maksimum terjadi ketika pengiriman paket no.13352 dan besarnya adalah 2711.05 ms. Gambar 4.6 menunjukkan grafik yang mendeskripsikan data hasil uji coba.



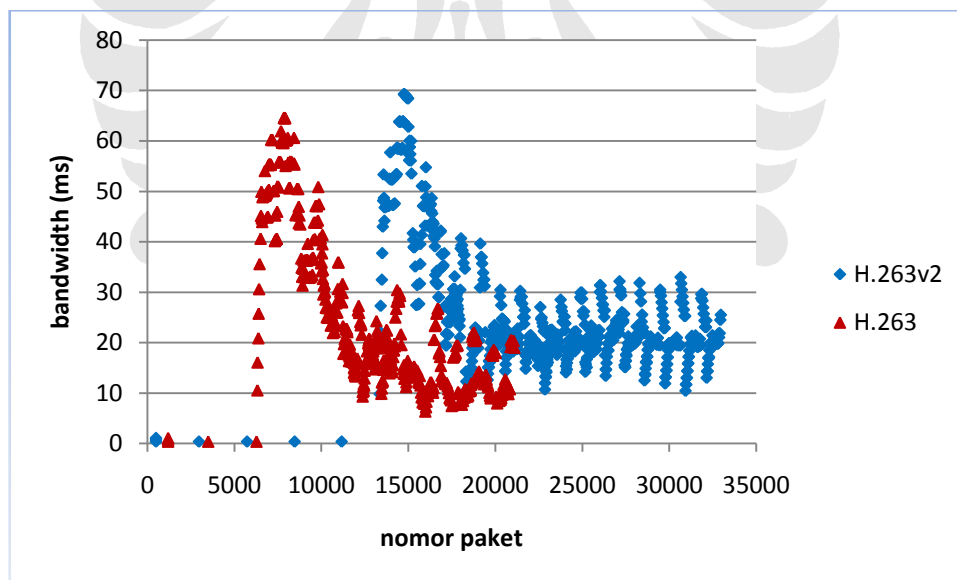
Gambar 4.6 Grafik nomor paket vs *Jitter* Uplink client 118.137.23.94 ke server

Pada pengiriman beberapa paket di awal, *jitter* yang terjadi cukup besar dan terdapat beberapa paket yang tidak terkirimkan pada penggunaan kedua

codec. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik. Penggunaan IP publik menyebabkan paket harus menunggu ketersediaan *bandwidth* yang memungkinkan paket dapat terkirimkan. *Jitter* yang dihasilkan dari paket yang pertama kali dikirimkan bernilai nol. Hal ini disebabkan paket yang pertama kali dikirim merupakan paket acuan penilaian *jitter* paket setelahnya. Grafik yang putus-putus memperlihatkan terjadinya interval yang besar diantara paket-paket yang dikirim dan menunjukkan adanya paket yang tidak terkirimkan di fase awal pengiriman.

b. Paket vs *Bandwidth*

Hasil uji coba menunjukkan bahwa *bandwidth* maksimum yang terpakai ketika menggunakan *codec* H.263 adalah 64.62 Kbps dan *bandwidth* rata-ratanya adalah 25 Kbps. Sedangkan *bandwidth* maksimum ketika menggunakan *codec* H.263v2 adalah 69.28 Kbps dan *bandwidth* rata-ratanya adalah 25.30 Kbps. Gambar 4.7 menunjukkan grafik *bandwidth* yang dipakai untuk pengiriman setiap paket.



Gambar 4.7 Grafik nomor paket vs *bandwidth* Uplink client 118.137.23.94 ke server

Bandwidth uplink yang terpakai ketika menggunakan *codec* H.263v2 lebih besar daripada *bandwidth* yang terpakai ketika menggunakan *codec* H.263. Hal ini disebabkan H263v2 memiliki *range bit rate* yang lebih besar

dibandingkan H.263. Hal ini menyebabkan pemakaian *bandwidth* H.263v2 lebih besar jika dibandingkan H.263. Tidak ada perbedaan besar *bandwidth uplink* yang signifikan untuk kedua *codec* karena H.263v2 lebih adaptif terhadap kondisi *bandwidth* yang disediakan ISP. Fastnet menyediakan *bandwidth* yang lebih kecil daripada jaringan UI sehingga penggunaan *bandwidth uplink* pada jaringan fastnet oleh H.263v2 sebagai *codec* lebih kecil jika dibandingkan dengan jaringan UI. Pada pengiriman beberapa paket di awal, *bandwidth* yang dapat digunakan relatif kecil. Perbandingan antara grafik pada gambar 4.6 dan 4.7 menunjukkan bahwa kecilnya *bandwidth* menyebabkan besarnya rata-rata *jitter* yang terjadi pada pengiriman paket atau bahkan paket mengalami *wrong sequence*. *Wrong sequence* adalah fenomena pengiriman paket yang tidak menempuh jalan yang telah ditentukan. Kecilnya *bandwidth* juga menyebabkan terjadinya *packet loss*. *Bandwidth* yang terpakai selama pengiriman paket terlihat kurang stabil. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik.

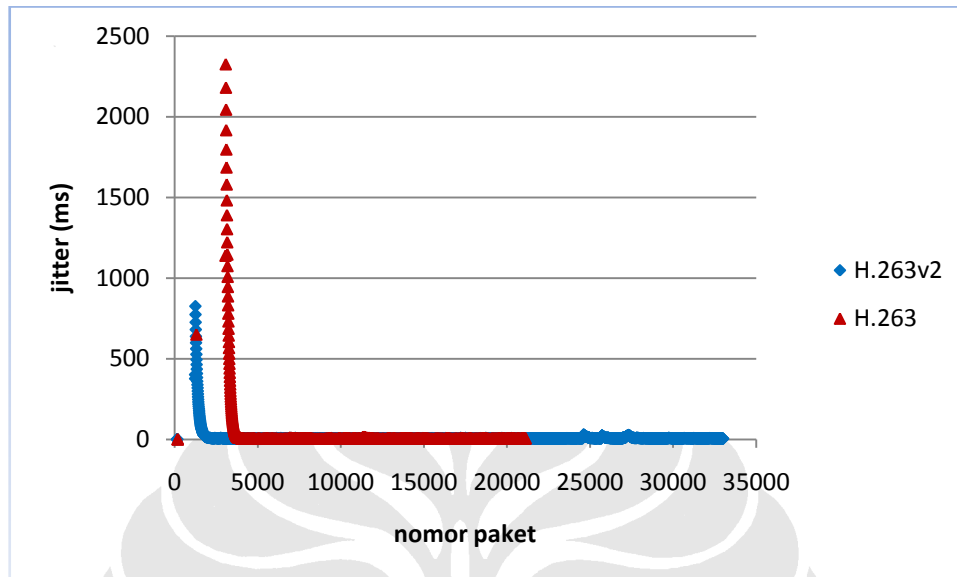
4.3.2. Uplink Client 152.118.101.206 ke Server

a. Paket vs Jitter

Hasil uji coba menunjukkan bahwa *jitter* rata-rata ketika menggunakan *codec* H.263 adalah 25.21 ms sedangkan *jitter* maksimum terjadi ketika pengiriman paket no. 3083 dan besarnya adalah 2325.49 ms. Sedangkan *jitter* rata-rata ketika menggunakan *codec* H.263v2 adalah 9.29 ms dan *jitter* maksimum terjadi ketika pengiriman paket no.1263 dan besarnya adalah 824.92 ms. Gambar 4.8 menunjukkan grafik yang mendeskripsikan *jitter* yang terjadi pada pengiriman setiap paket.

Pada pengiriman beberapa paket di awal, *jitter* yang terjadi cukup besar dan terdapat beberapa paket yang tidak terkirimkan pada penggunaan kedua *codec*. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik. Penggunaan IP publik menyebabkan paket harus menunggu ketersediaan *bandwidth* yang memungkinkan paket dapat terkirimkan. *Jitter* yang dihasilkan dari paket yang pertama kali dikirimkan bernilai nol. Hal ini disebabkan paket yang pertama kali dikirim merupakan paket acuan penilaian *jitter* paket setelahnya. Grafik yang putus-putus memperlihatkan terjadinya interval yang besar antarpaket yang

dikirim atau menunjukkan adanya paket yang tidak terkirimkan di fase awal pengiriman paket.



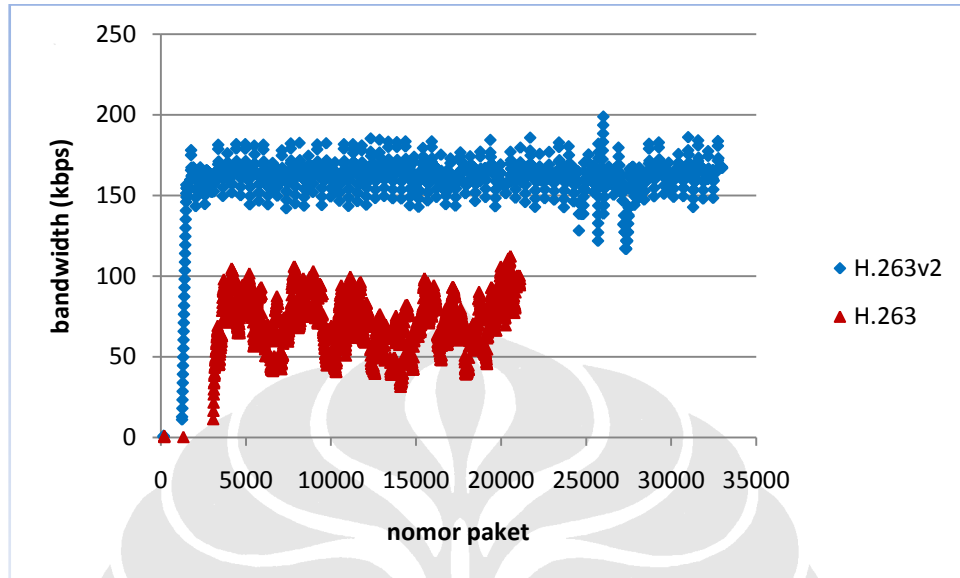
Gambar 4.8 Grafik nomor paket vs Jitter Uplink client 152.118.101.206 ke server

b. Paket vs Bandwidth

Hasil uji coba menunjukkan bahwa *bandwidth* maksimum yang dipakai ketika menggunakan *codec* H.263 adalah 112.3 Kbps dan *bandwidth* rata-ratanya adalah 72.02 Kbps. Sedangkan *bandwidth* maksimum ketika menggunakan *codec* H.263v2 adalah 198.82 Kbps dan *bandwidth* rata-ratanya adalah 161.25 Kbps. Gambar 4.9 menunjukkan grafik kondisi *bandwidth* yang dipakai untuk setiap paket.

Bandwidth uplink yang terpakai ketika menggunakan *codec* H.263v2 lebih besar daripada *bandwidth* yang terpakai ketika menggunakan *codec* H.263. Hal ini disebabkan H263v2 memiliki *range bit rate* yang lebih besar dibandingkan H.263. Hal ini menyebabkan pemakaian *bandwidth* H.263v2 lebih besar jika dibandingkan H.263. Pada pengiriman beberapa paket di awal, *bandwidth* yang dapat digunakan relatif kecil. Kecilnya *bandwidth* menyebabkan besarnya rata-rata *jitter* yang terjadi pada pengiriman paket atau bahkan paket mengalami *wrong sequence*. *Wrong sequence* adalah fenomena pengiriman paket yang tidak menempuh jalan yang telah ditentukan. Kecilnya *bandwidth* juga menyebabkan terjadinya *packet loss*. Fenomena ini dapat dilihat pada gambar 4.8. *Bandwidth*

yang terpakai selama pengiriman paket terlihat kurang stabil. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik.



Gambar 4.9 Grafik nomor paket vs *bandwidth uplink client* 152.118.101.206 ke *server*

4.3.3. Perbandingan Unjuk Kerja Kedua *Client*

Tabel 4.3 Perbandingan *jitter uplink* jaringan internal UI dan fastnet

IP Client	Mean Jitter	Mean Jitter	Max. Jitter	Max. Jitter
	H.263	H.263v2	H.263	H.263v2
152.118.101.206	25.21 ms	9.29 ms	2325.49 ms	842.92 ms
118.137.23.94	74.82 ms	90.99 ms	1320.85 ms	2711.05 ms

Tabel 4.4 Perbandingan *bandwidth uplink* jaringan internal UI dan fastnet

IP Client	Mean	Mean	Max.	Max.
	Bandwidth	Bandwidth	Bandwidth	Bandwidth
	H.263	H.263v2	H.263	H.263v2
152.118.101.206	72.02 Kbps	161.25 Kbps	112.3 Kbps	198.82 Kbps
118.137.23.94	25 Kbps	25.30 Kbps	64.62 Kbps	69.28 Kbps

Tabel 4.3 dan 4.4 menunjukkan perbandingan unjuk kerja *client* dengan IP *address* 152.118.101.206 dan 118.137.23.94. Walaupun *jitter* maksimum yang terjadi pada fase awal pengiriman paket menggunakan H.263 *uplink client* 152.118.101.206 lebih besar daripada *client* 118.137.23.94 *jitter* rata-rata *uplink*

client 152.118.101.206 lebih kecil daripada *client* 118.137.23.94 untuk semua konfigurasi. Pamakaian *bandwidth uplink client* 152.118.101.206 lebih besar daripada *client* 118.137.23.94. *Bandwidth* yang disediakan jaringan UI adalah 100 Mbps dan *bandwidth* yang disediakan fastnet adalah 746 Kbps. Semakin besar *bandwidth* yang disediakan *provider*, maka semakin besar pula *bandwidth* yang dapat digunakan untuk komunikasi melalui video *telephony*. *Available bandwidth* fastnet yang lebih kecil dibandingkan *available bandwidth* jaringan UI juga menyebabkan tidak ada perbedaan yang signifikan nilai maksimum dan rata-rata *bandwidth* yang terpakai pada penggunaan H.263 dan H.263v2. Oleh karena itu, kualitas *uplink client* 152.118.101.206 lebih baik daripada *client* 118.137.23.94.

4.4. UJI COBA ANTARA LAN UI DAN SPEEDY

Uji coba ini melibatkan dua *client* yang memiliki IP *address* dan SIP URI masing-masing adalah 152.118.101.146 (sip:3000@152.118.148.17) yang merupakan *client* yang berada pada jaringan UI dan 125.161.136.172 (sip:3001@152.118.148.17) yang merupakan *client* yang menggunakan speedy.

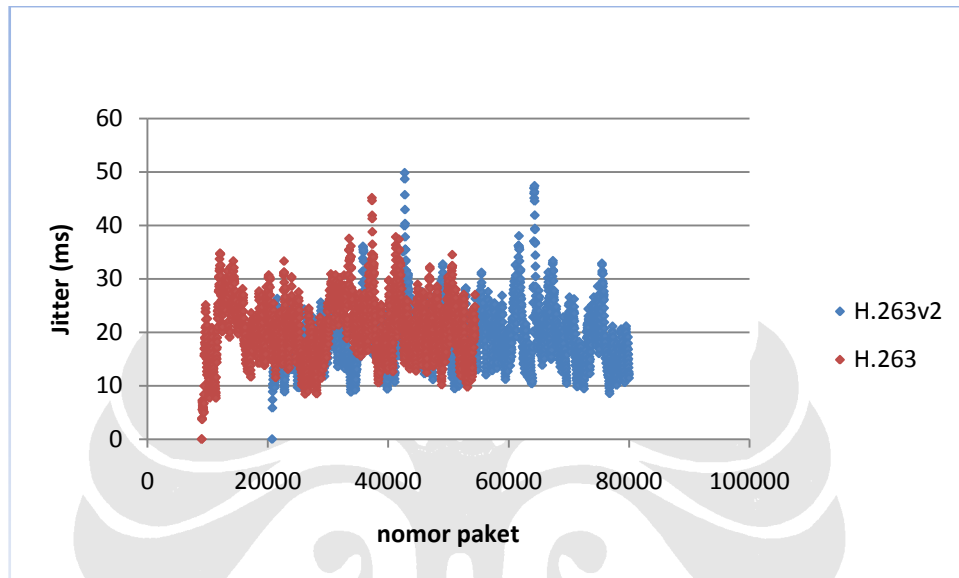
4.4.1. Uplink Client 152.118.101.146 ke Server

a. Paket vs Jitter

Hasil uji coba menunjukkan bahwa *jitter* rata-rata ketika menggunakan *codec* H.263 adalah 20.64 ms dan *jitter* maksimum terjadi ketika pengiriman paket no. 37266 dan besarnya adalah 45.18 ms. Sedangkan *jitter* rata-rata ketika menggunakan *codec* H.263v2 adalah 19.22 ms dan *jitter* maksimum terjadi ketika pengiriman paket no.42729 dan besarnya adalah 49.82 ms. Gambar 4.11 menunjukkan grafik yang mendeskripsikan *jitter* yang terjadi pada pengiriman setiap paket.

Jitter yang dihasilkan dari paket yang pertama kali dikirimkan bernilai nol. Hal ini disebabkan paket yang pertama kali dikirim merupakan paket acuan penilaian *jitter* paket setelahnya. Grafik yang putus-putus memperlihatkan terjadinya interval diantara paket-paket yang dikirim dan menunjukkan adanya paket yang tidak terkirimkan. *Jitter* rata-rata yang terjadi antara penggunaan

H.263 dan H.263v2 tidak menunjukkan perbedaan yang signifikan. Hal ini disebabkan *bandwidth* yang disediakan speedy hanya 144.10 Kbps sehingga kedua *codec* ini menyesuaikan dengan *bandwidth* yang disediakan ini. Grafik pada gambar 4.11 juga memperlihatkan tidak banyak paket yang tidak terkirim. Hal ini diperlihatkan dengan sedikitnya plot grafik yang putus-putus.



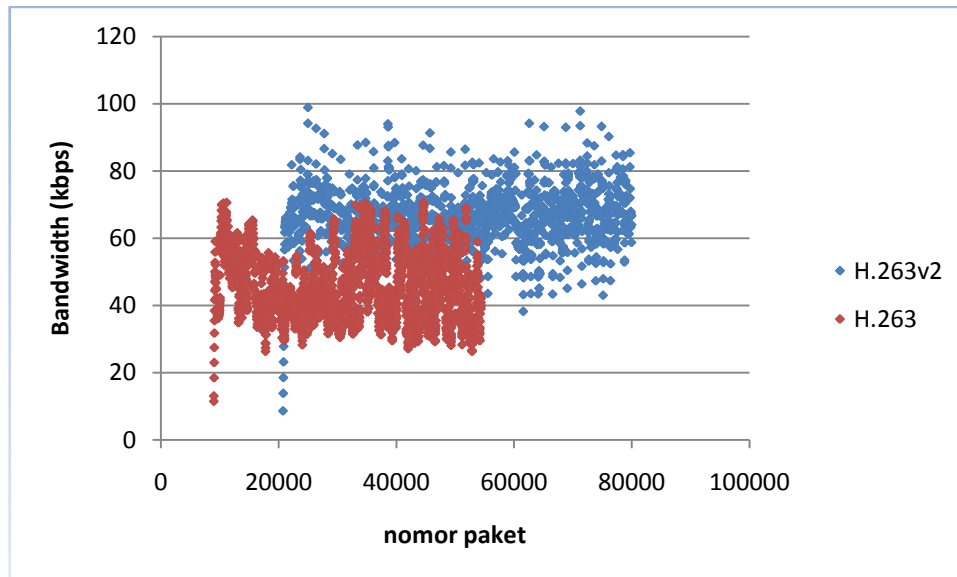
Gambar 4.10 Grafik nomor paket vs Jitter Uplink client 152.118.101.146 ke server

b. Paket vs *Bandwidth*

Hasil uji coba menunjukkan bahwa *bandwidth* maksimum yang terpakai ketika menggunakan *codec* H.263 adalah 70.55 Kbps dan *bandwidth* rata-ratanya adalah 45.18 Kbps. Sedangkan *bandwidth* maksimum ketika menggunakan *codec* H.263v2 adalah 98.88 Kbps dan *bandwidth* rata-ratanya adalah 66.65 Kbps. Gambar 4.12 menunjukkan grafik kondisi *bandwidth* yang terpakai untuk pengiriman setiap paket.

Bandwidth uplink yang terpakai ketika menggunakan *codec* H.263v2 lebih besar daripada *Bandwidth* yang terpakai ketika menggunakan *codec* H.263. Hal ini disebabkan H263v2 memiliki *range bit rate* yang lebih besar dibandingkan H.263. Hal ini menyebabkan pemakaian *bandwidth* H.263v2 lebih besar jika dibandingkan H.263. Pada pengiriman beberapa paket di awal, *bandwidth* yang dapat digunakan relatif kecil. Kecilnya *bandwidth* menyebabkan besarnya rata-rata *jitter* yang terjadi pada pengiriman paket atau bahkan paket

mengalami *wrong sequence*. *Wrong sequence* adalah fenomena pengiriman paket yang tidak menempuh jalan yang telah ditentukan. Kecilnya *bandwidth* juga menyebabkan terjadinya *packet loss*. Fenomena ini dapat dilihat pada gambar 4.11.



Gambar 4.11 Grafik nomor paket vs *bandwidth Uplink client 152.118.101.146 ke server*

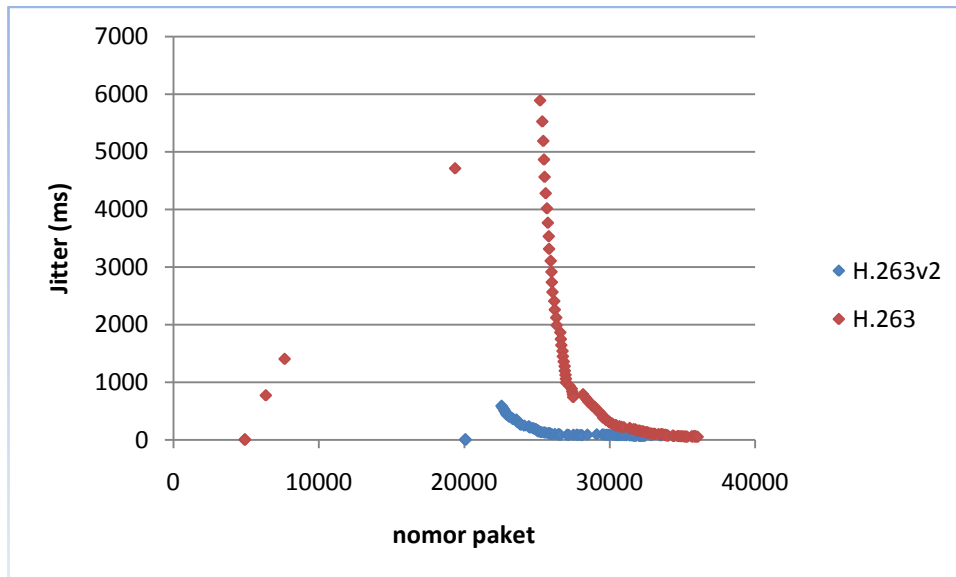
4.4.2. *Uplink Client 125.161.136.172 ke Server*

a. *Paket vs Jitter*

Hasil uji coba menunjukkan bahwa *jitter* rata-rata ketika menggunakan *codec* H.263v2 adalah 165.19 ms sedangkan *jitter* maksimum terjadi ketika pengiriman paket no. 22576 dan besarnya adalah 588.38 ms. Sedangkan *jitter* rata-rata ketika menggunakan *codec* H.263 adalah 955.82 ms dan *jitter* maksimum terjadi ketika pengiriman paket no.25221 dan besarnya adalah 5888. ms. Gambar 4.6 menunjukkan grafik yang mendeskripsikan data hasil uji coba.

Pada pengiriman beberapa paket di awal, *jitter* yang terjadi cukup besar dan terdapat beberapa paket yang tidak terkirimkan pada penggunaan kedua *codec*. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik. Penggunaan IP publik menyebabkan paket harus menunggu ketersediaan *bandwidth* yang memungkinkan paket dapat terkirimkan. *Jitter* yang dihasilkan dari paket yang pertama kali dikirimkan bernilai nol. Hal ini disebabkan paket yang pertama kali dikirim merupakan paket acuan penilaian *jitter* paket setelahnya. Grafik yang

putus-putus memperlihatkan terjadinya interval yang besar diantara paket-paket yang dikirim dan menunjukkan adanya paket yang tidak terkirimkan di fase awal pengiriman paket. Hal ini disebabkan tidak ada *bandwidth* yang dapat terpakai untuk pengiriman paket. Kondisi *bandwidth* seperti ini ditunjukkan pada gambar 4.13.



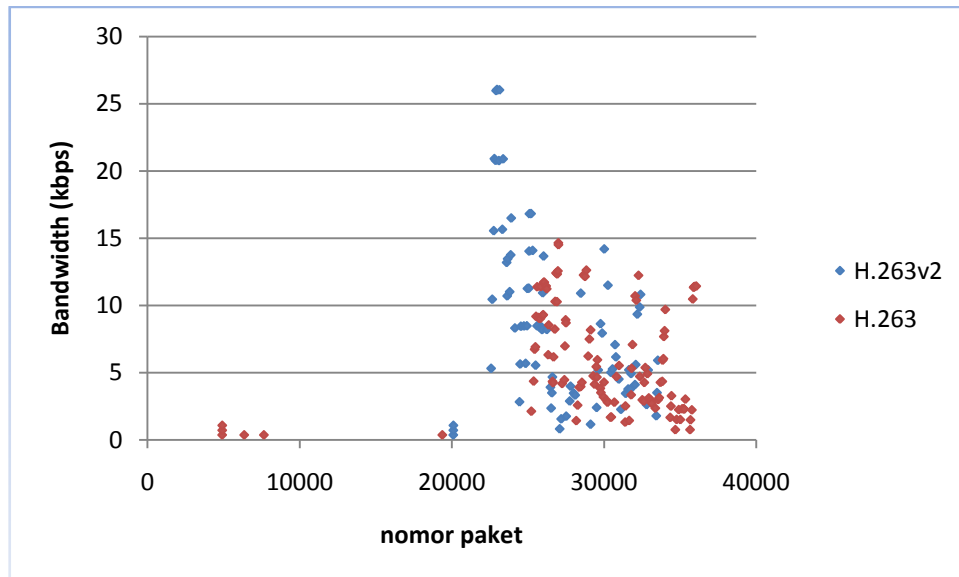
Gambar 4.12 Grafik Paket vs Jitter Uplink client 125.161.136.172 ke server

b. Paket vs *Bandwidth*

Hasil uji coba menunjukkan bahwa *bandwidth* maksimum yang dipakai ketika menggunakan *codec* H.263 adalah 14.64 Kbps dan *bandwidth* rata-rata 5.91 Kbps. Sedangkan *bandwidth* maksimum ketika menggunakan *codec* H.263v2 adalah 26.04 Kbps sedangkan *bandwidth* rata-ratanya adalah 8.7 Kbps. Gambar 4.14 menunjukkan grafik kondisi *bandwidth* yang dipakai untuk setiap paket.

Bandwidth uplink yang terpakai ketika menggunakan *codec* H.263v2 lebih besar daripada *bandwidth* yang terpakai ketika menggunakan *codec* H.263. Hal ini disebabkan H263v2 memiliki *range bit rate* yang lebih besar dibandingkan H.263. Hal ini menyebabkan pemakaian *bandwidth* H.263v2 lebih besar jika dibandingkan H.263. Pada pengiriman beberapa paket di awal, *bandwidth* yang dapat digunakan relatif kecil. Perbandingan antara grafik pada gambar 4.13 dan 4.14 menunjukkan bahwa kecilnya *bandwidth* menyebabkan besarnya rata-rata *jitter* yang terjadi pada pengiriman paket atau bahkan paket

mengalami *wrong sequence*. *Wrong sequence* adalah fenomena pengiriman paket yang tidak menempuh jalan yang telah ditentukan. Kecilnya *bandwidth* juga menyebabkan terjadinya *packet loss*. *Bandwidth* yang terpakai selama pengiriman paket terlihat kurang stabil. Hal ini disebabkan uji coba dilakukan pada jaringan IP publik.



Gambar 4.13 Grafik nomor paket vs *bandwidth* Uplink client 125.161.136.172 ke server

4.4.3. Perbandingan Unjuk Kerja Kedua Client

Tabel 4.5 Perbandingan *Jitter* uplink jaringan internal UI dan speedy

IP Client	Mean Jitter	Mean Jitter	Max. Jitter	Max. Jitter
	H.263	H.263v2	H.263	H.263v2
152.118.101.146	20.64 ms	19.22 ms	45.18 ms	49.82 ms
125.161.136.172	955.82 ms	165.19 ms	5888.19 ms	588.38 ms

Tabel 4.6 Perbandingan *bandwidth* uplink jaringan internal UI dan speedy

IP Client	Mean	Mean	Max.	Max.
	Bandwidth	Bandwidth	Bandwidth	Bandwidth
	H.263	H.263v2	H.263	H.263v2
152.118.101.146	45.18 Kbps	66.65 Kbps	70.55 Kbps	98.88 Kbps
125.161.136.172	5.91 Kbps	8.7 Kbps	14.64 Kbps	26.04 Kbps

Tabel 4.5 dan 4.6 menunjukkan perbandingan unjuk kerja *client* dengan IP address 152.118.101.146 dan 125.161.136.172. Terdapat perbedaan *jitter* rata-rata

dan *jitter* maksimum antara *client* 152.118.101.146 dan *client* 125.161.136.172. *Uplink jitter* 152.118.101.146 jauh lebih kecil dibandingkan *uplink jitter client* 125.161.136.172. Hal ini disebabkan *bandwidth* yang terpakai untuk *uplink* oleh *client* 152.118.101.146 jauh lebih besar daripada *client* 125.161.136.172. hal ini disebabkan *available bandwidth* jaringan UI yang bernilai 100 Mbps jauh lebih besar dibandingkan *available bandwidth* speedy yang hanya bernilai 144.10 Kbps. Walaupun demikian, kedua *codec* tetap menyesuaikan kondisi *bandwidth* untuk proses *coding* dan *decoding* sehingga informasi sampai walaupun dalam dalam kondisi *bandwidth* yang terbatas.

4.5. PERBANDINGAN UNJUK KERJA UPLINK UI NETWORK, FASTNET, DAN SPEEDY

Tabel 4.7 Perbandingan *jitter uplink* UI, Fastnet, dan Speedy

ISP	Mean <i>Jitter</i> (ms)		Maximum <i>Jitter</i> (ms)	
	H.263	H.263v2	H.263	H.263v2
UI	16.11 - 38.35	9.29 - 25.56	45.18 - 2325.49	49.82 - 859.67
Fastnet	74.82	90.99	1320.85	2711.05
Speedy	955.82	165.19	5888.19	588.38

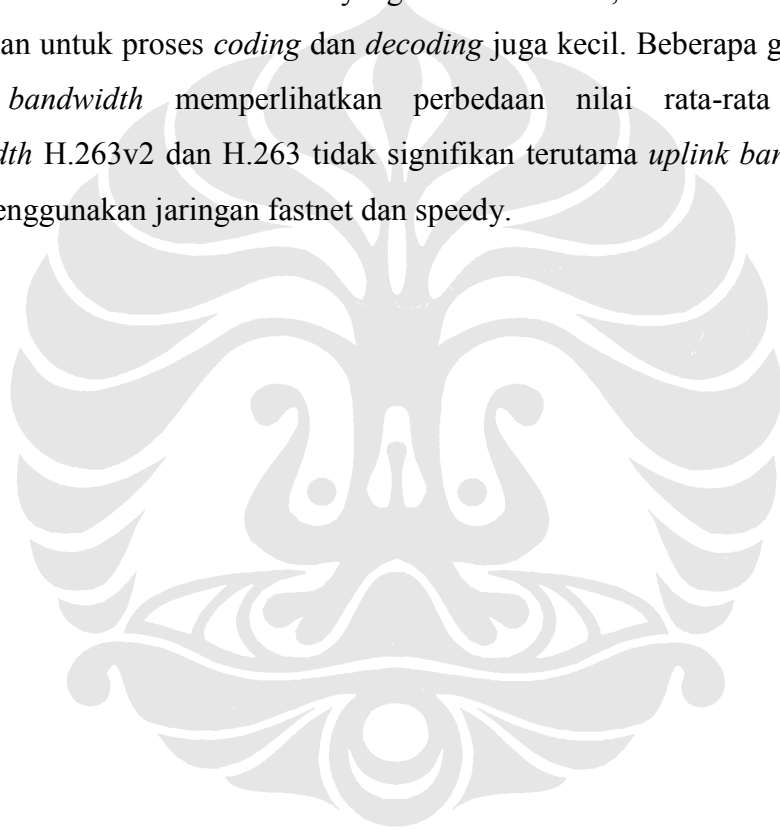
Tabel 4.8 Perbandingan *bandwidth uplink* UI, Fastnet, dan Speedy

ISP	Mean <i>Bandwidth</i> (Kbps)		Maximum <i>Bandwidth</i> (Kbps)	
	H.263	H.263v2	H.263	H.263v2
UI	35.23-72.02	55.72-161.55	59.87-112.3	84.98-205.3
Fastnet	25	25.30	64.62	69.28
Speedy	5.91	8.7	14.64	26.04

Tabel 4.7 dan 4.8 menunjukkan perbandingan unjuk kerja jaringan UI, fastnet, dan speedy dalam penggunaan video *telephony*. *Jitter uplink* yang terjadi ketika menggunakan jaringan UI lebih kecil dibandingkan fastnet dan speedy. Sedangkan speedy merupakan ISP yang menghasilkan *jitter uplink* paling besar dibandingkan jaringan UI dan fastnet. Hal ini disebabkan jaringan UI memiliki *available bandwidth* yang paling besar. Semakin besar *available bandwidth*, maka

semakin besar *bandwidth* yang dapat terpakai untuk video *telephony* sehingga *jitter* yang terjadi semakin kecil.

Tabel 4.8 juga menunjukkan video *codec* H.263 dan H.263v2 dapat menyesuaikan pemakaian *bandwidth* untuk *coding* dan *decoding* sesuai *bandwidth* yang tersedia. Semakin besar *bandwidth* yang tersedia, maka semakin besar pula *bandwidth* yang dapat dipakai. Selain itu, grafik hasil uji coba menunjukkan tingkat adaptasi H.263v2 lebih baik dibandingkan H.263. Walaupun sifat dasar H.263v2 memiliki bit rate lebih besar daripada H.263, tetapi H.263v2 bersifat *conditional*. Ketika *bandwidth* yang tersedia kecil, maka *bandwidth* yang digunakan untuk proses *coding* dan *decoding* juga kecil. Beberapa grafik uji coba *uplink bandwidth* memperlihatkan perbedaan nilai rata-rata penggunaan *bandwidth* H.263v2 dan H.263 tidak signifikan terutama *uplink bandwidth client* yang menggunakan jaringan *fastnet* dan *speedy*.



BAB 5

KESIMPULAN

1. H.263 dan H.263v2 merupakan video *codec* yang dapat dimanfaatkan untuk video *telephony* karena dapat menyesuaikan unjuk kerja yang tidak buruk sesuai dengan ketersediaan *bandwidth* namun tetap mengutamakan sampainya informasi ke tujuan.
2. Ketika menggunakan jaringan UI, *codec* yang menghasilkan *jitter* paling kecil adalah H.263v2 dengan kisaran 9.29 ms sampai 25.56 dengan pemakaian *bandwidth* rata-rata antara 55.72 kbps sampai 161.55 kbps.
3. Ketika menggunakan jaringan fastnet, *codec* yang menghasilkan *jitter* rata-rata paling kecil adalah H.263 yaitu sebesar 74.82 ms dengan pemakaian *bandwidth* rata-rata sebesar 25 kbps.
4. Ketika menggunakan jaringan speedy, *codec* yang menghasilkan *jitter* rata-rata paling kecil adalah H.263v2 yaitu sebesar 165.19 ms dengan pemakaian *bandwidth* rata-rata sebesar 8.7 kbps
5. Unjuk kerja *uplink* video *telephony* berbasis SIP dipengaruhi oleh *bandwidth* yang tersedia. Semakin besar *bandwidth* yang disediakan, maka unjuk kerja akan semakin baik.
6. Secara umum, H.263v2 memiliki unjuk kerja yang lebih baik daripada H.263 pada berbagai konfigurasi ditinjau dari kecilnya *jitter* dan penyesuaian penggunaan *bandwidth*.

DAFTAR ACUAN

- [1] Budi Raharjo, "Teknologi VoIP", 2003. Diakses 15 Maret 2008 dari website Budi Raharjo
<http://budi.insan.co.id>
- [2] W.C. Hardy, *QoS Measurement and Evaluation of Telecommunication Quality of Service*, John Wiley & Sons, 2001.
- [3] W.C. Hardy, *VOIP Service Quality: Measuring and Evaluating Packet-Switched Voice*, McGraw-Hill, 2003.
- [4] Anton Raharja, "VoIP Fundamental", 2006. Diakses 15 Maret 2008 dari website voip-rakyat
<http://voiprakyat.or.id/data/files/materi-voip-fundamental.pdf>
- [5] Mudji Basuki, "Voice over IP", 2007. Diakses 15 Maret 2008 dari Lintasarta
<http://i-networking.net/wp-content/uploads/2007/07/voice-over-ip-voip.pdf>
- [6] Alan B. Johnston, "SIP: Understanding the Session Initiation *Protocol*", Artech House, 2001
- [7] 3rd Generation Partnership Project (3GPP), "Technical Specification Group Services and System Aspects. *Quality of Service (QoS) concept and architecture*, release 5". 3GPP TS 23.107, Version 5.6.0, September 2002.
- [8] Apridian, "VoIP, Sejarah, dan Perkembangannya", Channel-11.net, 2007
- [9] Adnan Basalamah et.al., "Voice over IP". Diakses 15 Maret 2008 dari web Computer Network Research Group
<http://xxx.itb.ac.id/~cnrg>
- [10] Akhmad Ludfy, "Protocol Signaling SIP (Session Initiation *Protocol*)", 2005. Diakses 15 Maret 2008 dari website PT. TELKOM
<http://www.ristinet.com>
- [11] Jyh-Cheng Chen dan Tao Zhang, "IP-Based Next-Generation Wireless Networks, System, Architecture, and *Protocols*", John Wiley & Sons, 2004
- [12] P.Goyal, H.M. Vin, and H. Cheng. Start-time fair queuing: a scheduling algorithm for integrated services packet switching networks. In Proc. of ACM SIGCOMM, pp.157–168, Stanford, CA, August 1996.
- [13] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, pp. 397–413, August 1993.

- [14] W.T. Chen and L.C. Huang. RSVP mobility support: a signaling *protocol* for integrated services Internet with mobile hosts. In Proc. of IEEE INFOCOM, pp.1283–1292, 2000.
- [15] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. “The COPS (common open policy service) *protocol*”. IETF RFC 2748, January 2000.
- [16] I. Cidon and I.S. Gopal. “PARIS: an approach to integrated high-speed private networks”. International Journal of Digital and Analog Cabled Systems, pp. 77–86, April 1988.
- [17] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. “Assured forwarding PHB group”. IETF RFC 2597, June 1999.
- [18] ITU-T. *Draft Recommendation H.263: Video coding for low bitrate communication*. Due to be approved 1995/96. <http://tools.ietf.org/html/draft-ietf-avt-rtp-h263-Video-01>

