

**PERANCANGAN JARINGAN VoIP BERBASIS IAX (INTER  
ASTERISK EXCHANGE) PADA LAN MELALUI NAT(NETWORK  
ADDRESS TRANSLATION)**

**SKRIPSI  
KONSENTRASI TELEKOMUNIKASI**

Diajukan untuk memenuhi persyaratan  
Memperoleh gelar Sarjana Teknik



Disusun oleh :

**IG. JAROT FEBRI SETYO WIBOWO  
NIM 0410630049-63**

**DEPARTEMEN PENDIDIKAN NASIONAL  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
MALANG  
2009**

LEMBAR PERSETUJUAN

PERANCANGAN JARINGAN VoIP BERBASIS IAX (*INTER ASTERISK EXCHANGE*) PADA LAN MELALUI NAT (*NETWORK ADDRESS TRANSLATION*)

**SKRIPSI**  
KONSENTRASI TEKNIK TELEKOMUNIKASI

Diajukan untuk memenuhi persyaratan  
Memperoleh gelar Sarjana Teknik



Disusun oleh :

**IG. JAROT FEBRI SETYO WIBOWO**  
NIM 0410630049-63

Telah diperiksa dan disetujui oleh

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Wahyu Adi Priyono, MT.  
NIP 19600518 198802 1 001

Ali Mustofa, ST., MT.  
NIP 19710601 200003 1 001

**LEMBAR PENGESAHAN**

**PERANCANGAN JARINGAN VoIP BERBASIS IAX (INTER ASTERISK EXCHANGE) PADA LAN MELALUI NAT (NETWORK ADDRESS TRANSLATION)**

**SKRIPSI**

**KONSENTRASI TEKNIK TELEKOMUNIKASI**

Diajukan untuk memenuhi persyaratan  
Memperoleh gelar Sarjana Teknik

Disusun oleh :

**IG. JAROT FEBRI SETYO WIBOWO**  
**NIM 0410630049-63**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 12 Oktober 2009

Skripsi 1

Skripsi 2

Dr. Ir. Sholeh Hadi Pramono, MS.  
NIP. 19580728 198701 1 001

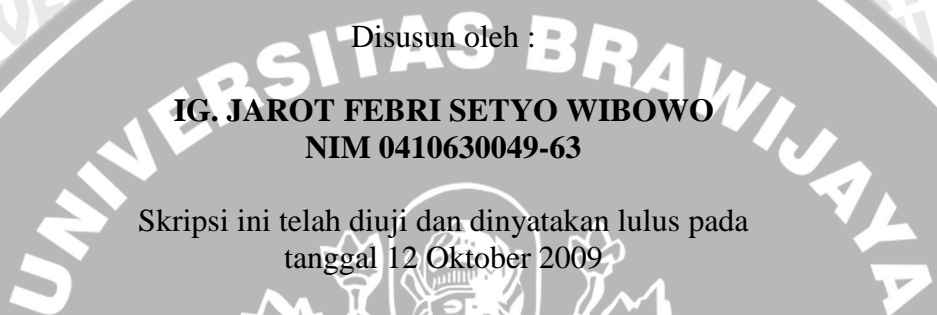
Gaguk Asmungi, ST., MT.  
NIP. 19670627 199802 1 001

Skripsi 3

Rusmi Ambarwati, ST., MT.  
NIP. 19720204 200003 2 002

Mengetahui :  
Ketua Jurusan Teknik Elektro

Rudy Yuwono, ST., M.Sc.  
NIP. 19710615 199802 1 003





UNIVERSITAS BRAWIJAYA

*Sebuah Persembahan Kepada  
Ayah dan Ibu  
Tercinta*



## PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, yang dengan kuasa dan kehendak-Nya, sehingga penulis dapat menyelesaikan Skripsi ini dengan judul “Perancangan Jaringan VoIP Berbasis IAX (*Inter Asterisk Exchange*) pada LAN Melalui NAT (*Network Address Translation*)”. Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Ir. Wahyu Adi P., MSc, dan Bapak Ali Mustofa, ST.,MT. sebagai dosen pembimbing atas saran, konsultasi, kesabaran dan waktu dalam pengerjaan hingga penyelesaian skripsi ini. Tidak terkecuali, ucapan terima kasih penulis sampaikan kepada :

1. Bapak Rudy Yuwono, ST., MSc. selaku Ketua Jurusan Teknik Elektro.
2. Bapak M. Azis Muslim, ST.,MT.,Ph.D selaku Sekretaris Jurusan Teknik Elektro.
3. Ibu Ir. Endah Budi Purnomowati, MT. selaku Ketua Laboratorium Telekomunikasi.
4. Bapak Ir. Nanang Sulistiyanto selaku dosen pembimbing akademik, yang telah memberikan nasehat selama penulis belajar di Jurusan Teknik Elektro.
5. Mas Iswanto, ST selaku laboran Laboratorium Telekomunikasi.
6. Seluruh Dosen dan Karyawan Jurusan Teknik Elektro.
7. Ayah, Ibu dan adik-kakakku (Tika dan Uky) tercinta beserta seluruh keluarga besar atas doa, dan semangat yang selalu diberikan pada penulis.
8. May, Feri, Dina, Uus, Qori, Debby , Pepenk, Onice, Gamma, Rio, Pak Doz, Mbah, Dion, Luthfi, Eja, Feri Meidianto, Icha, Rgb, dan semua Generator 04 sebagai keluarga yang selalu berbagi dengan suka maupun duka.
9. Anggota workshop, Asisten Laboratorium Telekomunikasi '05, '06 dan seluruh rekan-rekan mahasiswa Jurusan Teknik Elektro atas kebersamaan dan pengetahuannya, serta semua pihak yang tidak dapat penulis sebutkan satu-persatu.

Dalam penyusunan skripsi ini, penulis menyadari akan adanya kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk kelengkapan dan kesempurnaan skripsi ini.

Malang, Oktober 2009

Penulis



## ABSTRAK

Ig. Jarot Febri Setyo Wibowo, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Oktober 2009, *Perancangan Jaringan VoIP Berbasis IAX (Inter Asterisk Exchange) pada LAN Melalui NAT(Network Address Translation)*, Dosen Pembimbing: Ir. Wahyu Adi P., MSc dan Ali Mustofa.,ST.,MT

*Voice over Internet Protocol (VoIP)* adalah teknologi yang mampu melewati trafik suara yang berbentuk paket melalui jaringan IP. Keuntungan yang dapat diambil dengan mengirim dan menerima suara menggunakan VoIP adalah biaya lebih murah dari tarif telepon tradisional. VoIP dapat digunakan pada jaringan global, namun VoIP masih memiliki kelemahan, yaitu performansi VoIP rentan terhadap *delay*, *jitter* dan paket *loss*. Oleh karena itu, setiap perencanaan jaringan VoIP, parameter *delay*, *jitter* dan paket *loss* harus diperhatikan agar sesuai dengan standar VoIP. Jaringan LAN memiliki kualitas yang lebih baik dalam *delay*, *jitter* maupun paket *loss*, karena LAN hanya menangani jaringan komputer pada daerah terbatas. Pada beberapa jaringan digunakan NAT (*Network Address Translation*) dengan tujuan untuk melindungi keamanan jaringan didalamnya dari akses pihak luar .

Protokol yang digunakan dalam jaringan VoIP pada skripsi ini adalah protokol IAX (*Inter Asterisk eXchange*) yang dirancang untuk lebih mudah untuk menembus NAT. Dalam skripsi ini akan dibandingkan performansi protokol IAX dan protokol VoIP lainnya, yaitu SIP (*Session Initiation Protocol*) untuk melihat pengaruh NAT pada protokol IAX dan SIP. Perancangan jdan pengujian jaringan dilakukan di Laboratorium Telekomunikasi jurusan Teknik Elektro Universitas Brawijaya menggunakan 1 buah *server* VoIP Trixbox dan 8 buah *client* dengan *softphone* Idefisk, *software capture* menggunakan Wireshark dan Unsniiff. Pengambilan data dilakukan pada saat terjadi panggilan, baik panggilan 2 *client* maupun 8 *client*. Parameter performansi VoIP yang dibandingkan adalah *delay*, *jitter*, dan paket *loss*.

Dari pengujian dapat disimpulkan bahwa jaringan VoIP dapat berjalan sesuai dengan perancangan, yaitu dapat menerima dan melakukan panggilan antar *client*. SIP, sebagai protokol pembanding tidak dapat menembus NAT, sedangkan IAX dapat menembus NAT. Performansi jaringan, yaitu *delay* dan *jitter* dengan menggunakan protokol IAX memberikan hasil yang lebih baik daripada SIP. Packet *loss* pada kedua protokol hampir berimbang, yaitu maksimal 0,03 %, karena kedua protokol menggunakan jaringan lokal yang sama. Secara Umum performansi IAX dan SIP berada dalam kategori “good” untuk semua parameter (*delay end to end < 150 ms*, *jitter < 20 ms*, *paket loss < 0,05%*) dan memenuhi standar VoIP.

Kata kunci: VoIP, IAX (*Inter Asterisk eXchange*), LAN, NAT (*Network Address Translation*), Performansi.

## DAFTAR ISI

Pengantar .....	i
Abstrak .....	ii
Daftar isi .....	iii
Daftar Tabel .....	vi
Daftar Gambar .....	vii
Daftar Lampiran.....	ix
Daftar Singkatan .....	x
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Manfaat .....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 <i>VoIP (Voice over Internet Protocol)</i> .....	4
2.1.1 Konfigurasi Jaringan <i>VoIP</i> .....	4
2.1.2 Protokol <i>VoIP</i> .....	5
2.1.2.1 H.323.....	5
2.1.2.2 <i>SIP (Session Initiation Protocol)</i> .....	6
2.1.2.3 <i>IAX (Inter Asterisk eXchange)</i> .....	10
2.1.2.3.1 Proses Pembentukan Koneksi.....	12
2.1.2.3.2 Frame pada IAX.....	13
2.1.3 <i>Voice Coding VoIP</i> .....	15
2.1.3.1 G.711.....	15
2.1.3.2 G.723.1.....	16
2.1.4 OSI Layer .....	17
2.2 Jaringan Komputer.....	19
2.3.1 LAN ( <i>Local Area Network</i> ).....	19
2.3.2 MAN ( <i>Metropolitan AreaNetwork</i> ) .....	20
2.3.3 WAN ( <i>Wide Area Network</i> ) .....	20
2.3 Ethernet 802.3 .....	21



2.4 NAT ( <i>Network Address Translation</i> ) .....	25
2.5 Parameter Performansi Jaringan .....	29
2.5.1 Throughput .....	30
2.5.2 Delay .....	31
2.5.2.1 Delay Enkapsulasi dan Dekapsulasi .....	32
2.5.2.2 Delay Transmisi .....	35
2.5.2.3 Delay Propagasi .....	35
2.5.2.4 Delay Antrian .....	36
2.5.3 Jitter .....	38
2.5.4 Packet Loss .....	38
2.5.4 Pengukuran Trafik VoIP .....	39
<b>BAB III METODOLOGI</b>	
3.1 Studi literatur .....	43
3.2 Pengambilan Data .....	43
3.2.1 Pengambilan Data Primer .....	43
3.2.2 Pengambilan Data Sekunder .....	43
3.3 Perancangan .....	44
3.3.1 Perencanaan Topologi Jaringan .....	44
3.3.2 Perencanaan Hardware .....	44
3.3.3 Perencanaan Software .....	45
3.4 Pengujian dan Pengambilan Data .....	46
3.4.1 Pengujian Interkoneksi Jaringan .....	47
3.4.1.1 Tujuan Pengujian .....	47
3.4.1.2 Peralatan Pengujian .....	47
3.4.1.1 Prosedur Pengujian <i>client-client</i> dan <i>client-server</i> .....	47
3.4.2 Pengujian Performansi Jaringan VoIP berbasis IAX .....	49
3.4.2.1 Delay <i>End to End</i> .....	49
3.4.2.1.1 Tujuan Pengujian .....	49
3.4.2.1.2 Peralatan Pengujian .....	49
3.4.2.1.3 Prosedur Pengujian .....	49
3.4.2.2 Jitter .....	52
3.4.2.2.1 Tujuan Pengujian .....	52
3.4.2.2.2 Peralatan Pengujian .....	52
3.4.2.2.3 Prosedur Pengujian .....	53



3.4.2.3Paket Loss .....	55
3.4.2.3.1 Tujuan Pengujian.....	55
3.4.2.3.2 Peralatan Pengujian.....	55
3.4.2.3.3 Prosedur Pengujian.....	56
3.5 Kesimpulan dan Saran .....	47
<b>BAB IV PERANCANGAN JARINGAN VoIP</b>	
4.1 Perancangan Server.....	57
4.2 Perancangan Client .....	60
4.3 Perancangan Topologi Jaringan.....	64
4.4 Konfigurasi Router dan NAT .....	66
4.5 Penerapan Protokol IAX pada Jaringan VoIP .....	72
<b>BAB V HASIL PENGUJIAN</b>	
5.1 Pengujian .....	74
5.2 Data Hasil Pengujian .....	74
5.2.1 Data Hasil Pengujian Interkoneksi Jaringan .....	75
5.2.1.1Kesimpulan Hasil Pengujian.....	77
5.2.2 Data Hasil Pengujian Protokol IAX dan SIP dalam NAT.....	77
5.2.2.1Kesimpulan Hasil Pengujian .....	80
5.2.3 Pengujian Performansi Jaringan VoIP berbasis IAX.....	80
5.2.3.1Data Hasil Pengujian <i>Delay End to End</i> .....	80
5.2.3.1.1 Kesimpulan Hasil Pengujian .....	90
5.2.3.2Data Hasil Pengujian <i>Jitter</i> .....	90
5.2.3.2.1 Kesimpulan Hasil Pengujian .....	93
5.2.3.3Data Hasil Pengujian <i>Paket Loss</i> .....	93
5.2.3.3.1 Kesimpulan Hasil Pengujian .....	97
5.2.3.4 Data Hasil Pengujian Bandwidth dan Trafik.....	98
5.2.3.4.1 Kesimpulan Hasil Pengujian .....	105
<b>BAB VI KESIMPULAN DAN SARAN</b>	
6.1 Kesimpulan .....	106
6.2 Saran .....	107
<b>DAFTAR PUSTAKA</b> .....	108
<b>LAMPIRAN</b> .....	110



**DAFTAR TABEL**

No	Judul	Halaman
Tabel 2.1	Fungsi SIP .....	7
Tabel 2.2	Penjelasan full frame .....	15
Tabel 2.3	Penjelasan Mini Frame .....	15
Tabel 2.4	Perbandingan Codec .....	17
Tabel 2.5	Spesifikasi beberapa jenis kabel twisted pair .....	25
Tabel 2.6	Keuntungan dan Kerugian Menggunakan NAT .....	26
Tabel 2.7	Cara Kerja NAT .....	27
Tabel 2.8	Faktor Kali Komponen Delay Fast Ethernet .....	36
Tabel 2.9	Perbandingan model trafik .....	40
Tabel 3.1	Kebutuhan Hardware .....	45
Tabel 3.2	perangkat lunak yang dibutuhkan .....	46
Tabel 3.3	Pengalamatan IP .....	48
Tabel 4.1	Registrasi Client .....	64
Tabel 4.2	Pengalamatan IP .....	66
Tabel 4.3	Konfigurasi Jaringan Publik .....	69
Tabel 4.4	Konfigurasi Jaringan Lokal .....	70
Tabel 5.1	Pengujian Interkoneksi Jaringan Tanpa Menggunakan NAT .....	76
Tabel 5.2	Pengujian Interkoneksi Jaringan menggunakan NAT .....	76
Tabel 5.3	Data Pengujian Protokol IAX dan SIP dalam Melalui NAT .....	80
Tabel 5.4	Hasil Analisis Delay End to End .....	84
Tabel 5.5	Hasil Analisis Jitter .....	92
Tabel 5.6	Hasil Analisis Paket Loss Tanpa Menggunakan NAT .....	97
Tabel 5.7	Hasil Analisis Paket Loss Menggunakan NAT .....	97
Tabel 5.8	Throughput pada Jaringan VoIP menggunakan IAX .....	98
Tabel 5.9	Throughput pada Jaringan VoIP menggunakan SIP .....	99
Tabel 5.10	Kecepatan Transmisi Rata – Rata Client pada Jaringan VoIP ...	100





## DAFTAR GAMBAR

No	Judul	Halaman
Gambar 2.1	Jaringan VoIP Secara Umum .....	5
Gambar 2.2	Set-up Flow of Signalling SIP .....	8
Gambar 2.3	Flow Chart Set-Up Signalling SIP .....	9
Gambar 2.4	Beberapa Panggilan Melalui UDP Tunggal .....	10
Gambar 2.5	IAX Call Origination State Diagram melalui NAT.....	11
Gambar 2.6	Proses Call Setup Menggunakan Protokol IAX .....	12
Gambar 2.7	Flow Chart Set-up Signaling IAX.....	13
Gambar 2.8	Media Flow IAX.....	14
Gambar 2.9	Format binary dari full frame .....	14
Gambar 2.10	OSI layer.....	19
Gambar 2.11	Jaringan LAN .....	19
Gambar 2.12	Jaringan MAN .....	20
Gambar 2.13	Jaringan WAN .....	21
Gambar 2.14	Aliran Data pada Jaringan Berbentuk Bus .....	21
Gambar 2.15	MAC CSMA/CD.....	23
Gambar 2.16	Peristiwa Tabrakan dalam IEEE 802.3.....	24
Gambar 2.17	Format frame IEEE 802.3.....	24
Gambar 2.18	Peletakan NAT dalam Jaringan .....	25
Gambar 2.19	Flowchart Mekanisme NAT, IP Privat Menjadi IP Publik, Flowchart Mekanisme NAT, IP publik Menjadi IP Privat.....	29
Gambar 2.20	Visualisasi Keadaan Paket pada Suatu Jaringan .....	30
Gambar 2.21	Komponen Delay pada Implementasi VoIP di Local Area Network .....	31
Gambar 2.22	Proses Enkapsulasi Paket Data.....	32
Gambar 2.23	Kebutuhan <i>Bandwidth</i> Paket IAX .....	33
Gambar 2.24	Model Antrian Single Client.....	37
Gambar 3.1	Arsitektur jaringan VoIP menggunakan protocol IAX.....	44
Gambar 3.2	Metode Pengujian .....	46
Gambar 3.3	Arsitektur Pengujian Interkoneksi Jaringan .....	47
Gambar 3.4	Setting Pengalamatan IP pada Client.....	48
Gambar 3.5	Konfigurasi Pengujian Jaringan VoIP .....	50

Gambar 3.6	Flowchat Prosedur Pengujian Performansi .....	51
Gambar 3.7	Flowchart Analisis <i>Jitter</i> .....	54
Gambar 4.1	Flowchart Proses Konfigurasi Server .....	57
Gambar 4.2	Tampilan Server Trixbox Diakses Menggunakan Web Browser .....	59
Gambar 4.3	Membuat Extension IAX.....	60
Gambar 4.4	Softphone Idefisk.....	60
Gambar 4.5	Diagram Alir Perancangan Client.....	61
Gambar 4.6	Add Account pada Softphone Idefisk.....	62
Gambar 4.7	Arsitektur Jaringan VoIP Menggunakan Protocol IAX .....	65
Gambar 4.8	Flowchart Proses Penentuan Routing .....	67
Gambar 4.9	Flowchart Proses untuk Outbound dan Inbound Trafik, Flowchart Proses Translasi Paket.....	68
Gambar 4.10	Konfigurasi Router dan NAT .....	70
Gambar 4.11	Penggunaan Protokol IAX .....	72
Gambar 4.12	Signaling dan Data Stream pada Protokol IAX.....	73
Gambar 5.1	Proses Pengujian Interkoneksi Client ke Server.....	75
Gambar 5.2	Panggilan SIP ke SIP tanpa melewati NAT .....	78
Gambar 5.3	Panggilan SIP ke SIP melewati NAT.....	78
Gambar 5.4	Panggilan IAX ke IAX tanpa melewati NAT.....	79
Gambar 5.5	Panggilan IAX ke IAX melewati NAT .....	79
Gambar 5.6	Hasil Capture Unsiff.....	81
Gambar 5.7	Flowchart Prosedur Analisis Delay IAX.....	82
Gambar 5.8	Perhitungan <i>Delay End to End</i> Menggunakan Matlab .....	89
Gambar 5.9	Hasil Pengujian Jitter.....	91
Gambar 5.10	Grafik Hasil Pengujian Jitter.....	93
Gambar 5.11	Analisis Aliran Paket IAX.....	94
Gambar 5.12	Flowchart Analisis <i>Loss</i> .....	95
Gambar 5.13	Paket Loss SIP Menggunakan Wireshark .....	96
Gambar 5.14	Kapasitas Panggilan Maksimal berdasarkan Kapasitas Kanal ...	103
Gambar 5.15	Perhitungan Paket Loss Menggunakan Matlab .....	104



### DAFTAR LAMPIRAN

No	Judul	Halaman
Lampiran 1	Program Perhitungan <i>Delay End to End</i> .....	110
Lampiran 2	Program Perhitungan Jumlah User berdasarkan Kapasitas Kanal.....	111
Lampiran 3	Program Perhitungan Paket <i>Loss</i> .....	112



## DAFTAR SINGKATAN

CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>
DTE	<i>Digital Terminal Equipment</i>
FIFO	<i>First In First Out</i>
ISO	<i>International Standard Organization</i>
ISP	<i>Internet Service Provider</i>
IAX	<i>Inter Asterisk eXchange</i>
ITU-T	<i>International Telecommunication Union - Telecommunication Sector</i>
LAN	<i>Local Area Network</i>
LCC	<i>Lost Calls Cleared</i>
LCD	<i>Lost Calls Delayed</i>
LCH	<i>Lost Calls Held</i>
LCR	<i>Lost Calls Retried</i>
MAC	<i>Medium Access Control</i>
MAN	<i>Metropolitan Area Network</i>
NAT	<i>Network Address Translation</i>
NIC	<i>Network Interface Card</i>
PCM	<i>Pulse Code Modulation</i>
PSTN	<i>Public Switched Telephone Network</i>
QoS	<i>Quality of Service</i>
RSVP	<i>Resource Reservation Protocol</i>
RTCP	<i>Real-Time Control Protocol</i>
RTP	<i>Real-Time Protocol</i>
SIP	<i>Session Initiation Protocol</i>
TE	<i>Terminal Equipment</i>
UDP	<i>User Datagram Protocol</i>
VoIP	<i>Voice over Internet Protocol</i>
WAN	<i>Wide Area Network</i>



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

*Voice over Internet Protocol (VoIP)* adalah teknologi yang melewatkan trafik suara berbentuk paket melalui jaringan IP. Jaringan IP merupakan jaringan komunikasi data yang berbasis *packet-switch*, sehingga dalam mengirim dan menerima suara, VoIP menggunakan komunikasi data yang berbasis *packet-switch*. Keuntungan yang diperoleh bila mengirim dan menerima suara menggunakan VoIP adalah dari segi biaya lebih murah dari tarif telepon tradisional. Jaringan IP merupakan jaringan yang bersifat *global*, sehingga dengan berkomunikasi menggunakan jaringan IP yang telah ada, memungkinkan *provider* untuk menekan biaya hubungan internasional. Penghematan ini dapat dilihat dari daftar tarif yang diberikan *provider* SLI (Sambungan Langsung Internasional). Selain itu, biaya *maintenance* dapat ditekan karena *voice* dan *data network* tidak terpisah, sehingga VoIP dapat ditambah, dipindah dan diubah. Hal ini karena VoIP dapat dipasang di semua jaringan internet dan *IP address*, tidak seperti telepon tradisional yang harus mempunyai *port* tersendiri di sentral atau PBX.

Keterbatasan alamat IPv4 merupakan masalah pada jaringan *global*. Untuk memaksimalkan penggunaan alamat IP yang diberikan oleh *Internet Service Provider (ISP)* dapat digunakan *Network Address Translation* atau NAT. NAT digunakan untuk membatasi akses dari luar jaringan. NAT membuat jaringan yang menggunakan alamat lokal, alamat yang tidak ada dalam tabel routing internet dan dikhususkan untuk jaringan lokal, dapat berkomunikasi ke internet dengan jalan memakai sementara alamat IP internet yang dialokasikan oleh ISP.

Protokol adalah komponen berupa seperangkat aturan komunikasi antar user agent, antar proxy atau user agent dengan *proxy* dan sebaliknya. *Proxy* adalah suatu aplikasi yang menjembatani antara *client* dengan ISP. Saat ini dan sebelumnya protokol yang digunakan adalah H. 323, namun protokol ini tidak dapat melewati NAT karena menggunakan dua buah *port* dan pensinyalannya menggunakan model biner yang relatif sulit diubah. Protokol SIP atau *Session Initiation Protocol* diperkenalkan sebagai pengganti H.323.

Protokol lain yang mulai sering digunakan adalah *IAX* atau *Inter Asterisk eXchange*. Protokol IAX telah direvisi menjadi IAX2 untuk menggantikan IAX yang sebelumnya. Sehingga penyebutan protokol IAX dalam skripsi ini selanjutnya,

menunjuk pada protokol IAX2. Protokol IAX dapat melewati NAT dengan mudah dibandingkan dengan protokol lainnya. Hal ini dikarenakan IAX hanya menggunakan satu *port* saja untuk membentuk *session* dan *media transfer*. Protokol sebelumnya menggunakan dua *port*, satu *port* untuk *signalling* dan sisanya untuk mengirim *audio*.

LAN (*Local Area Network*) adalah suatu jaringan komputer pada suatu daerah tertentu dan memiliki kecepatan yang lebih cepat daripada MAN (*Metropolitan Area Network*) atau WAN (*Wide Area Network*). LAN memiliki kualitas yang lebih baik dalam *delay* maupun *throughput*, karena LAN hanya menangani jaringan komputer pada daerah terbatas.

Jaringan LAN dapat digunakan sebagai salah satu *media transfer* dalam membangun jaringan VoIP. Pada skripsi ini, kami akan merancang jaringan VoIP pada LAN di lingkungan Teknik Elektro Universitas Brawijaya, menggunakan protokol baru, yaitu protokol IAX dan melalui NAT, sebagai simulasi apabila protokol IAX digunakan sebagai media komunikasi suara secara luas.

## 1.2 Rumusan Masalah

Berdasarkan masalah yang telah dijelaskan sebelumnya, maka dapat disusun suatu rumusan masalah:

1. Bagaimana arsitektur jaringan VoIP (*Voice over IP*) yang dirancang menggunakan Protokol IAX (*Inter Asterisk Exchange*) pada jaringan LAN (*Local Area Network*) di Teknik Elektro Universitas Brawijaya.
2. Dimana dan bagaimana implementasi NAT (*Network Address Translation*) pada jaringan VoIP.
3. Bagaimana implementasi IAX dan kompatibilitasnya pada jaringan VoIP.
4. Berapa nilai *delay*, *jitter* dan paket *loss* sebagai parameter performansi jaringan VoIP menggunakan protokol IAX yang melalui NAT.
5. Berapa jumlah pengguna maksimum dengan menggunakan *bandwidth* yang tersedia bila menggunakan protokol IAX dan bila dibandingkan dengan jaringan VoIP menggunakan protokol lainnya.

## 1.3 Batasan Masalah

Dalam analisis performansi jaringan VoIP menggunakan protokol IAX, terdapat beberapa pembatasan masalah, antara lain :



1. Jaringan yang dirancang adalah terdapat 1 buah *server* dalam jaringan LAN (Teknik Elektro Universitas Brawijaya)
2. Parameter performansi yang akan di hitung adalah *delay*, *jitter*, dan *packet loss*. Data diambil pada rentang pukul 10.00 sampai 12.00 WIB.
3. Jaringan VoIP menggunakan *interface ethernet card* dan kabel LAN tipe RJ45
4. Menggunakan *codec G.711*
5. Delay router;  $t_{\text{rou}} = 2 \text{ ms}$ , delay DTE;  $t_{\text{DTE}} = 1 \text{ ms}$
6. *Software* yang digunakan sebagai *softphone* adalah Idefisk
7. Penangkapan paket data yang berjalan saat terjadi panggilan menggunakan *software* Unsniff dan Wireshark.
8. Protokol yang akan dijadikan komparasi adalah SIP

#### 1.4 Tujuan

Tujuan penulisan skripsi ini adalah untuk merancang jaringan VoIP berbasis IAX pada LAN di jurusan Teknik Elektro Universitas Brawijaya melalui NAT.

#### 1.5 Manfaat

Manfaat penulisan ini antara lain:

1. Bagi pembaca, diharapkan dapat memberikan informasi mengenai performansi jaringan VoIP berbasis protokol IAX dengan maupun tanpa melewati NAT pada jaringan LAN, sehingga dapat menentukan penggunaan protokol yang tepat sesuai dengan kebutuhan.
2. Bagi penulis, mampu memahami cara merencanakan jaringan LAN dengan menggunakan NAT sebagai media komunikasi layanan multimedia VoIP berbasis protokol IAX, sehingga bisa dihitung performansinya.
3. Bagi masyarakat, dapat mengimplementasikan jaringan VoIP berbasis IAX pada LAN di lingkungan RT/RW.
4. Bagi operator telekomunikasi, hasil analisis komparasi antara SIP dan protokol IAX dapat digunakan sebagai pertimbangan untuk menentukan protokol yang tepat dalam pengembangan jaringan VoIP.

## BAB II

### TINJAUAN PUSTAKA

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

*Voice over Internet Protocol (VoIP)* adalah teknologi yang mampu melewati trafik suara, yang berbentuk paket melalui jaringan IP. Jaringan IP merupakan jaringan komunikasi data yang berbasis *packet-switch*. Sinyal suara sebelum dipaketkan mengalami *voice coding* atau perubahan format suara kedalam bentuk digital agar dapat dilewatkan melalui jaringan IP.

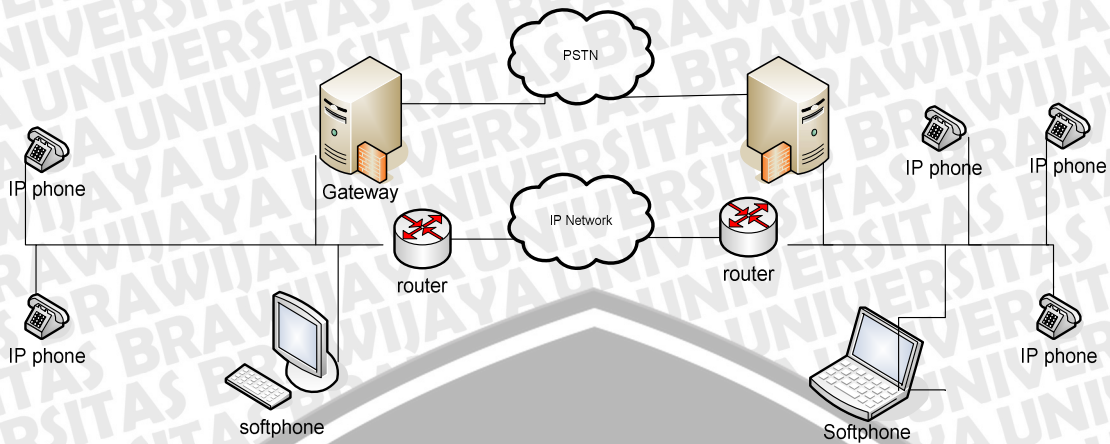
VoIP saat ini telah menjadi bagian dari dunia komunikasi seluruh dunia, karena VoIP memberikan nilai tambah pada jaringan internet yang telah ada. VoIP juga memberikan nilai ekonomis bila dibandingkan dengan telekomunikasi menggunakan jaringan telekomunikasi secara terpisah. Nilai ekonomis dapat dicapai karena jaringan VoIP tidak membutuhkan pemasangan jaringan baru, melainkan dapat melalui jaringan internet yang telah ada dengan sistem sewa jaringan.

Dalam perancangan jaringan VoIP, hal yang perlu mendapat perhatian adalah masalah *delay*, *jitter*, *packet loss* dan *bandwidth*. *Delay* didefinisikan sebagai waktu yang dibutuhkan untuk mengirimkan data dari sumber (pengirim) ke tujuan (penerima), sedangkan *bandwidth* adalah kecepatan maksimum yang dapat digunakan untuk melakukan transmisi data antar komputer pada jaringan IP atau internet.

##### 2.1.1 Konfigurasi Jaringan VoIP

Jaringan VoIP dapat dirancang sesuai kebutuhan dan perangkat pengguna. Beberapa bentuk jaringan dapat diaplikasikan dan memiliki fungsi serupa, namun tidak menutup kemungkinan bahwa konfigurasi jaringan yang digunakan sangat berbeda. Salah satu contoh bentuk jaringan VoIP adalah jaringan VoIP yang menggunakan komputer sebagai TE (*Terminal Equipment*).





Gambar 2.1 : Jaringan VoIP Secara Umum  
 Sumber: *VoIP Implementation & Management*, [www.netiq.com](http://www.netiq.com)

Jaringan ini banyak digunakan dalam gedung atau tempat yang memiliki koneksi LAN dan tidak memakai jaringan PSTN (*Public Switched Telephone Network*) dari sambungan telepon kabel.

### 2.1.2 Protokol VoIP

Protokol adalah aturan yang menjelaskan secara detail bagaimana komputer berinteraksi, termasuk komunikasi antar computer yang memerlukan banyak fungsi protokol dalam membangun hubungan komunikasi, mengakhiri hubungan komunikasi dan penanganan bila terjadi masalah. Tiga aspek utama komunikasi yang diperhatikan oleh protokol adalah representasi data dan pengkodean, petransmisian, dan identifikasi maupun penanganan kesalahan atau kegagalan.

Jaringan VoIP dapat saling berkomunikasi dengan *server* VOIP ditempat lain. Oleh karena itu, VoIP harus memiliki protokol yang dapat saling mengerti antara kedua komputer tersebut, sehingga dapat saling bertukar paket data *voice*. Protokol memiliki peran penting karena protokol memberikan aturan dan perintah yang sama sehingga dapat dikenali oleh *server*, *client* atau perangkat komunikasi lainnya. Beberapa protokol VoIP adalah H.323, SIP dan IAX.

#### 2.1.2.1 H.323

Pada H.323 terdapat beberapa protokol dalam pengiriman data yang mendukung agar data terkirim *real-time*. Protokol tersebut adalah *RTP* (*Real-Time Protocol*), *RTCP* (*Real-Time Control Protocol*), dan *RSVP* (*Resource Reservation Protocol*)

*RTP (Real-Time Protocol)* adalah protokol yang dibuat untuk mengkompensasi *jitter* yang terjadi pada jaringan IP. RTP dapat digunakan untuk beberapa macam *data stream* yang *realtime* seperti data suara dan data *video*. RTP berisi informasi tipe data yang di kirim, *timestamps* yang digunakan untuk pengaturan waktu suara percakapan agar terdengar seperti yang diucapkan, dan *sequence numbers* yang digunakan untuk pengurutan paket data dan mendeteksi adanya paket yang hilang

*RTCP(Real-Time Control Protocol)* merupakan suatu protokol yang biasanya digunakan bersama-sama dengan RTP. RTCP digunakan untuk mengirimkan *paket control* setiap *client* yang berpartisipasi pada percakapan dan digunakan sebagai informasi untuk kualitas transmisi pada jaringan.

*RSVP(Resource Reservation Protocol)* RSVP bekerja pada *layer transport*, digunakan untuk menyediakan *bandwidth* agar data suara yang dikirimkan tidak mengalami *delay* ataupun kerusakan saat mencapai alamat tujuan *unicast* maupun *multicast*.

#### 2.1.2.2 SIP (*Session Initiation Protocol*)

SIP merupakan kepanjangan dari *Session Initiation Protocol*, yaitu *protocol call setup* yang beroperasi pada *layer* aplikasi. Protokol lain dengan fungsi yang sama adalah H.323 yang dikeluarkan oleh ITU. SIP berbasis teks dan sangat fleksibel dan mendukung semua layanan, dan hampir semua perangkat telekomunikasi saat ini sudah mendukung SIP.

Di dalam IP dan telepon tradisional, selalu dibedakan dengan jelas dua tahap panggilan *voice*. Tahap pertama adalah "*call setup*" yang mencakup semua detail keperluan agar dua perangkat telepon dapat berkomunikasi. Tahap selanjutnya adalah "*transfer data*" dimana *call setup* sudah terbentuk. Di dalam VoIP, SIP adalah protokol *call setup* yang beroperasi pada *layer* aplikasi.



Tabel 2.1. Fungsi SIP

Fungsi	Keterangan
Registrasi dan identifikasi lokasi <i>User</i>	<i>End points (IP Phones)</i> melakukan notifikasi lokasi ke <i>SIP proxies</i> . SIP juga menentukan <i>end point</i> mana yang akan berpartisipasi dalam panggilan.
Ketersediaan <i>User</i>	SIP digunakan oleh <i>end point</i> untuk menentukan apakah panggilan yang datang dijawab atau tidak.
Kemampuan <i>User</i>	SIP digunakan <i>end point</i> melakukan negosiasi dengan kemampuan network, seperti penggunaan <i>voice codec</i> .
<i>Set-up Session</i>	SIP memberitahu ke <i>end point</i> bahwa <i>end point</i> harus <i>ringing</i> , hal ini terkait dengan fitur seperti <i>conference</i> .
<i>Management Session</i>	SIP digunakan untuk <i>transfer calls</i> , memutuskan <i>calls</i> , dan merubah parameter panggilan di tengah <i>session</i> .

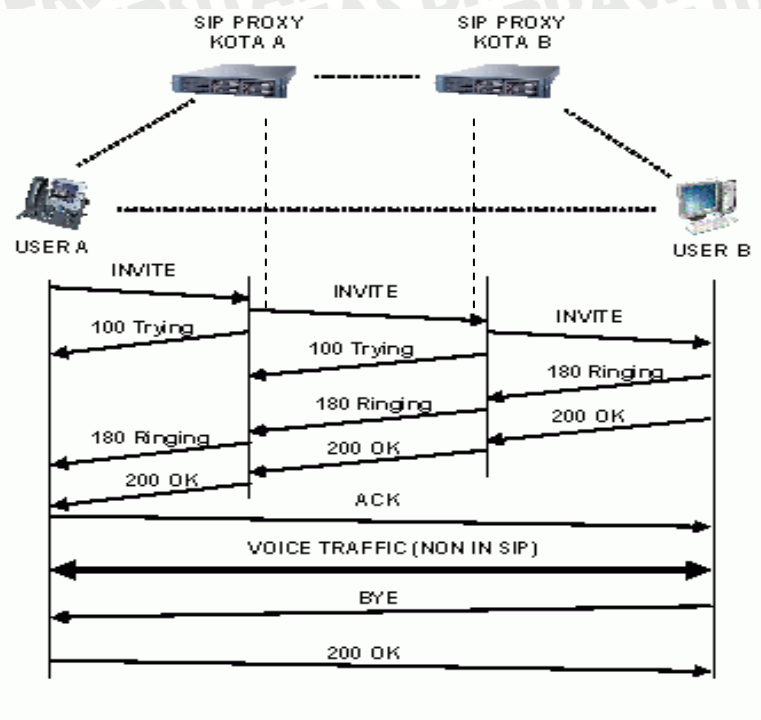
Sumber: <http://www.ristinet.com/index.php?lang=ind&ch=8>

SIP tidak hanya meng-*handle call setup*, tetapi juga memiliki kemampuan fungsi-fungsi lain untuk mendukung layanan VoIP. Namun kekurangan SIP yang menggunakan dua *port* untuk *signalling* dan transmisi data, mengakibatkan SIP relatif tidak mudah menembus NAT.

Kelebihan SIP adalah penggunaan *text-based model* pada *request/response* di HTTP. Hal ini akan memudahkan *debug* karena *messages* mudah untuk dilakukan rekonstruksi dan mudah dilihat dan dianalisa (bagi *network manager*). Berbeda dengan H.323, SIP adalah protokol yang sangat sederhana, namun demikian mempunyai fitur yang cukup banyak untuk lingkungan telepon PBX. SIP dapat berjalan diatas IPv4 and IPv6 dan dapat menggunakan TCP atau UDP. Implementasi VoIP pada umumnya menggunakan IPv4 dan UDP.

Meskipun dua perangkat SIP dapat berkomunikasi secara langsung, namun pada umumnya menggunakan intermediasi sistem sebagai *SIP Proxy*. *SIP proxy* hanya berpartisipasi dalam melakukan *set-up*, sekali *call set-up* terbentuk, perangkat dapat mengirimkan *voice traffic* secara langsung tanpa melibatkan *proxy* lagi. *SIP Proxy* sangat membantu dalam membagi tugas dan memudahkan dalam implementasi.

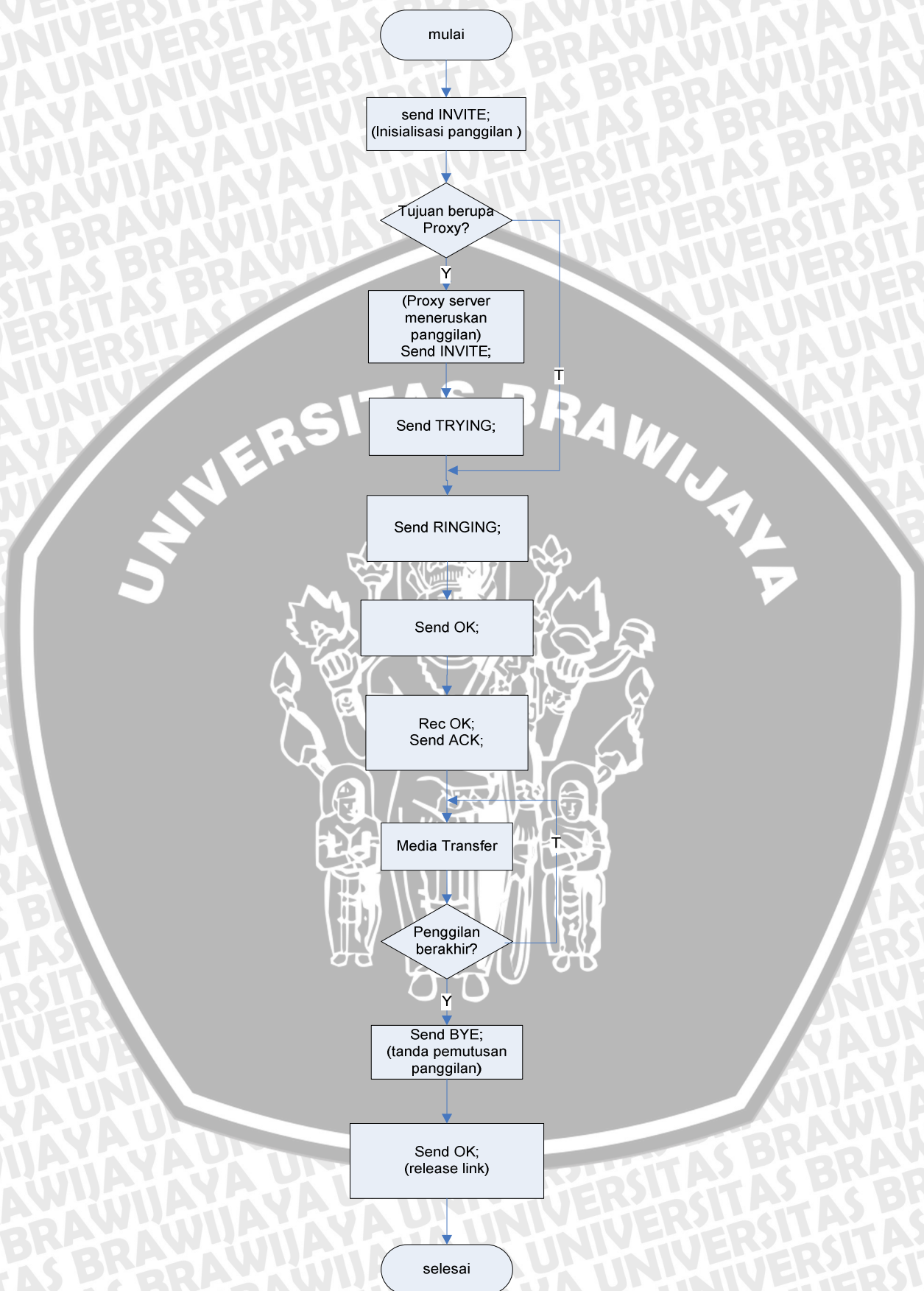
Gambar 2.2 menunjukkan dialog SIP yang melibatkan 2 pengguna VoIP dan *SIP Proxy server*. Dalam kasus ini pesan kontrol dalam SIP telah disederhanakan untuk memudahkan melihat aliran *traffic*.



Gambar 2.2. *Set-up Flow of Signaling SIP*  
 Sumber : <http://www.ristinet.com/index.php?lang=ind&ch=8>

Dalam diagram menunjukkan bahwa *proxy* tidak berperan lagi setelah *call set-up* terbentuk, namun hal ini tidak berlaku untuk semua panggilan. *Proxy* tetap dipertahankan sebagai media perantara untuk mengakomodasi beberapa *mid-call features*, seperti layanan *conference* dan *accounting*.





Gambar 2.3. Flow Chart Set-up Signaling SIP  
 Sumber: <http://www.ristinet.com/index.php?lang=ind&ch=8>

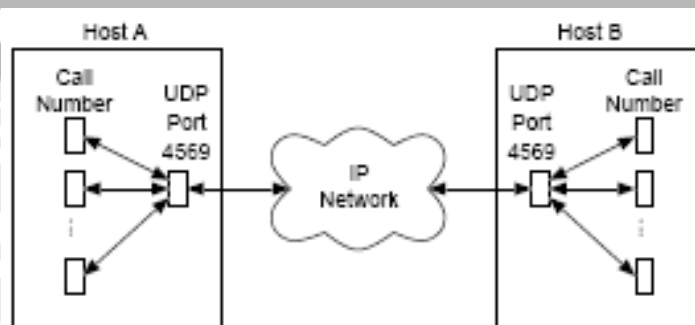
Kemampuan registrasi khususnya berguna di lingkungan dimana *end terminal* tidak memiliki *IP address* yang statis. Dalam *SIP registration server* dapat dijadikan satu dengan *proxy server* atau dapat juga dengan sistem terpisah. Registrasi *user* juga tidak terbatas dalam satu lokasi tetapi dapat dari berbagai lokasi, dan *proxy server* bertanggung jawab memutuskan *user* mana yang akan "ring" ketika panggilan datang. Dengan menggunakan *registration server SIP* dapat memilih salah satu *user* yang akan "ring" atau semua *user*.

### 2.1.2.3 IAX (Inter Asterisk eXchange)

Protokol *Inter-Asterisk eXchange (IAX)* menyediakan kontrol dan pengiriman aliran data melalui jaringan *Internet Protocol (IP)*. IAX dapat digunakan dengan berbagai aliran data termasuk *video* namun lebih difokuskan pada pengontrolan panggilan suara melalui IP.

IAX adalah protokol media dan pensinyalan *peer-to-peer*. Hal ini berarti bahwa setiap ujung-ujung sambungan dapat menerima operasi protokol IAX. Pada dasarnya, tetap ada dua protokol, sebuah protokol untuk pensinyalan dan protokol untuk pengiriman media itu sendiri. Hal ini berbeda dengan arsitektur yang dikembangkan oleh IETF (*Internet Engineering Task Force*) pada umumnya, yaitu pada pemisahan antara komponen kontrol dan aliran media menggunakan protokol yang berbeda. Karena *signalling* dan pertukaran media menggunakan *port* yang sama, maka IAX tidak mengalami masalah dalam melewati NAT.

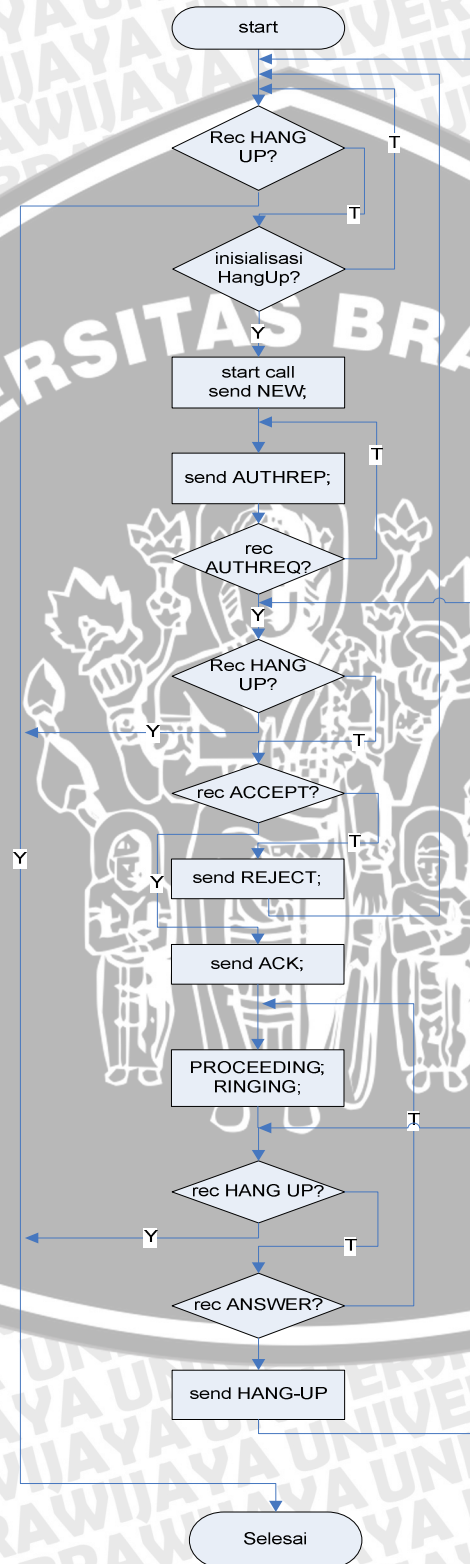
Gambar 2.4 menggambarkan hubungan antara dua *host* internet. Masing-masing *host* menggunakan UDP (*User Datagram Protocol*) *port* 4569 untuk saling komunikasi semua paket internet. IAX kemudian menggunakan *call number* 15-bit untuk menggabungkan berbagai aliran melalui UDP pada *port* tersebut.



Gambar 2.4 Beberapa Panggilan Melalui UDP Tunggal  
Sumber: IAX Protocol Description, [www.pdf-search-engine.com](http://www.pdf-search-engine.com)



UDP merupakan salah satu protokol utama diatas IP merupakan *transport protocol* yang lebih sederhana dibandingkan dengan TCP. UDP digunakan untuk situasi yang tidak mementingkan mekanisme reliabilitas.



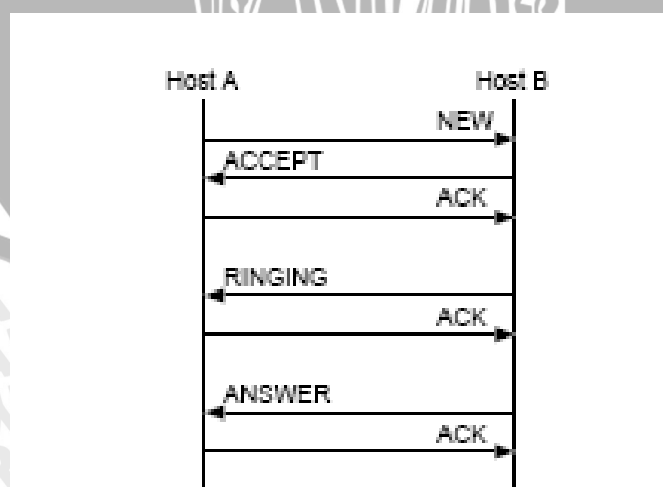
Gambar 2.5 IAX Call Origination State Diagram melalui NAT  
Sumber : Draft-IAX-05 Inter-Asterisk eXchange Version 2

IAX adalah protokol *binary*. Desain ini dipilih untuk efisiensi *bandwidth*. Sehingga, secara umum protokol ini dirancang agar dapat mengirimkan *voice call* dengan memakai *bandwidth* seperlunya.

IAX dirancang untuk menggunakan satu *port* untuk komunikasi data dan suara. Protokol ini dibuat karena NAT hanya dapat melewati data internet pada satu *port* saja secara bersamaan. Hal inilah yang menyebabkan IAX dapat dengan mudah menembus NAT, karena pada protokol VoIP yang lain, *port* untuk suara dan data dilewatkan secara terpisah. Ketika melewati NAT, komunikasi suara akan terhambat mulai dari suara yang tidak terkirim atau satu sisi suara saja yang terkirim. Cara yang dapat digunakan protokol VoIP lain untuk menembus NAT ini adalah dengan menggunakan *mediaproxy* dan *rtpproxy*.

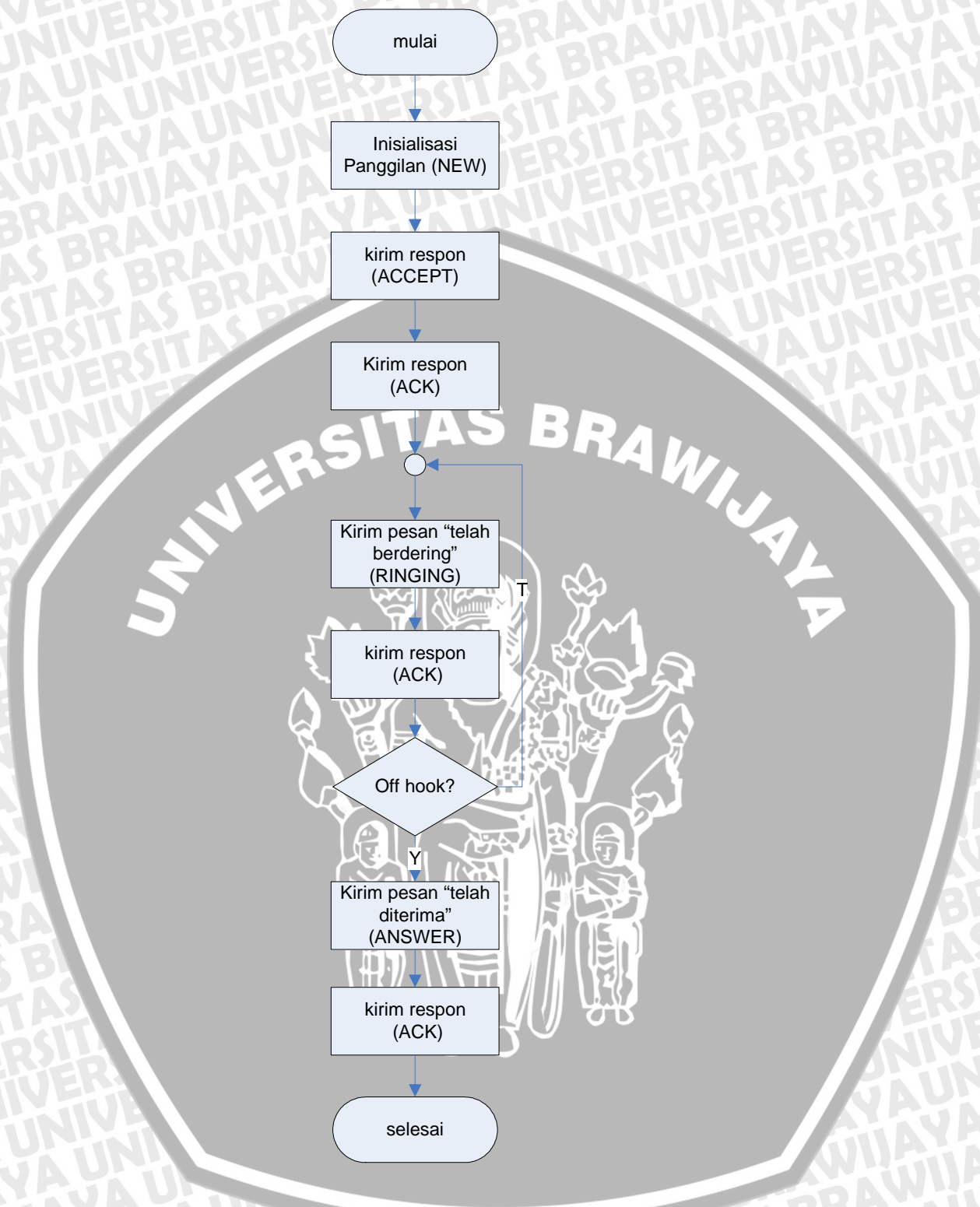
### 2.1.2.3.1 Proses Pembentukan Koneksi

Gambar 2.6 dan 2.7 menunjukkan aliran pesan dasar yang digunakan untuk membuat koneksi. Dalam contoh ini, *host A* memulai untuk mengirim pesan kepada *host B*. *Host B* kemudian mengirim pesan *ACCEPT*, yang mengindikasikan kepada *host A* bahwa telah menerima pesan dan siap untuk melayani panggilan. *Host A* kemudian mengirim pesan *ACK* kepada *host B* yang mengindikasikan telah menerima pesan *ACCEPT*. Sesudah *host B* berdering, *host B* juga mengirim pesan *RINGING* kepada *host A*. *Host A* kemudian mengirim pesan *ACK* kepada *host B* bahwa ia telah menerima pesan *RINGING*. Akhirnya, ketika telepon diangkat, *Host B* mengirim sebuah pesan *ANSWER* kepada *host A*. *Call setup* telah berhasil.



Gambar 2.6 Proses *Call Setup* Menggunakan Protokol IAX  
Sumber: IAX Protocol Description, [www.pdf-search-engine.com](http://www.pdf-search-engine.com)



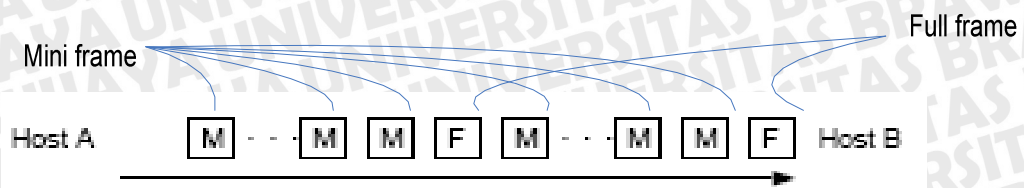


Gambar 2.7 Flow Chart Set-up Signaling IAX  
 Sumber : Draft-IAX-05 Inter-Asterisk eXchange Version 2

### 2.1.2.3.2 Frame pada IAX

Aliran Data pada IAX adalah seperti pada Gambar 2.8, dimana terdapat mini frame dan full frame. Pada setiap aliran terdapat *mini frame* (M) yang didalamnya

terdapat 4 *byte header* untuk efisiensi *bandwidth*. Kemudian secara periodik juga terdapat *Full Frame* (F) yang memuat informasi sinkronisasi.

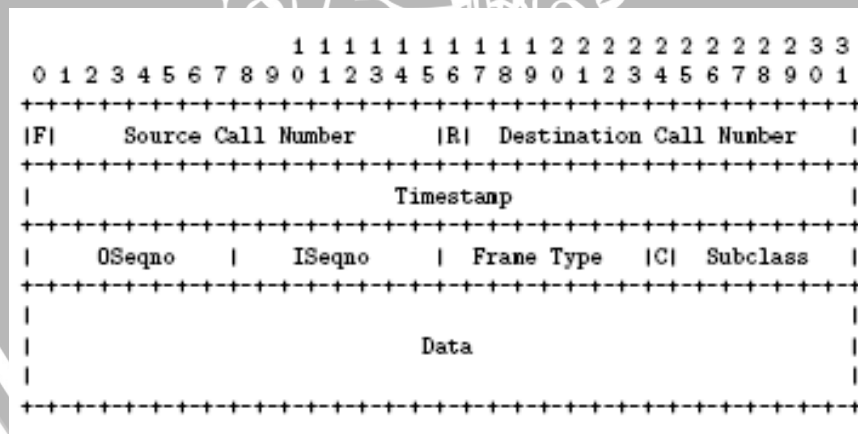


Gambar 2.8 : Media Flow IAX.

Sumber: *IAX Protocol Description*, [www.pdf-search-engine.com](http://www.pdf-search-engine.com)

Pesan-pesan dalam IAX disebut sebagai *frame*. Sebuah bit dalam *F frame* digunakan untuk mengidentifikasi apakah *frame* tersebut *full frame* atau bukan. *Call number* menggunakan 15 bit *unsigned integer*, yang digunakan untuk mengetahui tujuan *host*.

Sebuah *full frame* dapat digunakan untuk mengirim sinyal, *audio*, atau informasi *video*. *Full frame* adalah *frame* yang bersifat *reliably*, sehingga *host* penerima harus mengirimkan pesan yang menyatakan bahwa *host* tersebut sudah menerima pesan. Dalam Gambar 2.7 dijelaskan, setelah menerima pesan *NEW*, maka *host* penerima harus mengirimkan pesan *ACCEPT* secepatnya.



Gambar 2.9 Format binary dari full frame

Sumber: *IAX Protocol Description*, [www.pdf-search-engine.com](http://www.pdf-search-engine.com)





Tabel 2.2 Penjelasan *full frame*

Field	Description
F	Set to the value 1 indicating that this is a Full Frame
Source Call Number	Call number of the transmitting side of the Full Frame
R	Set to the value 1 if this frame is being retransmitted and the value 0 for the initial transmission
Destination Call Number	Call number of the receiving side of the Full Frame
Timestamp	Full 32-bit timestamp
OSeqno	Outbound stream sequence number
ISeqno	Inbound stream sequence number
Frame Type	Frame type
C	Subclass value format
Subclass	Subclass

Sumber: IAX Protocol Description, [www.pdf-search-engine.com](http://www.pdf-search-engine.com)

*Mini frame* digunakan untuk mengirimkan media dengan *overhead* protokol seminimal mungkin.

Tabel 2.3 penjelasan *mini frame*

Field	Description
F	Set to the value 0 indicating that this is not a Full Frame
Source Call Number	Call number of the transmitting side of the Mini Frame
Timestamp	16-bit timestamp

Sumber: IAX Protocol Description, [www.pdf-search-engine.com](http://www.pdf-search-engine.com)

### 2.1.3 Voice Coding VoIP

ITU-T (*International Telecommunication Union - Telecommunication Sector*) membuat beberapa standar untuk *voice coding* yang direkomendasikan untuk implementasi VoIP. Beberapa standar yang sering dikenal antara lain:

#### 2.2.3.1 G.711

G.711 adalah suatu standar internasional untuk kompresi *audio* dengan menggunakan teknik *Pulse Code Modulation (PCM)* dalam pengiriman suara. Standar ini banyak digunakan oleh operator telekomunikasi sebagai standar dalam pengkodean suara analog menjadi digital.

Sebuah kanal video yang baik tanpa di kompresi akan memerlukan *bandwidth* sekitar 9Mbps. Sebuah kanal suara (*audio*) yang baik tanpa di kompresi memerlukan *bandwidth* sekitar 64Kbps. Dengan adanya teknik kompresi, jaringan

dapat menghemat sebuah kanal video menjadi sekitar 30Kbps dan kanal suara menjadi 6Kbps (*half-duplex*), sehingga sebuah saluran Internet yang tidak cepat dapat digunakan untuk menyalurkan *video* dan *audio* sekaligus. Namun, untuk dibutuhkan konferensi dua arah dibutuhkan *bandwidth* sebesar dua kali, sehingga minimal harus menggunakan kanal 64Kbps ke Internet. Pengiriman Suara / *audio* menggunakan *bandwidth* lebih sedikit di banding pengiriman gambar / *video*.

PCM mengkonversikan sinyal analog ke bentuk digital dengan melakukan sampling sinyal analog tersebut 8000 kali/detik dan dikodekan dalam kode angka. Jarak antar sampel adalah  $125 \mu$  detik. Sinyal analog pada suatu percakapan diasumsikan berfrekuensi 300 Hz - 3400 Hz. Sinyal tersampel lalu dikonversikan ke bentuk diskrit. Sinyal diskrit ini direpresentasikan dengan kode yang disesuaikan dengan amplitudo dari sinyal sampel. Format PCM menggunakan 8 bit untuk penkodeannya. Laju transmisi diperoleh dengan mengkalikan 8000 sampel /detik dengan 8 bit/sampel, menghasilkan 64.000 bit/detik .

#### 2.2.3.2 G.723.1

Pengkode sinyal suara G.723.1 adalah jenis pengkode suara yang direkomendasikan untuk terminal multimedia dengan bit rate rendah. G.723.1 memiliki *dual rate speech coder* yang dapat di-switch pada batas 5.3 kbit/s dan 6.3 kbit/s. Dengan memiliki *dual rate speech coder* ini maka G.723.1 memiliki fleksibilitas dalam beradaptasi terhadap informasi yang dimiliki oleh sinyal suara. G.723.1 dilengkapi dengan fasilitas untuk memperbaiki sinyal suara hasil sintesis.

Rate yang lebih tinggi menghasilkan kualitas yang lebih baik. Masukan bagi G.723.1 adalah sinyal suara digital yang di-*sampling* dengan frekuensi sampling 8.000 Hz dan dikuantisasi dengan PCM 16 bit. *Delay* pemrosesannya sangat ditentukan oleh prosesor yang mengerjakan perhitungan-perhitungan pada algoritma G.723.1. Dengan menggunakan DSP prosesor maka *delay* pemrosesan dapat diperkecil. Selain itu kompresi data suara yang direkomendasikan ITU adalah G.726, merupakan teknik pengkodean suara ADPCM dengan hasil pengkodean pada 40, 32, 24, dan 16 kbps. Biasanya juga digunakan pada pengiriman paket data pada telepon publik maupun peralatan PBX yang mendukung ADPCM. G.728, merupakan teknik pengkodean suara CELP dengan hasil pengkodean 16 kbps. G.729 merupakan pengkodean suara jenis CELP dengan hasil kompresi pada 8kbps.



Tabel 2.4 menunjukkan perbandingan beberapa teknik kompresi standar ITU-T.

Tabel 2.4 Perbandingan *Codec*

Codec Information				Bandwidth Calculations					
Codec & Bit Rate (Kbps)	Codec Sample Size (Bytes)	Codec Sample Interval (ms)	Mean Opinion Score (MOS)	Voice Payload Size (Bytes)	Voice Payload Size (ms)	Packets Per Second (PPS)	Bandwidth MP or FRF.12 (Kbps)	Bandwidth w/cRTP MP or FRF.12 (Kbps)	Bandwidth Ethernet (Kbps)
G.711 (64 Kbps)	80 Bytes	10 ms	4.1	160 Bytes	20 ms	50	82.8 Kbps	67.6 Kbps	87.2 Kbps
G.729 (8 Kbps)	10 Bytes	10 ms	3.92	20 Bytes	20 ms	50	26.8 Kbps	11.6 Kbps	31.2 Kbps
G.723.1 (6.3 Kbps)	24 Bytes	30 ms	3.9	24 Bytes	30 ms	34	18.9 Kbps	8.8 Kbps	21.9 Kbps
G.723.1 (5.3 Kbps)	20 Bytes	30 ms	3.8	20 Bytes	30 ms	34	17.9 Kbps	7.7 Kbps	20.8 Kbps
G.726 (32 Kbps)	20 Bytes	5 ms	3.85	80 Bytes	20 ms	50	50.8 Kbps	35.6 Kbps	55.2 Kbps
G.726 (24 Kbps)	15 Bytes	5 ms		60 Bytes	20 ms	50	42.8 Kbps	27.6 Kbps	47.2 Kbps
G.728 (16 Kbps)	10 Bytes	5 ms	3.61	60 Bytes	30 ms	34	28.5 Kbps	18.4 Kbps	31.5 Kbps

Sumber : <http://www.cisco.com/en/US/tech/tk652/tk698>

### 2.1.4 OSI Layer

Dalam arsitektur jaringan komputer, terdapat suatu lapisan ( *layer* ) yang memiliki tugas khusus serta memiliki protokol tersendiri. ISO (*International Standard Organization*) telah mengeluarkan suatu standar untuk arsitektur jaringan komputer yang dikenal dengan nama *Open System Interconnection* ( OSI ) . Standar ini terdiri dari 7 lapisan protokol yang menjalankan fungsi komunikasi antara 2 komputer.

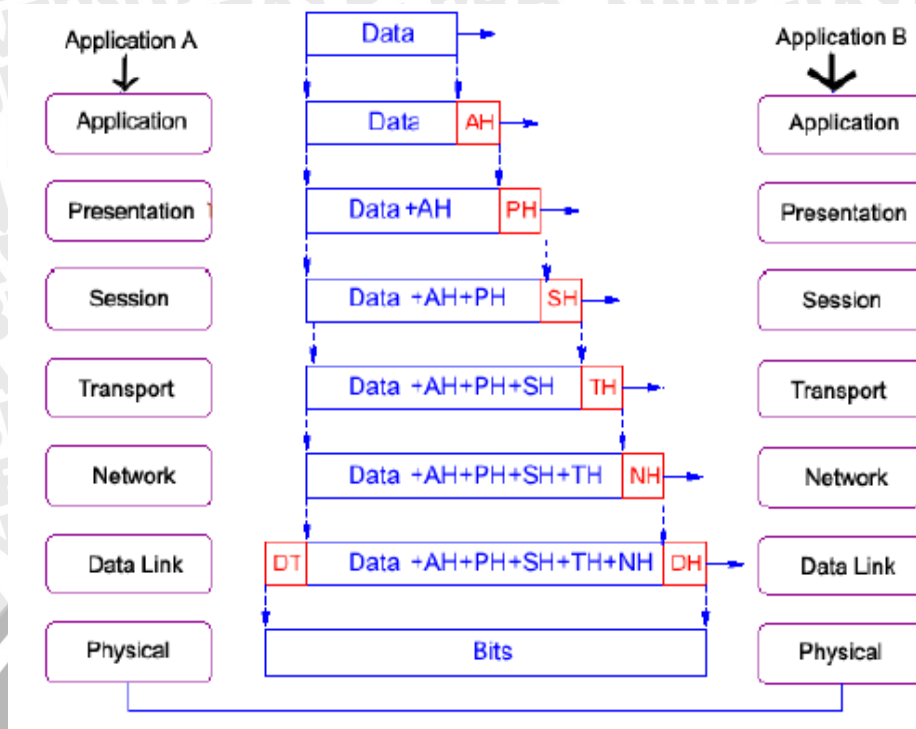
Setiap lapisan memiliki tugas yang berbeda satu sama lain. Masing-masing fungsi dan tugas dari tiap lapisan adalah:

- **Application layer** : menyediakan layanan untuk aplikasi misalnya transfer file, email, akses suatu komputer atau layanan.
- **Presentation layer** : bertanggung jawab untuk menyandikan informasi. Lapisan ini membuat dua host dapat berkomunikasi.

- **Session layer** : membuat sesi untuk proses dan mengakhiri sesi tersebut. Misalnya jika ada *login* secara interaktif maka sesi dimulai dan kemudian jika ada permintaan *log off* maka sesi berakhir. Lapisan ini juga menghubungkan lagi jika sesi login terganggu sehingga terputus.
- **Transport layer** : lapisan ini mengatur pengiriman pesan dari host-host di jaringan. Data dibagi-bagi menjadi paket-paket sebelum pengiriman dan kemudian penerima akan menggabungkan paket-paket tersebut menjadi data utuh kembali. Lapisan ini memastikan bahwa pengiriman data bebas kesalahan dan kehilangan paket data.
- **Network layer** : lapisan bertanggung jawab untuk menerjemahkan alamat logis jaringan ke alamat fisik jaringan. Lapisan ini juga memberi identitas alamat, jalur perjalanan pengiriman data, dan mengatur masalah jaringan misalnya pengiriman paket-paket data.
- **Data Link layer** : lapisan *data link* mengendalikan kesalahan antara dua komputer yang berkomunikasi lewat lapisan *physical*. *Data link* biasanya digunakan oleh Hub dan Switch.
- **Physical layer** : lapisan *physical* mengatur pengiriman data berupa bit lewat kabel. Lapisan ini berkaitan langsung dengan perangkat keras seperti kabel, dan kartu jaringan (*LAN CARD*).





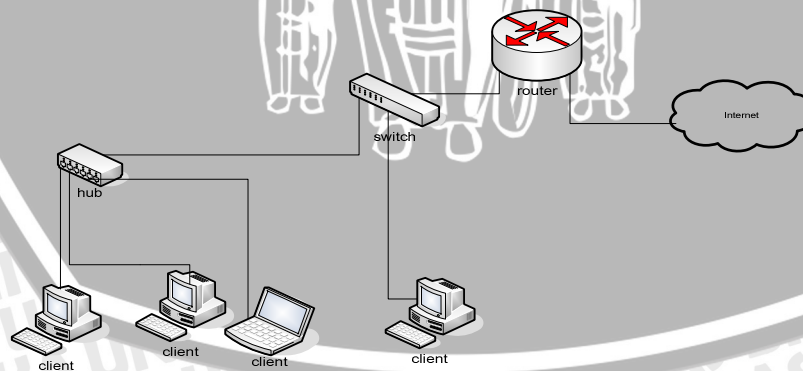


1 Gambar 2.10. / OSI layer  
 2 Sumber : Buku Jaringan Komputer , <http://www.eepis-its.edu/~dphoto/kuliah>

## 2.2 Jaringan Komputer

### 2.2.1 LAN (Local Area Network)

LAN digunakan untuk menghubungkan komputer yang berada di dalam suatu area yang kecil, misalnya di dalam suatu gedung perkantoran atau kampus. Jarak antar komputer yang dihubungkannya dapat mencapai 5 sampai 10 km.

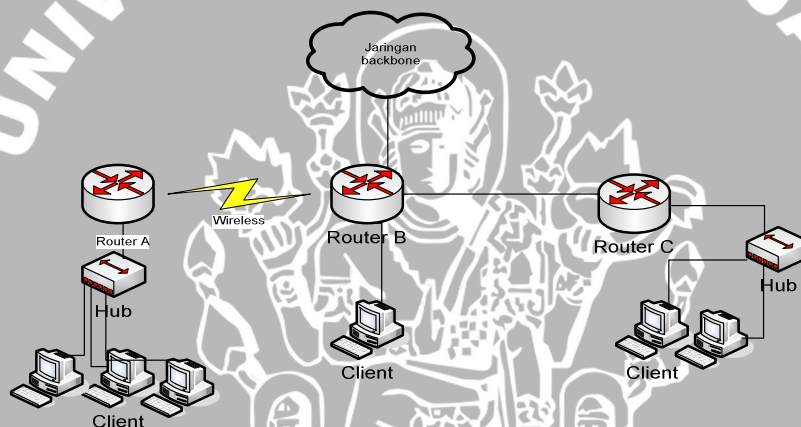


Gambar 2.11 Jaringan LAN  
 Sumber : "Buku Jaringan Komputer" <http://www.eepis-its.edu/~dphoto/kuliah>

Suatu LAN biasanya bekerja pada kecepatan mulai 10 Mbps sampai 100 Mbps. LAN menjadi populer karena memungkinkan banyak pengguna untuk memakai sumber daya secara bersama-sama. Penggunaan LAN misalnya pada *mainframe*, *file server*, *printer*, dan sebagainya.

### 2.2.2 MAN (Metropolitan Area Network)

MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang lokasinya berjauhan. Jangkauan MAN dapat mencapai 10 km sampai beberapa ratus km. MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps tergantung pada area dan kepadatan trafik yang dilalui. Pada Gambar 2.12 dapat dilihat suatu ilustrasi tentang MAN.



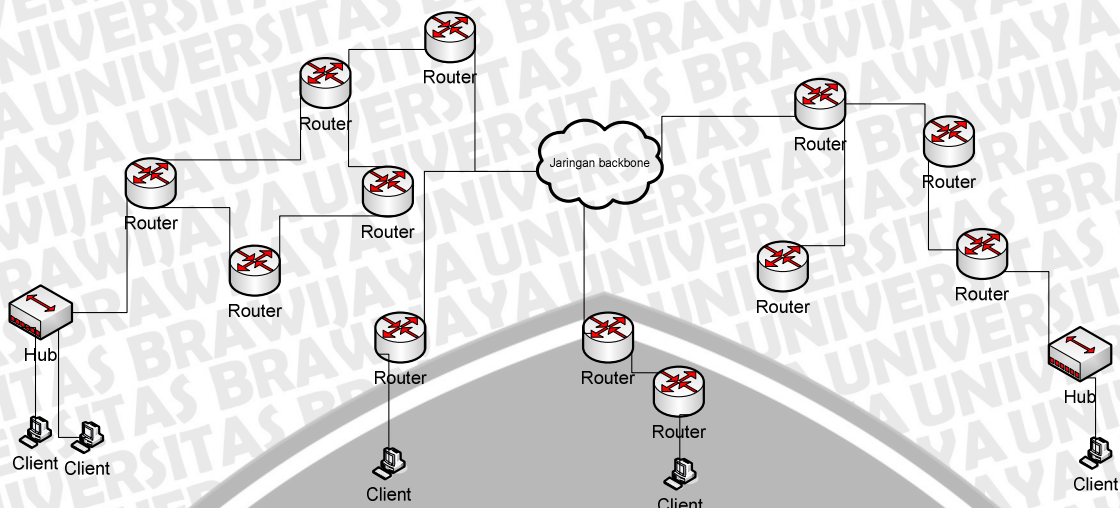
Gambar 2.12. Jaringan MAN

Sumber : "Buku Jaringan Komputer" <http://www.eepis-its.edu/~dphoto/kuliah>

### 2.2.3 WAN (Wide Area Network)

WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari satu kota ke kota lain di dalam suatu negara. Cakupan WAN dapat meliputi 100 km sampai 1.000 km, dan kecepatan antar kota dapat bervariasi antara 1,5 Mbps sampai 2,4 Gbps. Dalam WAN, biaya untuk peralatan transmisi sangat tinggi, dan biasanya jaringan WAN dimiliki dan dioperasikan sebagai suatu jaringan publik. Penggunaan WAN biasanya untuk menghubungkan antara kantor cabang dengan kantor pusat atau kantor cabang yang lain pada daerah yang berjauhan sehingga dapat saling terhubung.





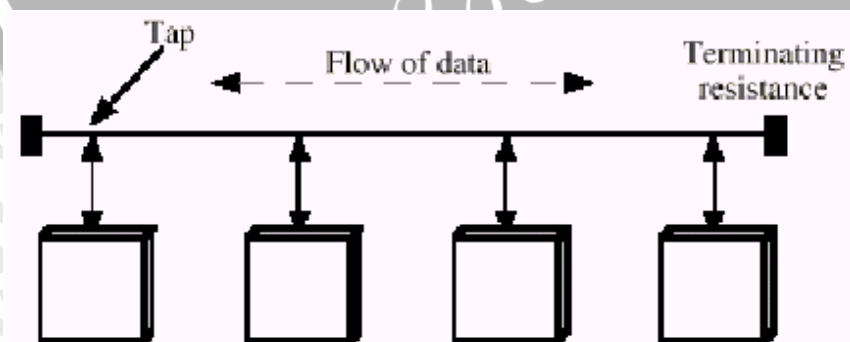
Gambar 2.13 Jaringan WAN

Sumber : “Buku Jaringan Komputer” <http://www.eepis-its.edu/~dphoto/kuliah>

### 2.3 Ethernet 802.3

Ethernet merupakan sebuah teknologi jaringan yang menggunakan metode transmisi *baseband* yang mengirim sinyalnya secara serial 1 bit pada satu waktu. Ethernet beroperasi dalam modus *half-duplex*, yang berarti setiap *station* dapat menerima atau mengirim data tapi tidak dapat melakukan keduanya secara sekaligus. *Fast Ethernet* serta *Gigabit Ethernet* dapat bekerja dalam modus *full-duplex* atau *half-duplex*.

Pada Ethernet, setiap *station* dihubungkan pada sebuah media transmisi yang disebut *ether*. Agar sinyal listrik dapat mengalir, pada ujung-ujung *ether* dipasang suatu *terminator*. Sinyal listrik yang berasal dari satu *station* akan dideteksi oleh semua *station* yang terhubung ke jaringan (*broadcast*). Gambar 2.14 memperlihatkan aliran sinyal listrik yang membawa data pada suatu jaringan berbentuk *bus*.



Gambar 2.14. Aliran Data pada Jaringan Berbentuk Bus

Sumber : “Buku Jaringan Komputer” <http://www.eepis-its.edu/~dphoto/kuliah>

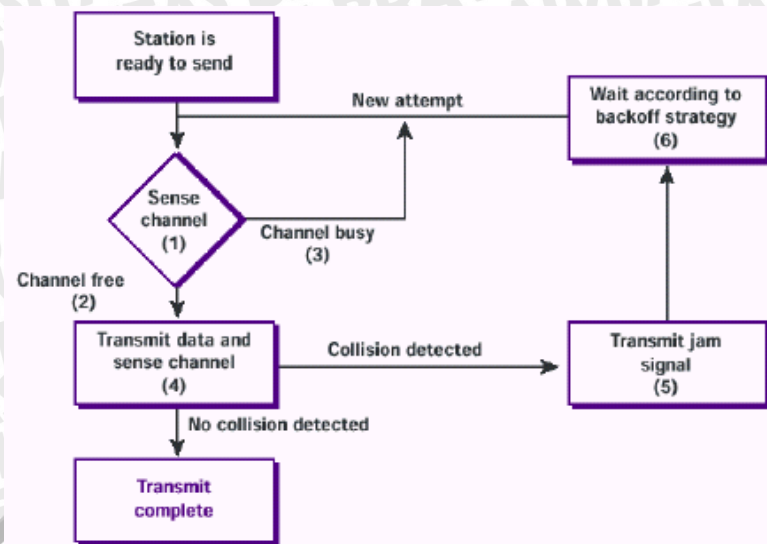
IEEE mengadopsi Ethernet dan kemudian distandarkan dengan nomor IEEE802.3 dengan nama *CSMA/CD bus*. LAN IEEE802.3 ini memiliki kecepatan 10 Mbps, dan menggunakan metoda akses CSMA/CD. Karena alasan sejarah maupun kebiasaan, LAN IEEE802.3 ini dikenal sebagai LAN Ethernet.

*Medium Access Control (MAC)* yang digunakan pada IEEE802.3 adalah *CSMA/CD (Carrier Sense Multiple Access with Collision Detection)*. Prosedur pada *MAC CSMA/CD* adalah sebagai berikut :

1. Suatu *station* yang akan mengirimkan data, melakukan proses *sensing* pada medium transmisi pada jaringan.
2. Jika terasa ada *carrier* (sinyal listrik yang sedang mengalir di media transmisi), hal ini berarti ada *station* lain yang sedang menggunakan jaringan, sehingga pengiriman akan ditunda.
3. Bila tidak terasa ada *carrier*, maka pengiriman data akan dilakukan.
4. Langkah 1 dilakukan untuk mencegah terjadinya pengiriman data yang dilakukan oleh lebih dari satu *station* sehingga tabrakan antar data dapat dicegah. Meskipun demikian, bisa jadi bahwa lebih dari satu *station* yang akan mengirimkan data merasakan jaringan sedang *idle*. Akibatnya *station* tersebut mengirimkan data pada saat yang bersamaan sehingga tabrakan akan terjadi. Oleh karena itu, *MAC CSMA/CD* dilengkapi oleh kemampuan mendeteksi adanya tabrakan. Apabila tabrakan terjadi maka *station* yang pertama kali merasakan adanya tabrakan akan menghentikan pengiriman data sekaligus memberitahukan kondisi tabrakan ke *station* lain yang terlibat tabrakan dengan mengirimkan *jamming sequence*. Apabila ini terjadi, maka semua *station* akan menghentikan pengiriman data, kemudian menunggu kesempatan mengirimkan kembali data pada selang waktu yang acak (*backoff time*).

Secara lengkap, Gambar 2.15 memperlihatkan diagram alir *MAC CSMA/CD*.





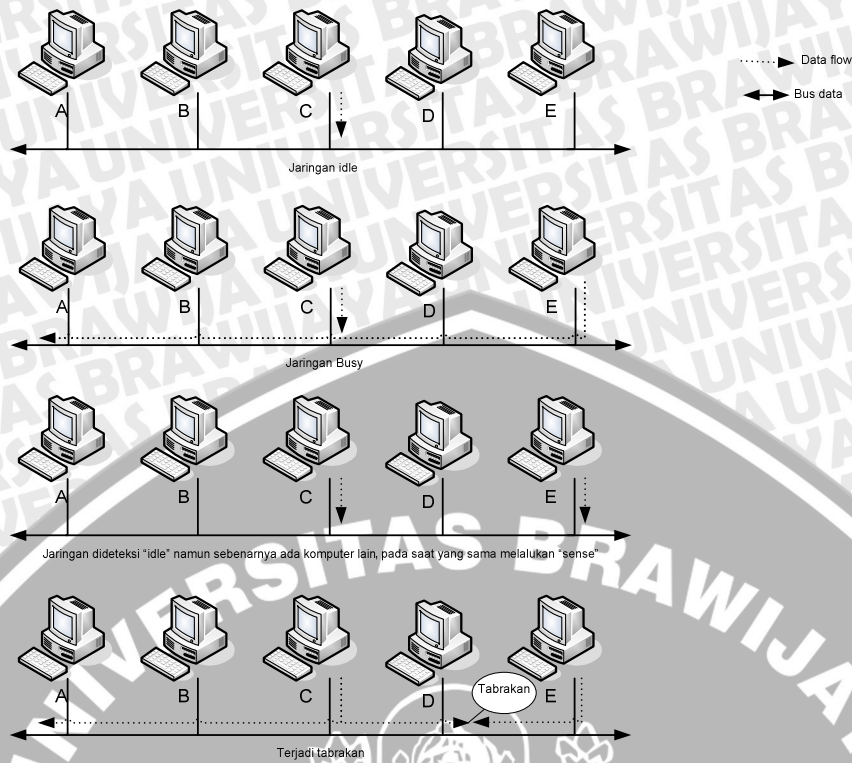
Gambar 2.15 MAC CSMA/CD

Sumber : "Buku Jaringan Komputer" <http://www.eepis-its.edu/~dphoto/kuliah>

Gambar 2.16 ini memperlihatkan tabrakan yang dapat terjadi pada CSMA/CD bus. Diasumsikan *station* A dan C akan mengirimkan data. *Station* A merupakan *station* yang pertama kali mengirimkan data setelah melalui proses *sensing* jaringan ( $t_0$ ). Kemudian *station* C akan mengirimkan data juga sehingga melakukan proses *mendeteksi* jaringan. Pada saat ini, sinyal yang berasal dari A masih berpropagasi dan belum mencapai lokasi *station* C. Akibatnya *station* C akan mendeteksi bahwa medium idle. Oleh karena itu *station* C langsung mengirimkan data ( $t_1$ ).

Pada waktu  $t_2$  terjadi tabrakan antara data yang berasal dari A dengan data yang berasal dari C. C akan terlebih dahulu mendeteksi adanya tabrakan sehingga akan menghentikan pengiriman data kemudian mengirimkan *jamming sequence*. Sedangkan A baru merasakan terjadinya tabrakan pada saat  $t_3$ .

Ukuran jaringan akan mempengaruhi kemampuan suatu *station* untuk mendeteksi tabrakan. Tabrakan masih dapat terdeteksi bila suatu *station* belum selesai mengirimkan data (*station* A masih dapat mendeteksi tabrakan karena belum selesai mengirimkan data)



8 Gambar 2.16. Peristiwa Tabrakan dalam IEEE 802.3  
 9 Sumber : "Buku Jaringan Komputer" <http://www.eepis-its.edu/~dphoto/kuliah>

Agar deteksi tabrakan dapat dilakukan maka ukuran jaringan maksimum (jarak maksimum antar *interface* IEEE802.3) adalah 500 m dengan menggunakan empat *repeater*, sedangkan ukuran *frame* minimum adalah 64 bytes (512 bit).

Format *frame* IEEE802.3 dapat dilihat pada Gambar 2.17

7	1	2 atau 6	2 atau 6	2	≤ 1500	0-46	4	Bytes
Preamble	Start of frame delimiter	Destination Address	Source Address	Length of data	Data	Pad	FCS	

- Keterangan :
- preamble* : untuk sinkronisasi
  - start of frame delimiter* : menunjukkan awal suatu *frame*
  - destination address* : alamat tujuan
  - source address* : alamat pengirim
  - length of data* : menunjukkan panjang data
  - data* : berisi informasi
  - pad* : digunakan bila ukuran *frame* kurang dari 512 bit
  - FCS : untuk error detection

Gambar 2.17 Format *frame* IEEE 802.3  
 Sumber : "Buku Jaringan Komputer" <http://www.eepis-its.edu/~dphoto/kuliah>





Terdapat dua jenis kabel yang digunakan pada fast ethernet, yaitu *twisted pair* dan serat optik *multimode*. Tabel 2.5 memperlihatkan spesifikasi beberapa jenis dapat diimplementasikan menggunakan UTP dengan kategori 3 atau lebih. Sedangkan 100 Base TX hanya dapat diimplementasikan menggunakan UTP Category 5. 100 Base-T4 dan 100 Base T-2 dapat pula diimplementasikan menggunakan UTP Category 3.

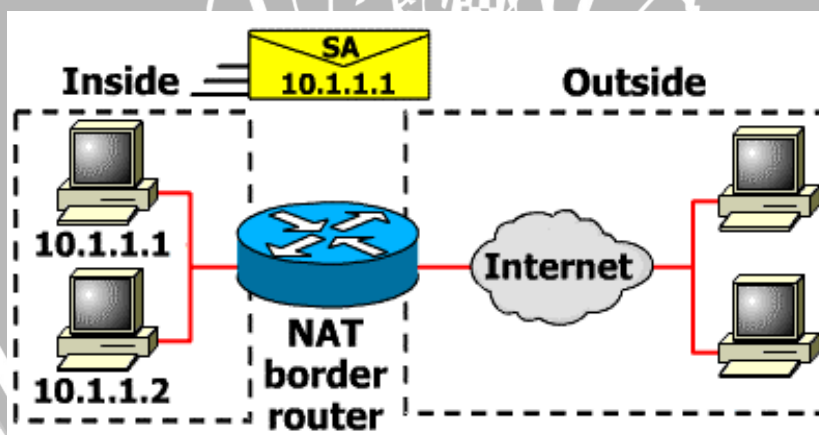
Tabel 2.5 Spesifikasi beberapa jenis kabel *twisted pair*

Frekuensi (MHz)	Redaman (dB per 100 m)			Redaman NEXT (dB)		
	Category 3 UTP	Category 5 UTP	STP 150 ohm	Category 3 UTP	Category 5 UTP	STP 150 ohm
1	2,6	2,0	1,1	41	62	58
4	5,6	4,1	2,2	32	53	58
16	13,1	8,2	4,4	23	44	50,4
25	-	10,4	6,2	-	32	47,5
100	-	22,0	12,3	-	-	38,5
300	-	-	21,4	-	-	31,3

Sumber : "Buku Jaringan Komputer" <http://www.eepis-its.edu/~dphoto/kuliah>

Besar *header* untuk melewati seluruh paket termasuk paket IAX adalah sebesar 34 byte/paket yang terdiri dari *header ethernet* 14 bytes dan *header IP* 20 bytes.

#### 2.4 NAT (*Network Address Translation*)



Gambar 2.18 Peletakan NAT dalam Jaringan

Sumber : *Network Address Translation (NAT)*, [www.infoteknologi.com](http://www.infoteknologi.com)

Keterbatasan alamat IPv.4 merupakan masalah pada jaringan global atau internet. Untuk memaksimalkan penggunaan alamat IP yang diberikan oleh *Internet Service Provider (ISP)* dapat digunakan *Network Address Translation* atau NAT. Cisco mengimplementasikan dengan menggunakan RFC 1631. NAT membuat jaringan yang menggunakan alamat lokal (*private*), alamat yang tidak boleh ada dalam tabel routing

internet dan dikhususkan untuk jaringan lokal/intranet, dapat berkomunikasi ke internet dengan cara memakai alamat IP Internet yang dialokasikan oleh ISP.

Dua tipe NAT adalah *static* dan dinamik yang keduanya dapat digunakan secara terpisah maupun bersamaan. Translasi *static* terjadi ketika sebuah alamat lokal (*inside*) di petakan ke sebuah alamat global/internet (*outside*). Alamat lokal dan global dipetakan satu lawan satu secara statis.

Translasi dinamik terjadi ketika *router* NAT diset untuk memahami alamat lokal yang harus ditranslasikan, dan kelompok (*pool*) alamat global yang akan digunakan untuk terhubung ke internet. Proses NAT Dinamik ini dapat memetakan beberapa kelompok alamat lokal ke beberapa kelompok alamat global.

Sejumlah IP lokal/internal dapat ditranslasikan ke satu alamat IP *global/outside*. Hal ini sangat menghemat penggunaan alokasi IP dari ISP. *Sharing*/pemakaian bersama satu alamat IP ini menggunakan metode *port multiplexing*, atau perubahan *port* ke *packet outbound*.

Teknologi NAT memungkinkan alamat IP lokal/*private* terhubung ke jaringan publik seperti internet. Sebuah *router* NAT ditempatkan antara jaringan lokal (*inside network*) dan jaringan publik (*outside network*), dan mentranslasikan alamat lokal/internal menjadi alamat IP global yang unik sebelum mengirimkan paket ke jaringan luar seperti internet.

NAT memungkinkan menambah alamat IP, tanpa merubah alamat IP pada *host* atau komputer Dengan demikian akan menghilangkan duplikat IP tanpa pengalamatan kembali *host* atau komputer .

Tabel 2.6 Keuntungan dan Kerugian Menggunakan NAT

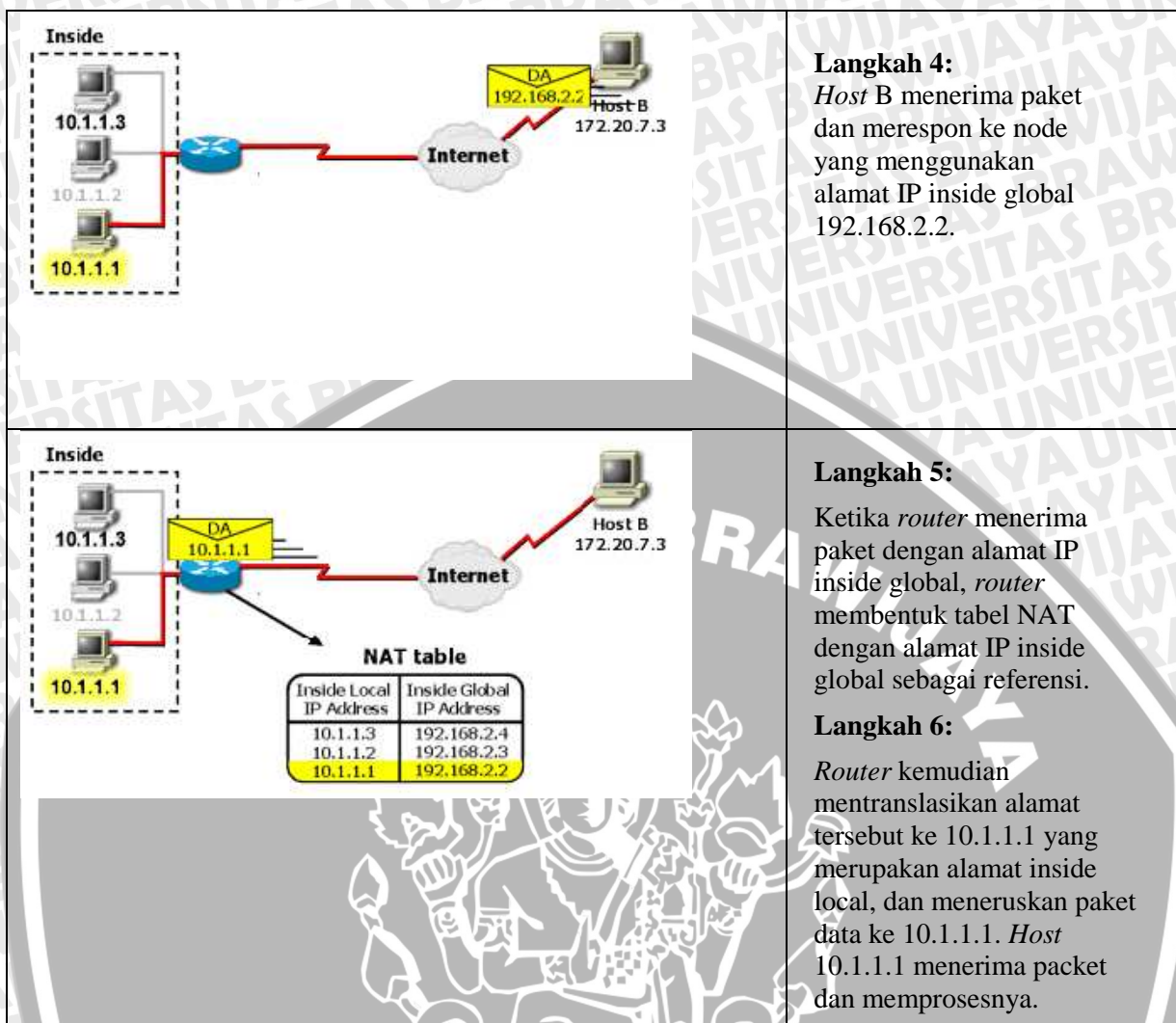
Keuntungan	Kerugian
Menghemat alamat IP legal (ditetapkan oleh NIC atau service provider)	Translasi menimbulkan delay switching.
Mengurangi terjadinya duplicate alamat jaringan IP.	Menghilangkan kemampuan 'trace' (traceability) end-to-end IP.
Meningkatkan fleksibilitas untuk koneksi ke Internet	Aplikasi tertentu tidak dapat berjalan jika menggunakan NAT.

Sumber : Network Address Translation (NAT), [www.infoteknologi.com](http://www.infoteknologi.com)



Tabel 2.7 Cara Kerja NAT

Gambar	Keterangan								
	<p><b>Langkah 1:</b> Pengguna pada <i>host</i> 10.1.1.1 membuat koneksi ke <i>outside host</i> B.</p>								
<table border="1" data-bbox="587 981 874 1108"> <thead> <tr> <th>Inside Local IP Address</th> <th>Inside Global IP Address</th> </tr> </thead> <tbody> <tr> <td>10.1.1.3</td> <td>192.168.2.4</td> </tr> <tr> <td>10.1.1.2</td> <td>192.168.2.3</td> </tr> <tr> <td>10.1.1.1</td> <td>192.168.2.2</td> </tr> </tbody> </table>	Inside Local IP Address	Inside Global IP Address	10.1.1.3	192.168.2.4	10.1.1.2	192.168.2.3	10.1.1.1	192.168.2.2	<p><b>Langkah 2:</b> Paket pertama yang diterima oleh <i>router</i> dari <i>host</i> 10.1.1.1 dicocokkan dengan tabel NAT. Jika translasi ada, (karena sudah diset statik), maka <i>router</i> melanjutkan ke Langkah 3.  Jika translasi tidak ditemukan. <i>Router</i> mengalokasikan sebuah alamat baru dan menset sebuah translasi dari <i>inside local</i> 10.1.1.1 ke sebuah alamat legal <i>inside global</i> dari kelompok (<i>pool</i>) alamat dinamik.</p>
Inside Local IP Address	Inside Global IP Address								
10.1.1.3	192.168.2.4								
10.1.1.2	192.168.2.3								
10.1.1.1	192.168.2.2								
<table border="1" data-bbox="587 1518 874 1646"> <thead> <tr> <th>Inside Local IP Address</th> <th>Inside Global IP Address</th> </tr> </thead> <tbody> <tr> <td>10.1.1.3</td> <td>192.168.2.4</td> </tr> <tr> <td>10.1.1.2</td> <td>192.168.2.3</td> </tr> <tr> <td>10.1.1.1</td> <td>192.168.2.2</td> </tr> </tbody> </table>	Inside Local IP Address	Inside Global IP Address	10.1.1.3	192.168.2.4	10.1.1.2	192.168.2.3	10.1.1.1	192.168.2.2	<p><b>Langkah 3:</b> <i>Router</i> mengganti alamat 10.1.1.1 <i>inside local</i> IP dengan alamat <i>inside global</i> yang telah dipilih, 192.168.2.2, dan meneruskan paket data.</p>
Inside Local IP Address	Inside Global IP Address								
10.1.1.3	192.168.2.4								
10.1.1.2	192.168.2.3								
10.1.1.1	192.168.2.2								

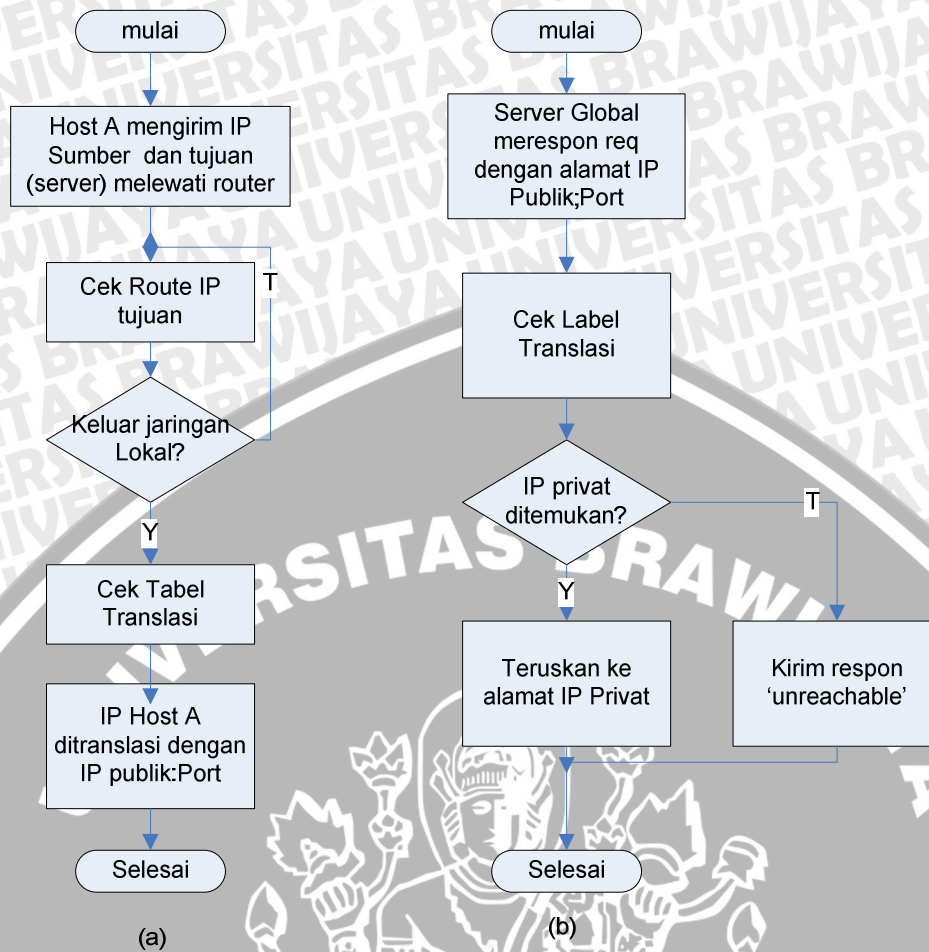


Sumber : Network Address Translation (NAT), [www.infoteknologi.com](http://www.infoteknologi.com)

Dalam flowchart gambar 2.19 dijelaskan proses pembentukan IP *privat* menjadi IP publik dan sebaliknya untuk akses koneksi internet yang memiliki keterbatasan IP publik. Dalam contoh tersebut, *host* A ingin berkomunikasi dengan *server* yang berada dalam jaringan global.

Pada Gambar 2.19(a), dijelaskan mengenai proses pembentukan IP publik. *Host* A hanya dapat berkomunikasi dengan *server* global bila memiliki IP publik karena IP privat tidak memiliki *route*, sehingga tidak dapat dikenali dalam jaringan global. *Host* A mengirimkan IP privat kepada *router* yang juga bertindak sebagai NAT, IP privat ini diubah menjadi IP publik dengan tambahan nomor *port*, sesuai dengan tabel translasi yang tersedia.





Gambar 2.19 (a) Flowchart Mekanisme NAT, IP Privat Menjadi IP Publik  
 (b) Flowchart Mekanisme NAT, IP publik Menjadi IP Privat

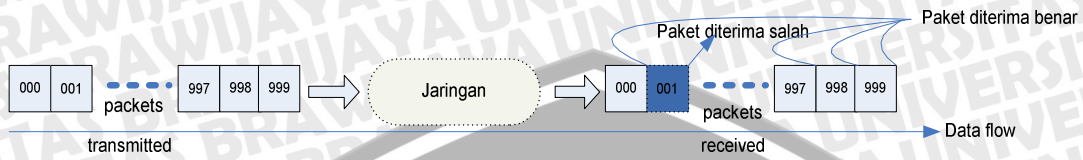
Setelah memakai IP publik, *route* dapat dikenali *router* global untuk kemudian diteruskan ke *server*. *Route* IP dari *server* menuju *host* A juga mengalami proses translasi seperti terlihat pada Gambar 2.19 (b). *Router* yang juga bertindak sebagai NAT mengubah alamat IP publik dan nomor *port* menjadi IP privat sesuai dengan tabel translasi.

### 2.5 Parameter Performansi Jaringan

Performansi jaringan merupakan kumpulan dari berbagai besaran teknis yang merujuk pada tingkat kecepatan dan keandalan penyampaian berbagai jenis beban data di dalam suatu sistem komunikasi. Performansi merupakan kumpulan berbagai besaran teknis.

### 2.5.1 Throughput

*Throughput* didefinisikan sebagai rata-rata paket data yang diterima benar yang dapat ditransmisikan pada satu waktu sama. *Throughput* pada jaringan komputer diukur dengan melihat jumlah paket data per detik.



Gambar 2.20 Visualisasi Keadaan Paket pada Suatu Jaringan  
 Sumber: IAX Protocol Description, www.pdf-search-engine.com

Persamaan *throughput* menurut rumus, dapat ditulis (Mischa Schwartz, 1987 : 129):

$$\lambda = \frac{1}{t_v} = \frac{(1-\rho)}{t_l [1 + (\alpha - 1)\rho]} \quad (2.1)$$

Dengan;

- $\lambda$  = *throughput* (paket/s)
- $t_v$  = waktu rata-rata untuk mentransmisikan 1 paket yang benar (s)
- $= \frac{T_{total}}{N_{Paket}}$
- $t_l$  = waktu yang dibutuhkan untuk mentransmisikan 1 paket (s)
- $\rho$  = probabilitas *bit error* yang diterima
- $\alpha$  = konstanta =  $1 + (t_{out}/t_l)$
- $t_{out} = t_{prop} + 2t_l$

Kecepatan transmisi rata-rata yang diterima merupakan jumlah bit yang dapat diterima di *client* setiap detik pada jaringan *intranet*. Kecepatan transmisi rata-rata ditentukan dengan Persamaan 2-2 (Mischa Schwartz, 1987 : 132):

$$D = \lambda l x 8 = \frac{(1-\rho) l x 8}{t_l [1 + (\alpha - 1)\rho]} \quad (2-2)$$

dengan :

- $D$  = kecepatan transmisi rata-rata yang diterima (bps)
- $\lambda$  = *throughput* (paket/s)
- $l$  = panjang paket data (*byte*/paket)

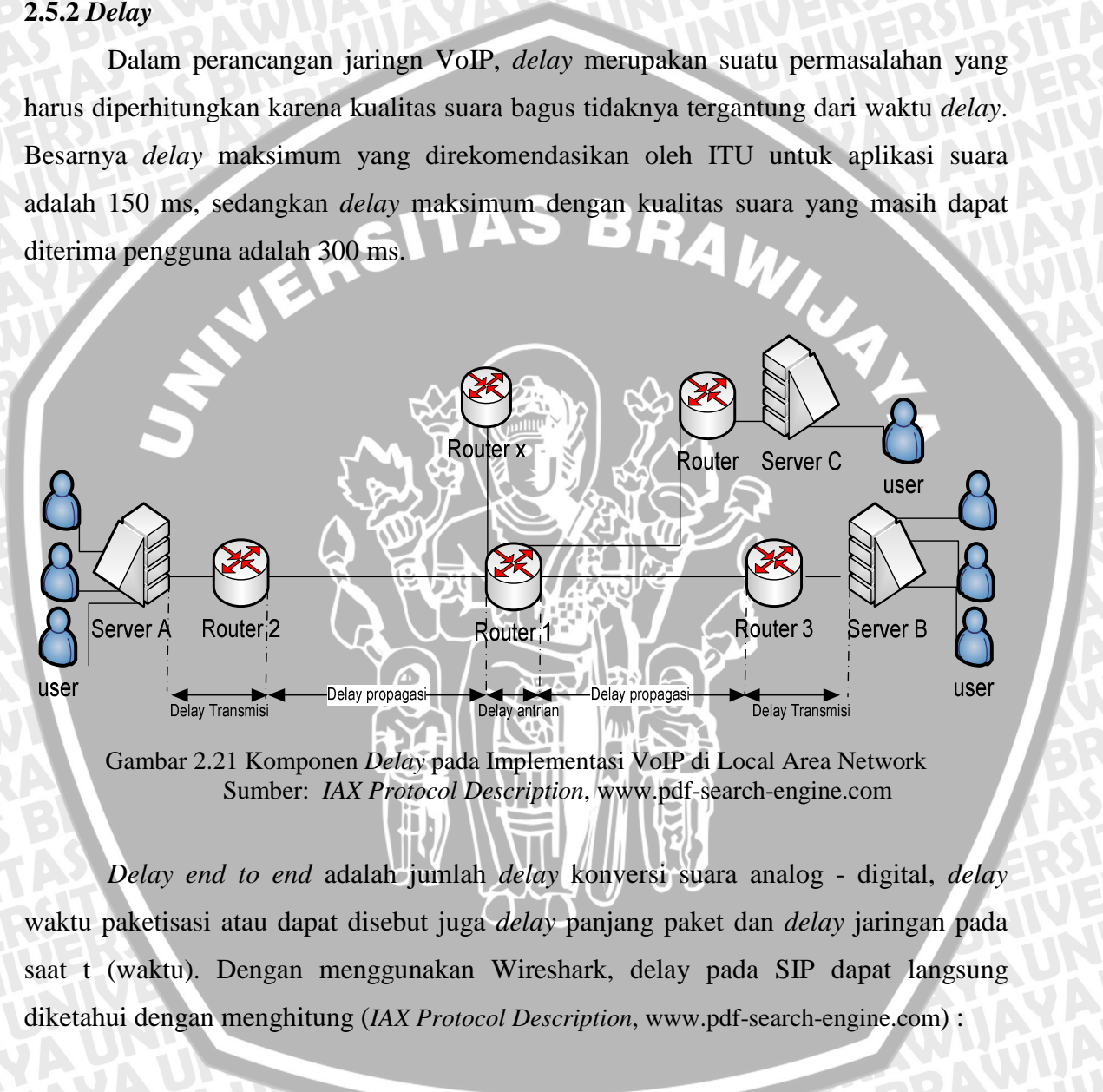




- $t_i$  = waktu yang dibutuhkan untuk mentransmisikan 1 paket (s)
- $p$  = probabilitas kesalahan paket
- $\alpha$  = konstanta transmisi ternormalisasi

### 2.5.2 Delay

Dalam perancangan jaringan VoIP, *delay* merupakan suatu permasalahan yang harus diperhitungkan karena kualitas suara bagus tidaknya tergantung dari waktu *delay*. Besarnya *delay* maksimum yang direkomendasikan oleh ITU untuk aplikasi suara adalah 150 ms, sedangkan *delay* maksimum dengan kualitas suara yang masih dapat diterima pengguna adalah 300 ms.



Gambar 2.21 Komponen *Delay* pada Implementasi VoIP di Local Area Network  
 Sumber: *IAX Protocol Description*, www.pdf-search-engine.com

*Delay end to end* adalah jumlah *delay* konversi suara analog - digital, *delay* waktu paketisasi atau dapat disebut juga *delay* panjang paket dan *delay* jaringan pada saat  $t$  (waktu). Dengan menggunakan Wireshark, *delay* pada SIP dapat langsung diketahui dengan menghitung (*IAX Protocol Description*, www.pdf-search-engine.com) :

$$t_{end\_to\_end}(client\_1 - client\_4) = t_{end\_to\_end}(client\_1 - server) + t_{end\_to\_end}(server - client\_4) \tag{2.3}$$

Pada IAX, paket dianalisa berdasarkan *time capture*, dan pola kedatangannya dengan *time sampling codec* (*IAX Protocol Description*) :

$$T_s = \frac{1}{\Sigma \text{Paket\_per\_sekon}} \tag{2.4}$$

Secara Teori, delay pada jaringan dapat dibedakan dalam beberapa komponen yang berbeda. Total *delay end to end* adalah (IAX Protocol Description, www.pdf-search-engine.com):

$$t_{end\ to\ end} = t_{enc} + t_{trans} + t_p + t_w + t_{dec} \tag{2-5}$$

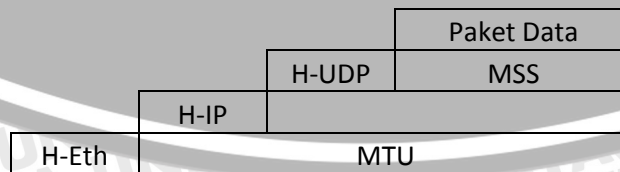
dengan :

- $t_{end\ to\ end}$  = delay end to end
- $t_{enc}$  = delay encapsulasi
- $t_{trans}$  = delay transmisi
- $t_p$  = delay propagasi
- $t_w$  = delay antrian
- $t_{dec}$  = delay dekapsulasi

### 2.5.2.1 Delay Enkapsulasi dan Dekapsulasi

Enkapsulasi adalah proses menambahkan *header* pada paket data, sehingga paket data tersebut dapat tepat sampai ke *host* tujuan. Proses enkapsulasi ditunjukkan dalam gambar 2.22

*Delay* enkapsulasi adalah waktu yang dibutuhkan untuk menambahkan keseluruhan *header* pada sebuah paket data. Sedangkan *delay* dekapsulasi adalah waktu yang dibutuhkan untuk melepaskan keseluruhan *header* dari sebuah paket data.



Gambar 2.22 Proses Enkapsulasi Paket Data

Sumber : [http://www.eepisits.edu/~dphoto/kuliah/Buku\\_Jaringan\\_Komputer.pdf](http://www.eepisits.edu/~dphoto/kuliah/Buku_Jaringan_Komputer.pdf)

Pada saat pembentukan hubungan (*connection established*) antara *host* tujuan dengan *host* sumber terjadi kesepakatan dalam enkapsulasi dan dekapsulasi paket data.





Sesuai dengan draft IAX2, sebuah paket IAX terdiri dari 34 byte untuk *header* IP dan Ethernet, 8 byte sebagai header UDP, serta 4 byte untuk *trunking*. Sehingga dapat dituliskan menjadi :

$$\text{Panjang paket IAX} = \{34 + 8 + (4 + \text{Payload}) \times \text{ncalls}\} \text{ bytes}$$

Sedangkan untuk paket IAX per detik dituliskan dalam persamaan :

$$\text{Besarnya paket IAX} = \text{Panjang paket IAX} \times \text{banyak paket per detik (pps)}$$

34 byte (IP/UDP)	8 byte (UDP Header)	4byte (Trunking)	160 byte(Audio) PAYLOAD
---------------------	---------------------------	------------------	----------------------------

Gambar 2.23 Kebutuhan *Bandwidth* Paket IAX

Sumber: IAX Protocol Description, [www.pdf-search-engine.com](http://www.pdf-search-engine.com)

Apabila *host* sumber ingin mengirim paket data ke *host* tujuan, maka proses yang terjadi adalah paket data tersebut dikirimkan ke *transport layer*. Di *transport layer* dengan menggunakan protokol UDP, paket data dikapsulasi menjadi segmen UDP. Panjang segmen UDP ditentukan dengan persamaan 2-6 (Onno W. Purbo, et al. (2001:24))

$$W_{\text{segmen}} = \text{MSS} + \text{Header}_{\text{UDP}} \quad (2-6)$$

dengan :

$$W_{\text{segmen}} = \text{Panjang segmen UDP (byte/paket)}$$

$$\text{MSS} = \text{Maximal Segment Size (byte/paket)}$$

$$\text{Header}_{\text{UDP}} = 8 \text{ byte/paket}$$

Dari *transport layer*, segmen UDP kemudian diteruskan ke *internet layer*. Di *internet layer* dengan menggunakan protokol IP, segmen UDP dikapsulasi menjadi datagram IP. Panjang datagram IP ditentukan dengan persamaan 2.7 (Onno W. Purbo, et al. (2001:24)):

$$W_{\text{datagram}} = W_{\text{segmen}} + \text{Header}_{\text{IP}} \quad (2-7)$$

dengan :

$$W_{\text{datagram}} = \text{panjang datagram IP (byte/paket)}$$

$$W_{\text{segmen}} = \text{panjang segmen UDP (byte/paket)}$$

$$\text{Header}_{\text{IP}} = 20 \text{ byte/paket}$$

Dari internet *layer* , datagram IP dikirimkan ke *network interface layer*. Di *network interface layer* dengan menggunakan protocol *Ethernet II*, datagram IP dienkapsulasi menjadi *frame Ethernet*. Panjang *frame Ethernet* ditentukan dengan persamaan 2.8 (Onno W. Purbo, et al. (2001:24)):

$$W_{frame} = W_{datagram} + Header_{ethernet} \quad (2-8)$$

dengan :

$W_{frame}$  = panjang *frame Ethernet* (byte/paket)

$W_{datagram}$  = panjang datagram IP (byte/paket)

$Header_{ethernet}$  = 14 byte/paket

*Delay* enkapsulasi ditentukan dengan Persamaan (2-9) (Onno W. Purbo, et al. (2001:24)):

$$t_{enc} = \frac{W_{frame} - MSS}{C_{pros}} \times 8 \quad (2-9)$$

Dengan :

$t_{enc}$  = *delay* enkapsulasi (s)

$W_{frame}$  = panjang *frame Ethernet* (byte/paket)

MSS = *Maximum Segment Size* (byte/paket)

$C_{pros}$  = kecepatan pemrosesan data di *server* (bps)

*Delay* dekapsulasi ditentukan dengan Persamaan (2-10) (Onno W. Purbo, et al. (2001:24)):

$$t_{dec} = \frac{W_{frame} - MSS}{C_{pros}} \times 8 \quad (2-10)$$

Dengan :

$t_{dec}$  = *delay* dekapsulasi (s)

$W_{frame}$  = panjang *frame Ethernet* (byte/paket)

MSS = *Maximum Segment Size* (byte/paket)

$C_{pros}$  = kecepatan pemrosesan data di *client* (bps)



### 2.5.2.2 Delay transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket multimedia ke media transmisi. Dipengaruhi ukuran paket data dan kapasitas kanal intranet. Delay transmisi ditentukan dengan persamaan 2-11

(<http://www.cisco.com/warp/public/788/voip/>)

$$t_{trans} = \frac{(L+L')}{C} \times 8 = \frac{W_{frame}}{C} \times 8 \quad (2-11)$$

dengan :

- $t_{trans}$  = delay transmisi (s)
- L = Panjang paket data (byte/paket)
- L' = panjang header (byte/paket)
- C = kapasitas kanal (bps)
- $W_{frame}$  = panjang frame ethernet (byte/paket)

### 2.5.2.3 Delay Propagasi

Delay propagasi adalah waktu yang dibutuhkan untuk merambatkan paket multimedia melalui media transmisi dari server ke client. Delay propagasi ditentukan dengan persamaan 2-12 (Grinham And Pratt (1995 : 34))

$$t_p = t_{DTE} + t_{UTP} + t_{rou} \quad (2-12)$$

dengan :

- $t_p$  = delay propagasi (s)
- $t_{DTE}$  = delay pada NIC (s)  
= 1 ms
- $t_{UTP}$  = delay pada UTP Cat.5 (s)  
=  $\alpha \times P_{UTP} \times \frac{1}{C}$
- $\alpha$  = faktor kali
- $P_{UTP}$  = Panjang kabel UTP
- $C$  = Kecepatan Transmisi
- $t_{rou}$  = delay pada router  
= 2 ms

komponen *delay fast ethernet* ditunjukkan pada tabel 2.8

Tabel 2.8 Faktor Kali Komponen *Delay Fast Ethernet*

Komponen	Faktor kali per meter
UTP Cat 3	0,57
UTP Cat 4	0,57
UTP Cat 5	0,556
STP	0,556

Sumber : [http://www.eepisits.edu/~dphoto/kuliah/Buku\\_Jaringan\\_Komputer.pdf](http://www.eepisits.edu/~dphoto/kuliah/Buku_Jaringan_Komputer.pdf)

Dengan melihat pada tabel 2.8 maka persamaan 2-12 dapat dituliskan kembali menjadi

$$\begin{aligned}
 t_p &= t_{UTP} + t_{DTE} + t_{rou} \\
 &= \left( 0,556 \times P_{UTP} \times \frac{1}{C} \right) + 0,001 + 0,002 \quad (2-13)
 \end{aligned}$$

dengan :

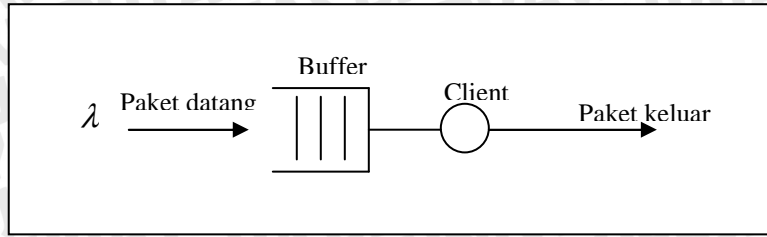
$C$  = kapasitas media transmisi

$P_{UTP}$  = panjang kabel UTP dari *node* sumber (*server*) ke *node* tujuan

#### 2.5.2.4 Delay Antrian

*Delay* antrian adalah waktu tiap paket untuk menunggu *delay* transmisi paket sebelumnya, lamanya waktu tiap paket untuk menunggu *delay* transmisi paket sebelumnya, lamanya waktu ini bergantung pada kecepatan saluran dan keadaan antrian. *Delay* antrian hanya timbul pada suatu interface ke saluran keluaran, dimana trafik yang masuk lebih besar dari /kecepatan saluran keluaran. *Delay* antrian dapat dihitung dengan menggunakan model antrian M/M/1. M pertama menunjukkan distribusi kedatangan *Poisson*, M kedua berarti distribusi waktu pelayanan eksponensial, dan 1 menunjukkan bahwa jumlah *client* adalah tunggal. Disiplin antrian yang digunakan dalam skripsi ini yaitu FIFO (*First In First Out* ), yaitu tiap paket *multimedia* yang datang lebih dulu pada suatu interface jaringan akan ditransmisikan lebih dulu.





Gambar 2.24 Model Antrian *Single Client*

Sumber : *Telecommunication Network : Protocols, Modelling, and Analysis*,

Besarnya *delay* antrian yang terjadi pada *client* yaitu (I Made Wiryana, 1992:2):

$$t_w = t_{queue} + t_{serv} \tag{2-14}$$

dengan:

$t_w$  = *delay* antrian pada *client* (s)

$t_{queue}$  = waktu tunggu paket pada *client* (s)

$t_{serv}$  = waktu rata-rata pelayanan *client* (s)

$$t_{serv} = \frac{1}{\mu} \tag{2-15}$$

dengan :

$t_{serv}$  = waktu rata-rata pelayanan *client* (s)

$\mu$  = kecepatan pelayanan *client* (paket/s)

$$t_{queue} = \frac{\lambda}{\mu(\mu - \lambda)} \tag{2-16}$$

dengan :

$t_{queue}$  = waktu tunggu paket pada *client* (s)

$t_w$  = *delay* antrian (s)

$\lambda$  = kecepatan kedatangan paket pada *client* (paket/s)

$\mu$  = kecepatan pelayanan *client* (paket/s)

Dari Persamaan (2-16) dan (2-17) maka *delay* antrian dapat dituliskan sebagai :

$$t_w = \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} \tag{2-17}$$

dengan :

$t_w$  = *delay* antrian (s)

$\lambda$  = kecepatan kedatangan paket pada *client* (paket/s)

$\mu$  = kecepatan pelayanan *client* (paket/s)



Dengan persamaan-persamaan tersebut, nantinya akan didapatkan waktu *delay* jaringan, laju pesan, jumlah pesan yang dapat dilayani, sehingga dapat menjadi acuan apakah *server* VoIP telah sesuai dengan standar yang ditetapkan.

### 2.5.3 Jitter

*Jitter* didefinisikan sebagai variasi *delay* antara blok-blok yang berurutan. Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya tabrakan dengan demikian nilai *jitter*-nya akan semakin besar. Semakin besar nilai *jitter* akan mengakibatkan nilai QoS akan semakin turun. Untuk mendapatkan nilai QoS jaringan yang baik, nilai *jitter* harus seminimum mungkin. *Jitter* dapat diketahui nilainya dengan mengukur nilai puncaknya, yang dijadikan standar dalam menentukan kualitas jaringan.

Perhitungan *jitter* diberikan pada persamaan (*IAX Protocol Description*, [www.pdf-search-engine.com](http://www.pdf-search-engine.com)):

$$\Delta t_n = t_{end\_to\_end(n+1)} - t_{end\_to\_end(n)} \quad (2-18)$$

### 2.5.4 Packet Loss

Paket *loss* adalah jumlah paket yang hilang dibandingkan dengan paket yang diterima benar. Umumnya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima. Jika terjadi tabrakan yang cukup banyak, *buffer* akan penuh, dan data baru tidak dapat diterima. Paket yang hilang ini harus ditransmisi ulang, yang akan membutuhkan waktu tambahan.

*Packet loss* adalah banyaknya paket multimedia yang hilang saat proses *download* oleh *client*. Pada *streaming*, ketika *bandwidth* multimedia lebih besar daripada kecepatan transmisi di intranet maka akan terjadi kongesti sehingga mengakibatkan beberapa paket hilang. Prosentase *packet loss* maksimum yang diperbolehkan oleh ISO untuk aplikasi multimedia adalah 5 %. Prosentase *packet loss* ditentukan dengan Persamaan 2-19

$$packet\ loss(\%) = \frac{N_{packet\ loss}}{N_{paket} + N_{packet\ loss}} \times 100\% \quad (2-19)$$



dengan :

$N_{\text{packet loss}}$  = jumlah paket multimedia yang hilang

$N_{\text{paket}}$  = jumlah paket multimedia yang diterima dengan benar

#### 2.5.4 Pengukuran Trafik VoIP

Pengukuran trafik diperlukan untuk menentukan jumlah saluran yang sesuai dengan perencanaan kanal VoIP yang tersedia. Kanal VoIP sendiri dibatasi dengan *bandwidth* saluran yang tersedia. Jumlah kanal VoIP yang disediakan mengikuti besarnya *traffic load* dan QOS (*Quality of Service*).

*Traffic load* merupakan perbandingan antara banyaknya panggilan yang datang dengan waktu rata-rata yang digunakan untuk melayani setiap panggilan pada periode waktu tertentu. Lamanya panggilan dibanding dengan jumlah panggilan yang terjadi dalam rentang waktu tersebut, atau disebut sebagai AHT (*Average Hold Time*) dan dipakai dalam menentukan *traffic load*. AHT dituliskan dalam persamaan (Anonim. *VoIP Implementation & Management, 2007*)

$$AHT = \frac{\Sigma \text{call\_seconds}}{\Sigma \text{calls}} \quad (2.20)$$

Dengan :

AHT = *Average Hold Time* (detik/panggilan)

$\Sigma$  call seconds = Total waktu semua panggilan (detik)

$\Sigma$  call = Banyaknya panggilan (panggilan)

Satuan *traffic load* yang digunakan disini adalah erlang. Satu erlang didefinisikan sebagai 3600 detik panggilan dalam *circuit* yang sama atau *traffic load* yang membuat satu *circuit* sibuk selama satu jam.

$$\text{Traffic(erlang)} = \frac{\Sigma \text{calls} \times AHT}{3600} \quad (2.21)$$

Dalam melakukan analisa trafik dalam suatu jaringan, ada empat hal yang harus diperhatikan, yaitu :

a. Pola datangnya panggilan

Terdapat tiga pola datangnya panggilan, yaitu :

- *Smooth Call Arrival Pattern*
- *Peaked Call Arrival Pattern*
- *Random Call Arrival Pattern*

b. *Blocked Calls*

*Blocked calls* adalah panggilan yang tidak langsung dilayani, dalam artian di *routing* ulang ke kelompok *trunk* lain, ditempatkan dalam antrian atau diberi nada pemberitahuan. Terdapat empat tipe *blocked calls*, yaitu :

- *Lost Calls Held (LCH)*
- *Lost Calls Cleared (LCC)*
- *Lost Calls Delayed (LCD)*
- *Lost Calls Retried (LCR)*

c. Banyak sumber (*sources*)

d. *Holding times*

Berdasarkan tabel 2.9 maka pemodelan yang tepat untuk jaringan *VoIP* adalah pemodelan menggunakan Erlang B. Erlang B digunakan ketika *blocked calls* yang di *routing* ulang tidak akan pernah kembali ke kelompok *trunk* asal. Pemodelan ini menggunakan pola datangnya panggilan *random* (acak). Pemanggil hanya akan melakukan satu kali panggilan, jika panggilannya diblokir maka panggilan tersebut akan di *routing* ulang. Tabel 2.9 menunjukkan perbandingan pemodelan trafik :

Tabel 2.9 Perbandingan model trafik

Model trafik	Sources	Pola Kedatangan	Disposisi Blocked Calls	Holding Times
Poisson	Infinite	Random	Held	Exponential
Erlang B	Infinite	Random	Cleared	Exponential
Extended Erlang B	Infinite	Random	Retried	Exponential
Erlang C	Infinite	Random	Delayed	Exponential
Engset	Finite	Smooth	Cleared	Exponential
EART/EARC	Infinite	Peaked	Cleared	Exponential
Neal-Wilkerson	Infinite	Peaked	Held	Exponential
Crommelin	Infinite	Random	Delayed	Constant
Binomial	Finite	Random	Held	Exponential
Delay	Finite	Random	Delayed	Exponential

Sumber : *Implementing Voice Over IP*





Untuk menghitung trafik dengan pemodelan Erlang B digunakan rumus sesuai dengan persamaan 2.22.

Persamaan distribusi Erlang

$$P(n) = \frac{A^n / n!}{\sum_{i=0}^n A^i / i!} \quad (2.22)$$

Dimana :

$n$  = jumlah saluran yang tersedia (kanal/saluran)

$A$  = *Trafik load* (erlang)

Pada saat  $N$  buah saluran diduduki, maka semua panggilan ditolak.  $P(N)$  tidak lain adalah nilai probabilitas dari trafik yang hilang.  $P(N)$  disebut juga sebagai rugi Erlang atau GOS (*Grade Of Service*).

Banyaknya paket yang ditransmisikan per detik (pps) bergantung pada jenis *codec* yang digunakan, pps dapat dihitung dengan menggunakan persamaan 2.23.

$$pps = \frac{BR}{P} \quad (2.23)$$

Dengan :

pps = *packet per second* (pps)

BR = *codec bit rate* (bps)

P = *payload* (bit)

*Bandwidth* transmisi merupakan *bandwidth* yang dibutuhkan paket suara yang ditransmisikan dalam periode satu detik. *Bandwidth* transmisi dapat dihitung dengan menggunakan persamaan 2.24.

$$BW_T = L_{PI} \times pps \quad (2.24)$$

Dengan :

$BW_T$  = *bandwidth transmisi* (bps)

$L_{PI}$  = *panjang paket internet* (bit per paket)

pps = *packet per second* (paket per detik)

Kapasitas jaringan bergantung pada kapasitas transmisi. Kapasitas transmisi merupakan banyaknya circuit yang dapat digunakan secara bersama-sama untuk melakukan proses transmisi data atau banyaknya proses komunikasi yang mungkin terjadi dalam waktu yang sama. Kapasitas transmisi dapat dihitung dengan menggunakan persamaan 2.25 (Anonim. *VoIP Implementation & Management*, 2007)

$$K_T = \frac{BW_{ST}}{BW_T} \quad (2.25)$$

Dengan :

$K_T$  = Kapasitas Transmisi (komunikasi)

$BW_{ST}$  = Bandwidth saluran transmisi (kbps)

$BW_T$  = Bandwidth transmisi (kbps)

Banyaknya user yang menempati jaringan mempengaruhi performansi jaringan tersebut. Semakin banyak user yang mengakses jaringan, utilitas jaringan akan semakin kecil sesuai dengan persamaan 2.26 (Anonim. *VoIP Implementation & Management* 2007)

$$\begin{aligned} \text{Paket loss} &= \frac{\Sigma_{error}}{\Sigma_{diterima}} \times 100\% \\ &= \left(1 - \frac{1}{1 + 3,44 \frac{dx \Sigma_{diterima}}{LV}}\right) \times 100\% \end{aligned} \quad (2.26)$$

Dengan :

$\Sigma_{error}$  = Jumlah paket yang rusak (paket)

$\Sigma_{diterima}$  = Jumlah paket total semua user (paket)

$d$  = Jarak antar 2 user (meter)

$V$  = Kecepatan rambat sinyal ( $3 \times 10^8$  meter)

$L$  = Panjang paket rata-rata (byte)



## BAB III METODOLOGI

### 3.1 Studi Literatur

Mempelajari teori tentang jaringan VoIP menggunakan protokol IAX, jaringan komputer, trafik *voice*, serta parameter-parameter yang digunakan dalam pengukuran performansi jaringan VoIP yang diperoleh dari buku panduan kuliah, sumber lain berupa buku literatur, RFC (*Request For Comment*) maupun *ebook*.

### 3.2 Pengambilan Data

Kualitas suara VoIP yang diterima sangat dipengaruhi oleh, *delay*, *jitter*, dan *packet loss*. Pada skripsi ini, ketiga faktor tersebut akan di uji dan dianalisis untuk mendapatkan hubungan dan pengaruh ketiganya dalam jaringan VoIP berbasis protokol IAX pada LAN yang melalui NAT, untuk dibandingkan dengan protokol VoIP lain, dalam hal ini SIP.

Data yang diambil dalam skripsi ini meliputi data primer dan data sekunder.

#### 3.2.1 Pengambilan Data Primer

Pengambilan data primer diperoleh dari data pada perencanaan yang akan digunakan. Perencanaan meliputi :

- Topologi jaringan berupa panjang jaringan
- Jenis kabel yang digunakan adalah kabel UTP
- *Codec* yang digunakan adalah G.711.

Pengambilan data performansi menggunakan software capture *unsniff* dan *wireshark*.

#### 3.2.2 Pengambilan Data Sekunder

Data primer belum cukup untuk menentukan performansi jaringan VoIP berbasis protokol IAX, oleh karena itu diperlukan data pendukung lainnya untuk melengkapi parameter-parameter yang tidak diketahui, dengan menggunakan buku literatur atau sumber-sumber lain.

Beberapa data sekunder yang diperoleh antara lain :

- Panjang *header* IP/Ethernet adalah sebesar 34 *byte* sesuai dengan draft-guy-iax-05.

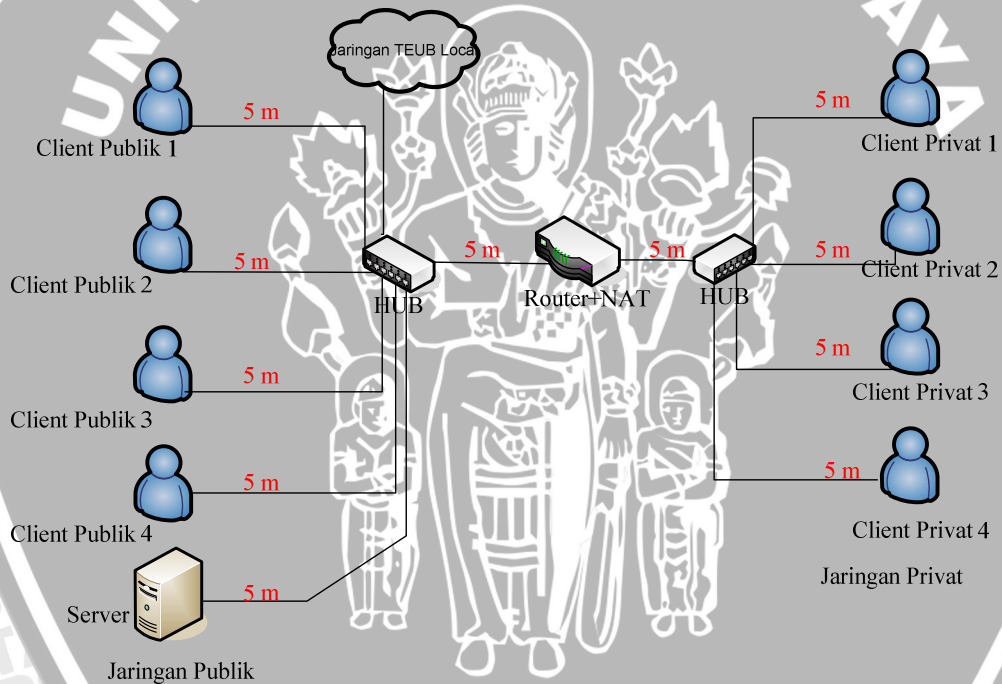
- Panjang *header* IAX adalah sebesar 4 *byte* sesuai dengan draft-guy-iax-05
- Panjang *header* untuk *UDP* adalah sebesar 8 *byte* sesuai dengan draft-guy-iax-05
- *Payload* G.711 sebesar 160 *byte*/paket.
- Kapasitas Hub yang digunakan 10 Mbit/s
- *Delay* router 2 ms
- *Delay* DTE 1 ms

### 3.3 Perancangan

Perancangan meliputi perancangan topologi jaringan, *hardware*, dan *software*.

#### 3.3.1 Perencanaan Topologi Jaringan

Perencanaan jaringan VoIP berbasis protokol IAX adalah seperti Gambar 3.1



Gambar 3.1 Arsitektur jaringan VoIP menggunakan protocol IAX  
Sumber : Perencanaan

#### 3.3.2 Perencanaan *Hardware*

Komponen jaringan secara *hardware* dipasang sesuai dengan gambar 3.1 dan disesuaikan dengan keadaan dan topologi jaringan yang telah terpasang pada jurusan Teknik Elektro Universitas Brawijaya.



Tabel 3.1 Kebutuhan *Hardware*

<i>Hardware</i>	<i>Client</i>	<i>Server</i>	<b>Router + NAT</b>	<b>Jaringan</b>
<b>Jumlah</b>	8 Buah	1 Buah	1 Buah	1 set
<b>Kebutuhan</b>	NIC Soundcard Speaker Mic	NIC	NIC (2 Buah)	HUB 8 <i>Port</i> (2 Buah) Kabel UTP kategori 5 • 12 x 5 meter Konektor RJ 45 (22 Buah)

Sumber : Perencanaan

Pada sisi *server*, digunakan sebuah komputer dengan sebuah NIC (*Network Interface Card*). Pada sisi *router* dan NAT, digunakan minimal dua buah NIC. Sedangkan pada sisi *client*, minimal terdapat sebuah NIC, dan juga perangkat *audio* berupa *sound card*, *mic* dan *speaker*, atau sebuah *headset*. Jaringan menggunakan kabel UTP kategori 5 dengan 2 buah HUB 8 *port* sebagai penghubung pada setiap NIC baik *client*, *server*, maupun *router*.

### 3.3.3 Perencanaan *Software*

*Software* yang digunakan untuk sisi *client* (*softphone*) menggunakan Idefisk versi 2.1 yang dapat mendukung protokol IAX dan *codec* G711, GSM, iLBC.

Tabel 3.2 Perangkat Lunak yang Dibutuhkan

<i>Hardware</i>	<i>Client</i>	<i>Server</i>	<b>Router</b>
<b>Sistem Operasi</b>	Windows XP SP2	Linux Kernel 2.6	Ubuntu <i>Server</i> Edition 8.04
<b>Aplikasi</b>	Idefisk 2.19	Trixbox 2.1.0	<i>IP</i> Tables
<b><i>Software</i> Analisis</b>	Unsniff + Wireshark 1.0.2.25698		-

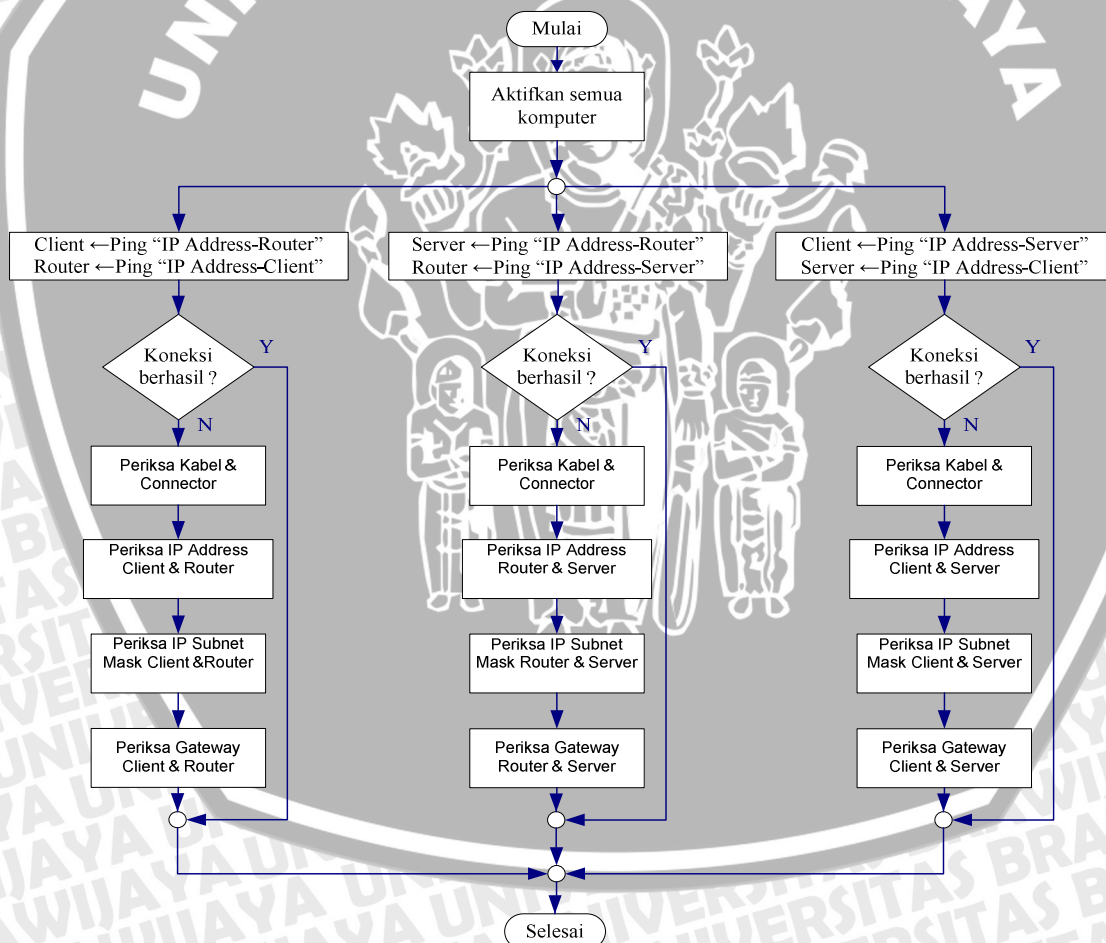
Sumber : Perencanaan

Pada sisi *router* dan NAT menggunakan linux Ubuntu Hardy 8.04. Sedangkan pada sisi *server* menggunakan Trixbox 2.1.0. Perencanaan pada sisi *software* juga meliputi tentang konfigurasi *software* dan konfigurasi jaringan yang dipakai.

### 3.4 Pengujian dan Pengambilan Data

Pengujian dilakukan setelah semua *hardware* dan *software* telah dipasang dan dikonfigurasi sesuai perencanaan. Pengujian dilakukan untuk memastikan bahwa semua sistem telah berjalan dengan baik sesuai dengan fungsi masing-masing dan dapat dilakukan pengambilan data untuk mengetahui performansi jaringan VoIP berbasis protokol IAX.

Pengujian dan pengambilan data menggunakan 8 *client* dengan 1 *server* dalam jaringan LAN untuk meminimalisir adanya kesalahan hasil pengujian akibat faktor jaringan internet. Pengujian dilakukan untuk melihat performansi sistem jaringan VoIP berbasis protokol IAX dengan mengambil data *delay*, *jitter* dan *packet loss* yang terjadi menggunakan *software* Wireshark dan Unsniff yang dapat menangkap paket-paket dalam jaringan.



Gambar 3.2 Metode Pengujian Koneksi Komputer

Sumber : perencanaan



### 3.4.1 Pengujian Interkoneksi Jaringan

#### 3.4.1.1 Tujuan Pengujian

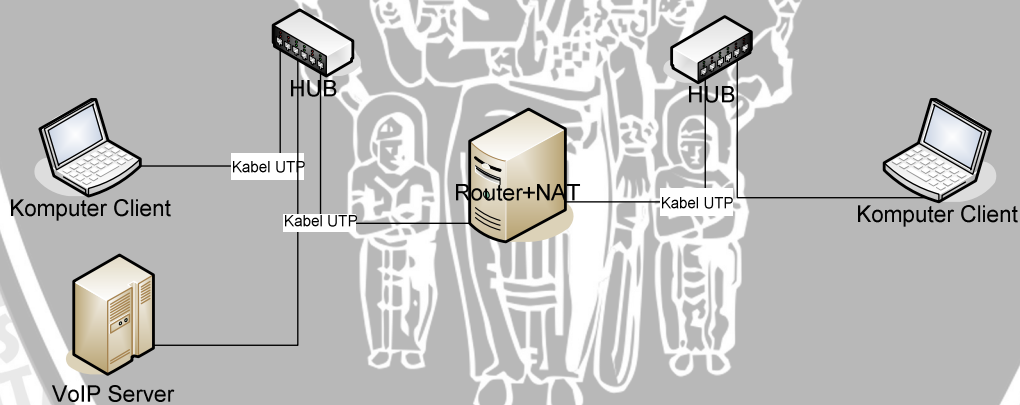
Pengujian interkoneksi jaringan ini dilakukan dengan tujuan untuk mengetahui apakah koneksi jaringan sudah dapat berjalan dengan baik dan apakah NAT telah berjalan sesuai fungsinya.

#### 3.4.1.2 Peralatan Pengujian

Pada pengujian Interkoneksi jaringan ini, peralatan yang dibutuhkan antara lain :

1. PC yang berfungsi sebagai *server*
2. PC yang berfungsi sebagai *router* dan NAT
3. Kabel UTP CAT 5 konfigurasi *straight*
4. 2 buah HUB 8 *port*
5. Seperangkat Laptop untuk *client*

Peralatan tersebut di hubungkan dengan jaringan pada jaringan lokal Teknik Elektro Universitas Brawijaya, lebih tepatnya di lingkungan Laboratorium Telekomunikasi, dan disusun seperti gambar 3.3



Gambar 3.3 Arsitektur Pengujian Interkoneksi Jaringan  
Sumber : Perencanaan

#### 3.4.1.3 Prosedur pengujian *client-client* dan pengujian *client-server*

1. Menghubungkan semua komputer, baik *server*, *router* NAT, dan *client* dengan jaringan melalui HUB atau Switch dengan menggunakan kabel UTP cat 5.
2. Mengkoneksikan semua komputer dengan mengatur IP sesuai networknya. IP diubah pada setiap *client* dengan cara membuka *icon local area connection*, klik *properties*, dan Internet Protokol (TCP/IP). Setting IP dapat dilihat pada gambar 3.4:



Gambar 3.4 Setting Pengalamatan IP pada Client  
Sumber : Perencanaan

IP diberikan sesuai jaringan yang akan dibuat dan fungsi komputer tersebut dalam jaringan. Jaringan ini akan dibedakan dalam jaringan privat, jaringan publik, komputer sebagai *server*, dan komputer *router* NAT. Pengalamatan IP diberikan sesuai tabel 3.3 :

Tabel 3.3 Pengalamatan IP

Host	IP	Mask	Gateway	Status
Client 1	172.17.63.111	253.453.455. 128	172.17.63.101	Publik
Client 2	172.17.63.112	255.255.255. 128	172.17.63.101	Publik
Client 3	172.17.63.113	255.255.255. 128	172.17.63.101	Publik
Client 4	172.17.63.114	255.255.255.128	172.17.63.101	Publik
Client 5	192.168.1.15	255.255.255.0	192.168.1.1	Privat
Client 6	192.168.1.16	255.255.255.0	192.168.1.1	Privat
Client 7	192.168.1.17	255.255.255.0	192.168.1.1	Privat
Client 8	192.168.1.18	255.255.255.0	192.168.1.1	Privat
Eth0 Router NAT	172.17.63.101	255.255.255.0	172.17.63.1	Router NAT
Eth1 Router NAT	192.168.1.1	255.255.255.0	172.17.63.101	Router NAT
<i>Server</i>	172.17.63.109	255.255.255.0	172.17.63.101	<i>Server</i>

Sumber : Perencanaan



3. Melakukan ping antar *client* dan *client* ke *server*. Ping dilakukan untuk sesama *host* yang berada pada jaringan yang sama untuk pengujian tanpa NAT maupun untuk jaringan yang berbeda untuk pengujian menggunakan NAT.

### 3.4.2 Pengujian Performansi Jaringan VoIP Berbasis IAX

Pengujian ini secara umum untuk mengetahui pengaruh penggunaan protokol pada jaringan VoIP dalam melewati NAT dengan melihat performansi jaringan.

#### 3.4.2.1 Delay End to End

Delay end to end merupakan delay yang berpengaruh pada kualitas suara yang diterima. Semakin besar *delay end to end*, mengakibatkan semakin buruk kualitas jaringan VoIP tersebut

##### 3.4.2.1.1 Tujuan Pengujian

Untuk melihat parameter performansi jaringan VoIP yang dirancang, yaitu parameter *delay end to end*.

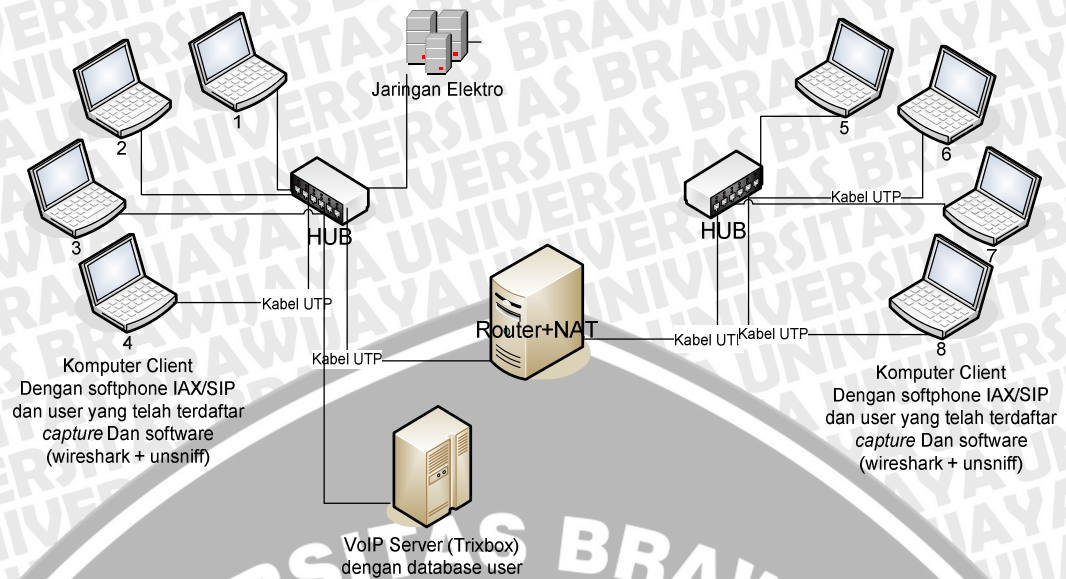
##### 3.4.2.1.2 Peralatan Pengujian

Pada pengujian performansi jaringan ini, peralatan yang dibutuhkan antara lain :

1. PC yang berfungsi sebagai *server* dengan *client* yang sudah terdaftar sebanyak 8 *client*.
2. PC yang berfungsi sebagai Router dan NAT dengan IP yang telah terkonfigurasi sesuai dengan tabel 3.3.
3. Kabel UTP CAT 5 konfigurasi *straight* sepanjang total 55 meter
4. 2 buah HUB 8 *port*
5. 8 buah laptop untuk *client* dengan *softphone* Idefisk, *software capture* Unsniff dan Wireshark

##### 3.4.2.1.3 Prosedur Pengujian

Prosedur pengujian dilakukan seperti pada perencanaan pengujian performansi. Pengujian dilakukan dengan menggunakan Unsniff yang dapat mengenali paket IAX.

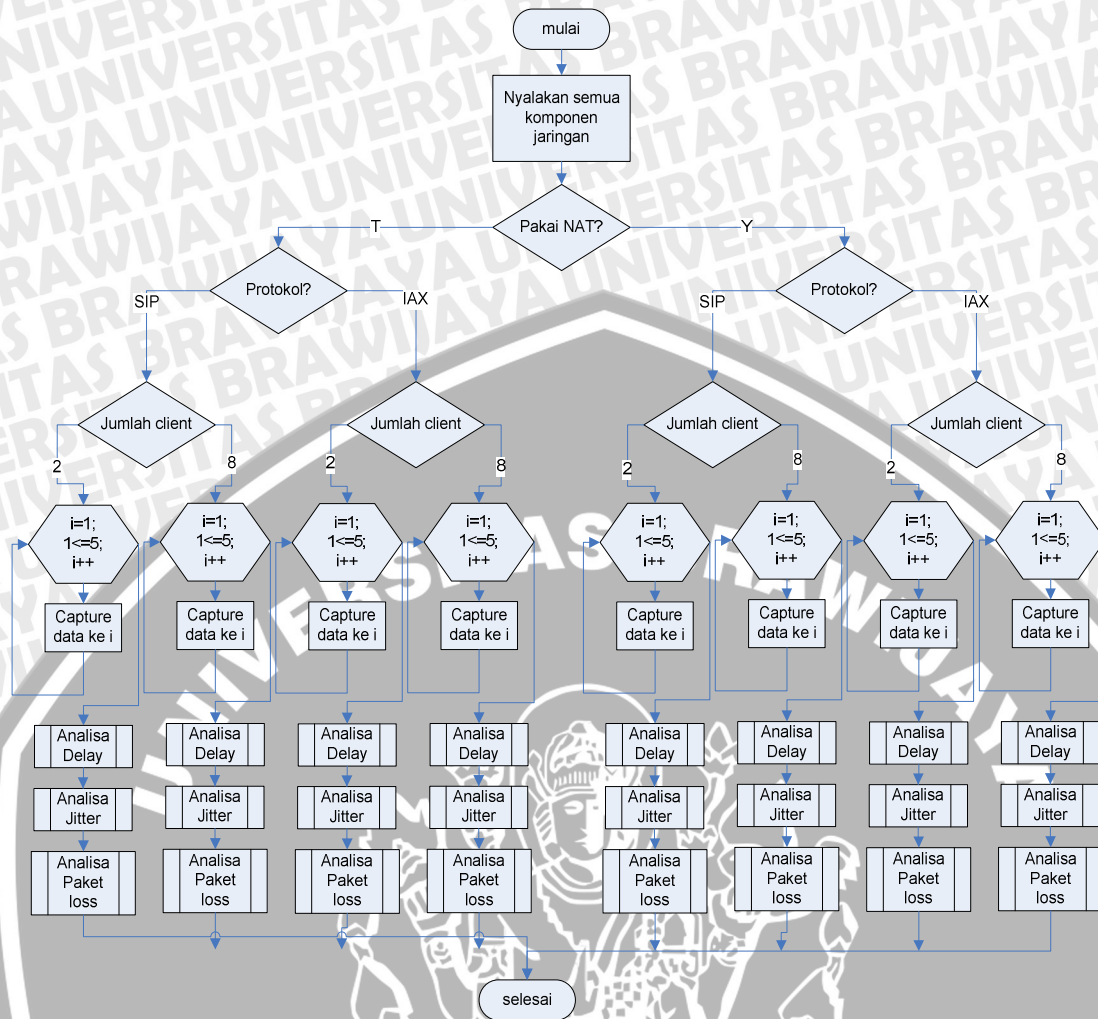


Gambar 3.5 Konfigurasi Pengujian Jaringan VoIP  
Sumber : Perencanaan

Pengujian dilakukan sebanyak dua kali, yaitu pengujian tanpa menggunakan NAT dan pengujian menggunakan NAT, masing-masing pengujian terdapat 2 sesi yaitu sesi pengujian menggunakan protokol IAX dan sesi pengujian menggunakan SIP (sebagai protokol pembandingan). Setiap sesi terdapat pengambilan data untuk 2 *client* dan pengambilan data untuk semua *client* ( total 8 *client* ) dengan kondisi jaringan dan client seperti pada gambar 3.5:

Data diambil pada saat salah satu *client* yang bertindak sebagai pemanggil melakukan panggilan, kemudian pada sisi yang lain berdering dan menerima panggilan dengan cara menekan tombol "*accept*". Pengujian performansi jaringan VoIP berbasis IAX dilakukan sesuai dengan *flowchart* pada gambar 3.6.





Gambar 3.6 Flowchat Prosedur Pengujian Performansi  
Sumber : Perencanaan

Sesuai dengan flowchart diatas, dapat dijelaskan;

1. Pengujian dilakukan berurutan untuk jaringan tanpa NAT dan menggunakan NAT.
2. Pada masing masing jaringan, baik menggunakan NAT ataupun tidak, dilakukan pengujian untuk SIP sebagai protokol pembandingan dan IAX, yang merupakan protokol utama yang dipakai dalam jaringan ini.
3. Pada setiap pengujian, baik menggunakan SIP maupun protokol IAX, dilakukan pengujian dengan jumlah 2 client (1 jalur percakapan) dan 8 client (4 jalur percakapan).
4. Pada setiap panggilan, baik 1 panggilan maupun 4 panggilan secara bersamaan, paket dalam jaringan ditangkap dengan menggunakan *software capture* unsniff. *Capture* dimulai saat sebelum panggilan dilakukan dengan

menekan tombol “*call*” dan berakhir sesudah panggilan berakhir dengan menekan tombol “*end call*” .

5. Hasil *capture* berupa paket-paket kemudian dianalisis *delay end to end*.
6. Pengujian 1-5 diulang untuk jaringan dengan menggunakan NAT.

### 3.4.2.2 Jitter

Seperti yang telah disebut sebelumnya, bahwa *delay* setiap paket berbeda-beda. Hal inilah yang kemudian menimbulkan *jitter*, yaitu perbedaan waktu *delay* antara paket pada ujung penerima. *Jitter* termasuk dalam salah satu parameter performansi VoIP, karena perbedaan *delay* antara paket suara yang diterima menyebabkan gangguan yang mengakibatkan suara yang diterima menjadi tidak jelas.

Standar *jitter* untuk VoIP adalah:

- *Good (0ms-20ms)*
- *Acceptable (20ms-50ms)*
- *Poor (> 50ms)*

#### 3.4.2.2.1 Tujuan Pengujian

Untuk mengetahui parameter *jitter* dalam performansi jaringan VoIP yang dirancang .

#### 3.4.2.2.2 Peralatan Pengujian

Pada pengujian performansi jaringan ini, peralatan yang dibutuhkan antara lain :

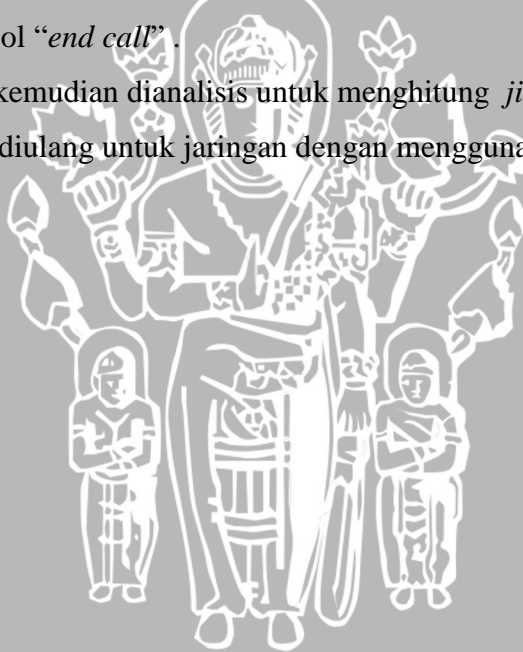
1. PC yang berfungsi sebagai *server* dengan *client* yang sudah terdaftar sebanyak 8 *client*.
2. PC yang berfungsi sebagai *Router* dan NAT dengan IP yang telah terkonfigurasi sesuai dengan tabel 3.3.
3. Kabel UTP CAT 5 konfigurasi *straight* sepanjang total 55 meter
4. 2 buah HUB 8 *port*
5. 8 buah laptop untuk *client* dengan *softphone* Idefisk, *software capture* Unsniff dan Wireshark

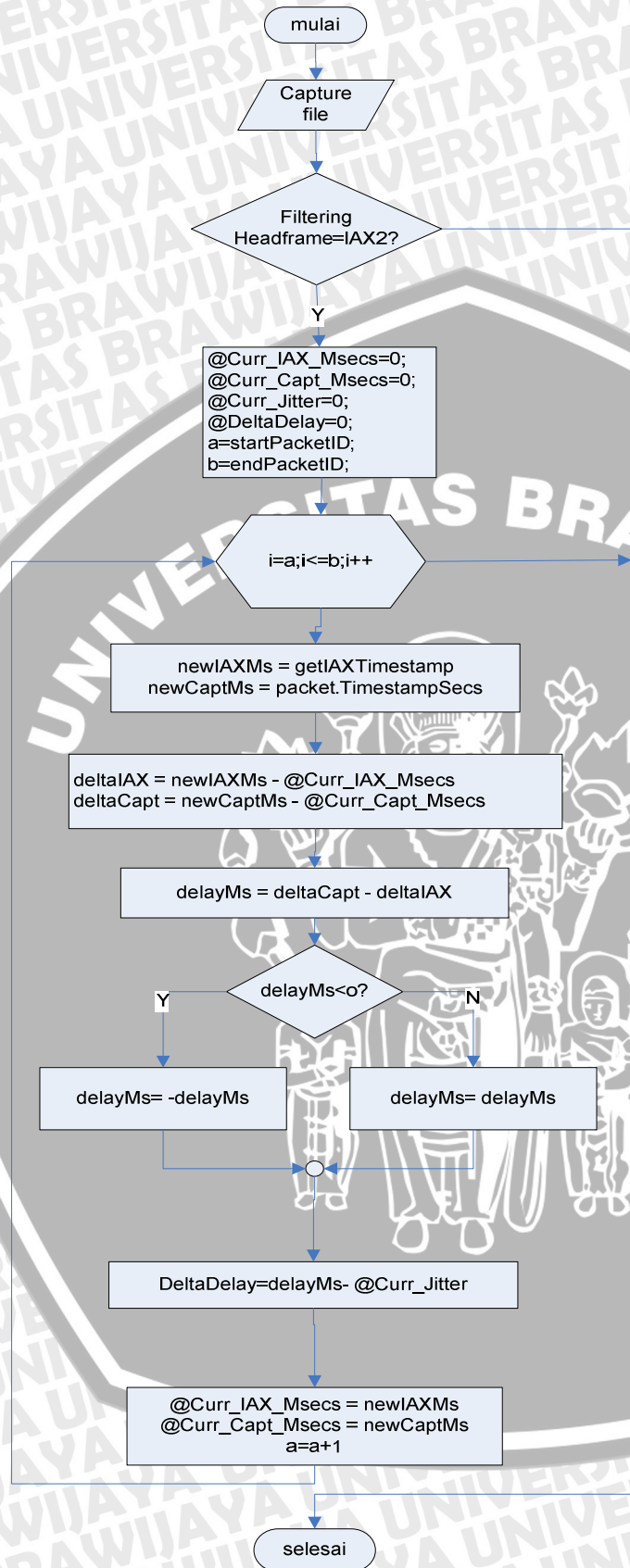


### 3.4.2.2.3 Prosedur Pengujian

Pengujian dilakukan dengan menggunakan Unsniff dan Wireshark yang dapat mengenali paket IAX. Prosedur pengujian adalah:

1. Pada masing masing jaringan, baik menggunakan NAT ataupun tidak, dilakukan pengujian untuk SIP sebagai protokol pembanding dan IAX, yang merupakan protokol utama yang dipakai dalam jaringan ini.
2. Pada setiap pengujian, baik menggunakan SIP maupun protokol IAX, dilakukan pengujian dengan jumlah 2 client (1 jalur percakapan) dan 8 client (4 jalur percakapan)
3. Pada setiap panggilan, baik 1 panggilan maupun 4 panggilan secara bersamaan, paket dalam jaringan ditangkap dengan menggunakan software *capture* unsniff. *Capture* dimulai saat sebelum panggilan dilakukan dengan menekan tombol “*call*” dan berakhir sesudah panggilan berakhir dengan menekan tombol “*end call*”.
4. Hasil *capture* kemudian dianalisis untuk menghitung *jitter*.
5. Pengujian 1-5 diulang untuk jaringan dengan menggunakan NAT.





Gambar 3.7 Flowchart Analisis Jitter  
Sumber : Perencanaan



### 3.4.2.3 Paket Loss

Pada *flowchart* analisis *loss* dapat dijelaskan bahwa paket *loss* diambil hanya pada paket dengan protokol IAX. Paket *loss* menyatakan banyaknya paket yang diterima salah dibanding dengan jumlah total paket yang dikirimkan dengan satuan paket per detik atau dalam satuan per seratus paket (persen). Standar *loss* untuk jaringan VoIP adalah :

- *Good* (0%-0.5%)
- *Acceptable* (0.5%-1.5%)
- *Poor* (> 1:5%)

#### 3.4.2.3.1 Tujuan Pengujian

Untuk mengetahui parameter paket *loss* pada performansi jaringan VoIP yang dirancang.

#### 3.4.2.3.2 Peralatan Pengujian

Pada pengujian performansi jaringan ini, peralatan yang dibutuhkan antara lain :

1. PC yang berfungsi sebagai *server* dengan *client* yang sudah terdaftar sebanyak 8 *client*.
2. PC yang berfungsi sebagai Router dan NAT dengan IP yang telah terkonfigurasi sesuai dengan tabel 3.3.
3. Kabel UTP CAT 5 konfigurasi *straight* sepanjang total 55 meter
4. 2 buah HUB 8 port
5. 8 buah laptop untuk *client* dengan *softphone* Idefisk, *software capture* Unsniff dan Wireshark

#### 3.4.2.3.3 Prosedur Pengujian

Pengujian dilakukan dengan menggunakan Unsniff dan Wireshark yang dapat mengenali paket IAX. Prosedur pengujian adalah sebagai berikut:

1. Pada masing masing jaringan, baik menggunakan NAT ataupun tidak, dilakukan pengujian untuk SIP sebagai protokol pembanding dan IAX, yang merupakan protokol utama yang dipakai dalam jaringan ini.

2. Pada setiap pengujian, baik menggunakan SIP maupun protokol IAX, dilakukan pengujian dengan jumlah 2 *client* (1 jalur percakapan) dan 8 *client* (4 jalur percakapan).
3. Pada setiap panggilan, baik 1 panggilan maupun 4 panggilan secara bersamaan, paket dalam jaringan ditangkap dengan menggunakan software *capture* *unsniff*. *Capture* dimulai saat sebelum panggilan dilakukan dengan menekan tombol “*call*” dan berakhir sesudah panggilan berakhir dengan menekan tombol “*end call*”.
4. Hasil *capture* kemudian dihitung untuk mengetahui paket *loss*.
5. Pengujian 1-5 diulang untuk jaringan dengan menggunakan NAT.

### 3.5 Kesimpulan dan Saran

Kesimpulan diambil berdasarkan perancangan yang dilakukan serta hasil yang diperoleh dari proses analisis untuk menjawab pertanyaan pada rumusan masalah.



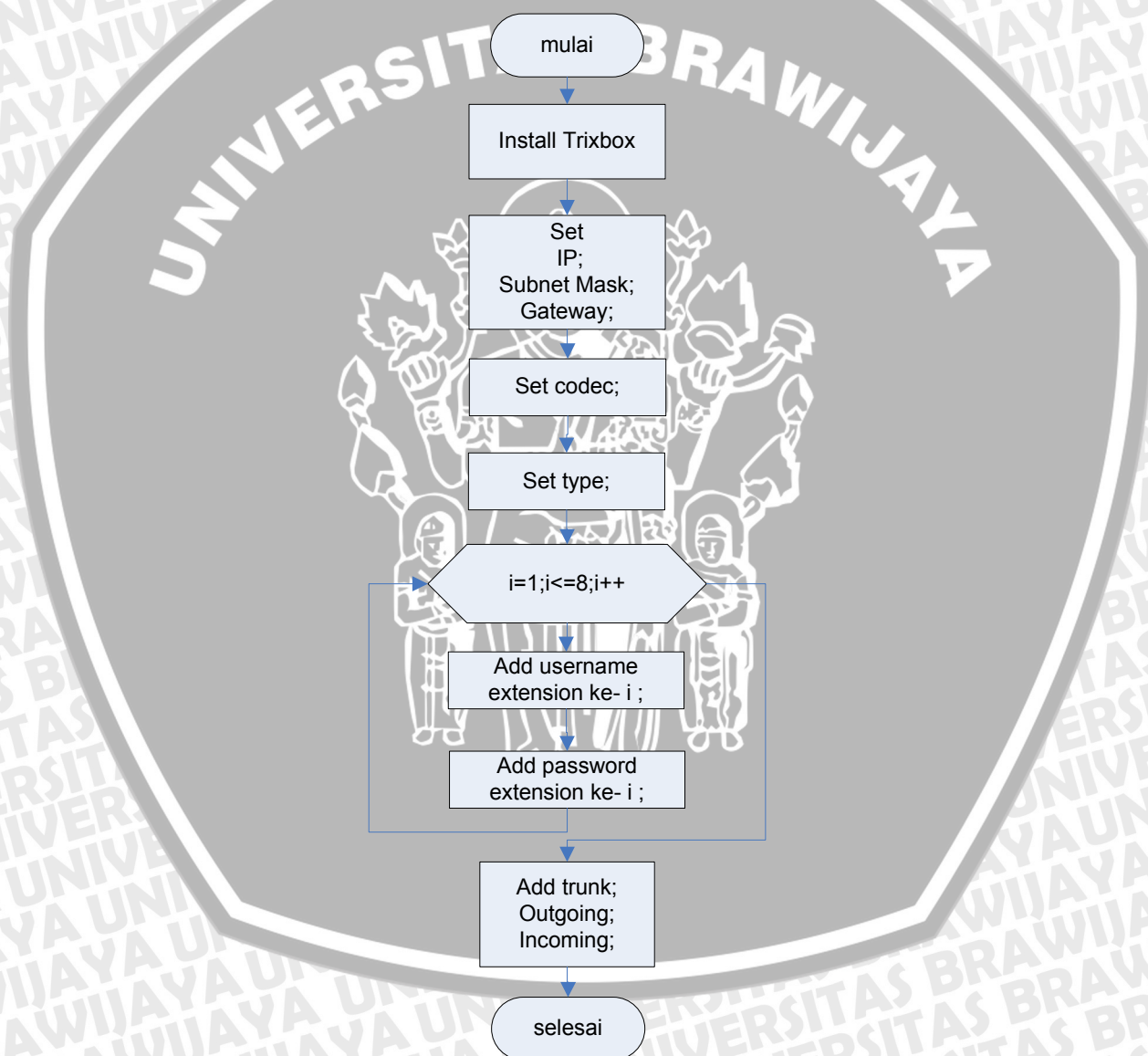


## BAB IV

### PERANCANGAN JARINGAN VOIP

#### 4.1 Perancangan Server

Pada skripsi ini digunakan satu buah *server* dan yang digunakan adalah trixbox. Trixbox sebelumnya dikenal sebagai “asterisk@home”, adalah *operating system* berbasis CentOS dan aplikasi *telephony* dengan menggunakan basis Asterisk *Open Source PBX Software*.



Gambar 4.1 Flowchart Proses Konfigurasi Server  
Sumber : Perencanaan

Sesuai dengan *flowchart* pada gambar 4.1, proses pemasangan *server* dimulai dengan memasang *trixbox*. *Trixbox* dipilih karena memberikan kemudahan dalam manajemen maupun kemampuan untuk mengembangkan lebih lanjut, karena *Trixbox* menyertakan *freePBX* yang bersifat *open source*. Selain itu, proses manajemennya menggunakan sistem web, sehingga manajemen *client* dan konfigurasi dapat dilakukan dengan bebas. Saat ini rilis terbaru dari *trixbox* adalah versi 2.4, menggunakan basis CentOS 5.1, Asterisk 1.4, dan FreePBX 2.3.

*Trixbox* berfungsi sebagai IP PBX. Fungsi ini sudah dapat dijalankan oleh sebuah komputer PII. Sesuai dengan persyaratan minimal yang dikeluarkan oleh *developer trixbox*, spesifikasi *hardware* minimal untuk membangun sebuah *server VoIP* dengan *platform* aplikasi *Trixbox* adalah :

- Prosesor : 350Mhz Pentium II
- Memory : 256MB RAM
- Harddisk : 4 GB *Hard disk space*
- NIC : 10/100 Mbps *Ethernet Card*
- Optical : CD-ROM Drive
- Network Device : 10/100 4 or 8 ports *Ethernet hub/switch* bila tidak menggunakan *router*

Sesuai dengan spesifikasi minimum yang disyaratkan untuk membangun *server trixbox*, maka dalam skripsi ini digunakan komputer *server* dengan spesifikasi sebagai berikut:

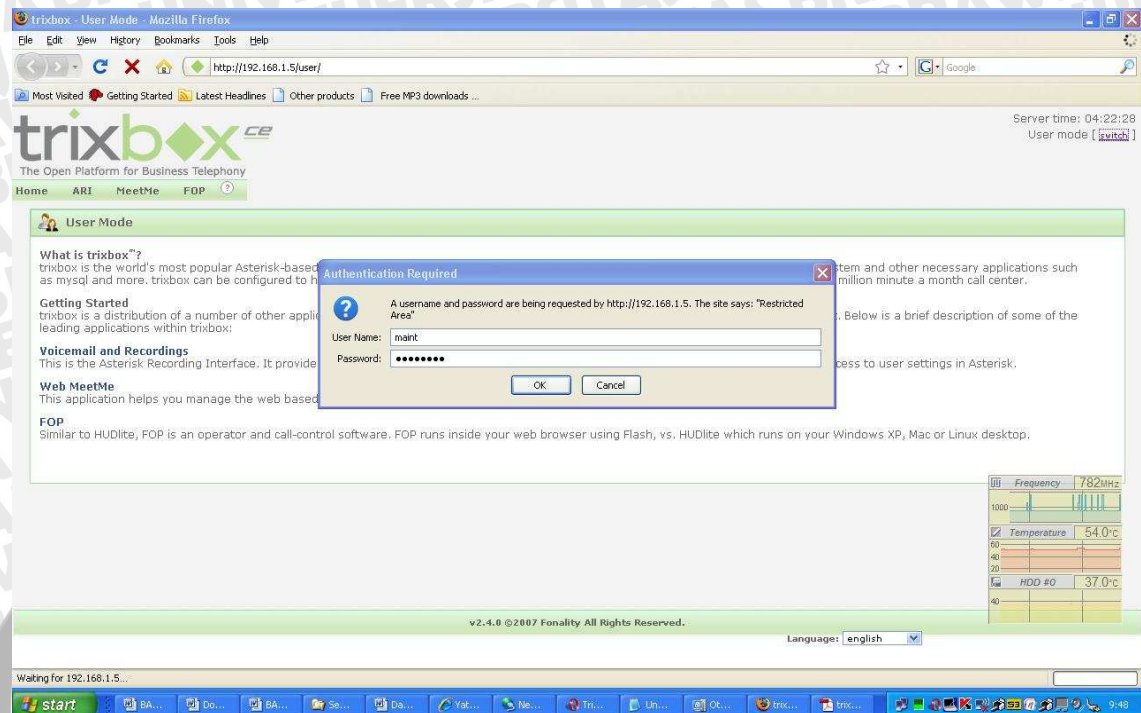
- Prosesor : Intel Pentium 4 1,7Ghz
- Memory : 256 MB RAM DDR1
- Harddisk : 20 GB 5400 rpm
- NIC : Ethernet Card 100Mbps
- Optical : CD-ROM Drive

*Trixbox* tidak menyediakan GUI pada komputer *servernya*, namun menggunakan console. Perubahan file-file berekstensi \*.conf dapat dilakukan melalui *console* ini, ataupun menggunakan fasilitas *web admin* yang merupakan fasilitas standar *Trixbox*.

*Trixbox* memberikan kemudahan karena *Trixbox* menyediakan pola administrasi dan konfigurasi *Trixbox* berbasis web. Komputer lain dapat mengakses, mengubah dan



menyimpan konfigurasi Trixbox dengan menuliskan alamat IP *server* pada *address browser*.

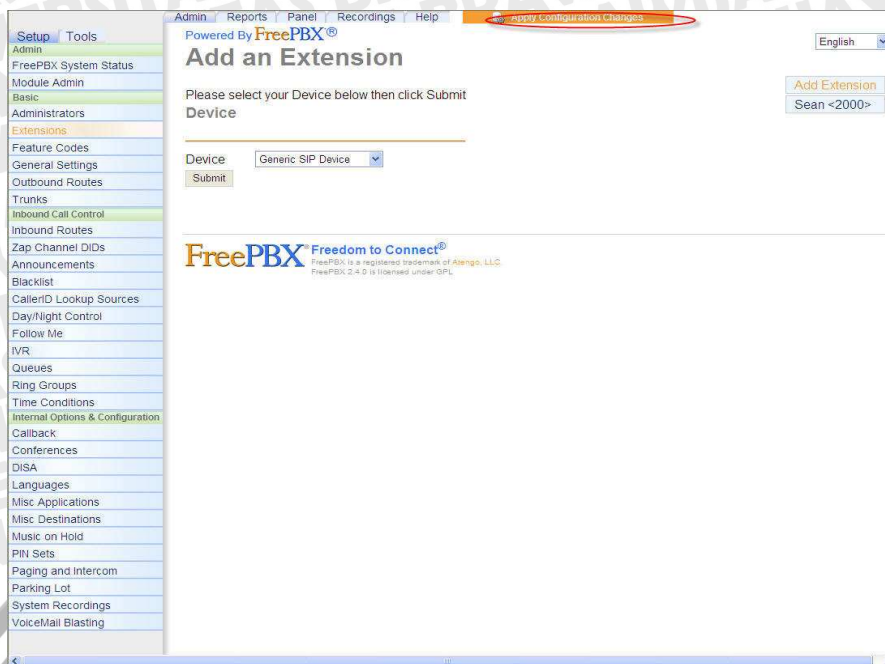


Gambar 4.2 Tampilan *Server* Trixbox Diakses Menggunakan *Web Browser*  
Sumber : Perancangan

Setelah selesai memasang Trixbox, dilanjutkan dengan konfigurasi IP sesuai dengan konfigurasi jaringan. Hal ini dilakukan agar komputer *server* dengan Trixbox dapat dikenali jaringan. Sesuai dengan *flowchart* proses konfigurasi *server*, selanjutnya adalah membuat *type client*. Tujuannya adalah untuk membatasi dan mengatur akses *client* untuk dipanggil saja, atau dapat pula memanggil. Kemudian dilanjutkan ke pemilihan *codec* untuk yang akan digunakan. Pembuatan *type client* dan pemilihan *codec* dapat dilakukan tidak berurutan.

Agar *client* dapat dikenali, maka harus dibuat *extension* untuk masing-masing *client*. *Client* didaftarkan pada PBX dengan membuat *extension* dengan menggunakan protokol IAX.

Registrasi dilakukan 2 kali, yaitu registrasi pada *client* dan registrasi pada *server*. Hal ini dilakukan agar *client* dan *server* dapat saling berkomunikasi walaupun IP yang digunakan berubah, sehingga *username* dan *password* sudah masuk dalam *database* saat *client* berubah IP-nya. Registrasi dalam *server* dilakukan dengan cara mengubah atau mengedit *file \*.conf*, sesuai dengan protokol dan *ekstension* yang digunakan.



Gambar 4.3 Membuat *Extension IAX*  
Sumber : Perancangan

Karena Trixbox menggunakan Free PBX yang bersifat open, maka pengubahan juga harus dilakukan pada *file* yang berektensi \*.conf. Dalam *file* ini, semua konfigurasi akan disimpan, kemudian *diload* pada waktu *booting*. *File* ini juga dapat diubah untuk memberikan konfigurasi baru, misalnya pada tambahan *codec* baru dan konfigurasi *trunk* baru.

#### 4.2 Perancangan *Client*

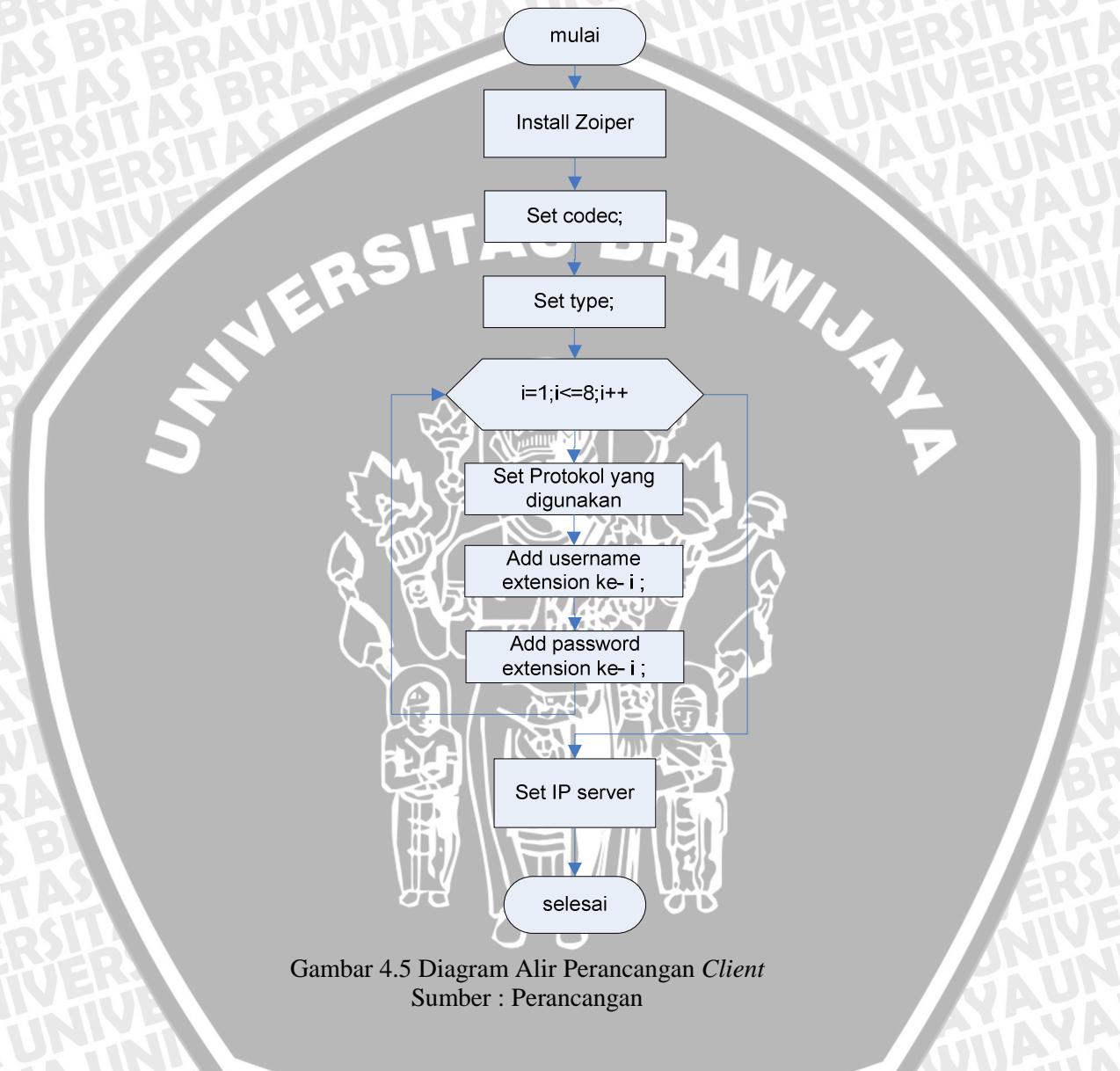
*Client* yang akan digunakan berjumlah 8 *client*, dengan 4 buah *client* pada jaringan *privat* dan 4 *client* pada jaringan publik. Agar memberikan kondisi yang sama, maka seluruh *client* menggunakan *softphone* Idefisk 2.0.



Gambar 4.4 *Softphone* Idefisk  
Sumber : Perancangan

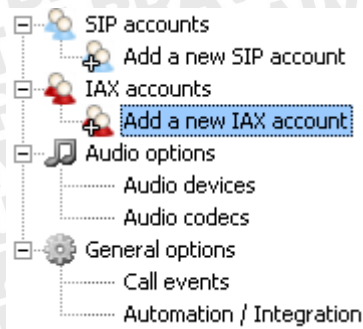


Idefisk dipilih karena memiliki kelebihan dalam hal fitur yang ditawarkan. *Softphone* ini mendukung protokol SIP dan IAX dan memiliki fitur penting diantaranya *hold button*, *transfer button*, *quick dial button*, *numpad*, *slide bars* untuk pengaturan volume *speaker* dan *microphone*, serta *history button*.



Gambar 4.5 Diagram Alir Perancangan *Client*  
Sumber : Perancangan

Dalam *flowchart* tersebut dapat dijelaskan bahwa *client* dibangun dengan cara menginstall *softphone* Idefisk pada semua komputer sebagai *client* yang berjumlah 8 *client*. Langkah selanjutnya adalah mengeset *codec* beserta *type client* sesuai dengan *codec* dan *type* yang telah diset pada *server*. Pada setiap Idefisk ditambahkan alamat IP *server* agar dapat melakukan registrasi pada *server* tersebut.



Gambar 4.6 Add Account pada *Softphone* Idefisk  
Sumber : Perancangan

*Client* dibuat dalam *softphone* ini sesuai dengan *client* yang telah didaftarkan pada *server*. Setelah dibuat, *client* harus diregistrasi terlebih dahulu dengan menekan tombol *register*. Pengaturan *client* adalah sesuai dengan *extensions* yang diinginkan.

Spesifikasi minimal untuk memasang software Idefisk adalah :

- Processor : Pentium II 300 Mhz
- Memory : 256 MB RAM
- Operating systems : Windows® 2000, XP
- Sound card : 16 bit *sound card*
- Internet connection : *wired* atau *wireless broadband*

Dalam perencanaan ini, digunakan komputer dengan spesifikasi sebagai berikut:

a. *Client* Publik 1

- Processor : Genuine Intel CPU T1350 1,86 GHz
- Memori : 896 MB
- Operating System : Microsoft Windows XP Professional version 2002
- LAN Card : VIA VT6102 Rhine II Fast Ethernet Adapter
- Sound Card : VIA VT8237A High Definition *Audip* Controller

b. *Client* Publik 2

- Processor : Intel ® Core (TM)2 CPU T5500 1,66 GHz
- Memori : 1.49 GB
- Operating System : Microsoft Windows XP Professional version 2002



- LAN Card : Marvell Yukon 88E8038 PCI-E Fast Internet Controller
- Sound Card : Intel ICH 7-High Definition *Audio* Controller

c. *Client Publik 3*

- Processor : Genuine Intel CPU T1350 1,86 GHz
- Memori : 896 MB
- Operating System : Microsoft Wiindows XP Professional version 2002
- LAN Card : VIA VT6102 Rhine II Fast Ethernet Adapter
- Sound Card : VIA VT8237A High Definition *Audio*Controller

d. *Client Publik 4*

- Processor : Intel ® Core™ 2 CPU T5500 1,66 GHz
- Memori : 1,49 GB
- Operating System : Microsoft Windows XP Professional version 2002
- LAN Card : Marvell Yukon 88E8038 PCI-E Fast Internet Controller
- Sound Card : Intel ICH7-High Definition *Audio* Controller

e. *Client Privat 1*

- Processor : Intel pentium 4 2.0 GHz
- Memori : DDR1 128 MB
- Operating System : Microsoft Windows XP Professional SP 2
- LAN Card : NIC 10/100
- Sound Card : int sound card

f. *Client Privat 2*

- Processor : Intel pentium 4 1.8 GHz
- Memori : DDR1 128 MB
- Operating System : Microsoft Windows XP Professional SP 2
- LAN Card : NIC 10/100
- Sound Card : int sound card

g. *Client Privat 3*

- Processor : Intel pentium 4 2.0 GHz
- Memori : DDR1 128 MB
- Operating System : Microsoft Windows XP Professional SP 2
- LAN Card : NIC 10/100
- Sound Card : int sound card

h. *Client* Privat 4

- Processor : Intel Core Duo T2130
- Memori : 1 GB DDR2
- Operating System : Microsoft Windows XP Professional SP 2
- LAN Card : Realtek gigabit ethernet
- Sound Card : SIS soundcard

Masing-masing *client* menggunakan registrasi username dan password sesuai dengan perencanaan. Registrasi dilakukan di *client*, kemudian dikonfirmasi pada *server*. Dalam registrasi ini juga diberikan nomor telepon (ID call) yang dapat diatur sesuai kehendak. Namun sesuai dengan standar VoIP local, maka ID call yang digunakan adalah ID call sebanyak 4 digit dengan ketentuan nomor 90XX untuk nomor IAX, dan 91XX untuk SIP. Dalam skripsi ini XX11-XX14 adalah *client* yang berada dalam nomor public, sedang XX15-XX18 berada dalam IP privat. Username, password, dan ID call diatur sesuai dengan table 4.1

Tabel 4.1 Registrasi *Client*

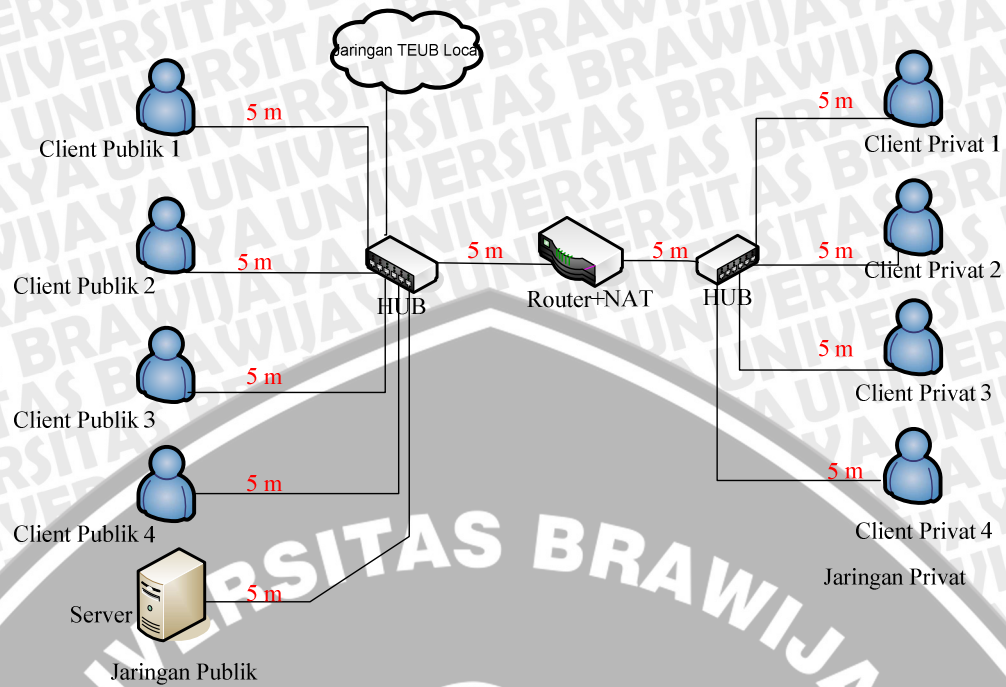
Host	IP	Username		Call ID		Password		Status
		IAX	SIP	IAX	SIP	IAX	SIP	
<i>Client</i> 1	172.17.63.111	IAX1	SIP	9011	9111	9011	9111	Publik
<i>Client</i> 2	172.17.63.112	IAX2	SIP	9012	9112	9012	9112	Publik
<i>Client</i> 3	172.17.63.113	IAX2	SIP	9013	9113	9013	9113	Publik
<i>Client</i> 4	172.17.63.114	IAX2	SIP	9014	9114	9014	9114	Publik
<i>Client</i> 5	192.168.1.15	IAX2	SIP	9015	9115	9015	9115	Privat
<i>Client</i> 6	192.168.1.16	IAX2	SIP	9016	9116	9016	9116	Privat
<i>Client</i> 7	192.168.1.17	IAX2	SIP	9017	9117	9017	9117	Privat
<i>Client</i> 8	192.168.1.18	IAX2	SIP	9018	9118	9018	9118	Privat

Sumber : Perancangan

### 4.3 Perancangan Topologi Jaringan

Jaringan VoIP yang akan dibangun menggunakan 1 buah *server* dengan 8 buah *client* yang terbagi atas *client* pada sisi privat dan *client* pada sisi publik. Keduanya dipisahkan oleh *router* yang juga bertindak sebagai NAT.





Gambar 4.7 Arsitektur Jaringan VoIP Menggunakan Protocol IAX  
Sumber : Perencanaan

Pada sisi publik terdapat 4 buah *client*, demikian pula untuk sisi *privat*. Penempatan *router* dan NAT dipasang pada sebuah PC. *Server* ditempatkan pada sisi jaringan publik, dengan tujuan untuk memodelkan sebuah *server* yang terdapat pada jaringan internet secara luas. Permodelan ini untuk mendapatkan data performansi jaringan VoIP berbasis protokol IAX pada saat melewati NAT.

Jaringan sebelumnya harus dikonfigurasi dengan alamat IP yang mewakili alamat IP publik dan *privat*. IP lokal TEUB yang dipakai akan mewakili IP publik dalam jaringan VoIP ini. Kemudian IP *privat* ditambahkan dengan membuat jaringan baru yang dibatasi dengan NAT. Pengalamanan IP diatur sesuai dengan Tabel 4.2.

Jaringan ini dirancang untuk mendapatkan data yang dapat menunjukkan performansi jaringan VoIP menggunakan protokol IAX. Penggunaan NAT dimaksudkan sebagai permodelan apabila jaringan VoIP digunakan secara luas dalam jaringan internet.

Pada skripsi ini, jaringan VoIP dibuat dalam jaringan LAN untuk meminimalkan gangguan transmisi serta parameter lain yang tidak dapat diketahui. Parameter-parameter tersebut misalnya *delay* dan *jitter* yang terjadi akibat penumpukan diaa pada *router* yang berada jauh dari *client*, yang sebenarnya *route* pengganti dari jalur yang sebenarnya.

Tabel 4.2 Pengalamatan IP

Host	IP	Mask	Gateway	Status
<i>Client 1</i>	172.17.63.111	253.453.455. 128	172.17.63.101	Publik
<i>Client 2</i>	172.17.63.112	253.455.255. 128	172.17.63.101	Publik
<i>Client 3</i>	172.17.63.113	255.255.255. 128	172.17.63.101	Publik
<i>Client 4</i>	172.17.63.114	255.255.255.128	172.17.63.101	Publik
<i>Client 5</i>	192.168.1.15	255.255.255.0	192.168.1.1	Privat
<i>Client 6</i>	192.168.1.16	255.255.255.0	192.168.1.1	Privat
<i>Client 7</i>	192.168.1.17	255.255.255.0	192.168.1.1	Privat
<i>Client 8</i>	192.168.1.18	255.255.255.0	192.168.1.1	Privat
Eth0 <i>Router NAT</i>	172.17.63.101	255.255.255.0	172.17.63.1	<i>Router NAT</i>
Eth1 <i>Router NAT</i>	192.168.1.1	255.255.255.0	172.17.63.101	<i>Router NAT</i>
<i>Server</i>	172.17.63.109	255.255.255.0	172.17.63.101	<i>Server</i>

Sumber : Perencanaan

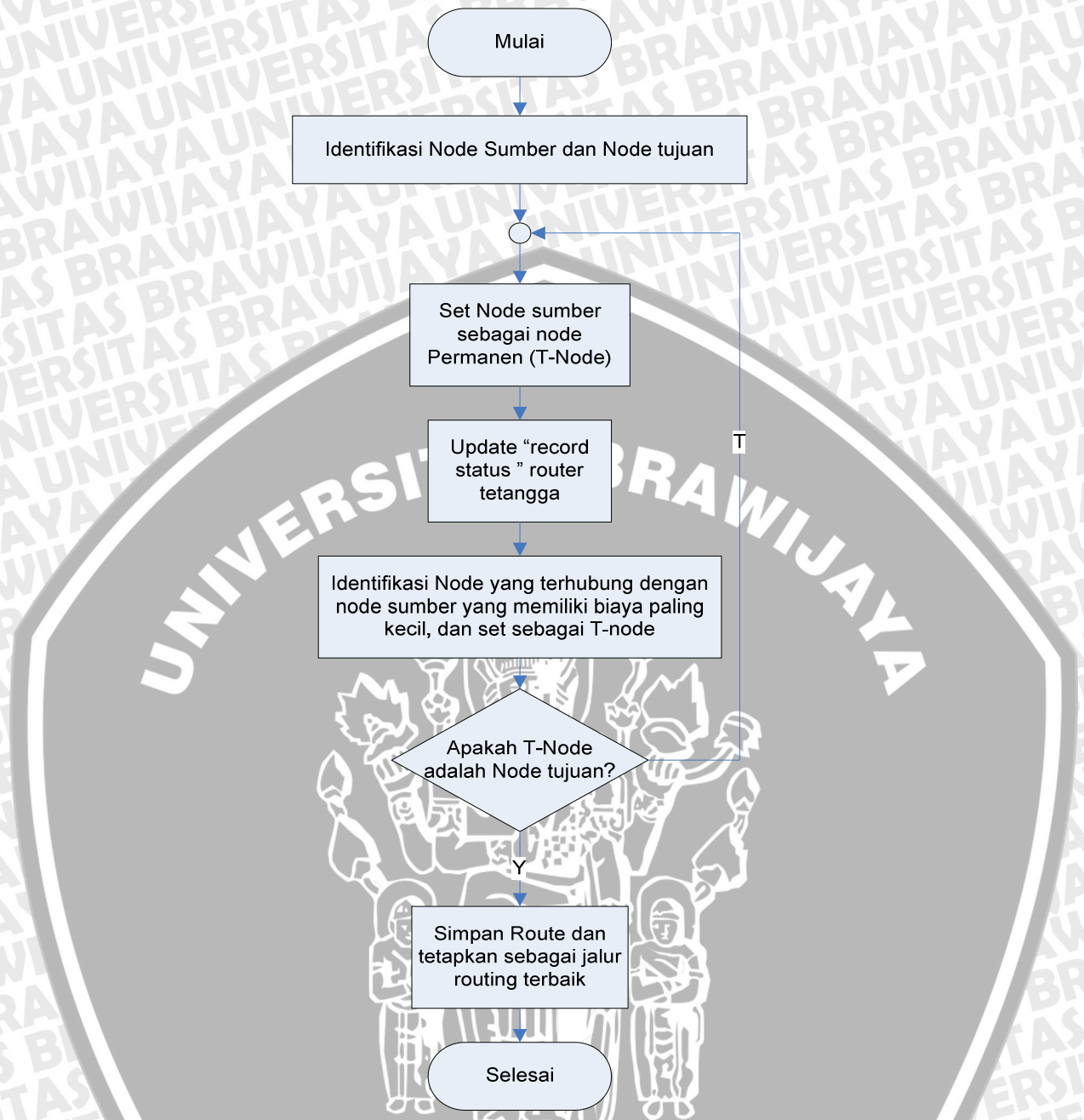
#### 4.4 Konfigurasi *Router* dan NAT

*Router* berfungsi untuk menentukan jalur terbaik dalam melewati paket data. *Route* terbaik ditentukan dari informasi yang diberikan oleh *router* tetangganya, atau mencari sendiri informasi yang diperlukan untuk menentukan *router* yang terbaik dengan cara mengirim paket data dan menghitung *cost* yang digunakan.

Pada skripsi ini, *router* dan NAT (*Network Address Translation*) diletakkan pada satu PC (*Personal Computer*). Spesifikasi Hardware untuk *router* dan NAT adalah sebagai berikut :

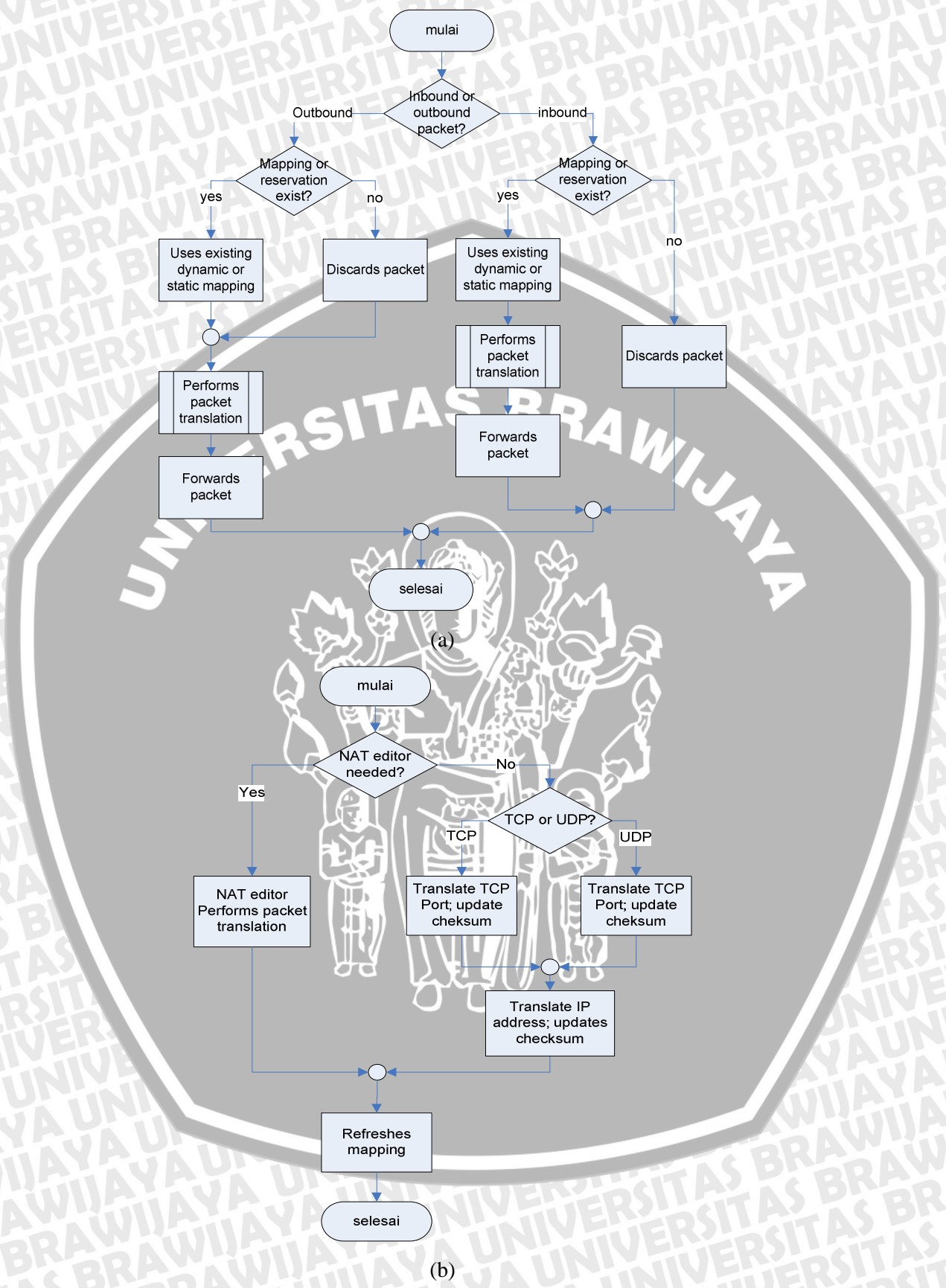
- Processor : Intel pentium 4 1,7 Ghz
- Memory : 256 MB DDR1
- Operating System : Linux Ubuntu *server* 8.04
- LAN Card : 1.D-Link DFE-528TX  
2.D-Link DFE-528TX
- IP Address : 172.17.63.101 (Public)  
192.168.1.1 (Private)





Gambar 4.8 Flowchart Proses Penentuan Routing  
Sumber : <http://web.njit.edu/>

IAX menggunakan *port* 4569 untuk melewati media, baik media kontrol maupun data yang dikirimkan. Hal inilah yang menyebabkan IAX relatif lebih mudah melewati NAT karena hanya mengatur media dalam 1 protokol bila dibandingkan dengan menggunakan protokol SIP yang minimal harus menggunakan 2 *port*, yaitu 1 *port* untuk *signalling* dan *port* yang lain untuk meneruskan aliran data berupa *voice*.



Gambar 4.9a Flowchart Proses untuk *Outbound* dan *Inbound* Trafik  
 4.9b Flowchart Proses Translasi Paket

Sumber : <http://technet.microsoft.com/en-us/library/cc785385.aspx>





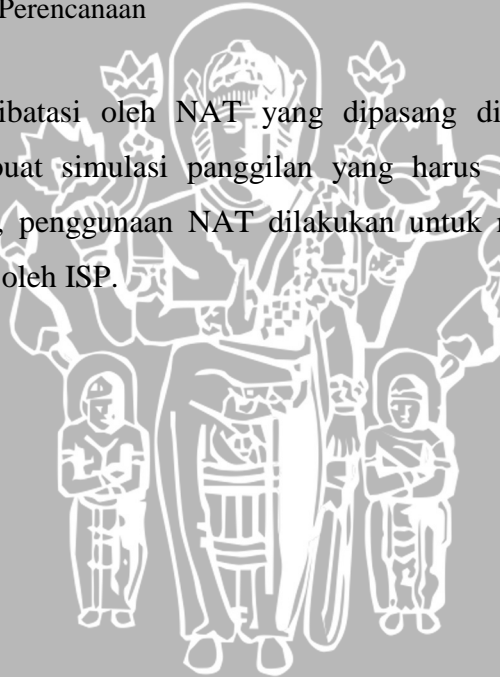
Konfigurasi *router* dilakukan dengan memberi alamat IP pada kedua *ethernet card* (eth0 dan eth1). Dalam *router* ini, eth1 dihubungkan pada jaringan lokal, sedang eth0 dihubungkan ke jaringan publik. Jaringan publik menggunakan jaringan yang dimiliki oleh Jurusan Teknik Elektro Universitas Brawijaya.

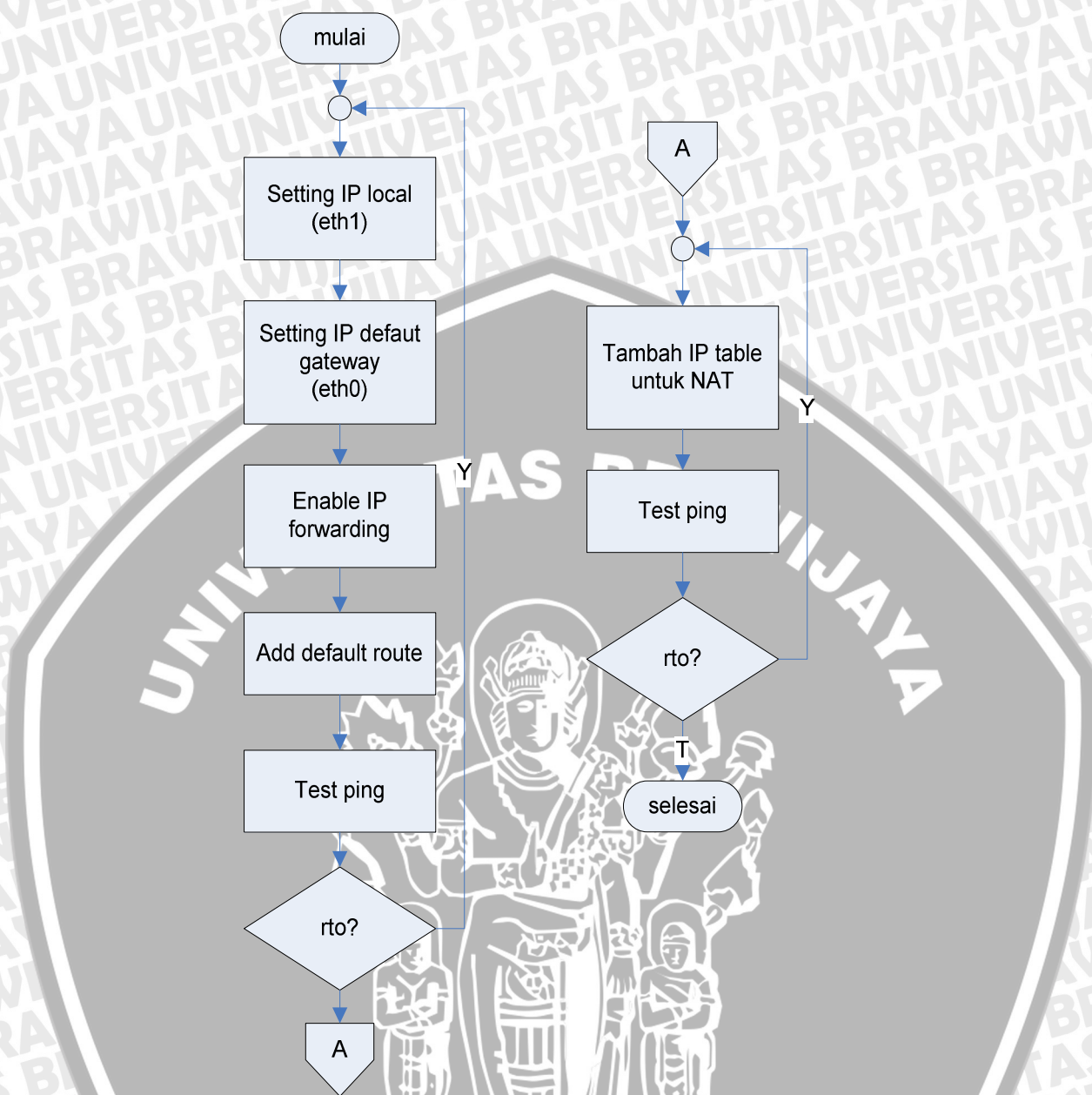
Tabel 4.3 Konfigurasi Jaringan Publik

IP	172.17.63.111- 172.17.63.114
Gateway	172.17.63.101
Net Mask	255.255.255.128
Broadcast	172.17.63.255
DNS 2	202.162.208.100

Sumber : Perencanaan

Jaringan lokal dibatasi oleh NAT yang dipasang di *router*. Jaringan ini dipisahkan untuk membuat simulasi panggilan yang harus melewati NAT. Pada keadaan yang sebenarnya, penggunaan NAT dilakukan untuk mengatasi kekurangan alamat IP yang disediakan oleh ISP.





Gambar 4.10 Konfigurasi Router dan NAT  
 Sumber : Perencanaan

Tabel 4.4 Konfigurasi Jaringan Lokal

IP	192.168.1.11 – 192.168.1.14
Gateway	192.168.1.1
Net Mask	255.255.255.0
Broadcast	192.168.1.255

Sumber : Perencanaan



Penulisan *script* pada *router* adalah sebagai berikut :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
# eth1 >> Description " LAN Connection "
auto eth1
iface eth1 inet static
address 192.168.0.1
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255

# eth0 >> Description " Outside Connection "
auto eth0
iface eth2 inet static
address 172.17.63.40
netmask 255.255.255.0
network 172.17.63.0
broadcast 172.17.63.255
gateway 172.17.63.1
```

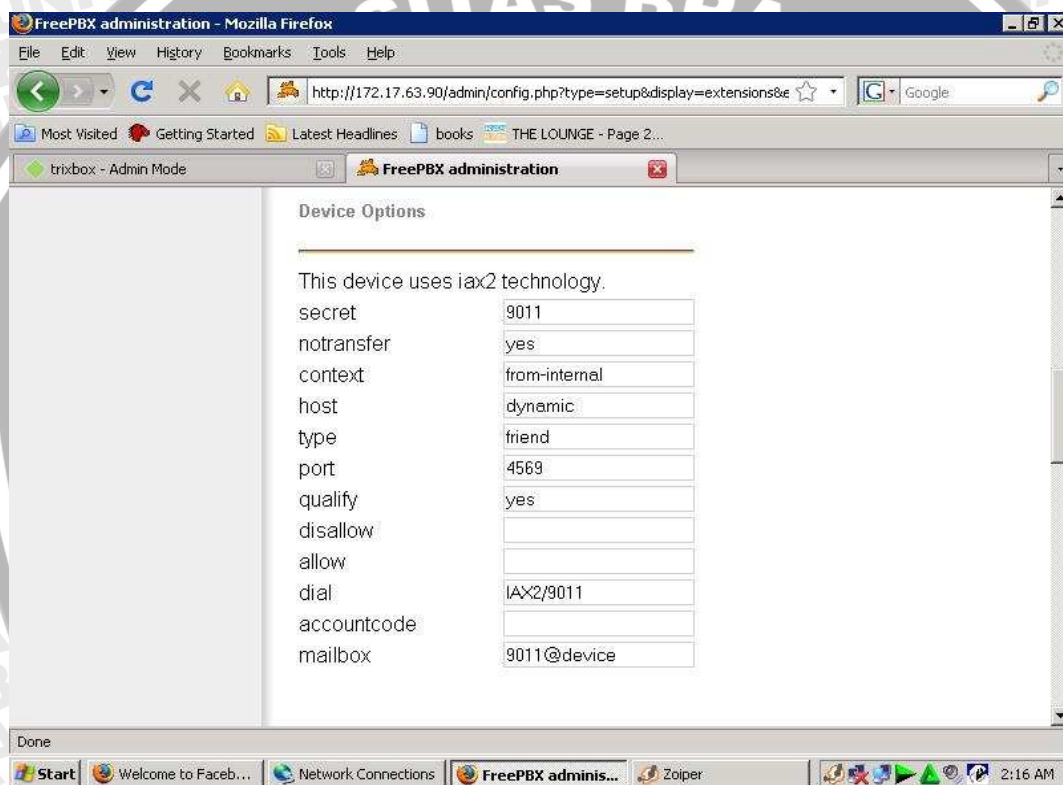
Setelah *eth0* dan *eth1* dikonfigurasi, paket-paket yang melewati ethernet harus diteruskan (*forwarding*). Pengaktifan fasilitas IP forwarding dilakukan dengan cara mengubah "*net.ipv4.ip\_forward = 0*" menjadi "*net.ipv4.ip\_forward = 1*"

Untuk menambahkan route, digunakan perintah "*route add -net 192.168.0.0 netmask 255.255.255.0 gw 172.17.63.1 dev eth0*". setelah itu *network* harus restart dengan perintah "*sudo /etc/init.d/networking restart*". Selanjutnya jaringan di test dengan menggunakan ping gateway.

Penambahan NAT pada *router* dilakukan dengan menambah *iptables* menggunakan perintah "*iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.0/24 -j SNAT --to-source 172.17.63.40*". Penambahan IPTables harus *restart* sebelum digunakan dengan menggunakan perintah "*/etc/init.d/iptables restart*".

#### 4.5 Penerapan Prorokol IAX Pada Jaringan VoIP

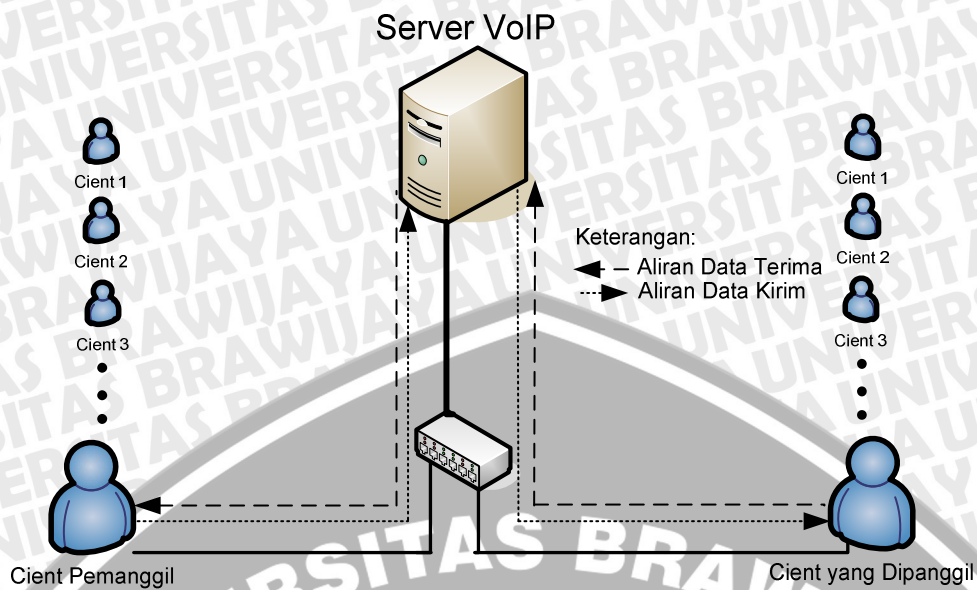
Penerapan protokol IAX dapat dilakukan dengan memilih opsi protokol yang disediakan, baik dari sisi *server* maupun *client*. Protokol ini dapat berjalan pada semua jaringan, dengan syarat *port* yang digunakan untuk signalling dan transmisi data tidak diblok oleh administrator jaringan. *Port* tersebut adalah *port* 4569. Sebenarnya *port* standar untuk media transmisi IAX dapat diubah sesuai dengan kehendak, namun hal itu tidak dilakukan karena beberapa *softphone* dan *server* tidak selalu kompatibel dengan *port* selain *port* 4569.



Gambar 4.11 Penggunaan Protokol IAX  
Sumber : Perencanaan

Sesuai dengan yang telah disebutkan sebelumnya, bahwa IAX hanya menggunakan 1 *port* untuk transmisi, baik signaling maupun data. Hal ini dimaksudkan agar IAX tidak mengalami masalah dengan NAT. Karena jalur yang digunakan untuk keduanya adalah sama, maka semua aliran signaling dan data dari *client* yang memanggil akan melewati *server* terlebih dahulu, baru kemudian diteruskan kepada *client* yang dituju.





Gambar 4.12 Signaling dan Data Stream pada Protokol IAX  
Sumber : Perencanaan

Sesuai dengan gambar 4.12, bandwidth menuju *server* harus disediakan sebesar-besarnya. Namun pada skripsi ini, untuk kepentingan pengujian dan pengambilan data pada jaringan VoIP global dengan sampel jaringan lokal yang diberi NAT, maka bandwidth jaringan yang digunakan sebesar 10 Mbps sesuai dengan kecepatan jaringan yang terpasang pada Laboratorium Telekomunikasi jurusan Teknik Elektro Universitas Brawijaya.

## BAB V

### HASIL PENGUJIAN

#### 5.1 Pengujian

Pada bab ini membahas mengenai pengujian, yaitu mengenai tujuan pengujian, peralatan yang digunakan, prosedur pengujian, data hasil pengujian maupun kesimpulan dari pengujian itu sendiri. Secara umum, tujuan dari pengujian ini adalah untuk mengetahui apakah jaringan VoIP berbasis IAX dapat memberikan performansi yang lebih baik dibanding protokol VoIP sebelumnya terutama dalam kemampuannya untuk menembus NAT.

Pengujian dilakukan pada hari Senin 29 Juni 2009 pukul 10.00 WIB sampai pukul 12.00 WIB. Seluruh proses pengujian dilakukan di Laboratorium Telekomunikasi Teknik Elektro Universitas Brawijaya Malang

Setelah pengujian dilakukan, jaringan VoIP akan dianalisis sesuai parameter Performansi yang telah disebutkan sebelumnya, yaitu *bandwidth*, *delay end to end*, *jitter*, dan *throughput*. Performansi ini akan dibandingkan dengan performansi jaringan VoIP yang menggunakan protokol sebelumnya, dalam hal ini SIP. Kedua performansi protokol ini digunakan untuk mengambil kesimpulan dan menyusun saran bila diperlukan..

#### 5.2 Data Hasil Pengujian

Pengujian Jaringan VoIP dilakukan melalui beberapa tahap pengambilan data untuk masing-masing parameter performansi yang dicari. Pengujian dilakukan di Laboratorium Telekomunikasi yang masih merupakan jaringan Teknik Elektro Universitas Brawijaya. Pengujian juga dilakukan pada saat jaringan VoIP berbasis SIP dan IAX tidak melewati NAT, maupun saat melewati NAT. Pengujian ini meliputi:

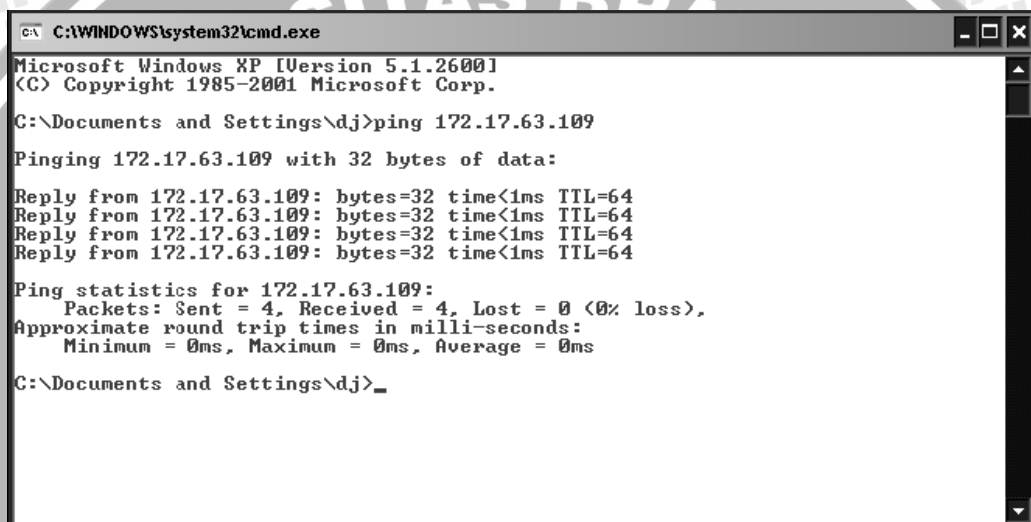
1. Pengujian Interkoneksi Jaringan
2. Pengujian Protokol IAX dan SIP dalam Menembus NAT
3. Pengujian Performansi Jaringan VoIP Berbasis IAX



### 5.2.1 Data Hasil Pengujian Pengujian Interkoneksi Jaringan

Dari pengujian yang dilakukan didapatkan data:

1. Setiap komputer dapat terkoneksi dalam jaringan dengan memberikan alamat IP sesuai dengan yang direncanakan.
2. Proses ping dilakukan satu persatu pada setiap komputer kecuali pada komputer yang bertindak sebagai *router* NAT. Hal ini dilakukan dengan asumsi bahwa bila paket data sudah sampai pada komputer *client* lain maupun pada komputer *server*, berarti paket data tersebut sudah terhubung dengan baik dengan router NAT.



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\dj>ping 172.17.63.109

Pinging 172.17.63.109 with 32 bytes of data:

Reply from 172.17.63.109: bytes=32 time<1ms TTL=64
Reply from 172.17.63.109: bytes=32 time<1ms TTL=64
Reply from 172.17.63.109: bytes=32 time<1ms TTL=64
Reply from 172.17.63.109: bytes=32 time<1ms TTL=64

Ping statistics for 172.17.63.109:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\dj>_
  
```

Gambar 5.1 Proses Pengujian Interkoneksi *Client* ke *Server*

Sumber : Pengujian

3. Pada proses pengujian ping *client* ke *client* maupun dari *client* ke *server* tanpa menggunakan NAT didapatkan data pada tabel 5.1:

Tabel 5.1 Pengujian Interkoneksi Jaringan Tanpa Menggunakan NAT

Tujuan		server	client 1	client 2	client 3	client 4	client 5	client 6	client 7	client 8
Sumber										
server		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 1		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 2		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 3		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 4		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 5		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 6		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 7		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 8		ok	ok	ok	ok	ok	ok	ok	ok	ok

Keterangan : ok= ping mendapat balasan reply (komputer dapat dikenali) -> Tersambung  
 rto= ping mendapat balasan rto (*request time out*) -> Tidak tersambung

Sumber : Pengujian

1. Pada proses pengujian ping-ing client ke client maupun dari client ke server menggunakan NAT didapatkan data pada tabel 5.2:

Tabel 5.2 Pengujian Interkoneksi Jaringan menggunakan NAT

Tujuan		server	client 1	client 2	client 3	client 4	client 5	client 6	client 7	client 8
Sumber										
server		ok	ok	ok	ok	ok	rto	rto	rto	rto
client 1		ok	ok	ok	ok	ok	rto	rto	rto	rto
client 2		ok	ok	ok	ok	ok	rto	rto	rto	rto
client 3		ok	ok	ok	ok	ok	rto	rto	rto	rto
client 4		ok	ok	ok	ok	ok	rto	rto	rto	rto
client 5		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 6		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 7		ok	ok	ok	ok	ok	ok	ok	ok	ok
client 8		ok	ok	ok	ok	ok	ok	ok	ok	ok

Keterangan : ok= ping mendapat balasan reply (komputer dapat dikenali) -> Tersambung  
 rto= ping mendapat balasan rto (*request time out*) -> Tidak tersambung

Sumber : Pengujian



### 5.2.1.1 Kesimpulan Hasil Pengujian

Dari data hasil pengujian pada tabel 5.1 dan pada table 5.2 didapatkan bahwa :

1. NAT dapat bekerja dengan baik, ditandai dengan hasil rto (*request time out*) pada tabel 5.2 saat ping dilakukan dari computer publik ke computer privat.
2. Pengujian client ke client menghasilkan bahwa computer privat dapat dilindungi dari akses kemputer public dengan menggunakan NAT.
3. Komputer *server* dapat diakses oleh semua client, dengan atau tanpa NAT

Hasil pengujian menunjukkan secara keseluruhan jaringan telah terkonfigurasi dengan baik, dan berjalan sesuai dengan perencanaan.

### 5.2.2 Data Hasil Pengujian Protokol IAX dan SIP dalam Menembus NAT

Pengujian ini dilakukan dengan untuk mengetahui pengaruh penggunaan NAT pada jaringan VoIP. Pengujian dilakukan setelah melakukan pengujian pertama terhadap NAT telah dilakukan dan dapat berjalan sesuai dengan fungsinya. Pengujian dilakukan dengan menggunakan *server* dan *softphone* yang dapat mendukung kedua protokol , baik SIP maupun IAX

Data diambil dengan melihat panggilan pada setiap protokol, baik saat melewati NAT ataupun tidak. Kesuksesan panggilan dapat diketahui pada masing-masing protokol dengan cara mencoba melakukan panggilan dan melihat langsung apakah panggilan sukses (ditunjukkan dengan nada sambung dan nada dering) atau gagal (ditunjukkan dengan pesan *cleared/ no route*).



Gambar 5.2 Panggilan SIP ke SIP tanpa melewati NAT  
Sumber : Pengujian



Gambar 5.3 Panggilan SIP ke SIP melewati NAT  
Sumber : Pengujian





Gambar 5.4 Panggilan IAX ke IAX tanpa melewati NAT  
Sumber : Pengujian



Gambar 5.5 Panggilan IAX ke IAX melewati NAT  
Sumber : Pengujian

Pengaruh NAT pada suatu panggilan VoIP dapat dilihat pada gambar 5.2, 5.3, 5.4, dan 5.5 dengan melihat pada status panggilan (ditandai dengan lingkaran warna biru). Panggilan sukses ditandai dengan status “up” atau “waiting”, sedangkan panggilan yang gagal ditandai dengan status “down”. Data yang diperoleh dapat dilihat pada tabel 5.3

Tabel 5.3 Data Pengujian Protokol IAX dan SIP dalam Melalui NAT

Panggilan	Keterangan
SIP ke SIP tanpa melewati NAT	sukses
SIP ke SIP melewati NAT	gagal
IAX ke IAX tanpa melewati NAT	sukses
IAX ke IAX melewati NAT	sukses

Sumber : Pengujian

#### 5.2.2.1 Kesimpulan Hasil Pengujian

Pengujian protokol IAX dan SIP ini dilakukan untuk mengetahui pengaruh NAT pada jaringan VoIP dengan menggunakan protokol IAX dan SIP. Menurut tabel 5.3 didapatkan bahwa:

- Jaringan VoIP menggunakan protokol SIP terbukti tidak dapat menembus NAT.
- Jaringan VoIP menggunakan protokol IAX terbukti dapat menembus NAT

#### 5.2.3. Pengujian Performansi Jaringan VoIP Berbasis IAX

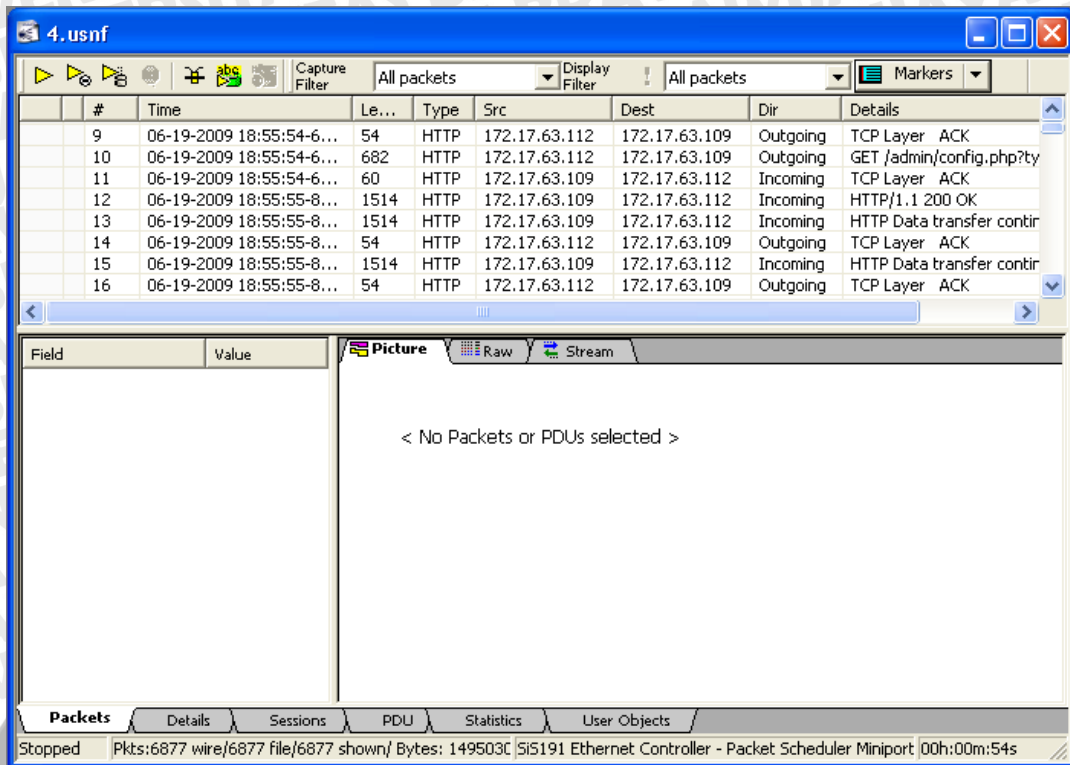
Pengujian ini secara umum untuk mengetahui pengaruh penggunaan protokol pada jaringan VoIP dalam melewati NAT dengan melihat performansi jaringan.

##### 5.2.3.1 Data Hasil Pengujian Delay End to End

Delay end to end merupakan delay yang berpengaruh pada kualitas suara yang diterima. Semakin besar *delay end to end*, mengakibatkan semakin buruk kualitas jaringan VoIP tersebut

Data yang didapatkan pada pengujian ini berupa file hasil *capture* dengan menggunakan *software* Wireshark dan Unsniff, seperti terlihat pada gambar 5.6.

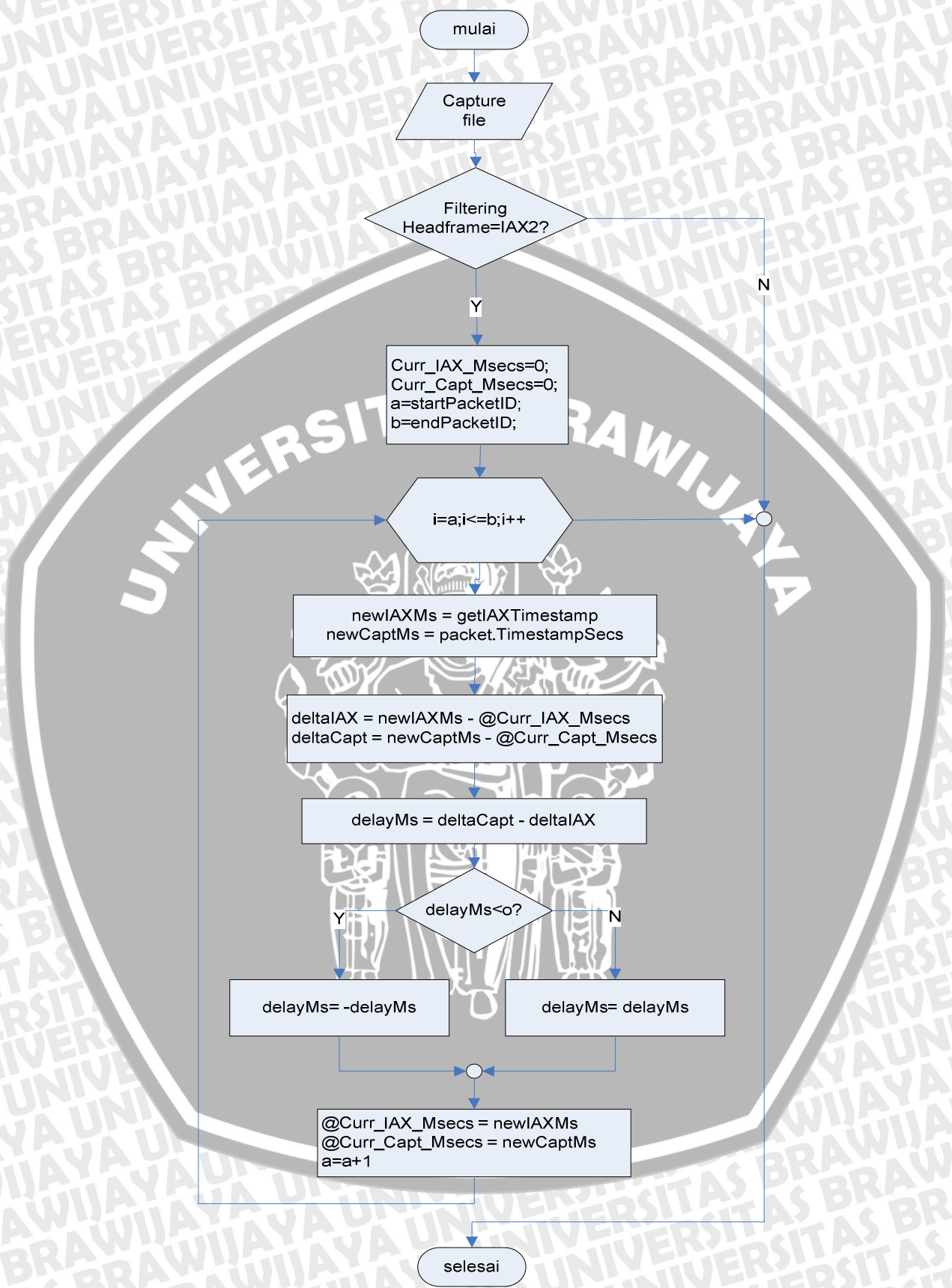




Gambar 5.6 Hasil *Capture* Unsniff  
Sumber : Pengujian

Hasil *capture* Unsniff dan Wireshark kemudian dianalisis satu persatu dengan melihat informasi yang terdapat dalam paket tersebut. Pada SIP, karena protokol SIP sudah lama dan sudah didukung sepenuhnya oleh Wireshark, pembacaan *delay end to end* dapat dilihat langsung pada statistik VoIP, dengan menambahkan *delay* dari client pemanggil ke *server* dan *delay* dari *server* ke *client* yang dipanggil.

Informasi tersebut diantaranya *timestamp*, *timecapture*, nomor paket, dan besar paket. Paket yang akan dianalisis adalah paket yang menggunakan protokol IAX, dan beberapa paket *signalling* lain yang diperlukan.



Gambar 5.7 Flowchart Prosedur Analisis Delay IAX  
Sumber : Perencanaan





Sesuai dengan *flowchart* analisis *delay* pada gambar 5.7, dapat dijelaskan bahwa perhitungan *delay* IAX dilakukan dengan menyaring paket yang akan dianalisis, yaitu paket dengan protokol IAX. Paket dilihat informasi *packet timestamp* dan IAX *timestamp* yang terdapat pada setiap paket.

Paket *timestamp* menunjukkan kapan paket tersebut diterima, dan IAX *timestamp* menunjukkan kapan paket tersebut seharusnya dikirim. Dengan mengetahui pewaktuan ini secara tepat, kita dapat mengetahui besarnya *delay* yang terjadi dalam paket tersebut. *Delay* pada tiap paket berbeda beda, karena itu pengambilan data dilakukan lebih dari 1 kali untuk mendapatkan *delay* rata-rata.

Standar *delay* untuk jaringan voip adalah:

- *Good* (0ms-150ms)
- *Acceptable* (150ms-300ms)
- *Poor* (> 300ms)

Dengan menghitung sesuai dengan data *timestamp* dan *timecapture* pada paket SIP dan IAX yang didapatkan dari hasil *capture* untuk masing-masing pengujian dengan dan tanpa NAT, 2 dan 8 *client*. Pada pengujian 1 pada 2 *client*, didapatkan:

- **SIP**

Sesuai dengan persamaan 2.3, *Delay* paket *end to end client* 1 ke client 4 :

$$= \text{Delay end to end client 1 ke client 4 (+) Delay server ke client 4}$$

$$= 15,53 \text{ ms} + 15,57 \text{ ms}$$

$$= 31,10 \text{ ms}$$

- **IAX**

Paket IAX tidak dapat dianalisis dengan menggunakan wireshark yang hanya dapat bekerja pada jaringan voip dengan menggunakan RTP. Oleh karena itu, perhitungan *delay* pada protokol IAX menggunakan Unsniff, yang dapat mengenali *mini frame* pada VoIP IAX. Sesuai dengan *flowchart* pada gambar 5.7 maka perhitungan *delay* sesuai dengan persamaan 2-4, jaringan VoIP menggunakan IAX pada pengujian 1 adalah:

$$\begin{aligned} \text{Banyak Paket pada codec G.711} &= 50 \text{ pps} \\ \text{Time sampling} &= \frac{1(\text{sekon})}{(\text{paket} \text{ _ per _ sekon})} \\ &= \frac{1(\text{sekon})}{50 \text{ pps}} = 20 \text{ ms/ paket} \end{aligned}$$

IAX *timestamp* pada paket yang dianalisis (paket ke 5095) = 15,020 s

Memiliki *capture time* pada 27756,22 ms

$$\begin{aligned}
 \text{IAX } t_{\text{timestamp}} \text{ paket berikutnya adalah} &= 15,020 \text{ s} + 20 \text{ ms} \\
 &= 15020 + 20 \text{ (ms)} \\
 &= 15040 \text{ ms} \\
 \text{capture time paket dengan } t_{\text{timestamp}} 15040 \text{ ms} &= 27773,11 \text{ ms} \\
 \text{sehingga } t_{\text{delay time}} &= 27773,11 - 27736,22 \text{ (ms)} \\
 &= 16,89 \text{ ms}
 \end{aligned}$$

Dengan melakukan perhitungan yang sama pada percobaan lainnya, didapatkan *delay end to end* seperti pada tabel 5.4:

Tabel 5.4 Hasil Analisis *Delay End to End*

pengujian ke	Tanpa Melalui NAT				Melalui NAT			
	IAX		SIP		IAX		SIP	
	2 client	8 client	2 client	8 client	2 client	8 client	2 client	8 client
	$t_{\text{endtoend}}$ (ms)	$t_{\text{endtoend}}$ (ms)	$t_{\text{endtoend}}$ (ms)	$t_{\text{endtoend}}$ (ms)	$t_{\text{endtoend}}$ (ms)	$t_{\text{endtoend}}$ (ms)	$t_{\text{endtoend}}$ (ms)	$t_{\text{endtoend}}$ (ms)
1	16,89	23,17	31,10	31,05	20,43	19,51	-	-
2	17,33	19,20	30,55	33,66	18,47	20,49	-	-
3	20,56	23,75	33,39	32,82	17,77	21,19	-	-
4	19,40	20,62	31,43	32,35	19,87	21,11	-	-
5	22,43	19,38	31,29	31,61	19,35	21,62	-	-
Rata-Rata	19,32	21,22	31,55	32,30	19,17	20,78	-	-

Sumber : Pengujian

### 1. Analisis *Delay* Enkapsulasi dan Dekapsulasi

Pada saat pembentukan hubungan (*connection established*) antara *client* dengan *server* terjadi kesepakatan dalam pemrosesan paket multimedia diantaranya adalah enkapsulasi dan dekapsulasi.

Dengan asumsi :

1. *IAX frame* 4 byte header + 160 byte *payload* (MSS)
2. *Header UDP* adalah 8 byte/paket
3. *Header IP* adalah 20 byte/paket
4. *Header Ethernet II* adalah 14 byte/paket

Maka analisis *delay* enkapsulasi dan dekapsulasinya adalah:



- **Analisis Delay Enkapsulasi**

Paket multimedia yang akan dikirim diberi *header* pada masing-masing *layer* TCP/IP. *Delay* enkapsulasi adalah waktu yang dibutuhkan untuk menambahkan keseluruhan *header* pada sebuah paket. Pada *transport layer*, paket multimedia diberi *header* UDP. Paket multimedia yang telah diberi *header* ini disebut segmen. Panjang segmen dihitung dengan menggunakan persamaan 2-6

$$W_{segment} = MSS + Header_{UDP}$$

Dengan mensubstitusikan nilai MSS dan *header* UDP diperoleh nilai panjang segmen

( $W_{segment}$ ):

$$\begin{aligned} W_{segment} &= MSS + Header_{UDP} \\ &= 164 + 8 \text{ (byte)} = 172 \text{ byte} \end{aligned}$$

Setelah melalui *transport layer*, paket melalui *internet layer*. Pada *internet layer*, paket multimedia diberi *header* IP. Paket multimedia yang telah diberi *header* IP disebut datagram. Panjang datagram dihitung dengan menggunakan persamaan 2-7

$$W_{datagram} = W_{segment} + Header_{IP}$$

Dengan mensubstitusikan nilai  $W_{segment}$  dan *header* IP diperoleh nilai panjang datagram ( $W_{datagram}$ ) sesuai persamaan 2-7

$$\begin{aligned} W_{datagram} &= W_{segment} + Header_{IP} \\ &= 172 + 20 \text{ (byte)} = 192 \text{ byte} \end{aligned}$$

Setelah itu paket melalui *network interface layer* baru kemudian dikirimkan ke *node* tujuan (*client*). Pada *network interface layer*, paket multimedia diberi *header* Ethernet II. Paket multimedia yang telah diberi *header* ini disebut *frame*. Panjang *frame* dihitung dengan menggunakan persamaan 2-8

$$W_{frame} = W_{datagram} + Header_{Ethernet}$$

Dengan mensubstitusikan nilai  $W_{datagram}$  dan *header* Ethernet II diperoleh nilai panjang

*frame* ( $W_{frame}$ ):

$$W_{frame} = W_{datagram} + Header_{Ethernet}$$

$$\begin{aligned}
 &= 192 + 14 \text{ (byte)} \\
 &= 206
 \end{aligned}$$

Delay enkapsulasi dihitung dengan menggunakan persamaan 2-9

$$t_{enc} = \frac{W_{frame} - MSS}{C_{pros}} \times 8$$

Dengan mensubstitusikan nilai  $W_{frame}$ , dan MSS serta mengasumsikan  $C_{pros}$  sama dengan kecepatan transmisi rata-rata yang diterima ke dalam persamaan 2-9, sesuai dengan kecepatan transmisi pada pengujian 1, maka *delay enkapsulasi* ( $t_{enc}$ ) adalah:

$$t_{enc} = \frac{W_{frame} - MSS}{C_{pros}} \times 8$$

Dengan mensubstitusikan nilai  $\lambda$  (*throughput*) dan  $l$  (kecepatan rata-rata) yang diperoleh dari hasil pengujian, ke dalam persamaan 2-2 diperoleh nilai kecepatan transmisi rata-rata diterima (D) :

$$D = \lambda \times l \times 8$$

Dari hasil pengujian 1 jaringan VoIP menggunakan IAX diketahui:

1. Jumlah paket yang diterima ( $N_{paket}$ ) = 6512 paket
2. Total data yang diterima  $M_{data\ diterima}$  = 1353758 byte
3. Total waktu transmisi ( $T_{total}$ ) = 57,450s

Dengan mensubstitusikan nilai  $t_v$  ke dalam persamaan 2-1 diperoleh nilai *throughput* ( $\lambda$ ) :

$$\begin{aligned}
 t_v &= \frac{T_{total}}{N_{paket}} \\
 &= \frac{57,450}{6512} \\
 &= 0,00882217 \text{ s / paket} \\
 \lambda &= \frac{1}{t_v} \\
 &= \frac{1}{0,00882217} \\
 &= 113,352 \text{ paket/s}
 \end{aligned}$$



$$\begin{aligned} \text{Panjang paket rata-rata : } l &= \frac{M_{\text{data diterima}}}{N_{\text{paket}}} \\ &= \frac{1353758}{6512} \\ &= 207,887 \text{ byte/paket} \end{aligned}$$

$$\begin{aligned} \text{Sehingga } D &= \lambda \times l \times 8 \\ &= 113,351 \times 207,887 \times 8 \\ &= 188,512.86 \text{ bps} \end{aligned}$$

Sesuai persamaan 2-10:

$$t_{\text{enc}} = \frac{W_{\text{frame}} - \text{MSS}}{C_{\text{pros}}} \times 8$$

$$\begin{aligned} t_{\text{enc}} &= \frac{206 - 160}{188.512,86} \times 8 \\ &= 0,001952 \text{ s} \end{aligned}$$

### 1. Analisis Delay Dekapsulasi

Paket multimedia yang sampai di *node* tujuan (*client*) akan dilepaskan *header*-nya pada masing-masing *layer* TCP/IP. *Delay* dekapsulasi adalah waktu yang dibutuhkan untuk melepaskan keseluruhan *header* dari sebuah paket. *Delay* dekapsulasi dihitung dengan menggunakan persamaan 2-10

$$t_{\text{dec}} = \frac{W_{\text{frame}} - \text{MSS}}{C_{\text{pros}}} \times 8$$

Dengan mensubstitusikan nilai  $W_{\text{frame}}$ , dan MSS serta mengasumsikan  $C_{\text{pros}2}$  sama dengan kecepatan transmisi rata-rata diterima ke dalam persamaan 2-10. Jika menggunakan *codec* G.711, maka kecepatan transmisi rata-rata diterima adalah 188,512.86 bps. Maka *delay* dekapsulasi :

$$t_{\text{enc}} = \frac{W_{\text{frame}} - \text{MSS}}{C_{\text{pros}}} \times 8$$

$$\begin{aligned} t_{\text{enc}} &= \frac{206 - 160}{188,512.86} \times 8 \\ &= 0,001952 \text{ s} \end{aligned}$$

## 2. Analisis Delay Transmisi

*Delay* transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket ke media transmisi. *Delay* transmisi dihitung dengan menggunakan persamaan 2-11

$$t_{trans} = \frac{(L+L')}{C} \times 8 = \frac{W_{frame}}{C} \times 8$$

Diketahui :

1. Panjang *frame* ( $W_{frame}$ ) adalah 206 byte
2. Kapasitas media transmisi (C) adalah 10 Mbps

Maka kapasitas media transmisi (C) dapat dituliskan kembali menjadi

$$C = 10.000 \text{ kbps}$$

Dengan mensubstitusikan nilai  $W_{frame}$ , dan C ke dalam persamaan 2-11 diperoleh nilai

*delay* transmisi ( $t_{trans}$ ) :

$$\begin{aligned} t_{trans} &= \frac{(L+L')}{C} \times 8 = \frac{W_{frame}}{C} \times 8 \\ &= \frac{206}{10.000} \times 8 \\ &= 0,001648 \end{aligned}$$

Sehingga *delay* transmisi ( $t_{trans}$ ) sebesar 0,001648 s

## 3. Delay Propagasi

*Delay* propagasi adalah waktu yang dibutuhkan untuk merambatkan sinyal yang berisi data multimedia dari *node* sumber (*server*) ke *node* tujuan (*client*). Dengan menghitung panjang kabel UTP CAT 5 untuk *server* dan *client* adalah 2x20 meter, jumlah *router* adalah 1 buah dengan *delay* 2 ms. Maka *delay* propagasi dengan menggunakan persamaan 2-12 adalah:

$$\begin{aligned} t_p &= t_{UTP} + t_{router} + t_{DTE} \\ &= 0,556 \times 40 \times \frac{1}{10000} + 0,002 + 0,001 \\ &= 0.00524 \text{ s} \end{aligned}$$

## 4. Delay Antrian

*Delay* antrian terjadi karena keterbatasan *client* dalam memproses paket multimedia yang datang. *Delay* antrian ditentukan dengan Persamaan (2-17)



$$t_w = \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu}$$

$$\mu = \frac{C}{l \times 8}$$

$$= \frac{10000000}{207,887 \times 8}$$

$$= \frac{10000000}{1663,096} = 6012,881 \text{ paket/s}$$

$$t_w = \frac{113,351}{6012,881(6012,881 - 113,351)} + \frac{1}{6012,881}$$

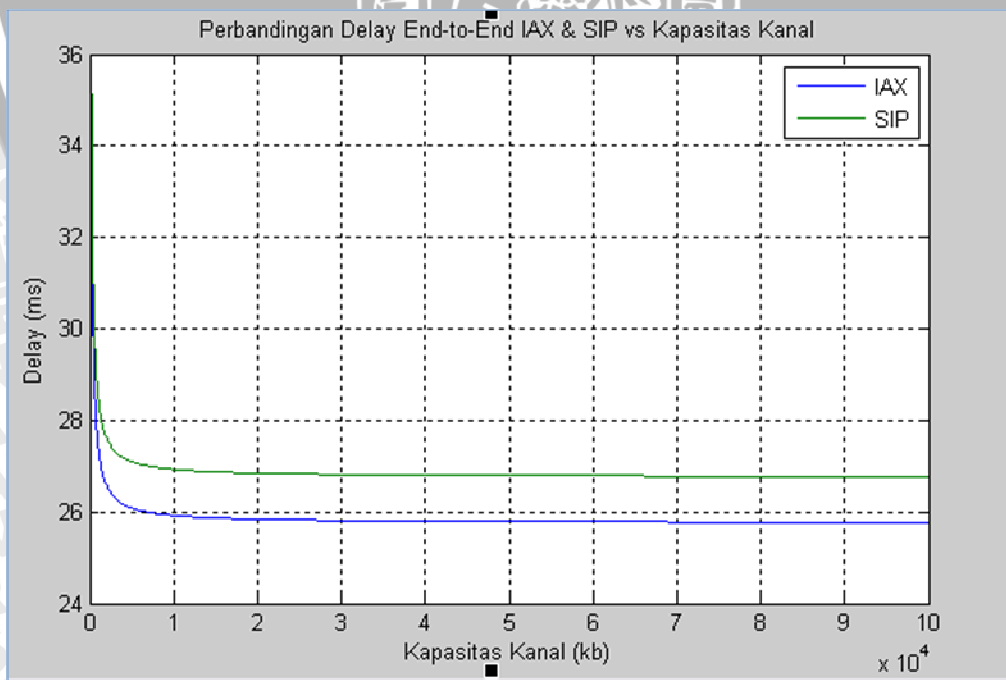
$$= \frac{113,351}{35473171,845} + \frac{1}{6012,881}$$

$$= 0.0169 \text{ detik}$$

Sehingga *delay end to end* sistem adalah

$$\begin{aligned} t_{\text{endtoend}} &= t_{\text{enc}} + t_{\text{trans}} + t_p + t_w + t_{\text{dec}} \\ &= 0,001952 + 0,001952 + 0,01648 + 0,00524 + 0,00169 \\ &= 0.027314 \text{ detik} \end{aligned}$$

Perhitungan *delay end to end* dengan menggunakan matlab dapat dilihat sebagai berikut:



Gambar 5.8 Perhitungan *Delay End to End* Menggunakan Matlab  
 Sumber : Pengujian

Gambar 5.8 menunjukkan perbandingan *delay end to end* jaringan VoIP dengan menggunakan protokol IAX dan SIP. Dalam gambar dapat diketahui bahwa semakin besar kapasitas kanal yang disediakan, maka semakin kecil *delay* yang terjadi, dan sebaliknya, *delay* tinggi terjadi saat kapasitas kanal yang disediakan rendah.

*Delay end to end* pada jaringan VoIP dengan menggunakan IAX lebih rendah daripada jaringan VoIP dengan menggunakan SIP pada kecepatan transmisi sama. Hal ini dikarenakan jaringan VoIP dengan menggunakan SIP menggunakan lebih banyak *header frame*, sedangkan IAX lebih efisien dalam menggunakan *header*.

### 5.2.3.1.1 Kesimpulan Hasil Pengujian

Kesimpulan yang diambil dari pengujian ini adalah bahwa *delay* jaringan VoIP memenuhi standar VoIP. Pengujian *delay end to end*, didapatkan *delay* pada jaringan VoIP menggunakan protokol IAX lebih kecil daripada *delay* pada jaringan VoIP dengan menggunakan SIP. *delay* yang terjadi pada IAX sebesar rata-rata 19,32 ms, sedang pada SIP *delay* yang terjadi rata-rata 31,55 ms.

### 5.2.3.2 Data Hasil Pengujian Jitter

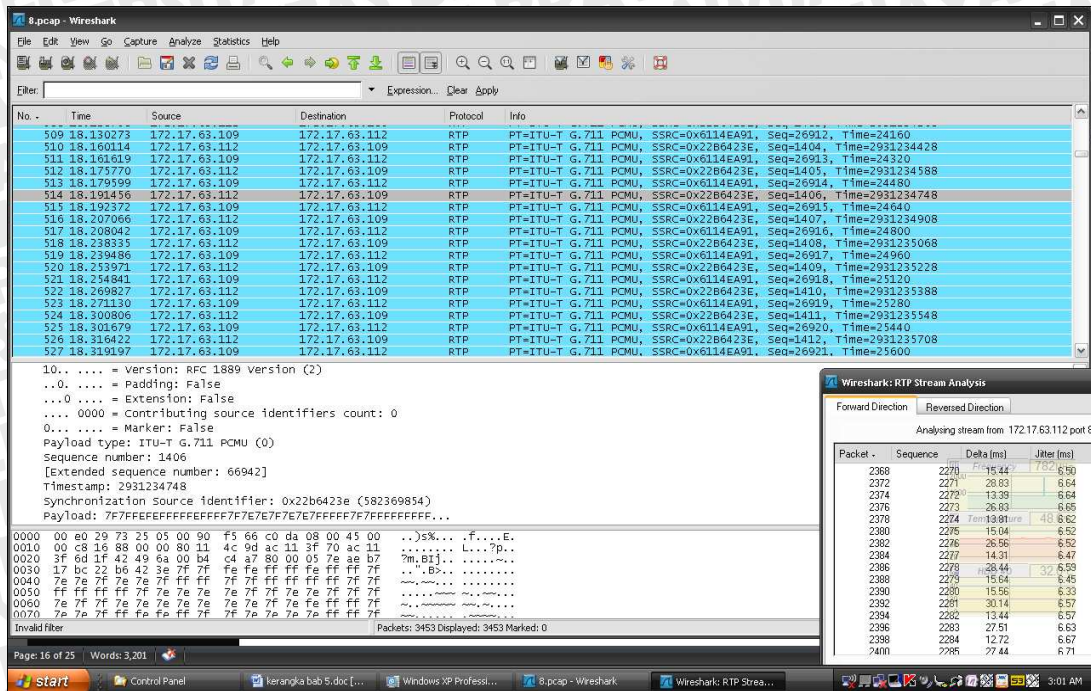
Seperti yang telah disebut sebelumnya, bahwa *delay* setiap paket berbeda-beda. Hal inilah yang kemudian menimbulkan *jitter*, yaitu perbedaan waktu *delay* antara paket pada ujung penerima. *Jitter* termasuk dalam salah satu parameter performansi VoIP, karena perbedaan *delay* antara paket suara yang diterima menyebabkan gangguan yang mengakibatkan suara yang diterima menjadi tidak jelas.

Standar *jitter* untuk VoIP adalah:

- *Good (0ms-20ms)*
- *Acceptable (20ms-50ms)*
- *Poor (> 50ms)*

Analisis *jitter* dapat dijelaskan bahwa secara umum untuk mendapatkan *jitter* adalah dengan mengetahui *delay* yang terjadi pada paket. *Delay* pada paket yang berurutan dibandingkan satu dan yang lainnya sehingga dapat diketahui selisih waktu yang terjadi.





Gambar 5.9 Hasil Pengujian Jitter  
Sumber: pengujian

Dengan menghitung sesuai dengan persamaan 2-18, dengan data *timestamp* dan *timecapture* pada paket SIP dan IAX yang didapatkan dari hasil *capture* untuk masing-masing pengujian dengan dan tanpa NAT, 2 dan 8 *client*, pada pengujian 1 dengan 2 *client* dapat dihitung:

- SIP

Jitter pada SIP dapat langsung diketahui dengan menganalisis RTP menggunakan Wireshark.

$$\begin{aligned}
 \text{Jitter paket client 1 ke client 4} &= \text{jitter client 1 ke client 4 (+) jitter ke client 4} \\
 &= 6,68 \text{ ms} + 6,74 \text{ ms} \\
 &= 13,42 \text{ ms}
 \end{aligned}$$

- IAX

IAX *timestamp* pada paket pertama yang dianalisis (paket ke 5095) = 15,020 s

Memiliki *capture time* pada 27756,22 ms

$$\begin{aligned}
 \text{IAX } \textit{timestamp} \text{ paket berikutnya adalah} &= 15,020 \text{ s} + 20 \text{ ms} \\
 &= 15020 + 20 \text{ (ms)} \\
 &= 15040 \text{ ms}
 \end{aligned}$$

$$\text{capture time paket dengan } \textit{timestamp} \text{ 15040 ms} = 27773,11 \text{ ms}$$

$$\begin{aligned} \text{sehingga } \textit{delay time} (1) &= 27773,11 - 27736,22 \text{ (ms)} \\ &= 16,89 \text{ ms} \end{aligned}$$

$$\text{IAX } \textit{timestamp} \text{ pada paket kedua yang dianalisis (paket ke 5103)} = 15,060 \text{ s}$$

Memiliki *capture time* pada 27789,65 ms

$$\begin{aligned} \text{IAX } \textit{timestamp} \text{ paket berikutnya adalah} &= 15,060 \text{ s} + 20 \text{ ms} \\ &= 15060 + 20 \text{ (ms)} \\ &= 15080 \text{ ms} \end{aligned}$$

$$\textit{capture time} \text{ paket dengan } \textit{timestamp} \text{ 15080 ms} = 27808,51 \text{ ms}$$

$$\begin{aligned} \text{sehingga } \textit{delay time} (2) &= 27808,51 - 27789,65 \text{ (ms)} \\ &= 18,86 \text{ ms} \end{aligned}$$

$$\begin{aligned} \textit{Jitter} = \text{delta } \textit{delay} &= \textit{delay time} (1) - \textit{delay time} (2) \\ &= 16,89 - 18,86 \text{ (ms)} \\ &= 1,97 \text{ ms} \end{aligned}$$

Dengan melakukan perhitungan yang sama pada percobaan lain, maka *jitter* pada jaringan VoIP didapatkan hasil:

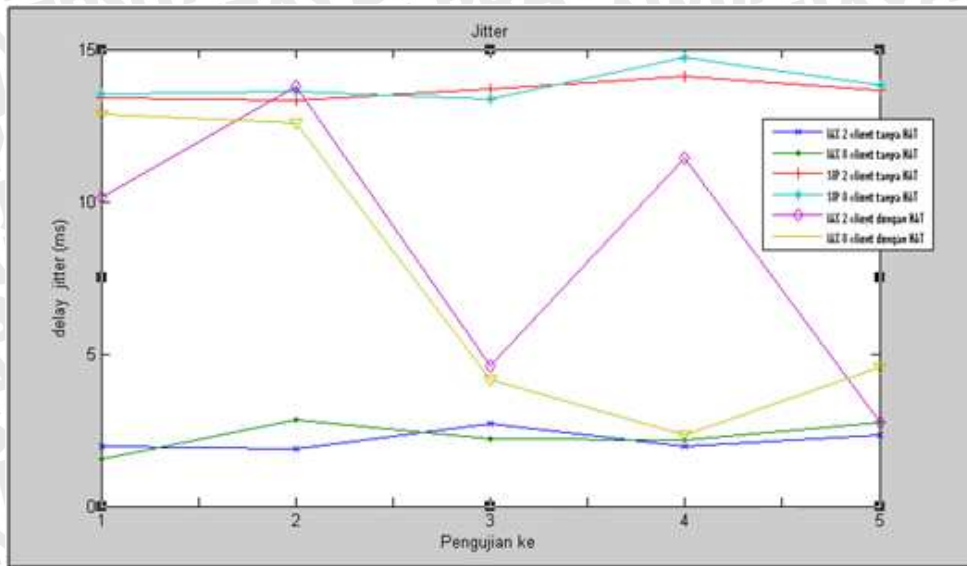
Tabel 5.5 Hasil Analisis *Jitter*

pengujian ke	Tanpa Melalui NAT				Melalui NAT			
	IAX		SIP		IAX		SIP	
	2 client	8 client	2 client	8 client	2 client	8 client	2 client	8 client
	$\Delta t$ (ms)	$\Delta t$ (ms)	$\Delta t$ (ms)	$\Delta t$ (ms)	$\Delta t$ (ms)	$\Delta t$ (ms)	$\Delta t$ (ms)	$\Delta t$ (ms)
1	1,97	1,53	13,42	13,55	10,13	12,87	-	-
2	1,88	2,81	13,31	13,61	13,77	12,60	-	-
3	2,72	2,21	13,71	13,36	4,59	4,16	-	-
4	1,98	2,19	13,27	14,74	11,43	2,35	-	-
5	2,34	2,75	14,10	13,81	2,76	4,56	-	-
Rata-Rata	2,18	2,30	13,56	13,67	8,37	7,31	-	-

Sumber : pengujian

Dari table 5.5 dapat dilihat bahwa IAX memberikan nilai *jitter* yang lebih kecil daripada SIP. Hal ini dikarenakan IAX menggunakan paket yang sama untuk *signaling* dan paket *audip*. Selain itu paket IAX menggunakan *binary* untuk mengurangi *size* pada tiap paket.





Gambar 5.10 Grafik Hasil Pengujian Jitter

Sumber : Pengujian

Grafik tersebut memberikan gambaran jitter yang terjadi pada jaringan VoIP. Jitter pada jaringan VoIP dengan menggunakan protokol IAX, baik melalui NAT atau tidak, memberikan jitter lebih kecil daripada jaringan SIP. Grafik ini diperoleh dari hasil pengujian.

#### 5.2.3.2.1 Kesimpulan Hasil Pengujian

Pada pengujian *jitter*, IAX memberikan nilai yang jauh lebih baik daripada SIP, yaitu rata-rata 2,18 ms. Sedangkan SIP sebesar rata-rata 13,56 ms. Kedua protokol masih sesuai dengan standar VoIP

#### 5.2.3.3 Data Hasil Pengujian Paket Loss

Pada *flowchart* analisis *loss* dapat dijelaskan bahwa paket *loss* diambil hanya pada paket dengan protokol IAX. Paket *loss* menyatakan banyaknya paket yang diterima salah dibanding dengan jumlah total paket yang dikirimkan dengan satuan paket per detik atau dalam satuan per seratus paket (persen).

Time	Le...	Type	Src	Dest	Dir	Details	
5093	06-19-2009 04:46:27-748563	206	IAX2	172.17.63.109	172.17.63.98	Sniffed	Mini frame (call #: 2) codec: G.711 mu-law
5094	06-19-2009 04:46:27-755933	206	IAX2	172.17.63.98	172.17.63.109	Sniffed	Mini frame (call #: 211) codec: G.711 mu-law
5095	06-19-2009 04:46:27-756223	206	IAX2	172.17.63.109	172.17.63.112	Incoming	Mini frame (call #: 1) codec: G.711 mu-law
5096	06-19-2009 04:46:27-760767	206	IAX2	172.17.63.112	172.17.63.109	Outgoing	Mini frame (call #: 240) codec: G.711 mu-law
5097	06-19-2009 04:46:27-762384	206	IAX2	172.17.63.109	172.17.63.98	Sniffed	Mini frame (call #: 2) codec: G.711 mu-law
5098	06-19-2009 04:46:27-772814	206	IAX2	172.17.63.98	172.17.63.109	Sniffed	Mini frame (call #: 211) codec: G.711 mu-law
5099	06-19-2009 04:46:27-773111	206	IAX2	172.17.63.109	172.17.63.112	Incoming	Mini frame (call #: 1) codec: G.711 mu-law
5100	06-19-2009 04:46:27-787601	206	IAX2	172.17.63.112	172.17.63.109	Outgoing	Mini frame (call #: 240) codec: G.711 mu-law
5101	06-19-2009 04:46:27-788274	206	IAX2	172.17.63.109	172.17.63.98	Sniffed	Mini frame (call #: 2) codec: G.711 mu-law
5102	06-19-2009 04:46:27-789347	206	IAX2	172.17.63.98	172.17.63.109	Sniffed	Mini frame (call #: 211) codec: G.711 mu-law
5103	06-19-2009 04:46:27-789652	206	IAX2	172.17.63.109	172.17.63.112	Incoming	Mini frame (call #: 1) codec: G.711 mu-law
5104	06-19-2009 04:46:27-801183	206	IAX2	172.17.63.112	172.17.63.109	Outgoing	Mini frame (call #: 240) codec: G.711 mu-law
5105	06-19-2009 04:46:27-801893	206	IAX2	172.17.63.109	172.17.63.98	Sniffed	Mini frame (call #: 2) codec: G.711 mu-law
5106	06-19-2009 04:46:27-808125	206	IAX2	172.17.63.98	172.17.63.109	Sniffed	Mini frame (call #: 211) codec: G.711 mu-law
5107	06-19-2009 04:46:27-808510	206	IAX2	172.17.63.109	172.17.63.112	Incoming	Mini frame (call #: 1) codec: G.711 mu-law
5108	06-19-2009 04:46:27-824491	206	IAX2	172.17.63.98	172.17.63.109	Sniffed	Mini frame (call #: 211) codec: G.711 mu-law
5109	06-19-2009 04:46:27-824785	206	IAX2	172.17.63.109	172.17.63.112	Incoming	Mini frame (call #: 1) codec: G.711 mu-law
5110	06-19-2009 04:46:27-827882	206	IAX2	172.17.63.112	172.17.63.109	Outgoing	Mini frame (call #: 240) codec: G.711 mu-law
5111	06-19-2009 04:46:27-828657	206	IAX2	172.17.63.109	172.17.63.98	Sniffed	Mini frame (call #: 2) codec: G.711 mu-law
5112	06-19-2009 04:46:27-841330	206	IAX2	172.17.63.98	172.17.63.109	Sniffed	Mini frame (call #: 211) codec: G.711 mu-law
5113	06-19-2009 04:46:27-841488	206	IAX2	172.17.63.109	172.17.63.112	Incoming	Mini frame (call #: 1) codec: G.711 mu-law
5114	06-19-2009 04:46:27-841850	206	IAX2	172.17.63.112	172.17.63.109	Outgoing	Mini frame (call #: 240) codec: G.711 mu-law
5115	06-19-2009 04:46:27-842482	206	IAX2	172.17.63.109	172.17.63.98	Sniffed	Mini frame (call #: 2) codec: G.711 mu-law
5116	06-19-2009 04:46:27-857685	206	IAX2	172.17.63.98	172.17.63.109	Sniffed	Mini frame (call #: 211) codec: G.711 mu-law
5117	06-19-2009 04:46:27-858045	206	IAX2	172.17.63.109	172.17.63.112	Incoming	Mini frame (call #: 1) codec: G.711 mu-law
5118	06-19-2009 04:46:27-869333	206	IAX2	172.17.63.112	172.17.63.109	Outgoing	Mini frame (call #: 240) codec: G.711 mu-law
5119	06-19-2009 04:46:27-870140	206	IAX2	172.17.63.109	172.17.63.98	Sniffed	Mini frame (call #: 2) codec: G.711 mu-law
5120	06-19-2009 04:46:27-878073	206	IAX2	172.17.63.98	172.17.63.109	Sniffed	Mini frame (call #: 211) codec: G.711 mu-law
5121	06-19-2009 04:46:27-878334	206	IAX2	172.17.63.109	172.17.63.112	Incoming	Mini frame (call #: 1) codec: G.711 mu-law

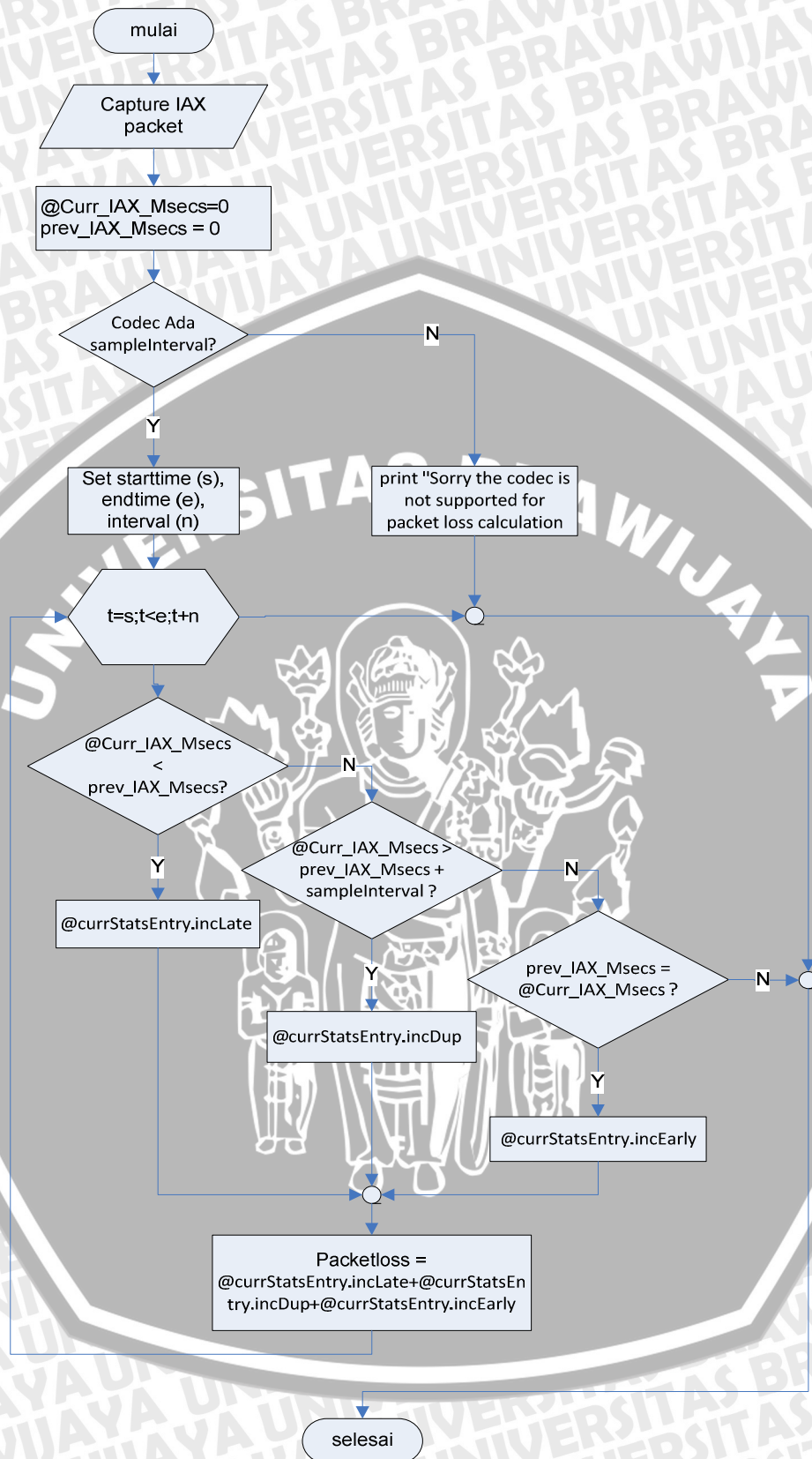
Gambar 5.11 Analisis Aliran Paket IAX  
Sumber : pengujian

Standar *loss* untuk jaringan VoIP adalah :

- *Good* (0%-0.5%)
- *Acceptable* (0.5%-1.5%)
- *Poor* (> 1:5%)

Dalam *flowchart* diperlihatkan bahwa pencarian paket yang hilang dilakukan dengan melihat *timestamp* yang ada dibandingkan dengan waktu kapan seharusnya paket tersebut datang. Hal ini sesuai dengan ketentuan VoIP yang menyatakan bahwa paket yang datang terlalu cepat, terlalu lambat atau mengirimkan paket yang sama dalam waktu yang berbeda, pakatnya akan segera dibuang, karena tidak memiliki nilai informasi lagi. Paket *loss* didapatkan dengan cara membagi paket yang hilang dengan paket total yang dikirim.





Gambar 5.12 Flowchart Analisis Loss  
Sumber : Perencanaan

Perhitungan paket *loss* pada jaringan VoIP berbasis SIP menggunakan wireshark dan hasilnya dapat langsung diketahui. Berbeda dengan IAX yang belum banyak mendapat dukungan dari *software sniffer*. Untuk itu perhitungan paket *loss* pada IAX dilakukan secara manual dengan melihat *timestamp* dan *time capture*.

Packet #	Sequence	Delta (ms)	Jitter (ms)	IP BW (kbps)	Marker	Status
161	20690	0.07	11.25	22.40		[Ok]
162	20690	0.34	11.78	24.00		[Ok]
164	20691	4.97	11.98	25.60		[Ok]
166	20692	19.31	11.28	27.20		[Ok]
170	20693	20.65	10.61	28.80		[Ok]
172	20694	20.65	9.99	30.40		[Ok]
176	20695	20.51	9.40	32.00		[Ok]
179	20696	18.90	8.88	33.60		[Ok]

Max delta = 0.050428 sec at packet no. 2022  
 Total RTP packets = 1654 (expected 1655) Lost RTP packets = 1 (0.06%) Sequence errors = 1

Gambar 5.13 Paket *Loss* SIP Menggunakan Wireshark  
 Sumber : pengujian

Dengan menghitung sesuai dengan persamaan 2-19 , dengan data *timestamp* dan *time capture* dibandingkan pada paket yang dikirim, pada paket SIP dan IAX yang didapatkan dari hasil *capture* untuk masing-masing pengujian tanpa dan menggunakan NAT, 2 dan 8 *client*, pada paket RTP untuk SIP dan IAX2 untuk IAX pada tabel 5.4 pengujian 1:

Jumlah paket yang diterima = 3398 Paket

Jumlah paket yang rusak atau tidak sesuai = 0 paket

$$\begin{aligned} \text{Sehingga paket loss} &= \frac{\sum \text{error}}{\sum \text{diterima}} \times 100\% \\ &= \frac{0}{3398} \times 100\% \\ &= 0\% \end{aligned}$$

Pada pengujian didapatkan hasil seperti pada tabel 5.6 dan 5.7:



Tabel 5.6 Hasil Analisis Paket Loss Tanpa Menggunakan NAT

penguji- an ke	TANPA NAT											
	IAX						SIP					
	2 client			8 client			2 client			8 client		
	$\Sigma$ terima	$\Sigma$ Error	Paket loss	$\Sigma$ terima	$\Sigma$ Error	Paket loss	$\Sigma$ terima	$\Sigma$ Error	Paket loss	$\Sigma$ terima	$\Sigma$ Error	Paket loss
	(paket)	(paket)	%	(paket)	(paket)	%	(paket)	(paket)	%	(paket)	(paket)	%
1	3398	0	0	3452	0	0	1616	0	0,00	1692	0	0
2	3207	0	0	3602	0	0	1624	0	0,00	1787	0	0
3	2989	0	0	3771	0	0	1655	1	0,06	1657	0	0
4	3387	0	0	3321	0	0	1528	0	0,00	1533	0	0
5	3211	0	0	3928	0	0	1620	0	0,00	1684	0	0

Sumber : Pengujian

Tabel 5.7 Hasil Analisis Paket Loss Menggunakan NAT

penguji- an ke	MENGUNAKAN NAT											
	IAX						SIP					
	2 client			8 client			2 client			8 client		
	$\Sigma$ terima	$\Sigma$ Error	Paket loss	$\Sigma$ terima	$\Sigma$ Error	Paket loss	$\Sigma$ terima	$\Sigma$ Error	Paket loss	$\Sigma$ terima	$\Sigma$ Error	Paket loss
	(paket)	(paket)	%	(paket)	(paket)	%	(paket)	(paket)	%	(paket)	(paket)	%
1	3039	0	0	3419	0	0	-	-	-	-	-	-
2	3122	0	0	3311	0	0	-	-	-	-	-	-
3	3227	0	0	3216	0	0	-	-	-	-	-	-
4	3198	0	0	3241	0	0	-	-	-	-	-	-
5	3111	1	0,03	3328	0	0	-	-	-	-	-	-

Sumber : Pengujian

Dalam tabel 5.6 dan tabel 5.7 menunjukkan bahwa tidak ada pengaruh signifikan yang diberikan oleh jaringan VoIP yang menggunakan IAX atau SIP, karena paket loss lebih dipengaruhi oleh media transmisi yang digunakan dan jumlah *node* yang dilewati. Karena kedua protokol melewati media yang sama dan jumlah *node* yang sama pula, maka kedua protokol VoIP memberikan nilai yang hampir sama pada nilai paket loss.

### 5.2.3.3.1 Kesimpulan Hasil Pengujian

Pada pengujian paket loss, keduanya memberikan nilai yang hampir sama, yaitu 0 %. Paket loss SIP dan IAX memenuhi standar VoIP.

#### 5.2.3.4. Data Hasil Pengujian *Bandwidth* dan Trafik

Trafik adalah banyaknya paket yang menduduki sebuah kanal dan membuat jaringan tersebut sibuk dalam selang waktu tertentu. Karena jaringan yang digunakan berbasis IP, maka jumlah kanal berhubungan erat dengan *bandwidth* yang disediakan.

Dari hasil pengujian 1 jaringan VoIP menggunakan IAX sesuai dengan tabel 5.9 dapat diketahui :

1. Jumlah paket yang diterima ( $N_{paket}$ ) = 6512 paket.
2. Total waktu transmisi ( $T_{total}$ ) = 57,450 s

Dengan mensubstitusikan nilai  $t_v$  ke dalam persamaan 2-1 diperoleh nilai *throughput* ( $\lambda$ ) :

$$t_v = \frac{T_{total}}{N_{paket}}$$

$$= \frac{57,450}{6512}$$

$$= 0,00882217 \text{ s / paket}$$

$$\lambda = \frac{1}{t_v}$$

$$= \frac{1}{0,00882217}$$

$$= 113,352 \text{ paket/s}$$

Sehingga untuk pengujian 1. pada jaringan VoIP menggunakan IAX diperoleh nilai *throughput* ( $\lambda$ ) sebesar 113,352 paket/s.

Tabel 5.8 *Throughput* pada Jaringan VoIP menggunakan IAX

Pengujian Ke	Durasi ( $T_{total}$ ) (s)	Jumlah Paket ( $N_{paket}$ ) (paket)	$t_{rata2}$ per paket $t_v$ (s)	<i>Throughput</i> ( $\lambda$ ) (paket/s)
1	57.450	6512	0.00882217	113.351
2	47.427	3533	0.01342400	74.493
3	39.679	3651	0.01086798	92.013
4	48.819	3619	0.01348964	74.131
5	49.477	3711	0.01333252	75.005

Sumber : Pengujian



Dengan perhitungan yang sama, pengujian 1 pada jaringan VoIP menggunakan SIP diperoleh nilai *throughput* ( $\lambda$ ) sebesar 131,426 paket/s

Tabel 5.9 *Throughput* pada Jaringan VoIP menggunakan SIP

Pengujian Ke	Durasi ( $T_{total}$ ) (s)	Jumlah Paket ( $N_{paket}$ ) (paket)	$t_{rata}$ per paket ( $t_v$ ) (s)	<i>Throughput</i> ( $\lambda$ ) (paket/s)
1	52.501	6900	0.00760884	131.426
2	50.674	6625	0.00764891	130.738
3	53.640	6791	0.00789869	126.603
4	51.612	3453	0.01494700	66.903
5	49.448	5768	0.00857282	116.648

Sumber : Pengujian

Dalam analisis penggunaan *bandwidth* rata-rata, dari hasil pengujian 1, yang terlihat dalam tabel 5.9 pada jaringan VoIP menggunakan protokol IAX diketahui :

1. Panjang data yang diterima ( $M_{data\ diterima}$ ) = 1353758 byte
2. Jumlah paket yang diterima ( $N_{paket}$ ) = 6512 paket
3. *Throughput* ( $\lambda$ ) = 113.351 paket/s

Panjang paket rata-rata ( $l$ ) dihitung dengan membagi panjang data yang diterima dengan jumlah paket yang diterima:

$$\begin{aligned}
 l &= \frac{M_{data\ diterima}}{N_{paket}} \\
 &= \frac{1353758}{6512} \\
 &= 207,887 \text{ byte/paket}
 \end{aligned}$$

Dengan mensubstitusikan nilai  $\lambda$  dan  $l$  yang telah dihitung sebelumnya ke dalam persamaan 2-2 diperoleh nilai kecepatan transmisi rata-rata diterima ( $D$ )

$$\begin{aligned}
 D &= \lambda \times l \times 8 \\
 &= 113,351 \times 207,887 \times 8 \\
 &= 188,512.86 \text{ bps}
 \end{aligned}$$

Dengan cara perhitungan yang sama untuk percobaan yang lain, didapatkan hasil pada tabel 5.10

Tabel 5.10 Kecepatan Transmisi Rata – Rata Diterima Setiap *Client* pada Jaringan VoIP

No	Protokol	Durasi ( $T_{total}$ ) (s)	Data Diterima ( $M_{data\_diterima}$ ) (byte)	Jumlah Paket ( $N_{paket}$ ) (paket)	Panjang Paket Rata-Rata ( $\ell$ ) (byte)	$t_{rata2}$ per paket $t_v$ (s)	Throughput ( $\lambda$ ) (paket/s)	Kecepatan Transmisi Rata-rata diterima ( $D$ ) (bps)
1	IAX	57.450	1353758	6512	207.887	0.00882217	113.351	188,512.86
2	IAX	47.427	732273	3533	207.267	0.01342400	74.493	123,520.02
3	IAX	39.679	755888	3651	207.036	0.01086798	92.013	152,400.61
4	IAX	48.819	769759	3619	212.699	0.01348964	74.131	126,140.89
5	IAX	49.477	775857	3711	209.070	0.01333252	75.005	125,449.32
6	SIP	52.501	1506198	6900	218.290	0.00760884	131.426	229,511.51
7	SIP	50.674	1464654	6625	221.080	0.00764891	130.738	231,227.69
8	SIP	53.640	1560289	6791	229.758	0.00789869	126.603	232,705.29
9	SIP	51.612	743902	3453	215.436	0.01494700	66.903	115,306.83
10	SIP	49.448	1383949	5768	239.936	0.00857282	116.648	223,903.74

Sumber : pengujian

- **AHT (Average Hold Time)**

*AHT* merupakan total waktu yang digunakan oleh semua panggilan pada periode tertentu dibagi dengan banyaknya panggilan. *AHT* dapat dihitung dengan menggunakan persamaan (2.20)

$$AHT = \frac{\Sigma call\_seconds}{\Sigma calls}$$

Sesuai dengan table 5.8, didapatkan:

$$\begin{aligned}
 AHT &= \frac{\Sigma call\_seconds}{\Sigma calls} \\
 &= \frac{57,450 + 47,427 + 39,679 + 48,819 + 49,477 + 52,501 + 50,674 + 53,640 + 51,612 + 49,448}{10} \\
 &= \frac{500,727}{10} \\
 &= 50,07 \text{ s}
 \end{aligned}$$

- **Traffic Load**

*Traffic load* merupakan perbandingan antara banyaknya panggilan yang datang dengan waktu rata-rata yang digunakan untuk melayani setiap panggilan pada periode waktu tertentu. *Traffic load* dapat dihitung dengan menggunakan persamaan (2.21).

$$A = \frac{\Sigma calls \times AHT}{3600}$$



$$\begin{aligned}
 &= \frac{10 \times 50,07}{3600} \\
 &= \frac{500,727}{3600} \\
 &= 0,140 \text{ erlang}
 \end{aligned}$$

Dengan menggunakan tabel erlang B, diketahui bahwa nilai N/B adalah 1,00 Dimana N adalah jumlah *server* , dalam skripsi ini berjumlah 1 dan B adalah probabilitas *blocking* dalam persen, yaitu 1%

- **Kapasitas Jaringan**

Kapasitas jaringan bergantung pada kapasitas transmisi. Kapasitas transmisi merupakan banyaknya saluran komunikasi yang dapat digunakan secara bersama-sama untuk melakukan proses transmisi data atau banyaknya proses komunikasi yang mungkin terjadi dalam waktu yang sama. Kapasitas transmisi dapat dihitung dengan menggunakan persamaan (2.25).

$$K_T = \frac{BW_{ST}}{BW_T}$$

Kapasitas media transmisi yang digunakan ( $BW_{ST}$ ) adalah 10 Mbps atau sama dengan 10.000.000 bps. Jumlah panggilan dalam jaringan sangat dipengaruhi oleh *bandwidth*. *Bandwidth* transmisi ( $BW_T$ ) merupakan *bandwidth* yang dibutuhkan paket suara yang ditransmisikan dalam periode satu detik. *Bandwidth* transmisi dapat dihitung dengan menggunakan persamaan (2.24).

$$BW_T = L_{PI} \times pps$$

Paket suara per second (*pps*) dihitung dengan menggunakan persamaan (2.23)

$$pps = \frac{BR}{P}$$

Keterangan:

- a. Bit rate *codec* (*BR*) G.711 adalah 64 kbps
- b. Payload *codec* (*P*) G.711 adalah 160 Byte

$$\begin{aligned} pps &= \frac{BR}{P} = \frac{64kbps}{160 \times 8} \\ &= 50 \text{ paket/s} \end{aligned}$$

Panjang paket internet rata-rata:

$$\begin{aligned} L_{PI} &= \frac{207,887 + 207,267 + 207,036 + 212,699 + 209,070}{5} \\ &= \frac{1043,959}{5} \\ &= 208,792 \text{ byte} \end{aligned}$$

Dengan mensubstitusikan panjang paket internet rata-rata ke persamaan (2.24), maka nilai *bandwidth* transmisi adalah:

$$\begin{aligned} BW_T &= L_{PI} \times pps \\ &= (208,792 \times 8) \times 50 \\ &= 83.516,72 \text{ bps} \end{aligned}$$

Kapasitas transmisi dapat dihitung dengan mensubstitusikan nilai  $BW_T$  pada persamaan (2.25):

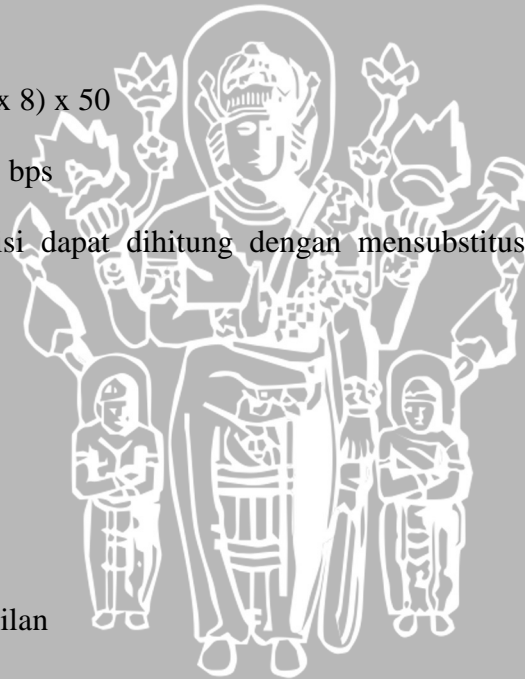
$$\begin{aligned} K_T &= \frac{BW_{ST}}{BW_T} \\ &= \frac{10.000.000}{83.516,72} \\ &= 119,7 \text{ Panggilan} \end{aligned}$$

- **Jumlah panggilan maksimal jaringan VoIP menggunakan protokol SIP**

Seperti pada perhitungan jumlah panggilan maksimal jaringan VoIP dengan menggunakan Protokol IAX, maka jumlah panggilan maksimal jaringan VoIP menggunakan protokol SIP dapat dihitung sebagai berikut:

Panjang paket rata-rata:

$$L_{PI} = \frac{218,29 + 221,08 + 229,758 + 215,436 + 239,936}{5}$$





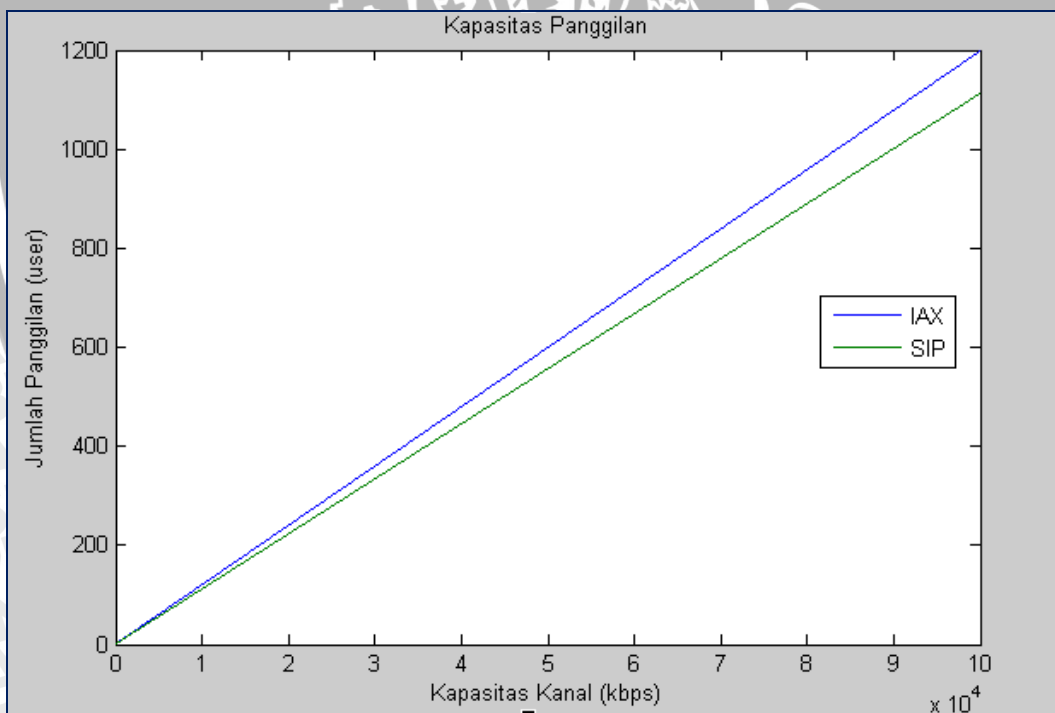
$$= \frac{1124,5}{5} = 224,9 \text{ byte}$$

Dengan mensubstitusikan panjang paket internet rata-rata ke persamaan (2.24), maka nilai *bandwidth* transmisi adalah :

$$\begin{aligned} BW_T &= L_{PI} \times pps \\ &= (224,9 \times 8) \times 50 \\ &= 89.960 \text{ bps} \end{aligned}$$

Kapasitas transmisi dapat dihitung dengan mensubstitusikan nilai  $BW_T$  pada persamaan (2.25):

$$\begin{aligned} K_T &= \frac{BW_{ST}}{BW_T} \\ &= \frac{10.000.000}{89960} = 111,1 \text{ Panggilan} \end{aligned}$$



**Gambar 5.14** Kapasitas Panggilan Maksimal berdasarkan Kapasitas Kanal  
 Sumber : pengujian

Gambar 5.14 memperlihatkan jumlah panggilan maksimal yang dapat dilakukan dengan menggunakan kapasitas kanal yang disediakan. Semakin besar kapasitas kanal, jumlah panggilan maksimal semakin meningkat. Grafik ini hanya menunjukkan jumlah

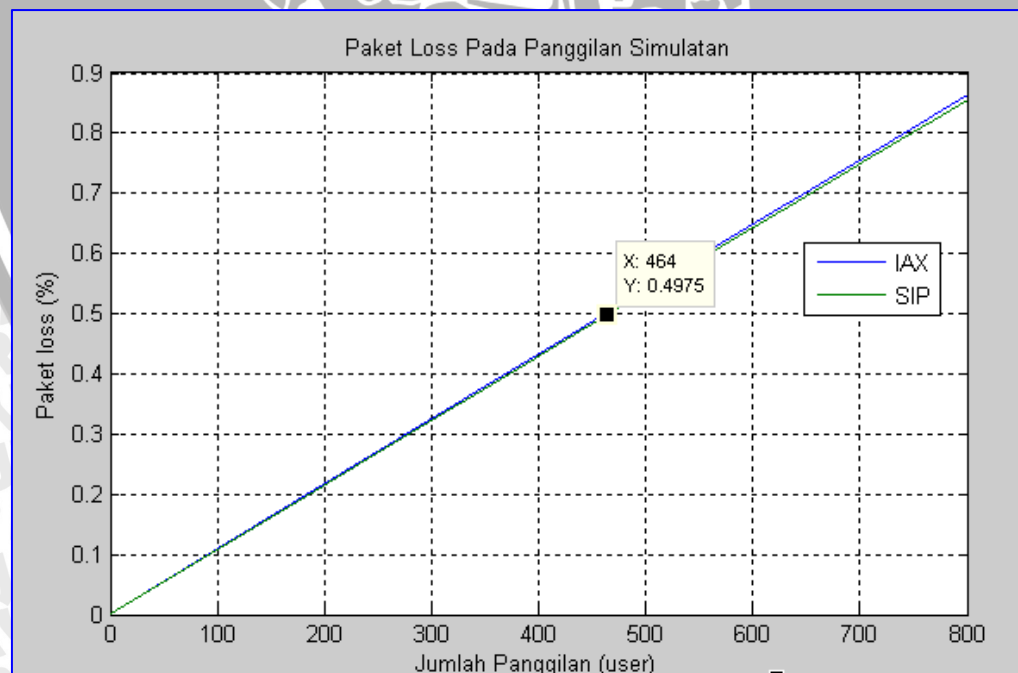
panggilan maksimal tanpa melihat performansi jaringan. Jumlah panggilan maksimal dengan memperhitungkan paket loss dapat dilihat pada gambar 5.15.

Paket loss yang terjadi saat panggilan maksimum terjadi dapat dihitung dengan persamaan 2.26.

$$\begin{aligned} \text{Paket loss (SIP)} &= \left(1 - \frac{1}{1 + 3,44 \frac{dx \Sigma \text{ diterima}}{LV}}\right) \times 100\% \\ &= \left(1 - \frac{1}{1 + 3,44 \frac{30 \times 111 \times 6900}{218,290 \times 3 \times 10^8}}\right) \times 100\% \\ &= 0,12055\% \end{aligned}$$

$$\begin{aligned} \text{Paket loss (IAX)} &= \left(1 - \frac{1}{1 + 3,44 \frac{dx \Sigma \text{ diterima}}{LV}}\right) \times 100\% \\ &= \left(1 - \frac{1}{1 + 3,44 \frac{30 \times 119 \times 6512}{207,887 \times 3 \times 10^8}}\right) \times 100\% \\ &= 0,12806\% \end{aligned}$$

Paket loss protokol SIP dan IAX masih memenuhi standar VoIP (< 0,5%)



Gambar 5.15 Perhitungan Paket Loss Menggunakan Matlab

Sumber : pengujian



Perhitungan jumlah user berdasarkan paket *loss* yang masih dapat diterima (0,5%) dengan menggunakan matlab, diketahui bahwa panggilan maksimum adalah 464 panggilan secara bersamaan.

#### 5.2.3.4.1 Kesimpulan Hasil Pengujian

Pada pengujian penggunaan *bandwidth*, didapatkan protokol IAX menggunakan lebih sedikit *bandwidth*, minimal 123.520 kbps dan maksimal 188.512 kbps per saluran percakapan. Sedangkan SIP menggunakan lebih banyak *bandwidth*, walaupun tidak terlalu signifikan, yaitu minimal sebesar 115.306 kbps dan maksimal 232.705 kbps. Jumlah *user* yang dapat menggunakan jaringan secara bersamaan, protokol IAX sebanyak 119 user, SIP 111 *user*. Penggunaan *bandwidth* IAX yang lebih disebabkan oleh penggunaan label biner yang mengurangi jumlah penggunaan *header*.



## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan pengujian jaringan VoIP berbasis protokol IAX pada LAN melalui NAT yang diujikan pada Laboratorium Telekomunikasi Teknik Elektro Universitas Brawijaya, dapat disimpulkan bahwa:

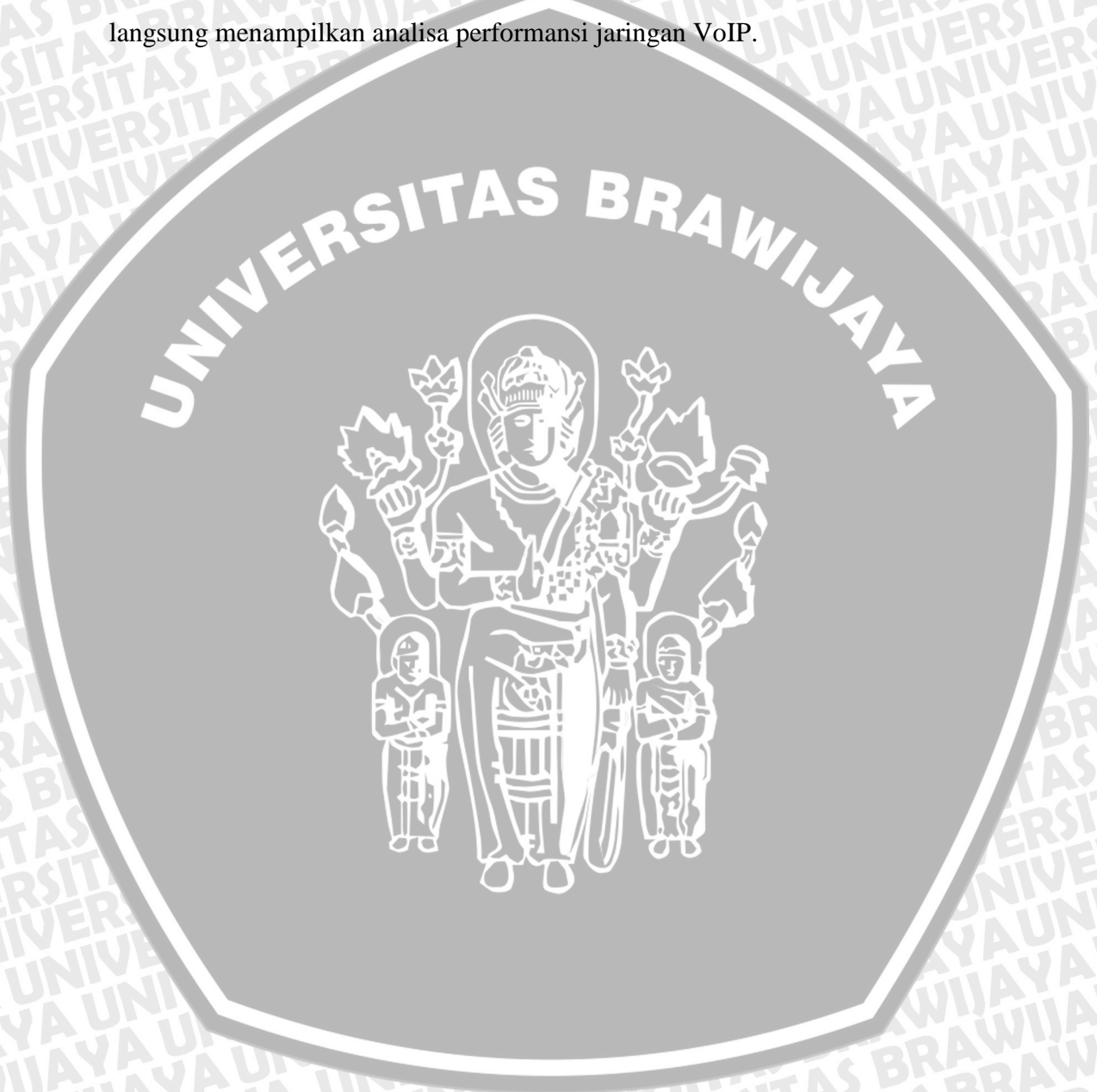
1. Perancangan jaringan VoIP menggunakan protokol IAX melewati NAT dapat diaplikasikan pada jaringan Teknik Elektro, dengan meregistrasikan 16 *client* pada sebuah server yang terletak di Laboratorium Telekomunikasi.
2. NAT dapat diimplementasikan dengan baik sesuai dengan perencanaan. NAT digunakan untuk membatasi akses jaringan publik ke jaringan privat pada jaringan internet.
3. Protokol IAX dapat diaplikasikan pada jaringan VoIP. Kapasitas maksimum yang dapat dilewati adalah 119 komunikasi secara bersamaan bila menggunakan protokol IAX dan 111 komunikasi bila menggunakan SIP.
4. Jaringan VoIP menggunakan protokol IAX tanpa melalui NAT didapatkan performansi; *delay end to end 2 client* rata-rata 19,32 ms; *delay end to end 8 client* rata-rata 21,22 ms; *jitter 2 client* rata-rata 2,18ms; *jitter 8 client* rata-rata 2,30 ms; *paket loss 0% untuk 2 client* dan *8 client*
5. Jaringan VoIP menggunakan protokol IAX melalui NAT didapatkan performansi; *delay end to end 2 client* rata-rata 19,17 ms; *delay end to end 8 client* rata-rata 20,78 ms; *jitter* rata-rata 8,37 ms; *jitter 8 client* rata-rata 7,31 ms; *paket loss maksimal 0,03% untuk 2 client* dan 0% untuk *8 client*.
6. Protokol IAX dan SIP dapat digunakan sebagai VoIP, kecuali bahwa SIP tidak dapat menembus NAT, Performansi IAX dan SIP berada dalam kategori “good” untuk semua parameter (*delay end to end < 150 ms, jitter < 20 ms, paket loss < 0,05%*) . Performansi protokol IAX lebih baik, dengan nilai; *delay end to end* rata-rata IAX 21,22 ms ; *delay end to end* rata-rata SIP 32,30 ms; *jitter* rata-rata IAX 2,30 ms ; *jitter* rata-rata SIP 13,67 ms; *paket loss* rata-rata IAX dan SIP 0%.



## 6.2 Saran

Saran yang dapat diberikan untuk pengembangan analisis performansi jaringan VoIP, antara lain:

1. Pengujian performansi pada media yang berbeda untuk mengetahui kualitas masing-masing media.
2. Merancang *software capture* yang dapat mengenali protokol IAX dan dapat langsung menampilkan analisa performansi jaringan VoIP.



## DAFTAR PUSTAKA

- Anonim, *Asterisk VoIP Call Analysis Tool*: <http://www.unleashnetworks.com/lib/IAX2AnalyzerWhitepaper3.pdf> (diakses 9 Oktober 2008)
- Anonim. *How to sniff a network*: <http://www.networkworld.com/columnists/2004/0830nutter.html?page=2> (diakses 20 Agustus 2008)
- Anonim. *Sampling*: <http://www.wikipedia.org/wiki/Sampling> (diakses 9 Oktober 2008)
- Anonim. *Session Initiation Protocol*: <http://www.ristinet.com/index.php?sip> (diakses 6 Oktober 2008)
- Anonim, *Understanding Delay in Packet Voice Networks*: <http://www.cisco.com/application/pdf/paw/5125/delay-details.pdf> (diakses 9 Oktober 2008)
- Anonim. *VoIP Implementation & Management*: <http://www.netiq.com/voip> (diakses 20 Agustus 2008)
- Asterisk, Sharif, Ben. 2007. *Trixbox-2 without Tears*: trixbox
- Basuki, Mudji. 2003. *Network Address Translation (NAT)*: <http://www.ilmukomputer.com> (diakses 6 Oktober 2008)
- Dhoto, 2005, *Buku Jaringan Komputer*: [http://www.eepisits.edu/~dphoto/kuliah/Buku\\_Jaringan\\_Komputer.pdf](http://www.eepisits.edu/~dphoto/kuliah/Buku_Jaringan_Komputer.pdf) (diakses 6 Oktober 2008)
- Jarkko Ahonen, Arttu Laine.1997 . *Realtime Speech and Voce Transmission on the Internet*: Helsinki University of Technology.
- Keith W. Ross. 2000. *Computer Networking: A Top Down Approach Featuring the Internet*. Addison Wesley Publishing Company.
- Khasnabish, Bhumip. 2003. *Implementing Voice Over IP*. New Jersey, USA: John Wiley & Sons, Inc
- Rosiani, Ulla Delfana. 2001. *Perancangan Aplikasi VoIP Pada Jaringan 100VG-AnyLAN di Unicersitas Brawijaya*. Skripsi tidak dipublikasikan. Malang: Universitas Brawijaya.
- Schwart, Mischa. 1987. *Telecommunication Network : Protocols, Modelling, and Analysis*. USA: Addison Wesley Publishing Company
- Spencer, Mark,2007. *draft-guy-iax-05 - IAX2: Inter-Asterisk eXchange Version 2*: <http://tools.ietf.org/html/draft-guy-iax-03> (diakses 9 Oktober 2008)



Spencer, Mark. 2004. *IAX Protocol Description*: [http://www.asteriskdocs.org/modules/tinycontent/content/docbook/current\\_v1/docs-html/x607.html](http://www.asteriskdocs.org/modules/tinycontent/content/docbook/current_v1/docs-html/x607.html)  
(diakses 6 Oktober 2008)

William Stallings. 1997. *Data And Communication Design Handbook*. New Jersey, USA: Prentice-Hall, Inc



## LAMPIRAN

### Lampiran 1: Program Perhitungan *Delay End to End*

#### Data yang diketahui:

panjang frame IAX ( $W_{frame}$ )	= 206 byte
panjang frame SIP ( $W_{frame}$ )	= 214 byte
maksimum segment size( $MSS$ )	= 160 byte
kecepatan transmisi ( $C_{pros}$ )	= 118512,86 bps
throughput( $\lambda$ )	= 113.351 kbps

#### Persamaan *Delay End to End*

$$t_{endoend} = t_{enc} + t_{trans} + t_p + t_w + t_{dec}$$

$$t_{dec} = \frac{W_{frame} - MSS}{C_{pros}} \times 8$$

$$t_{enc} = \frac{W_{frame} - MSS}{C_{pros}} \times 8$$

$$t_{trans} = \frac{(L+L')}{C} \times 8 = \frac{W_{frame}}{C} \times 8$$

$$t_p = t_{UTP} + t_{router}$$

$$t_w = \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu}$$

#### Listing Program

```
%delay end to end
c=206:100000; %kapasitas kanal
wf=206; %panjang frame iax
wfs=214; %panjang frame sip
mss=160; %iax's maximum segment size
th=113.351; %throughput
cpros=118512.86; %kecepatan transmisi per user

a=((wf-mss)*8)./cpros); %delay encapsulasi, delay decapsulasi

b=(8*wf./c); %delay transmisi
bs=(8*wfs./c);

f=((0.556*25./c)+0.003); %delay propagasi

d=(th./(c./8*wf.*((c./8*wf)- th))); %delay antrian
ds=( th./(c./8*wfs.*((c./8*wfs)- th)));
e=(1./8*wf); %waktu pelayanan server
es=(1./8*wfs);

t=2*a+b+f+d+e;
ts=2*bs+f+ds+es;

plot(c,t,c,ts)
```



## Lampiran 2: Program Perhitungan Jumlah User berdasarkan Kapasitas Kanal

### Data yang diketahui:

Paket per sekon ( $pps$ )	= 50 pps
Panjang paket rata-rata IAX ( $L_{PI}$ )	=0,208792 kBps
Panjang paket rata-rata SIP	=0,2249 kBps

### Persamaan Kapasitas Kanal

$$BW_T = L_{PI} \times pps$$

$$K_T = \frac{BW_{ST}}{BW_T}$$

### Listing Program

```
%Kapasitas Kanal Berbanding jumlah User
c=10:100000; %kanal LAN sampai dengan 100 Mbps
pp=50; %paket per sekon
pi=0.208792; %panjang paket rata2 IAX
ps=0.2249; %panjang paket rata2 SIP

ki=c./(8*(pi.*pp));
ks=c./(8*(ps.*pp));

plot(c,ki,c,ks)
```



### Lampiran 3: Program Perhitungan Paket Loss

#### Data yang diketahui:

Panjang Kabel UTP ( $d$ )	= 30 m
Jumlah Paket SIP diterima ( $\Sigma_{diterima}$ )	= 6900 paket
Jumlah Paket IAX diterima	= 6515 paket
Panjang Paket SIP ( $L$ )	= 218,290 byte
Panjang Paket IAX	= 207,887 byte

#### Persamaan Perhitungan Paket Loss

$$\left(1 - \frac{1}{1 + 3,44 \frac{dx \Sigma_{diterima}}{LV}}\right) \times 100\%$$

#### Listing Program

```
%paket loss
u=1:800;
d=30;
ri=6900;
rs=6512;
li=218.290;
ls=207.887;
v=3000000000;

pl=(1-(1./(1+(3.44.*d.*u.*ri)./(li.*v))))*100;
pls=(1-(1./(1+(3.44.*d.*u.*rs)./(ls.*v))))*100;

plot (u,pl,u,pls)
```

