

BAB I PENDAHULUAN

1.1 Latar Belakang

Teknologi telekomunikasi berkembang begitu cepat dan beragam seiring dengan perkembangan ilmu pengetahuan dan teknologi (IPTEK). Pada awalnya, teknologi telekomunikasi diharapkan dapat mempercepat penyampaian informasi, kemudian teknologi telekomunikasi berkembang agar dapat mengirimkan lebih banyak informasi dalam satu waktu dan dapat diakses secara *mobile*. Kini, teknologi telekomunikasi dituntut agar dapat memberikan fasilitas yang bermacam-macam, salah satunya adalah fasilitas *Voice over Internet Protocol (VoIP)*.

VoIP merupakan sebuah teknologi yang mendigitalisasi data suara ke dalam paket-paket data untuk ditransmisikan melalui jaringan komputer memanfaatkan *Internet Protocol (IP)*. *VoIP* merupakan sebuah terobosan dalam berkomunikasi, yang mampu menjawab kekurangan jaringan *PSTN* dalam hal biaya dan fasilitas. Karena teknologi ini memanfaatkan jaringan internet yang sudah ada, biaya komunikasi yang dikeluarkan sama dengan biaya penggunaan internet biasa. Di samping itu fasilitas-fasilitas yang bisa ditambahkan pada teknologi *VoIP* juga lebih banyak, antara lain *call waiting, call scheduling, call conference, video call, video conference, instant messaging*, serta juga bisa diintegrasikan dengan aplikasi internet yang telah ada seperti *chatting, email, file transfer*, dan aplikasi lainnya. (Tharom, 2002 : 21)

Jaringan berbasis *IP* yang sekarang ini sedang mengalami proses migrasi dari *IPv4* menuju *IPv6*. *IP* versi 6 (*IPv6*) adalah protokol Internet versi baru yang didesain sebagai pengganti dari *Internet protocol* versi 4 (*IPv4*). *IPv6* yang memiliki kapasitas alamat *IP* yang besar (128 bit) yaitu $3,4 \times 10^{38}$ alamat, tentu saja hal ini mampu mengatasi masalah keterbatasan alamat pada *IPv4* yang hanya memiliki kapasitas alamat sebesar 32 bit saja atau sebesar $4,294 \times 10^9$ alamat (Beijnum, 2006 : 2).

IPv6 mendukung penyusunan alamat secara terstruktur, yang memungkinkan Internet terus berkembang dan menyediakan kemampuan *routing* baru yang tidak terdapat pada *IPv4*. *IPv6* memiliki tipe *address anycast* yang dapat digunakan untuk pemilihan *route* secara efisien. *IPv6* juga dilengkapi oleh mekanisme penggunaan *address* secara lokal yang memungkinkan terwujudnya instalasi secara *Plug and Play*,

serta menyediakan *platform* bagi cara baru pemakaian Internet, seperti dukungan terhadap aliran data secara real-time, pemilihan provider, *mobilitas host*, *end-to-end security*, ataupun konfigurasi otomatis. (Jun Ichiro, 2005 : 2)

Protokol-protokol penunjang *VoIP* dewasa ini semakin berkembang. Perkembangan ini bertujuan untuk mengatasi kekurangan protokol sebelumnya dan sangat cocok digunakan untuk teknologi *VoIP*, salah satunya yaitu *SIP (Session Initiation Protocol)*. *SIP* merupakan protokol lapis aplikasi yang direkomendasikan oleh IETF (*Internet Engineering Task Force*), yang kemudian didokumentasikan ke dalam dokumen *request for command (RFC) 2543*. IETF memilih *SIP* karena memiliki sifat yang sederhana, berbasis teks dan sangat fleksibel terhadap pengembangan-pengembangan baru serta dapat mendukung implementasi berbagai layanan multimedia masa depan. Di dalam *VoIP*, *SIP* adalah protokol *call setup* yang beroperasi pada lapis aplikasi yang digunakan untuk membangun, memodifikasi dan memutuskan sesi komunikasi yang terdiri dari dua atau lebih partisipan. (Tabratas Tharom, 2002 : 55)

Alasan-alasan utama tentang keterbatasan pengalamatan pada *IPv4* dan ketersediaan alamat yang besar pada *IPv6* serta tuntutan-tuntutan penggunaan internet yang sangat tinggi menjadi dasar penelitian yang berjudul Perencanaan dan Realisasi Jaringan *Voice Over IP (VoIP)* berbasis *Session Initiation Protocol (SIP)* menggunakan *IP* versi 6 (*IPv6*).

1.2 Rumusan Masalah

Berdasarkan permasalahan yang terdapat pada latar belakang, dibuat rumusan masalah sebagai berikut:

1. Bagaimana merancang jaringan *VoIP* menggunakan *IPv6*?
2. Bagaimana mengimplementasikan *SIP* pada jaringan *VoIP*?
3. Berapa besar performansi *VoIP* berbasis *SIP* menggunakan *IPv6* yang meliputi *throughput*, *delay end to end*, *packet loss*, kapasitas jaringan dan *Mean Opinion Score (MOS)*?

1.3 Ruang Lingkup

Berdasarkan rumusan masalah diatas, agar pembahasan terfokus pada pokok pembahasan maka masalah yang akan dibahas pada pembuatan skripsi ini antara lain :

1. Penelitian dilakukan di kos-kosan jalan Candi Panggung Indah no.7 Malang.

2. Jaringan yang digunakan adalah *Local Area Network (LAN)*.
3. Software *server VoIP* menggunakan *Opensips*.
4. Software *client VoIP* menggunakan *SIP-communicator*.
5. Analisis performansi jaringan *VoIP* menggunakan software *Wireshark*.
6. Parameter performansi yang akan di hitung adalah *throughput*, *delay end to end*, *packet loss*, kapasitas jaringan dan *Mean Opinion Score (MOS)*.

1.4 Tujuan

Tujuan yang ingin dicapai dari penyusunan skripsi ini adalah merencanakan dan menerapkan *Voice Over IP (VoIP)* berbasis *Session Initiation Protocol (SIP)* menggunakan *IPv6* dengan parameter *throughput*, *delay*, *packet loss*, kapasitas jaringan dan *MOS*.

1.5 Sistematika Penulisan

Sistematika penulisan dan gambaran untuk setiap bab pada skripsi ini adalah sebagai berikut: BAB I PENDAHULUAN. Pada bab I ini memuat tentang latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika penulisan. BAB II yaitu *Voice over Internet Protocol (VoIP)* berbasis *SIP* pada *LAN* menggunakan *IPv6*, pada bab ini akan dibahas tentang dasar teori yang membahas tentang *VoIP*, *SIP*, *LAN*, dan *Internet Protocol*. BAB III METODOLOGI, pada bab ini memuat jenis data dan cara pengambilan data, variabel dan cara analisis data, dan kerangka solusi permasalahan dalam mengerjakan penelitian. BAB IV yaitu Perancangan Jaringan *VoIP*, bab ini menjelaskan secara terperinci tentang perancangan jaringan *VoIP*. BAB V Pengujian dan Analisis, pada bab ini akan dibahas tentang perhitungan performansi yang meliputi *throughput*, *delay*, *packet loss*, kapasitas jaringan dan *MOS*. BAB VI memuat tentang kesimpulan dan saran yang diperoleh dari analisis perhitungan.

BAB II

VOIP BERBASIS SIP PADA LAN MENGGUNAKAN IPV6

Bab 2 ini merupakan tinjauan teori tentang *Voice over Internet Protocol (VoIP)* berbasis *SIP* pada *LAN* menggunakan *Ipv6* yang berisi tentang *VoIP*, *SIP*, *LAN*, dan *Internet Protocol*. *VoIP* merupakan sebuah teknologi yang mendigitalisasi data suara ke dalam paket-paket data untuk ditransmisikan melalui jaringan komputer memanfaatkan *Internet Protocol (IP)*. Untuk menunjang *VoIP* digunakan protokol *SIP (Session Initiation Protocol)*, *SIP* merupakan protokol lapis aplikasi yang digunakan untuk membangun, memodifikasi dan memutuskan sesi komunikasi yang terdiri dari dua atau lebih partisipan. Untuk jaringan yang digunakan pada *VoIP* berbasis *SIP* adalah Jaringan *LAN*, sedangkan pengalamatan yang digunakan adalah *IPv6*. Secara lengkap uraian dari tinjauan teori ditunjukkan dalam sub bab dibawah ini.

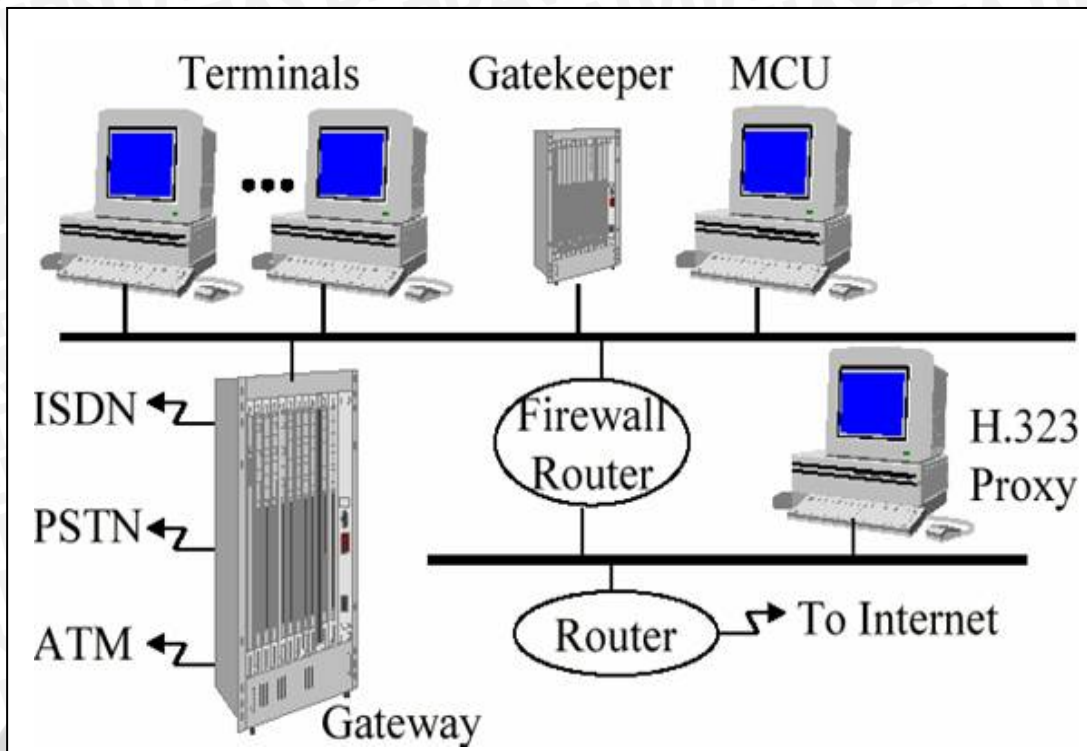
2.1 VoIP (Voice over Internet Protocol)

Voice over Internet Protocol (VoIP) adalah teknologi yang mampu melewati trafik suara, yang berbentuk paket melalui jaringan *IP*. Jaringan *IP* sendiri 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* (M. Iskandarsyah, 2003:1).

VoIP saat ini telah menjadi bagian dari dunia komunikasi seluruh dunia, karena *VoIP* memberikan nilai lebih 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.

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* dibawah ini adalah jaringan *VoIP* secara umum.



Gambar 2.1 : Jaringan VoIP Secara Umum
 Sumber: Teknis dan Bisnis VoIP, 2002:29

Pada gambar di atas ada tiga buah komponen utama pada jaringan VoIP, yaitu :

a. Terminal

Terminal atau *endpoint* adalah perangkat yang akan digunakan pengguna untuk melakukan komunikasi. Secara umum terminal adalah perangkat yang bertugas untuk mengubah sinyal analog dari suara pelanggan ke bentuk kode-kode digital (bit), membentuk paket dari kode-kode digital untuk ditransmisikan ke jaringan, dan melakukan proses sebaliknya di penerima. Terminal umumnya dilengkapi dengan sebuah *buffer* yang digunakan untuk mengkompensasi *jitter* (disebut *dejitter buffer*). Terminal dapat berupa *IP-Phone* dan *Softphone*.

- *IP-Phone*

IP-Phone adalah telepon digital yang dilengkapi dengan *codec* untuk mengubah sinyal suara ke format digital, dan *network interface card* (NIC) untuk melakukan koneksi ke jaringan berbasis *IP*.

- *Softphone*

Softphone adalah sebuah *personal computer* (PC) atau laptop yang memiliki *audio card* yang *full duplex* yang dilengkapi dengan *microphone* dan NIC untuk melakukan koneksi ke jaringan berbasis *IP*.



b. *Gateway*

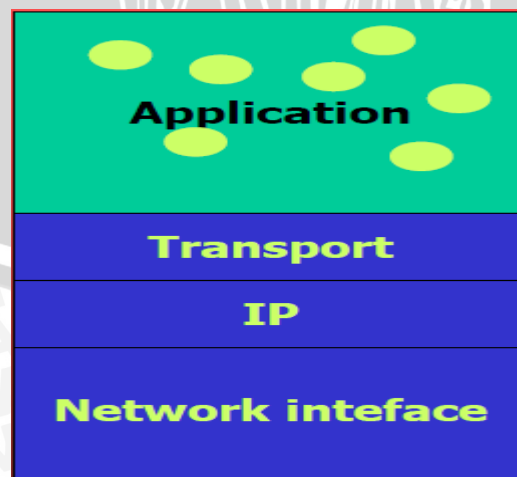
Gateway adalah perangkat yang digunakan untuk menghubungkan jaringan *VoIP* dengan jaringan lain yang tidak berbasis paket (misalnya *PSTN*). *Gateway* berfungsi untuk menterjemahkan format transmisi jaringan yang tidak berbasis paket ke jaringan *VoIP* dan sebaliknya. *Gateway* tidak diperlukan jika tidak ada interkoneksi dengan jaringan lain yang tidak berbasis paket.

c. *Router*

Router adalah perangkat yang digunakan untuk melakukan interkoneksi dengan jaringan lain yang juga berbasis paket. Pada jaringan, *router* digunakan untuk interkoneksi dengan jaringan *IP* lain, yaitu untuk melakukan proses *routing* antar jaringan.

2.1.2 Protokol Penunjang Jaringan *VoIP*

VoIP menggunakan protokol internet untuk melewatkan paket datanya, sedangkan protokol itu sendiri merupakan sekumpulan dari beberapa aturan yang berhubungan dengan komunikasi data antara beberapa alat komunikasi supaya komunikasi data dapat dilakukan dengan benar. Salah satunya adalah *TCP/IP* (*Transfer Control Protocol/Internet Protocol*) yang merupakan sebuah protokol yang digunakan pada jaringan internet. *TCP/IP protocol suite* terdiri dari 4 layer: *Applikasi*, *Transport*, *Internetwork*, dan *network interface*. Layer tersebut dapat dilihat sebagai hirarki seperti di bawah ini : (Onno W.Purbo, 1998:23)



Gambar 2.2 : hirarki layer pada *TCP/IP*

Sumber : *TCP/IP*, 1998:23

a. *Application layer*

Fungsi utama dari layer ini adalah pemindahan file. Pemindahan file dari sebuah sistem ke sistem lainnya yang berbeda memerlukan suatu sistem pengendalian untuk mengatasi adanya ketidak kompatibelan sistem file yang berbeda-beda. protokol ini berhubungan dengan aplikasi. Salah satu contoh aplikasi yang telah dikenal misalnya *HTTP (Hypertext Transfer Protocol)* untuk web dan *FTP (File Transfer Protocol)* untuk perpindahan file.

b. *Transport layer*

Transport Layer bertanggung jawab untuk komunikasi antara aplikasi. Layer ini mengatur aliran informasi dan mungkin menyediakan pemeriksaan *error*. Data dibagi kedalam beberapa paket yang dikirim ke *internet layer* dengan sebuah *header*. *Header* mengandung alamat tujuan, alamat sumber dan *checksum*. *Checksum* diperiksa oleh mesin penerima untuk melihat apakah paket tersebut ada yang hilang pada rute.

c. *Internetwork layer*

Internetwork Layer bertanggung jawab untuk komunikasi antara mesin. Protocol yang berada pada layer ini bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada layer ini terdapat tiga macam protocol, yaitu *IP, ARP dan ICMP*.

IP (Internet Protocol) berfungsi untuk menyampaikan paket data ke alamat yang tepat. *ARP (Address Resolution Protocol)* ialah protocol digunakan untuk menemukan alamat hardware dari host/komputer yang terletak pada network yang sama. Sedangkan *ICMP (Internet Control Message Protocol)* ialah protocol yang digunakan untuk mengirimkan pesan & melaporkan kegagalan pengiriman data

d. *Interface layer*

Lapisan/layer terbawah, yaitu *Network Interface layer* bertanggung jawab mengirim dan menerima data ke dan dari media fisik. Media fisiknya dapat berupa kabel, serta optik atau gelombang radio. Karena tugasnya ini, *protocol* pada layer ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasal dari peralatan lain yang sejenis.

2.1.2.1 Protokol H.323

Salah satu standar komunikasi pada *VoIP* menurut rekomendasi ITU-T adalah H.323. Standar H.323 terdiri dari komponen, protokol, dan prosedur yang menyediakan komunikasi multimedia melalui jaringan *packet-based*. Bentuk jaringan *packet-based*

yang dapat dilalui antara lain jaringan internet, *internet packet Exchange (IPX)-based*, *Local Area Network (LAN)*, dan *Wide Area Network (WAN)* H.323 dapat digunakan untuk layanan-layanan multimedia seperti komunikasi suara, komunikasi *video* dengan suara, dan gabungan suara, *video*, dan data (Tabratas Tharom, 2002 : 40). Standar H.323 terdiri dari 4 komponen penting, yaitu : (Tabratas Tharom, 2002 : 44)

a. Terminal

Terminal digunakan untuk komunikasi multimedia *real time* dua arah. Terminal H.323 dapat berupa *PC*, laptop, dan alat komunikasi lain yang dapat berdiri sendiri yang dapat menjalankan aplikasi multimedia.

b. Gateway

Gateway digunakan untuk menghubungkan dua jaringan yang berbeda, yaitu antara jaringan H.323 dan jaringan bukan H.323. sebagai contoh *gateway* dapat menghubungkan dan menyediakan komunikasi antara terminal H.323 dengan jaringan *PSTN*. Dalam menghubungkan dua jaringan yang berbeda dilakukan dengan menterjemahkan protokol-protokol untuk *call setup* dan *release* serta mengirimkan informasi antara jaringan yang terhubung dengan *gateway*. Namun *gateway* tidak dibutuhkan untuk komunikasi antara dua terminal H.323.

c. Gatekeeper

Gatekeeper dapat dianggap sebagai otak pada jaringan H.323 karena merupakan titik yang penting pada jaringan H.323. *Gatekeeper* merupakan titik fokus dari semua *call* yang terjadi pada *network* H.323. *Gatekeeper* menyediakan pelayanan-pelayanan yang penting seperti pengalamatan, otorisasi dan otentifikasi dari terminal dan *gateway*, manajemen *bandwidth*, pembiayaan dan rekening. *Gatekeeper* juga bisa menyediakan layanan *call-routing*. (Tabratas Tharom, 2002 : 46).

d. MCU (*Multipoint Control Unit*)

MCU digunakan untuk layanan konferensi tiga terminal H.323 atau lebih. Semua terminal yang ingin berpartisipasi dalam konferensi dapat membangun hubungan dengan *MCU* yang mengatur bahan-bahan untuk konferensi, negosiasi antara terminal-terminal untuk memastikan *audio*, *video coder/decoder* (codec).

2.2 SIP (*Session Initiation Protocol*)

SIP (Session Initiation Protocol) merupakan protokol lapis aplikasi yang direkomendasikan oleh IETF (*Internet Engineering Task Force*), awalnya didefinisikan pada RFC 2543 (*obsolete*) dan kemudian diperbaharui melalui RFC 3261. Badan

standartisasi dunia yang mengembangkan standartisasi *NGN (Next Generation Network)* telah memilih *SIP* karena sifatnya yang sederhana, berbasis teks dan sangat fleksibel terhadap pengembangan-pengembangan baru serta dapat mendukung implementasi berbagai layanan multimedia masa depan.

Di dalam *IP* dan telepon tradisional, selalu dibedakan dengan jelas dua tahap panggilan suara. Tahap pertama adalah “*call setup*” yang mencakup semua detail keperluan agar dua perangkat telepon dapat berkomunikasi. Tahap selanjutnya adalah “*transfer data*” di mana *call setup* sudah terbentuk. Di dalam *VoIP*, *SIP* adalah protokol *call setup* yang beroperasi pada lapis aplikasi yang digunakan untuk membangun, memodifikasi dan memutuskan sesi komunikasi yang terdiri dari dua atau lebih partisipan. Sesi komunikasi di sini dapat berupa panggilan teleponi, konferensi multimedia maupun aplikasi-aplikasi lainnya berbasis media sebagai data, audio dan video. (Tabratas Tharom, 2002 : 55)

2.2.1 Komponen *SIP*

Dalam hubungannya dengan *IP Telephony*, komponen *SIP* dibagi menjadi 2 kategori utama, *server SIP* dan *user agent SIP*. (Tabratas Tharom, 2002 : 59).

- *Server SIP*

Server SIP dibagi lagi menjadi 3 macam yaitu *proxy*, *registrar*, dan *redirect*. Setiap tipe *server* melakukan tugas yang berbeda, ditunjukkan pada tabel 2.1. Fungsi spesifik *server* menunjukkan request *SIP* mana yang akan diproses.

Tabel 2.1 Fungsi tipe-tipe *SIP server*

<i>SIP server</i>	Fungsi
<i>Proxy server</i>	Bertindak sebagai mediator antara <i>client</i> dan <i>server user agent</i> . <i>Server</i> ini bertugas sebagai <i>client user agent</i> atau <i>server user agent</i> . Tergantung dari arah komunikasi antara <i>client</i> dengan <i>server</i> . <i>Proxy server</i> dapat secara mudah melakukan <i>forward request SIP</i> atau mengubahnya sebelum mengirim.
<i>Registrar server</i>	Menerima permintaan <i>REGISTER</i> , yang berisi alamat <i>IP</i> dan alamat <i>SIP</i> dengan kata lain <i>Uniform Resource Locator (URL)</i> dari <i>user agent</i> . Hal ini mengijinkan

	<i>server</i> ini untuk melacak lokasi <i>user agent</i> dari <i>server registrar</i> manakah permintaan REGISTER diterima.
<i>Redirect server</i>	Menerima inisiasi, dalam bentuk <i>request SIP INVITE</i> , bentuk sesi <i>SIP</i> dari <i>user agent</i> yang memanggil, mendapatkan alamat <i>SIP</i> yang benar dari <i>user agent</i> yang dipanggil, dan me-reply ke <i>user agent</i> yang memanggil menggunakan alamat <i>SIP</i> tersebut. <i>User agent</i> yang memanggil kemudian menggunakan alamat <i>SIP</i> yang benar tersebut untuk memulai sesi <i>SIP</i> dengan <i>user agent</i> yang dipanggil.

Sumber : Analisa Perbandingan Kinerja IAX2 dan SIP :32

- *User agent SIP*

User agent merupakan komponen *SIP* yang memulai, menerima, dan menutup sesi komunikasi. Tabel 2.2 menampilkan 2 tipe *user agent SIP* dan fungsi yang dilakukan. Setiap *user agent* mempunyai sebuah alamat *SIP*.

Tabel 2.2. Fungsi tipe-tipe *SIP user agent*

<i>User Agent SIP</i>	Fungsi
<i>User Agent Client (UAC)</i>	Menginisiasi <i>request SIP</i>
<i>User Agent Server (UAS)</i>	Menerima <i>request SIP</i>

Sumber : Analisa Perbandingan Kinerja IAX2 dan SIP :32

2.2.2 Operasi Dasar SIP

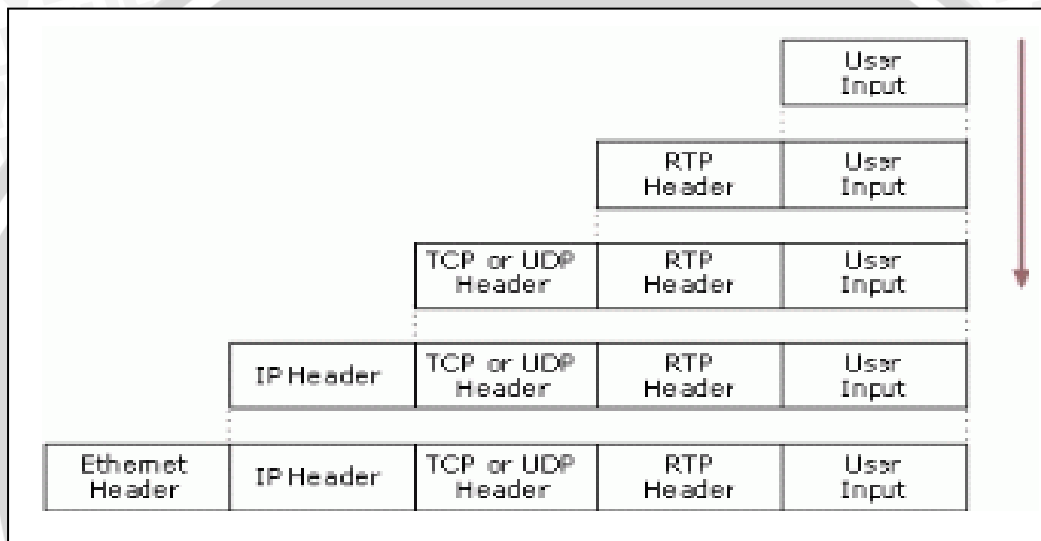
Proses komunikasi antar *user agent* pada jaringan VoIP berbasis SIP melalui beberapa tahap. Tahapan dalam proses komunikasi tersebut adalah : (Noviatul, 2005:16)

- Salah satu tipe *signaling* dan *call control* dalam hal ini SIP dibutuhkan untuk menentukan, merubah, dan mengakhiri sebuah sesi panggilan.
- Setelah sesi panggilan dibuat, suara akan disampling dan dikonversikan ke format digital.
- Sampel data tersebut kemudian dibungkus atau dienkapsulasi ke dalam paket RTP (*Real Time Transport Protocol*). RTP didesain secara khusus untuk digunakan dalam komunikasi *real-time* melalui jaringan *packet-based*.



- Dari *RTP* kemudian dienkapsulasi lagi ke dalam protokol transport, yaitu *UDP* (*User Datagram Protocol*). Dalam transmisinya, *RTP* menggunakan *RTCP* (*Real-time Control Protocol*) yang digunakan untuk memonitor kualitas dari sesi *RTP*.
- *UDP* kembali dienkapsulasi lagi ke dalam protocol network layer, yaitu *IP*.
- *IP* dibungkus lagi kedalam *data link layer* semisal Ethernet.

Gambar 2.3 menunjukkan proses enkapsulasi, dari paket *RTP* ke frame link layer.



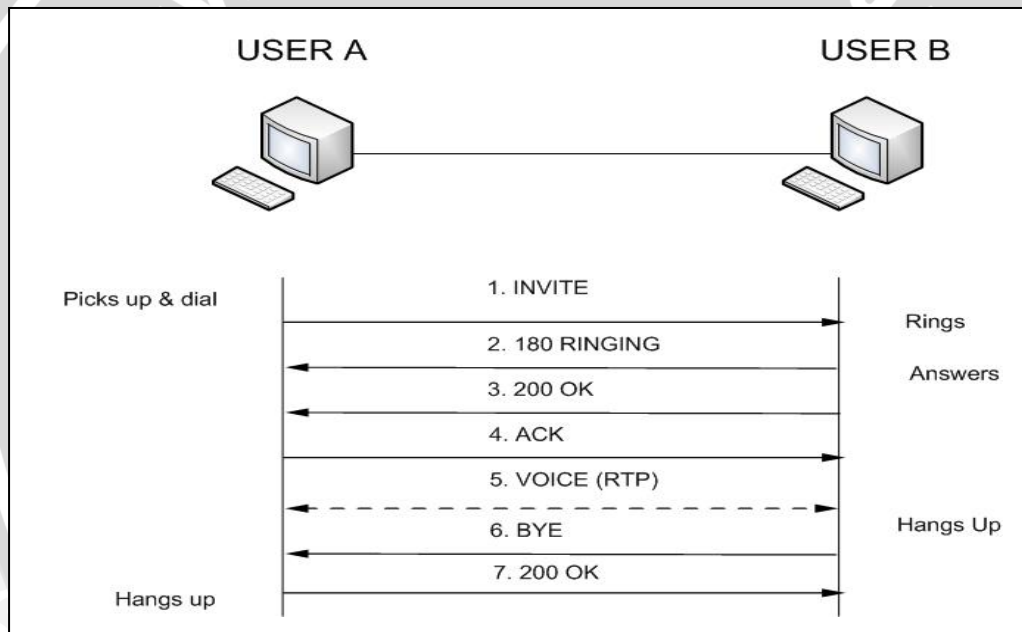
Gambar 2.3 : Enkapsulasi protokol pada *voip*

Sumber : Muchammad Husni– aplikasi *QOS analyzer* pada jaringan *voip*

Hubungan yang dibangun oleh *SIP* pada proses signaling bersifat client-server. Dengan demikian ada 2 jenis *message*, yaitu *request* dan *response*. (Tabratas Tharom, 2002 : 66,).

- *SIP Request* :
 - ✓ *INVITE* : Mengundang *user agent* yang lain untuk bergabung dalam sesi komunikasi.
 - ✓ *ACK* : Konfirmasi bahwa *user agent* telah menerima pesan terakhir dari serangkaian pesan *INVITE*.
 - ✓ *BYE* : Terminasi sesi.
 - ✓ *CANCEL* : Membatalkan *INVITE*.
 - ✓ *REGISTER* : Registrasi di *Register Server*.
 - ✓ *OPTIONS* : Meminta informasi tentang kemampuan *server*.

- ✓ **INFO** : Digunakan untuk membawa pesan informasi lainnya, seperti informasi inline *DTMF*.
 - **SIP Response** :
 - ✓ 1xx : *Informational Message*.
 - ✓ 2xx : *Successful Response*.
 - ✓ 3xx : *Redirection Response*.
 - ✓ 4xx : *Request Failure Response*.
 - ✓ 5xx : *Server Failure Response*.
 - ✓ 6xx : *Global Failure Response*.
- Contoh : User A menggunakan aplikasi *SIP* pada *PC (Softphone)* untuk memanggil User B (juga menggunakan *Softphone*) melalui internet.



Gambar 2.4 : Diagram alir sebuah panggilan
 Sumber : Mekanisme dan implementasi keamanan pada *SIP* : 6

Tabel 2.3. Penjelasan diagram alir panggilan

Step	Deskripsi
1.	<i>INVITE</i> : Pelanggan A melakukan inisialisasi panggilan ke pelanggan B
2.	180 ringing : Pelanggan B memberi respon dengan mengirim sinyal dering ke pelanggan A
3.	200 OK : Pelanggan B siap menerima panggilan

4.	<i>ACK</i> : Pelanggan A memberitahu pelanggan B bahwa ia telah menerima pesan 200
5.	<i>VOICE</i> : terjadi komunikasi dua arah melalui <i>RTP (Real Time Transport Protocol)</i>
6.	<i>BYE</i> : Pelanggan B menutup telepon
7.	200 OK : Panggilan berakhir pelanggan A menutup telepon

Sumber : Mekanisme dan implementasi keamanan pada *SIP* : 6

2.2.3 Aplikasi dan Kelebihan *SIP*

SIP mempunyai banyak keunggulan antara lain dapat digunakan untuk berbagai macam aplikasi, serta memiliki banyak kelebihan. Di bawah ini adalah macam-macam aplikasi dan kelebihan dari *SIP*.

- Aplikasi dari *SIP* antara lain :
 - ✓ *Voice over Internet Protocol (VoIP)*
 - ✓ Konferensi multimedia
 - ✓ *Text-messaging*
 - ✓ *Event-notification*
- Kelebihan dari *SIP* antara lain :
 - ✓ *General-purpose*
SIP dapat diintegrasikan dengan protokol standar *IETF* lainnya untuk membuat suatu aplikasi yang berbasis *SIP*.
 - ✓ Sederhana
 Pengiriman *message* berbasis *HTTP (text-based)*, bukan *binary-based*. Hal ini membuat *SIP* mudah diimplementasikan.
 - ✓ *Mobility*
 - Seorang pengguna dapat menerima *message/call* yang ditujukan kepadanya, meskipun berpindah dari satu lokasi ke lokasi lainnya. *Proxy-server* akan meneruskan *call* ke lokasi pengguna pada saat ini.
 - *Device* yang digunakan dapat berupa *PC, wireless phone, dan IP-phone*.
 - ✓ Layanan yang dapat dibuat dengan *CPL (Call Processing Language)* dan *CGI (Common Gateway Interface)* antara lain :
 - *Call waiting, call forwarding, call blocking (basic feature)*
 - *Call-forking* (melakukan call kepada beberapa endpoint)
 - *Instant-messaging*

- *Find me / follow me*
- ✓ Arsitektur yang terdistribusi dan *scalable*
- *Proxy-server*

Menerima *request* dari *user-agent-client*, melakukan *autentikasi*, memprosesnya, dan mengirimkan *request* tersebut kepada hop selanjutnya atas nama *client* tersebut.
- *Redirect-server*

Menerima *request* dari *client*, membandingkan alamat tujuan yang ingin dicapai, setelah ditemukan, alamat tersebut dikembalikan kepada *client*.
- *Registrar-server*

Menerima *REGISTER request* dari *client*.
- *Location-server*

Menyimpan data yang diperoleh dari *registrar-server*. *Location-server* digunakan oleh *proxy/redirect server* untuk mendapatkan informasi mengenai alamat tujuan yang ingin dicapai.

2.3 *Local Area Network (LAN)*

LAN adalah suatu jaringan yang menghubungkan beberapa komputer dalam suatu local area. Biasanya digunakan di dalam rumah, perkantoran, perindustrian, universitas, rumah sakit dan daerah yang sejenis (Behrouz A. Farouzan, 2001:30).

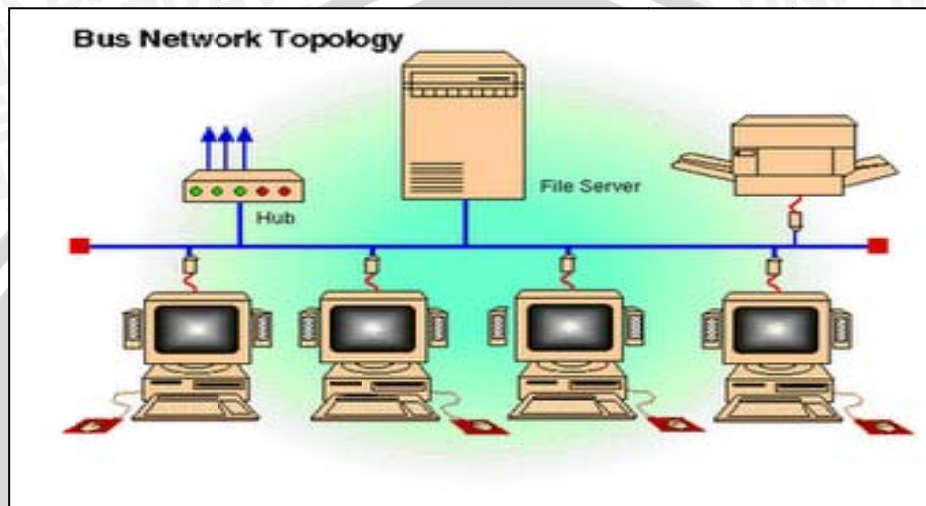
2.3.1 Topologi Jaringan pada *LAN*

Topologi adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Konfigurasi yang digunakan untuk membentuk jaringan secara fisik disebut topologi jaringan. Cara yang saat ini banyak digunakan adalah *bus, stars, cincin, pohon (Tree), dan Mesh*. Masing-masing topologi ini mempunyai ciri khas, dengan kelebihan dan kekurangannya masing-masing. (Warren Hioki, 1998:390)

- Topologi BUS

Topologi ini adalah topologi yang awal di gunakan untuk menghubungkan komputer. Dalam topologi ini masing masing komputer akan terhubung ke satu kabel panjang dengan beberapa terminal, dan pada akhir dari kabel harus di akhiri dengan satu *terminator*. Topologi ini sudah sangat jarang digunakan didalam

membangun jaringan komputer biasa karena memiliki beberapa kekurangan diantaranya kemungkinan terjadinya tabrakan aliran data, jika salah satu perangkat putus atau terjadi kerusakan pada satu bagian komputer maka jaringan langsung tidak akan berfungsi sebelum kerusakan tersebut di atasi. Gambar 2.5 menunjukkan topologi jaringan BUS.



Gambar 2.5 : Topologi jaringan BUS
Sumber : Telecommunications,1998:393

Keuntungan dan kekurangan dari topologi BUS adalah sebagai berikut:

Keuntungan :

- Hemat kabel.
- Layout kabel sederhana.
- Mudah dikembangkan.

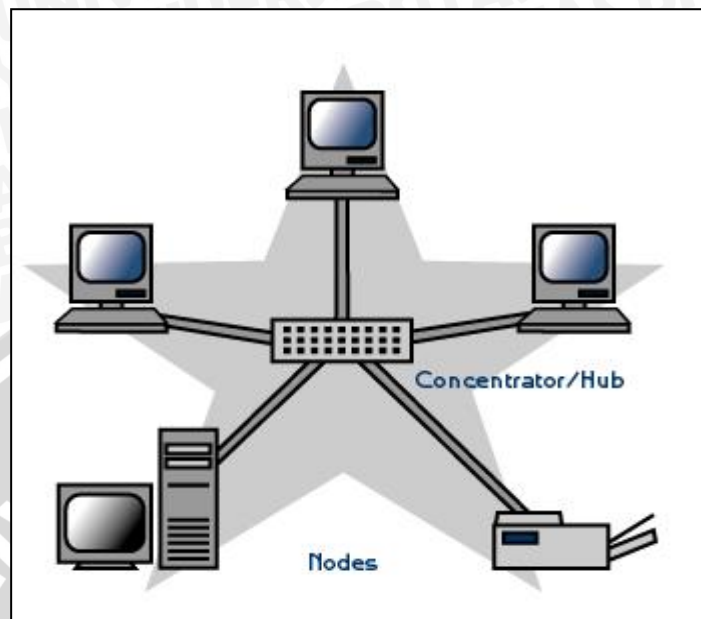
Kekurangan :

- Deteksi dan isolasi kesalahan sangat kecil.
- Bila ada gangguan di sepanjang kabel pusat maka keseluruhan jaringan akan mengalami gangguan.
- Diperlukan repeater untuk jarak jauh.

- Topologi *Star*

Topologi bintang atau yang lebih sering disebut dengan topologi *star*. Pada topologi ini kita sudah menggunakan bantuan alat lain untuk mengkoneksikan jaringan komputer. Contoh alat yang di pakai disini adalah hub, switch, dll. HUB

atau Switch digunakan untuk menghubungkan setiap node dalam jaringan LAN. Gambar 2.6 menunjukkan topologi jaringan *Star*.



Gambar 2.6 : Topologi jaringan *Star*
Sumber : Telecommunications,1998:392

Keuntungan dan kekurangan dari topologi *Star* adalah sebagai berikut:

Keuntungan :

- Paling fleksibel.
- Pemasangan/perubahan stasiun sangat mudah dan tidak mengganggu bagian jaringan lain.
- Kontrol terpusat.
- Kemudahan deteksi kerusakan.
- Kemudahan pengelolaan jaringan.

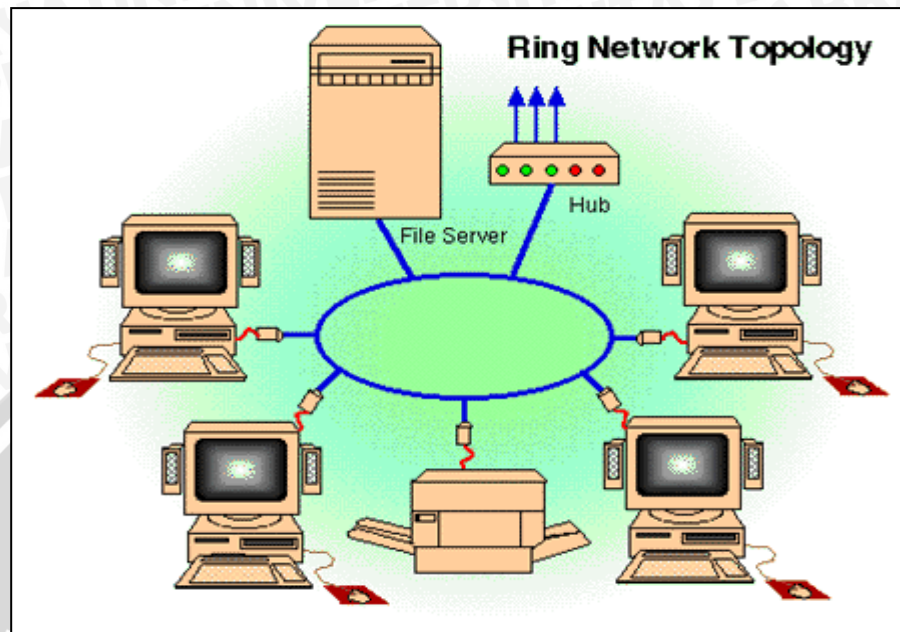
Kekurangan :

- Butuh jumlah kabel yang lebih banyak dibandingkan topologi yang lain.
- Kontrol terpusat (HUB) jadi elemen kritis.
- Perlu penanganan khusus

- Topologi Cincin

Topologi cincin atau yang sering disebut dengan *ring topologi* adalah topologi jaringan dimana setiap komputer yang terhubung membuat lingkaran. Dengan artian setiap komputer yang terhubung kedalam satu jaringan saling terkoneksi ke

dua komputer lainnya sehingga membentuk satu jaringan yang sama dengan bentuk cincin. Gambar 2.7 menunjukkan topologi jaringan cincin (ring).



Gambar 2.7 : Topologi jaringan Cincin (Ring)
Sumber : Telecommunications,1998:395

Keuntungan dan kekurangan dari topologi Ring adalah sebagai berikut:

Keuntungan :

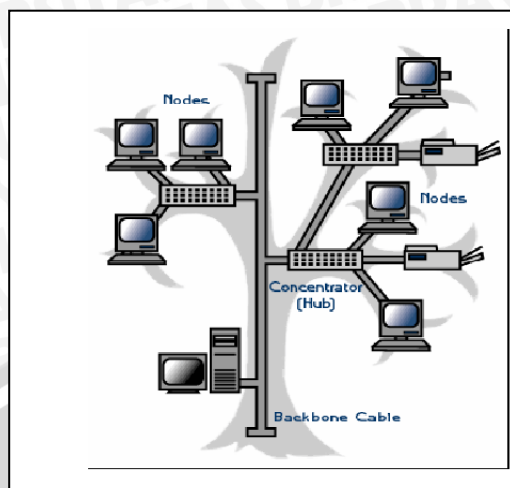
- Hemat kabel.

Kekurangan :

- Setiap node dalam jaringan selalu ikut serta mengelola informasi yang dilewatkan dalam jaringan.
- Pengembangan jaringan lebih kaku.
- Bila terjadi gangguan pada suatu node, maka seluruh jaringan akan terganggu.

- Topologi Pohon (*tree*)

Topologi jaringan ini disebut juga sebagai topologi jaringan bertingkat. Topologi ini biasanya digunakan untuk interkoneksi antar sentral dengan hirarki yang berbeda. Untuk hirarki yang lebih rendah digambarkan pada lokasi yang rendah dan semakin keatas mempunyai hirarki semakin tinggi. Topologi jaringan jenis ini cocok digunakan pada sistem jaringan komputer. Gambar 2.8 menunjukkan topologi jaringan pohon (*tree*).



Gambar 2.8 : Topologi jaringan pohon (tree)
Sumber : Telecommunications,1998:394

2.3.2 Komponen-komponen pada LAN

Pada dasarnya LAN terdiri dari beberapa komponen utama, yaitu:

- *Server*

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan atau *network operating system*. *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (*printer*), dan memberikan akses kepada *workstation* anggota jaringan.

Umumnya, di dalam sistem operasi *server* terdapat berbagai macam servis yang menggunakan arsitektur *client/server*. Contoh dari servis ini adalah *DHCP Server*, *Mail Server*, *HTTP Server*, *FTP Server*, *DNS Server* dan lain sebagainya. Setiap sistem operasi *server* umumnya membundel layanan-layanan tersebut atau layanan tersebut juga dapat diperoleh dari pihak ketiga. Setiap layanan tersebut akan merespons terhadap *request* dari klien. Sebagai contoh, klien *DHCP* akan memberikan *request* kepada *server* yang menjalankan servis *DHCP server*; ketika sebuah klien membutuhkan alamat *IP*, klien akan memberikan perintah/*request* kepada *server*, dengan bahasa yang dipahami oleh *server DHCP*, yakni protokol *DHCP* itu sendiri.

Server biasanya terhubung dengan *client* dengan kabel *UTP* dan sebuah *Network Card*. Kartu jaringan ini biasanya berupa kartu *PCI* atau *ISA*. Dilihat dari fungsinya,

server bisa di kategorikan dalam beberapa jenis, seperti: *server* aplikasi (*application server*), *server* data (*data server*) maupun *server* proxy (*proxy server*). *Server* aplikasi adalah *server* yang digunakan untuk menyimpan berbagai macam aplikasi yang dapat diakses oleh *client*, *server* data sendiri digunakan untuk menyimpan data baik yang digunakan *client* secara langsung maupun data yang diproses oleh *server* aplikasi. *Server* proxy berfungsi untuk mengatur lalu lintas di jaringan melalui pengaturan proxy. Orang awam lebih mengenal *proxy server* untuk mengkoneksikan komputer *client* ke Internet. (Warren Hioki, 1998:446)

- *NIC (Network Interface card)*

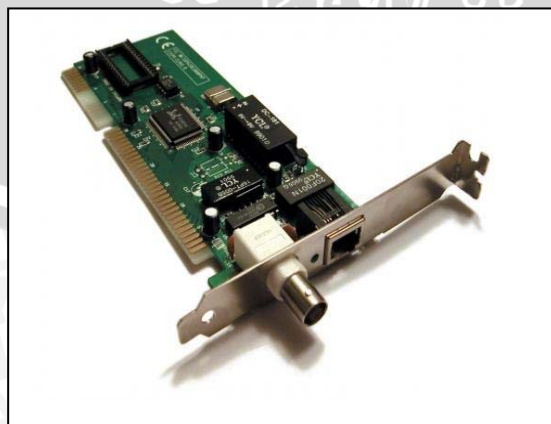
NIC adalah sebuah kartu yang berfungsi sebagai jembatan dari komputer ke sebuah jaringan komputer. Jenis *NIC* yang beredar, terbagi menjadi dua jenis, yakni *NIC* yang bersifat fisik, dan *NIC* yang bersifat logis. Contoh *NIC* yang bersifat fisik adalah *NIC Ethernet*, *Token Ring*, dan lainnya; sementara *NIC* yang bersifat logis adalah *loopback adapter* dan *Dial-up Adapter*. Disebut juga sebagai *Network Adapter*. Setiap jenis *NIC* diberi nomor alamat yang disebut sebagai *MAC address*, yang dapat bersifat statis atau dapat diubah oleh pengguna. Kartu *NIC* Fisik terbagi menjadi dua jenis, yakni:

- Kartu *NIC* dengan media jaringan yang spesifik (*Media-specific NIC*): yang membedakan kartu *NIC* menjadi beberapa jenis berdasarkan media jaringan yang digunakan. Contohnya adalah *NIC Ethernet*, yang dapat berupa *Twisted-Pair (UTP atau STP)*, *Thinnet*, atau *Thicknet*, atau bahkan tanpa kabel (*Wireless Ethernet*).
- Kartu *NIC* dengan arsitektur jaringan yang spesifik (*architecture-specific NIC*): yang membedakan kartu *NIC* menjadi beberapa jenis, sesuai dengan arsitektur jaringan yang digunakan. Contohnya adalah *Ethernet*, *Token Ring*, serta *FDDI (Fiber Distributed Data Interface)*, yang kesemuanya itu menggunakan *NIC* yang berbeda-beda. Kartu *NIC Ethernet* dapat berupa *Ethernet 10 Megabit/detik*, *100 Megabit/detik*, *1 Gigabit/detik* atau *10 Gigabit/detik*.

Tugas *NIC* adalah untuk mengubah aliran data paralel dalam bus komputer menjadi bentuk data serial sehingga dapat ditransmisikan di atas media jaringan. Media yang umum digunakan, antara lain adalah kabel *UTP Category 5* atau *Enhanced Category 5 (Cat5e)*, kabel *fiber-optic*, atau radio (jika memang tanpa kabel). Komputer dapat berkomunikasi dengan *NIC* dengan menggunakan beberapa metode, yakni *I/O*

yang dipetakan ke memori, Direct Memory Access (DMA), atau memory yang digunakan bersama-sama. Sebuah aliran data paralel akan dikirimkan kepada kartu NIC dan disimpan terlebih dahulu di dalam memori dalam kartu sebelum dipaketkan menjadi beberapa frame berbeda-beda, sebelum akhirnya dapat ditransmisikan melalui media jaringan. Proses pembuatan frame ini, akan menambahkan header dan trailer terhadap data yang hendak dikirimkan, yang mengandung alamat, pensinyalan, atau informasi pengecekan kesalahan. Frame-frame tersebut akan kemudian diubah menjadi pulsa-pulsa elektronik (voltase, khusus untuk kabel tembaga), pulsa-pulsa cahaya yang dimodulasikan (khusus untuk kabel fiber-optic), atau gelombang mikro (jika menggunakan radio/jaringan tanpa kabel). NIC yang berada dalam pihak penerima akan memproses sinyal yang diperoleh dalam bentuk terbalik, dan mengubah sinyal-sinyal tersebut ke dalam aliran bit (untuk menjadi frame jaringan) dan mengubah bit-bit tersebut menjadi aliran data paralel dalam bus komputer penerima. Beberapa fungsi tersebut dapat dimiliki oleh NIC secara langsung, diinstalasikan di dalam firmware, atau dalam bentuk perangkat lunak yang diinstalasikan dalam sistem operasi.

NIC logis merupakan jenis NIC yang tidak ada secara fisik dan menggunakan sepenuhnya perangkat lunak yang diinstalasikan di atas sistem operasi dan bekerja seolah-olah dirinya adalah sebuah NIC. Contoh dari perangkat NIC logis adalah loopback adapter (dalam sistem operasi Windows, harus diinstalasikan secara manual atau dalam sistem operasi keluarga UNIX, terinstalasi secara default, dengan nama interface **lo**) dan Dial-up adapter (yang menjadikan modem sebagai sebuah alat jaringan dalam sistem operasi Windows). Kartu NIC logis ini dibuat dengan menggunakan teknik emulasi. (http://id.wikipedia.org/wiki/Kartu_jaringan)



Gambar 2.9 : Network Interface Card

Sumber : <http://dim4siv3.wordpress.com/2008/10/28/perangkat-wireline/>

- Router

Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya, mencarikan jalur terbaik bagi paket yang terkirim dari suatu jaringan ke jaringan lain baik yang sama maupun berbeda protokolnya. *Router* sangat banyak digunakan dalam jaringan berbasis teknologi protokol *TCP/IP*, dan *router* jenis itu disebut juga dengan **IP Router**. Selain *IP Router*, ada lagi *AppleTalk Router*, dan masih ada beberapa jenis *router* lainnya. Internet merupakan contoh utama dari sebuah jaringan yang memiliki banyak *router IP*. *Router* dapat digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang disebut dengan *internetwork*, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya. *Router* juga kadang digunakan untuk mengoneksikan dua buah jaringan yang menggunakan media yang berbeda (seperti halnya *router wireless* yang pada umumnya selain ia dapat menghubungkan komputer dengan menggunakan radio, ia juga mendukung penghubungan komputer dengan kabel UTP), atau berbeda arsitektur jaringan, seperti halnya dari Ethernet ke Token Ring.

Router juga dapat digunakan untuk menghubungkan *LAN* ke sebuah layanan telekomunikasi seperti halnya telekomunikasi *leased line* atau *Digital Subscriber Line (DSL)*. *Router* yang digunakan untuk menghubungkan *LAN* ke sebuah koneksi *leased line* seperti T1, atau T3, sering disebut sebagai *access server*. Sementara itu, *router* yang digunakan untuk menghubungkan jaringan lokal ke sebuah koneksi *DSL* disebut juga dengan **DSL router**. *Router-router* jenis tersebut umumnya memiliki fungsi *firewall* untuk melakukan penapisan paket berdasarkan alamat sumber dan alamat tujuan paket tersebut, meski beberapa *router* tidak memilikinya. *Router* yang memiliki fitur penapisan paket disebut juga dengan *packet-filtering router*. *Router* umumnya memblokir lalu lintas data yang dipancarkan secara *broadcast* sehingga dapat mencegah adanya *broadcast storm* yang mampu memperlambat kinerja jaringan. (Onno W.Purbo,1998:34)

- Kabel *UTP (Unshielded twisted-pair cable)*

Unshielded twisted-pair cable (UTP) merupakan media yang tersusun atas 4 atau 8 pasang *wire* dan digunakan untuk bermacam-macam *network*. *Twisting* pada setiap pasang kabel dilakukan untuk menghasilkan efek *cancellation*, sehingga dapat

membatasi *degradasi* yang disebabkan oleh interferensi elektromagnetik dan interferensi frekuensi radio. Untuk lebih mengurangi *crosstalk* antara *pair* dalam *UTP*, jumlah lilitan (*twisting*) pada setiap *pair* berbeda. Gambar 2.10 menunjukkan gambar dari kabel *UTP*. (Warren Hioki, 1998:409)



Gambar 2.10 : kabel *UTP*

Sumber : Telecommunications,1998:410

Keuntungan dan kekurangan dari kabel *UTP* adalah sebagai berikut :

Kelebihan *UTP*:

- Mudah diinstal
- Lebih murah dibandingkan tipe media yang lain
- Memiliki diameter kecil, sehingga mempermudah dalam membuat saluran kabel

Kekurangan *UTP*:

- Lebih mudah terkena interferensi elektromagnetik dan noise
- Jarak maksimum kabel lebih kecil dibandingkan dengan kabel koaksial
- Lebih lambat dalam transmisi data
- *Switch*

Switch adalah media komunikasi jaringan yang menyerupai hub. *Switch* adalah peralatan sentral yang berfungsi sebagai penghubung komputer selain sebagai sarana komunikasi data. Perbedaannya antara *switch* dan hub adalah *switch* dapat beroperasi dengan mode *full-duplex* dan mampu mengalihkan jalur dan memfilter informasi ke dan dari tujuan yang spesifik. Gambar 2.12 menunjukkan gambar dari *Switch*. (<http://comstp.wordpress.com/2009/07/>)



Gambar 2.11 : switch

Sumber : (<http://comstp.wordpress.com/2009/07/>)

2.4 *Internet Protocol*

Berdasarkan pengalamatannya saat ini *Internet Protocol* terbagi atas dua jenis pengalamatan, yaitu *Internet Protocol versi 4 (Ipv4)* dan *Internet protocol versi 6 (Ipv6)*. *IP* versi 6 (*IPv6*) adalah protokol Internet versi baru yang didesain sebagai pengganti dari *Internet protocol* versi 4 (*IPv4*).

2.4.1 *Internet Protocol versi 4 (Ipv4)*

Ipv4 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IP versi 4. Panjang totalnya adalah 32-bit, dan secara teoritis dapat mengalami hingga 4 miliar *host* komputer atau lebih tepatnya 4.294.967.296 *host* di seluruh dunia, jumlah *host* tersebut didapatkan dari 256 (didapatkan dari 8 bit) dipangkat 4(karena terdapat 4 oktet) sehingga nilai maksimal dari alamat *IP* versi 4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga nilai nilai *host* yang dapat ditampung adalah $256 \times 256 \times 256 \times 256 = 4.294.967.296$ *host*. (Onno W.Purbo,1998:63)

❖ Representasi Alamat

Alamat *IP* versi 4 umumnya diekspresikan dalam notasi desimal bertitik (*dotted decimal notation*), yang dibagi ke dalam empat buah oktet berukuran 8-bit. Alamat *IP* yang dimiliki oleh sebuah *host* dapat dibagi dengan menggunakan subnet mask jaringan ke dalam dua buah bagian, yakni: (Onno W.Purbo,1998:65)

- *Network Identifier/NetID* atau *Network Address* (alamat jaringan) yang digunakan khusus untuk mengidentifikasi alamat jaringan di mana host berada. Dalam banyak kasus, sebuah alamat *network identifier* adalah sama dengan segmen jaringan fisik dengan batasan yang dibuat dan didefinisikan oleh *router* IP. Meskipun demikian, ada beberapa kasus di mana beberapa jaringan logis terdapat di dalam sebuah segmen jaringan fisik yang sama dengan menggunakan sebuah praktek yang disebut sebagai **multinetting**. Semua sistem di dalam sebuah jaringan fisik yang sama harus memiliki alamat *network identifier* yang sama. *Network identifier* juga harus bersifat unik dalam sebuah *internetwork*. Jika semua node di dalam jaringan logis yang sama tidak dikonfigurasi dengan menggunakan *network identifier* yang sama, maka terjadilah masalah yang disebut dengan *routing error*. Alamat *network identifier* tidak boleh bernilai 0 atau 255.
- *Host Identifier/HostID* atau *Host address* (alamat host) yang digunakan khusus untuk mengidentifikasi alamat host (dapat berupa *workstation*, *server* atau sistem lainnya yang berbasis teknologi *TCP/IP*) di dalam jaringan. Nilai *host identifier* tidak boleh bernilai 0 atau 255 dan harus bersifat unik di dalam *network identifier*/segmen jaringan di mana ia berada.

❖ Kelas-kelas Alamat

Alamat IP versi 4 dibagi ke dalam beberapa kelas (RFC 791), dilihat dari oktet pertamanya, seperti terlihat pada tabel. Sebenarnya yang menjadi pembeda kelas *IP* versi 4 adalah pola biner yang terdapat dalam oktet pertama (utamanya adalah bit-bit awal/*high-order bit*), tapi untuk lebih mudah mengingatnya, akan lebih cepat diingat dengan menggunakan representasi desimal. Tabel 2.4 menunjukkan pembagian kelas dalam IPv4 dan penggunaannya. (Onno W.Purbo,1998:67)

Tabel 2.4 : Pembagian kelas pada *IPv4* dan penggunaannya

Kelas Alamat IP	Oktet pertama (desimal)	Oktet pertama (biner)	Digunakan oleh
Kelas A	1–126	0xxx xxxx	Alamat <i>unicast</i> untuk jaringan skala besar
Kelas B	128–191	10xx xxxx	Alamat <i>unicast</i> untuk jaringan skala menengah hingga skala besar
Kelas C	192–223	110x xxxx	Alamat <i>unicast</i> untuk jaringan skala kecil
Kelas D	224–239	1110 xxxx	Alamat <i>multicast</i> (bukan alamat <i>unicast</i>)
Kelas E	240–255	1111 xxxx	Direservasikan; umumnya digunakan sebagai alamat percobaan (eksperimen); (bukan alamat <i>unicast</i>)

Sumber : TCP/IP, 1998:67

- Kelas A

Alamat-alamat kelas A diberikan untuk jaringan skala besar. Nomor urut bit tertinggi di dalam alamat *IP* kelas A selalu diset dengan nilai **0** (nol). Tujuh bit berikutnya—untuk melengkapi oktet pertama—akan membuat sebuah *network identifier*. 24 bit sisanya (atau tiga oktet terakhir) merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16,777,214 *host* tiap jaringannya. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme *Interprocess Communication (IPC)* di dalam mesin yang bersangkutan.

- Kelas B

Alamat-alamat kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua bit pertama di dalam oktet pertama alamat *IP* kelas B selalu diset ke bilangan biner **10**. 14 bit berikutnya (untuk melengkapi dua oktet pertama), akan membuat sebuah *network identifier*. 16 bit sisanya (dua oktet terakhir) merepresentasikan *host identifier*. Kelas B dapat memiliki 16,384 *network*, dan 65,534 *host* untuk setiap *network*-nya.

- Kelas C

Alamat IP kelas C digunakan untuk jaringan berskala kecil. Tiga bit pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner **110**. 21 bit selanjutnya (untuk melengkapi tiga oktet pertama) akan membentuk sebuah *network identifier*. 8 bit sisanya (sebagai oktet terakhir) akan merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2,097,152 buah network, dan 254 *host* untuk setiap *network*-nya.

- Kelas D

Alamat IP kelas D disediakan hanya untuk alamat-alamat *IP multicast*, sehingga berbeda dengan tiga kelas di atas. Empat *bit* pertama di dalam IP kelas D selalu diset ke bilangan biner **1110**. 28 *bit* sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali host. Untuk lebih jelas mengenal alamat ini, lihat pada bagian Alamat *Multicast IPv4*.

- Kelas E

Alamat IP kelas E disediakan sebagai alamat yang bersifat "eksperimental" atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat bit pertama selalu diset kepada bilangan biner **1111**. 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali host.

Tabel 2.5. Alamat IP khusus

Alamat	Kegunaan
127.0.0.1	localhost
HA semua 1 (202.46.249.255)	Ke semua host dalam network ini (multicast)
255.255.255.255	broadcast
10.0.0.0 – 10.255.255.255	Private Network (Class A)
172.16.0.0 – 172.31.255.255	Private Network (Class B)
192.168.0.0 – 192.168.255.255	Private Network (Class C)

Sumber : http://id.wikipedia.org/wiki/Alamat_IP_versi_4

❖ Subnetting

Subnetting adalah membagi suatu *network* menjadi sub-sub *network* yang lebih kecil. Cara kerja *subnetwork* mirip dengan cara kerja pada *network*, hanya pada skala yang lebih kecil (Onno W.Purbo,1998:78). Berikut ini adalah contoh dari perhitungan subnet pada kelas C :

Pada *IP* kelas C hanya 8 bit yang tersedia untuk *host* kelas C yaitu sebesar 256 *host*.

Subnet mask secara default pada kelas C adalah :

11111111 11111111 11111111 00000000 = 255.255.255.0

Subnet mask untuk *subnetting* pada kelas C bisa menjadi :

11111111 11111111 11111111 11000000 = 255.255.255.192

11111111 11111111 11111111 11100000 = 255.255.255.224

11111111 11111111 11111111 11110000 = 255.255.255.240

11111111 11111111 11111111 11111000 = 255.255.255.248

11111111 11111111 11111111 11111100 = 255.255.255.252

Contoh : kita punya jaringan kelas C dengan network address = 192.168.10.0 . network tersebut kita subnet dengan subnet mask 255.255.255.224 (x.x.x.11100000).

- Berarti jumlah subnet yang digunakan adalah $2^n - 2 = 2^3 - 2 = 6$, n adalah banyaknya angka binary 1 pada oktat terakhir dari subnet untuk kelas C yaitu 1 oktat terakhir.
- Banyaknya host per subnet adalah $2^n - 2 = 2^5 - 2 = 30$ host per subnet, dimana n adalah banyaknya angka binary 0 pada oktat terakhir dari subnet untuk kelas C yaitu 1 oktat terakhir.

	Network ID	Host ID	Broadcast ID
Subnet 1 (255.255.255.32)	192.168.10.32	x.x.x.33 – x.x.x.62	192.168.10.63
Subnet 2 (255.255.255.64)	192.168.10.64	x.x.x.65 – x.x.x.94	192.168.10.95
Subnet 3 (255.255.255.96)	192.168.10.96	x.x.x.97 – x.x.x.126	192.168.10.127
Subnet 4 (255.255.255.128)	192.168.10.128	x.x.x.128 – x.x.x.158	192.168.10.159
Subnet 5 (255.255.255.160)	192.168.10.160	x.x.x.161 – x.x.x.190	192.168.10.191
Subnet 6	192.168.10.192	x.x.x.193 – x.x.x.190	192.168.10.223

(255.255.255.192)			
-------------------	--	--	--

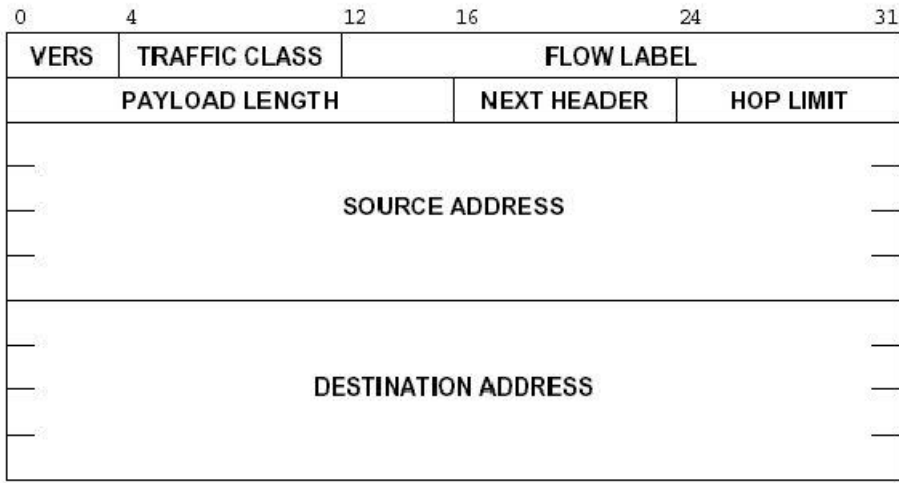
2.4.2 Internet Protocol versi 6 (Ipv6)

Ipv6 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IP versi 6. IPv6 merupakan penyempurnaan dari IP yang sudah digunakan digunakan sebelumnya, yaitu IPv4.

Berbeda dengan IPv4 yang hanya memiliki panjang 32-bit (jumlah total alamat yang dapat dicapainya mencapai 4.294.967.296 alamat), alamat IPv6 memiliki panjang 128-bit sehingga memiliki total alamat $2^{128}=3,4 \times 10^{38}$ alamat. Total alamat yang sangat besar ini bertujuan untuk menyediakan ruang alamat yang tidak akan habis (hingga beberapa masa ke depan), dan membentuk infrastruktur routing yang disusun secara hierarkis, sehingga mengurangi kompleksitas proses routing dan tabel routing.

❖ Arsitektur IPv6

Header IPv6 didesain mempunyai lebih sedikit *field* dibandingkan dengan IPv4, panjang header yang selalu tetap, dan fragmentasi yang terbatas pada paket IPv6 akan membuat *router* menjadi lebih cepat dalam memproses paket IPv6. Gambar 2.12 menunjukkan *header* IPv6 (Dikshie Fauzie,2003:2).



Gambar 2.12 : Header IPv6

Sumber : (Pengantar IPv6 dan implementasinya pada Free BSD,2003:2)

Header IPv6 mempunyai panjang yang tetap sebesar 40 bytes. Fields dalam header IPv6 dijelaskan sebagai berikut: (Dikshie Fauzie,2003:3)



- Field Version digunakan untuk menandai versi dari IP yang digunakan. Dalam IPv6 field ini berisi angka 6. Panjang field ini 4 bit.
- Field Traffic Class untuk menandai kelas atau prioritas dari paket IPv6. Ukuran field ini 8 bit.
- Field Flow Label untuk menandai bahwa paket tersebut dimiliki oleh urutan spesifik tertentu dari paket IPv6 antra asal dan tujuan. Field ini digunakan untuk aplikasi tertentu seperti aplikasi data *real-time*.
- Field Payload Length untuk menandai panjang dari payload.
- Field Next Header menandai tambahan header pertama jika ada atau jenis protokol padalapis atas PDU (Protokol Data Unit).
- Field Extension Header digunakan untuk tambahan fungsionalitas yang dibutuhkan seperti security dan sebagainya.
- Field Hop Limit untuk menandai maksimum hop yang dapat digunakan oleh IPv6 dalam lalu lintas internet.
- Field Source Address digunakan untuk menyimpan alamat IPv6 dari host asal. Ukuran field ini 128 bit.
- Field Destination Address digunakan untuk menyimpan alamat IPv6 dari host tujuan. Ukuran field ini 128 bit.

❖ Struktur pengalamatan

Pada IPv6 alamat 128-bit akan dibagi ke dalam 8 blok berukuran 16-bit, yang dapat dikonversikan ke dalam bilangan *heksadesimal* berukuran 4-digit. Setiap blok bilangan *heksadesimal* tersebut akan dipisahkan dengan tanda titik dua (:). Karenanya, format notasi yang digunakan oleh IPv6 juga sering disebut dengan *colon-hexadecimal format*, berbeda dengan IPv4 yang menggunakan *dotted-decimal format*. (Dikshie Fauzie,2003:4)

Berikut ini adalah contoh alamat IPv6 dalam bentuk bilangan biner:

```
001000011101101000000000110100110000000000000000010111100111011000000
1010101010000000001111111111110001010001001110001011010
```

Untuk menerjemahkannya ke dalam bentuk notasi *colon-hexadecimal format*, angka-angka biner di atas harus dibagi ke dalam 8 buah blok berukuran 16-bit:

```
0010000111011010 0000000011010011 0000000000000000 0010111100111011
0000001010101010 0000000011111111 111111000101000 1001110001011010
```

Lalu, setiap blok berukuran 16-bit tersebut harus dikonversikan ke dalam bilangan heksadesimal dan setiap bilangan heksadesimal tersebut dipisahkan dengan menggunakan tanda titik dua. Hasil konversinya adalah sebagai berikut:

21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A

Penyederhanaan bentuk alamat : Alamat di atas juga dapat disederhanakan lagi dengan membuang angka 0 pada awal setiap blok yang berukuran 16-bit di atas, dengan menyisakan satu digit terakhir. Dengan membuang angka 0, alamat di atas disederhanakan menjadi: 21DA:D3:0:2F3B:2AA:FF:FE28:9C5A

Konvensi pengalamatan IPv6 juga mengizinkan penyederhanaan alamat lebih jauh lagi, yakni dengan membuang banyak karakter 0, pada sebuah alamat yang banyak angka 0-nya. Jika sebuah alamat IPv6 yang direpresentasikan dalam notasi *colon-hexadecimal* format mengandung beberapa blok 16-bit dengan angka 0, maka alamat tersebut dapat disederhanakan dengan menggunakan tanda dua buah titik dua (::). Untuk menghindari kebingungan, penyederhanaan alamat IPv6 dengan cara ini sebaiknya hanya digunakan sekali saja di dalam satu alamat, karena kemungkinan nantinya pengguna tidak dapat menentukan berapa banyak bit 0 yang direpresentasikan oleh setiap tanda dua titik dua (::) yang terdapat dalam alamat tersebut. Tabel berikut mengilustrasikan cara penggunaan hal ini.

Tabel 2.6. Contoh pengalamatan pada *Ipv6*

Alamat asli	Alamat asli yang disederhanakan	Alamat setelah dikompres
FE80:0000:0000:0000:02AA:00FF:FE9A:4CA2	FE80:0:0:0:2AA:FF:FE9A:4CA2	FE80::2AA:FF:FE9A:4CA2
FF02:0000:0000:0000:0000:0000:0000:0002	FF02:0:0:0:0:0:0:2	FF02::2

Sumber : (Pengantar IPv6 dan implementasinya pada Free BSD,2003:4)

Untuk menentukan berapa banyak bit bernilai 0 yang dibuang (dan digantikan dengan tanda dua titik dua) dalam sebuah alamat IPv6, dapat dilakukan dengan menghitung berapa banyak blok yang tersedia dalam alamat tersebut, yang kemudian dikurangkan dengan angka 8, dan angka tersebut dikalikan dengan 16. Sebagai contoh,

alamat FF02::2 hanya mengandung dua blok alamat (blok FF02 dan blok 2). Maka, jumlah bit yang dibuang adalah $(8-2) \times 16 = 96$ buah bit.

Alamat IPv6 yang mempunyai panjang 128 bit dalam hexadesimal tentunya sulit dihapalkan karena itu alamat numerik jarang digunakan, lebih mudah menggunakan hostname, untuk itu DNS memegang peranan penting. Alamat IPv6 sendiri terbagi atas beberapa macam, berdasarkan RFC 3513 : (Dikshie Fauzie,2003:4)

- Unspecified dengan notasi ::1/128
- Loopback dengan notasi ::1/128
- Multicast dengan notasi ff00::/8
- Link local unicast dengan notasi FE80::/8
- Site local unicast dengan notasi FEC0::/8
- Global unicast

❖ Sistem pengalamatan

Pada IP v6 tidak dikenal adanya istilah pengkelasan. IPv6 mendukung beberapa jenis *format prefix*, yakni sebagai berikut:

- Alamat *Unicast*, yang menyediakan komunikasi secara *point-to-point*, secara langsung antara dua *host* dalam sebuah jaringan.
- Alamat *Multicast*, yang menyediakan metode untuk mengirimkan sebuah paket data ke banyak *host* yang berada dalam *group* yang sama. Alamat ini digunakan dalam komunikasi *one-to-many*.
- Alamat *Anycast*, yang menyediakan metode penyampaian paket data kepada anggota terdekat dari sebuah *group*. Alamat ini digunakan dalam komunikasi *one-to-one-of-many*. Alamat ini juga digunakan hanya sebagai alamat tujuan (*destination address*) dan diberikan hanya kepada router, bukan kepada *host-host* biasa.

Jika dilihat dari cakupan alamatnya, alamat unicast dan anycast terbagi menjadi alamat-alamat berikut:

- *Link-Local*, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam satu subnet.
- *Site-Local*, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam sebuah intranet.

- *Global Address*, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam Internet berbasis IPv6.

Unicast address

Alamat IPv6 *unicast* dapat diimplementasikan dalam berbagai jenis alamat, yakni:

- Alamat *unicast global*
- Alamat *unicast site-local*
- Alamat *unicast link-local*
- Alamat *unicast* yang belum ditentukan (*unicast unspecified address*)
- Alamat *unicast loopback*
- Alamat *unicast 6to4*
- Alamat *unicast ISATAP*

- **Unicast global address**

Alamat *unicast global* IPv6 mirip dengan alamat publik dalam alamat IPv4. Dikenal juga sebagai **Aggregatable Global Unicast Address**. Seperti halnya alamat publik IPv4 yang dapat secara *global* dirujuk oleh *host-host* di Internet dengan menggunakan proses *routing*, alamat ini juga mengimplementasikan hal serupa. Struktur alamat IPv6 *unicast global* terbagi menjadi topologi tiga level (*Public, Site, dan Node*).

Tabel 2.7 Struktur alamat IPv6 *unicast global*

Field	Panjang	Keterangan
001	3 bit	Berfungsi sebagai tanda pengenalan alamat, bahwa alamat ini adalah sebuah alamat IPv6 <i>Unicast Global</i> .
<i>Top Level Aggregation Identifier</i> (TLA ID)	13 bit	Berfungsi sebagai level tertinggi dalam hierarki routing. TLA ID diatur oleh <i>Internet Assigned Name Authority (IANA)</i> , yang mengalokasikannya ke dalam daftar <i>Internet registry</i> , yang kemudian mengolasikan sebuah TLA ID ke sebuah ISP global.
Res	8 bit	Direservasikan untuk penggunaan pada masa yang akan



		datang (mungkin untuk memperluas <i>TLA ID</i> atau <i>NLA ID</i>).
<i>Next Level Aggregation Identifier</i> (NLA ID)	24 bit	Berfungsi sebagai tanda pengenal milik situs (<i>site</i>) pelanggan tertentu.
<i>Site Level Aggregation Identifier</i> (SLA ID)	16 bit	Mengizinkan hingga 65536 (2^{16}) subnet dalam sebuah situs individu. SLA ID ditetapkan di dalam sebuah <i>site</i> . ISP tidak dapat mengubah bagian alamat ini.
<i>Interface ID</i>	64 bit	Berfungsi sebagai alamat dari sebuah node dalam subnet yang spesifik (yang ditentukan oleh SLA ID).

Sumber : http://www.tcpipguide.com/free/t_IPv6Addressing.htm

- **Unicast site-local addresses**

Alamat *unicast site-local* IPv6 mirip dengan alamat privat dalam IPv4. Ruang lingkup dari sebuah alamat terdapat pada *internetwork* dalam sebuah site milik sebuah organisasi. Penggunaan alamat *unicast global* dan *unicast site-local* dalam sebuah jaringan adalah mungkin dilakukan. *Prefiks* yang digunakan oleh alamat ini adalah FEC0::/48.

Tabel 2.8 Struktur alamat IPv6 *unicast site-local*

Field	Panjang	Keterangan
11111110110000000000000000000000	48 bit	Nilai ketetapan alamat <i>unicast site-local</i>
<i>Subnet Identifier</i>	16 bit	Mengizinkan hingga 65536 (2^{16}) subnet dalam sebuah struktur subnet datar. Administrator juga dapat membagi bit-bit yang memiliki nilai tinggi (high-order bit) untuk membuat sebuah infrastruktur routing hierarkis.
<i>Interface Identifier</i>	64 bit	Berfungsi sebagai alamat dari sebuah node dalam subnet yang spesifik.

Sumber : http://www.tcpipguide.com/free/t_IPv6Addressing.htm



- **Unicast link-local address**

Alamat *unicast link-local* adalah alamat yang digunakan oleh *host-host* dalam *subnet* yang sama. Alamat ini mirip dengan konfigurasi *APIPA (Automatic Private Internet Protocol Addressing)* dalam sistem operasi Microsoft Windows XP ke atas. *host-host* yang berada di dalam *subnet* yang sama akan menggunakan alamat-alamat ini secara otomatis agar dapat berkomunikasi. Alamat ini juga memiliki fungsi resolusi alamat, yang disebut dengan **Neighbor Discovery**. Prefiks alamat yang digunakan oleh jenis alamat ini adalah FE80::/64.

Tabel 2.9 Struktur alamat IPv6 *unicast site-local*

Field	Panjang	Keterangan
11111110100000000000000000000000 00000000000000000000000000000000 000000	64 bit	Berfungsi sebagai tanda pengenal alamat unicast link-local.
Interface ID	64 bit	Berfungsi sebagai alamat dari sebuah node dalam subnet yang spesifik.

Sumber : http://www.tcpipguide.com/free/t_IPv6Addressing.htm

- **Unicast unspecified address**

Alamat *unicast* yang belum ditentukan adalah alamat yang belum ditentukan oleh seorang administrator atau tidak menemukan sebuah *DHCP Server* untuk meminta alamat. Alamat ini sama dengan alamat IPv4 yang belum ditentukan, yakni **0.0.0.0**. Nilai alamat ini dalam IPv6 adalah 0:0:0:0:0:0:0 atau dapat disingkat menjadi dua titik dua (::).

- **Unicast Loopback Address**

Alamat *unicast loopback* adalah sebuah alamat yang digunakan untuk mekanisme interprocess communication (IPC) dalam sebuah *host*. Dalam IPv4, alamat yang ditetapkan adalah 127.0.0.1, sementara dalam IPv6 adalah **0:0:0:0:0:0:0:1**, atau **::1**.

- **Unicast 6to4 address**

Alamat *unicast 6to4* adalah alamat yang digunakan oleh dua *host* IPv4 dan IPv6 dalam Internet IPv4 agar dapat saling berkomunikasi. Alamat ini sering digunakan sebagai pengganti alamat publik IPv4. Alamat ini aslinya menggunakan *prefiks* alamat 2002::/16, dengan tambahan 32 bit dari alamat publik IPv4 untuk membuat sebuah *prefiks* dengan panjang 48-bit, dengan format **2002:WWXX:YYZZ::/48**, di mana WWXX dan YYZZ adalah representasi dalam notasi *colon-decimal format* dari notasi *dotted-decimal format* w.x.y.z dari alamat publik IPv4. Sebagai contoh alamat IPv4 157.60.91.123 diterjemahkan menjadi alamat IPv6 2002:9D3C:5B7B::/48. Meskipun demikian, alamat ini sering ditulis dalam format *IPv6 Unicast global address*, yakni 2002:WWXX:YYZZ:SLA ID:Interface ID.

- **Unicast ISATAP address**

Alamat Unicast ISATAP adalah sebuah alamat yang digunakan oleh dua *host* IPv4 dan IPv6 dalam sebuah Intranet IPv4 agar dapat saling berkomunikasi. Alamat ini menggabungkan *prefiks* alamat *unicast link-local*, alamat *unicast site-local* atau alamat *unicast global* (yang dapat berupa *prefiks* alamat 6to4) yang berukuran 64-bit dengan 32-bit ISATAP Identifier (0000:5EFE), lalu diikuti dengan 32-bit alamat IPv4 yang dimiliki oleh *interface* atau sebuah *host*. *Prefiks* yang digunakan dalam alamat ini dinamakan dengan **subnet prefix**. Meski alamat 6to4 hanya dapat menangani alamat IPv4 publik saja, alamat ISATAP dapat menangani alamat pribadi IPv4 dan alamat publik IPv4.

Multicast address

Alamat *multicast* IPv6 sama seperti halnya alamat *multicast* pada IPv4. Paket-paket yang ditujukan ke sebuah alamat *multicast* akan disampaikan terhadap semua *interface* yang dikenali oleh alamat tersebut. *Prefiks* alamat yang digunakan oleh alamat *multicast* IPv6 adalah FF00::/8.

Tabel 2.10 Struktur alamat IPv6 *multicast address*

Field	Panjang	Keterangan
11111111	8 bit	Tanda pengenal bahwa alamat ini adalah alamat <i>multicast</i> .
Flags	4 bit	Berfungsi sebagai tanda pengenal apakah alamat ini adalah alamat transient atau bukan. Jika nilainya 0, maka alamat ini bukan alamat transient, dan alamat ini merujuk kepada alamat multicast yang ditetapkan secara permanen. Jika nilainya 1, maka alamat ini adalah alamat transient.
Scope	4 bit	Berfungsi untuk mengindikasikan cakupan lalu lintas <i>multicast</i> , seperti halnya <i>interface-local</i> , <i>link-local</i> , <i>site-local</i> , <i>organization-local</i> atau <i>global</i> .
Group ID	112 bit	Berfungsi sebagai tanda pengenal group multicast

Sumber : http://www.tcpipguide.com/free/t_IPv6Addressing.htm

Anycast address

Alamat *Anycast* dalam IPv6 mirip dengan alamat *anycast* dalam IPv4, tapi diimplementasikan dengan cara yang lebih efisien dibandingkan dengan IPv4. Umumnya, alamat *anycast* digunakan oleh Internet Service Provider (ISP) yang memiliki banyak klien. Meskipun alamat *anycast* menggunakan ruang alamat *unicast*, tapi fungsinya berbeda daripada alamat *unicast*.

IPv6 menggunakan alamat *anycast* untuk mengidentifikasi beberapa *interface* yang berbeda. IPv6 akan menyampaikan paket-paket yang dialamatkan ke sebuah alamat *anycast* ke *interface* terdekat yang dikenali oleh alamat tersebut. Hal ini sangat berbeda dengan alamat *multicast*, yang menyampaikan paket ke banyak penerima, karena alamat *anycast* akan menyampaikan paket kepada salah satu dari banyak penerima.

2.5 Mean Opinion Score (MOS)

Pada komunikasi suara dan video, kualitas pada umumnya dinyatakan apakah suara atau video dinyatakan baik atau buruk. Untuk mengatakan nilai dari kualitas suara tersebut maka digunakan cara penilaian yang disebut *Mean Opinion Score (MOS)*. *MOS* merupakan standart dari *International Telecommunication Union - Telecommunication (ITU-T)* P.800 berupa angka yang didapatkan berdasarkan survey. Survey tersebut dilakukan oleh jumlah pria dan wanita yang seimbang untuk menilai kualitas dari suara

kita sebagai hasil dari pentransmisiannya. MOS menggunakan nilai antara rentang waktu 1 sampai 5, nilai 1 merukan nilai terburuk dan 5 merupakan nilai terbaik. Parameter dari setiap nilai (http://en.wikipedia.org/wiki/Mean_opinion_score) adalah sebagai berikut :

- 5 - Sangat baik, hampir menyerupai kualitas suara percakapan.
- 4 - baik, suara terdengar jelas seperti saat melakukan komunikasi dengan media telepon.
- 3 - suara dapat didengar tapi diperlukan usaha yang lebih untuk memahaminya.
- 2 - buruk, suara hampir tidak bisa dilakukan berkomunikasi
- 1 - sangat buruk, sangat tidak mungkin dilakukan komunikasi

MOS pada umumnya dilakukan untuk menilai kualitas dari *codec* yang digunakan. Berikut ini merupakan beberapa standart penilaian untuk beberapa *codec* yang sering digunakan :

Tabel 2.11 Standart MOS pada masing-masing codec

Codec	Data rate[kbit/s]	Mean opinion score (MOS)
<u>G.711 (ISDN)</u>	64	4.3
<u>iLBC</u>	15.2	4.14
<u>AMR</u>	12.2	4.14
<u>G.729</u>	8	3.92
<u>G.723.1r63</u>	6.3	3.9
<u>GSM EFR</u>	12.2	3.8
<u>G.726 ADPCM</u>	32	3.8
<u>G.729a</u>	8	3.7
<u>G.723.1r53</u>	5.3	3.65
<u>GSM FR</u>	12.2	3.5

Sumber : http://en.wikipedia.org/wiki/Mean_opinion_score

2.6 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.

2.6.1 Throughput

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

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

$$\lambda = \frac{1}{t_v} \quad (1)$$

Dengan;

λ = *throughput* (packet/s)

t_v = waktu rata-rata untuk mentransmisikan 1 paket yang benar (s)

dimana;

$$t_v = \frac{T_{total}}{N_{Paket}} \quad (2)$$

T_{total} = Total waktu transmisi

N_{paket} = Jumlah paket yang diterima

2.6.2 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.

Delay end to end merupakan *delay* antara *node* sumber dan *node* tujuan. *Delay end to end* pada sistem jaringan VoIP dihitung dari *client* ke *client*.

Secara Teori, *delay* pada jaringan dapat dibedakan dalam beberapa komponen yang berbeda. Total *delay end to end* adalah:

$$t_{end\ to\ end} = t_{codec} + t_{trans} + t_p + t_w \quad (3)$$

dengan :

$t_{end\ to\ end}$ = *delay end to end*

t_{codec} = *delay codec*

t_{trans} = *delay transmisi*

t_p = *delay propagasi*

t_w = *delay antrian*

2.6.2.1 Delay Codec

Delay codec terdiri dari *delay* paketisasi (t_{pack}) dan *delay* enkapsulasi (t_{enc}). *Delay* paketisasi pada *codec* terdiri dari beberapa komponen, yaitu *delay* pada sisi *encoder* (sisi pengirim) dan *delay* pada sisi *decoder* (sisi penerima). *Delay* pada sisi pengirim terdiri dari waktu mengakumulasi sampel suara ke dalam frame suara (*speech frame*), waktu mengkompresi paket suara, waktu untuk memuat frame suara ke dalam paket dan mentransfer paket tersebut ke jaringan *transport* dan *delay hardware*. Beberapa jenis *vocoder* tertentu memerlukan tambahan *delay* untuk keperluan kompresi yang disebut dengan *delay look-ahead*, yaitu menunggu bagian pertama pada frame berikutnya untuk memperoleh informasi yang diperlukan untuk membantu proses rekonstruksi sample suara jika terjadi paket yang hilang.

Delay paketisasi pada *decoder* terdiri dari waktu yang diperlukan untuk mendekompres frame suara ke dalam bentuk sample suara dan *hardware delay*. Mengacu pada standarisasi ITU.G..114 besarnya *delay* paketisasi adalah : (Salah Khaled.2002:11)

Untuk *High speed connection* :

$$\delta = (Nf + 1) \times \text{ukuran frame} + Dl \text{ (ms)} \quad (4)$$

Untuk *low speed connection* :

$$\delta = (2Nf + 1) \times \text{ukuran frame} + Dl \text{ (ms)} \quad (5)$$

dengan :

δ : *delay* paketisasi (ms)

Nf : banyaknya frame suara perpaket, ukuran frame suara (ms)

Dl : *delay look ahead* dan *delay hardware* (ms)

Yang disebut dengan *high speed connection* adalah jika waktu yang diperlukan untuk mentransfer paket suara ke fasilitas *transport* yang digunakan besarnya tidak signifikan jika dibandingkan dengan ukuran frame suara. Jika ukuran yang digunakan 10 ms, dan fasilitas *transport* yang digunakan adalah LAN 10 Mbps, maka dapat

dikategorikan sebagai *high speed connection*, karena waktu yang digunakan untuk mentransfer paket suara (misalkan 68 byte) kedalam koneksi 10 Mbps jauh lebih kecil dari 10 ms.

Delay paketisasi merupakan *fixed delay* yang nilainya tetap. Sedangkan *delay* enkapsulasi merupakan *delay* yang dibutuhkan untuk mengkonversi sinyal suara analog menjadi sinyal digital pada sisi pengirim, atau dari sinyal digital menjadi sinyal analog pada sisi penerima. *Delay* enkapsulasi merupakan *delay* yang nilainya tetap, bergantung pada jenis *codec* yang digunakan. Tabel 2.12 menunjukkan nilai dari *delay* paketisasi dan *delay* enkapsulasi.

Tabel 2.12 Nilai *delay* paketisasi dan *delay* enkapsulasi

<i>Codec</i>	Paketisasi (ms)	Enkapsulasi (ms)
G.711	20	1
G.729	30	25
G.7231 (MP-MLQ)	48	67.5
G.7231 (ACELP)	60	67.5

Sumber : *Deploying VoIP in Existing IP network : 6*

2.6.2.2 *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

$$t_i = \frac{(L + L')}{C} \times 8 = \frac{W_{frame}}{C} \times 8 \quad (6)$$

dengan :

- t_i = *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)

2.6.2.3 Delay Propagasi

Delay propagasi adalah waktu yang dibutuhkan untuk merambatkan paket multimedia melalui media transmisi dari *server* ke *client*. *Delay* propagasi ditentukan dengan persamaan :

$$t_p = t_{DTE} + t_{UTP} + t_{rou} + t_{HUB} \quad (7)$$

dengan :

t_p = *delay* propagasi (s)

t_{DTE} = *delay* pada NIC (*Network Interface Card*) (s)

t_{HUB} = *delay* pada HUB

t_{UTP} = *delay* pada UTP Cat.5 (s)

t_{rou} = *delay* pada router

Nilai dari *delay* pada komunikasi multimedia ditunjukkan pada tabel 2.13.

Tabel 2.13 Nilai *delay* pada komunikasi multimedia

Komponen sistem	<i>Delay</i>
Pengcodean Audio/video	2-150 ms, tergantung codec audio/video
Paketisasi	8 ms, tergantung ukuran paket
Local Area Network	0.1-200 ms, tergantung protokol
Backbone	10 ms
Router/bridge	5-10 ms, jika menerapkan skala prioritas
Wide Area Network	5 ms/km

Sumber : Grinham and Pratt, 1995 : 34

Sedangkan komponen *delay fast ethernet* ditunjukkan pada tabel 2.14.

Tabel 2.14 standar *delay* propagasi berbagai media transmisi

Komponen	Delay per meter	Delay maksimum	Round – Trip per meter	Maksimum Round Trip delay
DTE	-	25	-	100
UTP cat.3	0,57	57 (100m)	1,14	114 (100m)
UTP cat.4	0,57	57 (100m)	1,14	114 (100m)
UTP cat.5	0,556	55,6 (100m)	1,112	111,2 (100m)
STP	0,556	55,6 (100m)	1,112	111,2 (100m)
Fiber Optic	0,5	206 (412m)	1,0	412 (412m)

Sumber : <http://www.oreilly.com/catalog/enettds/CH.13.htm>

Mengacu pada tabel 2.13 dan tabel 2.14 maka persamaan 7 dapat ditulis kembali menjadi : (Noviatul Fardiyah,2008 : 48)

$$T_p = [N_{DTE} \times 25 \times \frac{1}{C}] + [0.556 \times P_{up} \times \frac{1}{C}] + [N_{hub} \times \mu s] + [N_{router} \times 5ms] \quad (8)$$

2.6.2.4 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. Besarnya *delay* antrian yang terjadi pada *client* yaitu (I Made Wiryana, 1992 : 7)

$$t_w = t_{queue} + t_{serv} \quad (9)$$

$$t_w = \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} \quad (10)$$

dengan :

t_w = *delay* antrian (s)

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

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

λ = kecepatan kedatangan paket pada *client* (paket/s)

μ = kecepatan pelayanan *client* (paket/s)

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

C = kapasitas kanal (bps)

l = panjang paket rata-rata (byte/paket)

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.

2.6.3 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. Prosentase *packet loss* maksimum yang diperbolehkan

oleh ISO untuk aplikasi multimedia adalah 5 %. Prosentase *packet loss* ditentukan dengan Persamaan 15 (Wijaya Hendra A, 2005 : 34) :

$$packet\ loss(\%) = \frac{N_{packet\ loss}}{N_{paket} + N_{packet\ loss}} \times 100\% \quad (11)$$

dengan :

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

N_{paket} = jumlah paket multimedia yang diterima dengan benar

2.6.4 Kapasitas Jaringan

Kapasitas jaringan atau 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 (Rafidyan Rasyid :2004). Kapasitas transmisi dapat dihitung dengan menggunakan persamaan :

$$K_T = \frac{BW_{st}}{BW_t} \quad (12)$$

Dengan :

K_T = Kapasitas Transmisi (komunikasi)

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

BW_T = *Bandwidth* transmisi (kbps)

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

$$BW_T = L_{PI} \times pps \quad (13)$$

Dengan :

L_{PI} = panjang paket internet (bit/paket)

Pps = packet per second (paket/s)

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

$$pps = \frac{BR}{P} \quad (14)$$

Dengan :

BR = *codec bit rate* (bps)

P = *payload* (bit)

BAB III

METODOLOGI PENELITIAN

Kajian yang digunakan dalam penelitian ini adalah kajian yang bersifat aplikatif yaitu perencanaan dan penerapan jaringan *VoIP* yang berbasis *SIP* menggunakan *IPv6*. Tahapan dalam perencanaan dan aplikasi skripsi ini terdiri dari beberapa tahap yaitu: jenis data dan cara mendapatkan data, variabel dan cara analisis yang digunakan, rangka keseluruhan proses penyelesaian skripsi yang dijelaskan dalam bentuk *flowchart*. Pengujian jaringan *VoIP* dilaksanakan di Kos jalan Candi Panggung Indah no.7 Malang dan pengambilan data menggunakan program *Wireshark*.

3.1 Jenis Data dan Cara Pengambilan Data

Jenis data yang digunakan dalam penyusunan skripsi ini terdiri dari data primer dan data sekunder.

3.1.1 Data Primer

Data primer adalah data yang diperoleh langsung dari obyek yang diuji. Data primer bersumber dari hasil pengujian yang meliputi hasil pengujian terhadap parameter-parameter yang diuji. Data primer terdiri dari :

- Total waktu transmisi (T_{total})
- Jumlah paket yang diterima (N_{paket})
- Jumlah paket yang hilang (N_{paket_loss})
- Panjang data yang diterima ($M_{data_diterima}$)

Data primer didapatkan dari pengujian. Pengujian Jaringan VoIP berbasis SIP menggunakan IPv6 dilakukan di Kos jalan Candi Panggung Indah no.7 Malang. Pengujian ini menggunakan laptop dengan *Linux Ubuntu versi 10* sebagai *Operating System*-nya, software *SIP-Communicator* pada sisi *client*, software *Opensips* pada sisi *server* dan software *Wireshark*. Gambar 3.1 menunjukkan diagram alir pengambilan data pada penelitian ini.



Gambar 3.1 Diagram Alir pengambilan data primer
Sumber : Perancangan

3.1.2 Data Sekunder

Data sekunder adalah data yang bersumber dari buku referensi, jurnal, skripsi, internet, dan forum-forum resmi mengenai jaringan *VoIP* berbasis *SIP*. Beberapa data sekunder yang digunakan antara lain :

- Kapasitas Hub yang digunakan 100 Mbit/s : sebagai kapasitas kanal yang digunakan untuk menghitung nilai dari delay transmisi.
- Kapasitas media transmisi UTP kategori 5 maksimal 100 Mbps : sebagai kapasitas kanal yang digunakan untuk menghitung nilai dari delay transmisi.
- Panjang header IPv6 sebesar 40 *bytes* : digunakan untuk menghitung delay enkapsulasi dan delay dekapulasi.
- Panjang header UDP sebesar 8 *byte/paket* : digunakan untuk menghitung delay enkapsulasi dan delay dekapulasi.
- Panjang header ethernet sebesar 14 *byte/paket* : digunakan untuk menghitung delay enkapsulasi dan delay dekapulasi.
- *Payload* G.711 sebesar 160 *byte/paket*.
- Panjang header RTP sebesar 12 *byte/paket*.

3.2 Variabel dan Cara Analisis Data

3.2.1 Variabel Data

Variabel data yang digunakan dalam skripsi ini terdiri dari tiga variabel, yaitu: *delay end-to-end*, probabilitas *packet loss*, dan *throughput*.

- *Delay end-to-end* : Secara Teori, *delay* pada jaringan dapat dibedakan dalam beberapa komponen yang berbeda. Total *delay end to end* adalah:

$$t_{end\ to\ end} = t_{codec} + t_{trans} + t_p + t_w$$

dengan :

$$t_{end\ to\ end} = \text{delay end to end}$$

$$t_{codec} = \text{delay codec}$$

$$t_{trans} = \text{delay transmisi}$$

$$t_p = \text{delay propagasi}$$

$$t_w = \text{delay antrian}$$

- Paket *loss* adalah jumlah paket yang hilang dibandingkan dengan paket yang diterima benar.

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

dengan :

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

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

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

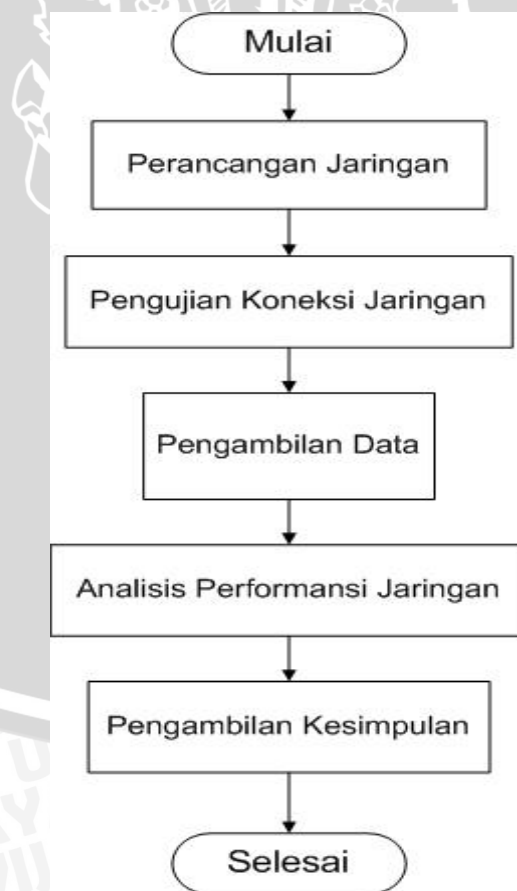
3.2.2 Cara Analisis Data

Cara analisis data yang digunakan dalam perencanaan dan penerapan jaringan *VoIP* yang berbasis *SIP* menggunakan *IPv6*. adalah sebagai berikut:

1. Analisis secara matematis yaitu mengumpulkan nilai dari parameter data-data primer dan sekunder dengan kesesuaian terhadap standar yang digunakan untuk kemudian dianalisis berdasarkan pembahasan yang telah diuraikan. Analisis yang dilakukan meliputi: perhitungan *delay end-to-end*, perhitungan probabilitas *packet loss*, perhitungan *throughput*, dan kapasitas jaringan.
2. Analisis secara grafis yaitu melakukan simulasi hasil perhitungan ke dalam bentuk grafis sehingga dapat diketahui karakteristik sistem yang diterapkan.

3.3 Kerangka Solusi Permasalahan

Rangka keseluruhan proses penyelesaian masalah yang telah dirumuskan dan ditelaah dari aspek tertentu, dan sesuai dengan tujuan yang telah ditetapkan dijabarkan dalam bentuk *flowchart*.



Gambar 3.2 Diagram Alir solusi permasalahan
Sumber : Perancangan

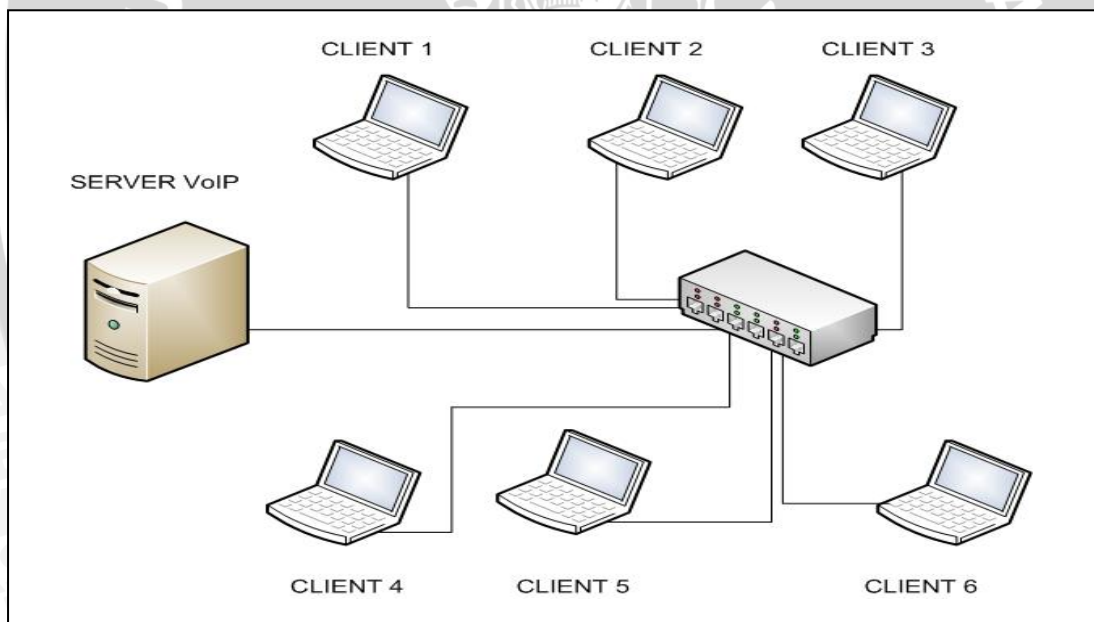
3.3.1 Perancangan Jaringan

Perancangan jaringan *VoIP* berbasis *SIP* menggunakan *IPv6* dibagi menjadi 3 bagian yaitu :

- Perancangan topologi jaringan
- Perancangan *server*
- Perancangan *client*

Perancangan Topologi Jaringan

Jaringan *VoIP* berbasis *SIP* yang dirancang pada skripsi ini diimplementasikan pada *LAN* menggunakan protokol *IPv6*. Jaringan terdiri dari satu *server*, satu *switch*, dan 6 *client*. Gambar 3.3 menunjukkan konfigurasi topologi jaringan yang akan digunakan pada penelitian ini.



Gambar 3.3 : konfigurasi jaringan *VoIP*
Sumber : Perancangan

Untuk merancang jaringan *VoIP* yang sesuai dengan topologi yang telah dirancang, maka diperlukan beberapa komponen, yaitu :

- e. 7 unit *Personal Computer* (PC) / Laptop (1 untuk *server* dan 6 untuk *client*)
- f. 1 unit HUB / *Switch*
- g. Kabel UTP Category 5
- h. 14 konektor RJ-45

Perancangan *Server-Client*

Perancangan *Server-Client* membutuhkan *software* yang meliputi *software* aplikasi *VoIP* berbasis *SIP* yang dapat menggunakan protokol *IPv6* pada server, Sedangkan pada sisi *Client* menggunakan *software Softphone*, untuk analisis jaringan menggunakan *software Wireshark* yang diaplikasikan pada sisi *Server* maupun *Client*.

Tabel 3.1 Spesifikasi perangkat lunak sistem

Perangkat Lunak	Server	Client
Sistem Operasi	<i>Linux Ubuntu v.10</i>	<i>Windows7</i>
Program Aplikasi	<i>Opensips</i>	<i>SIP-communicator</i>
Program Analisis	Wireshark	

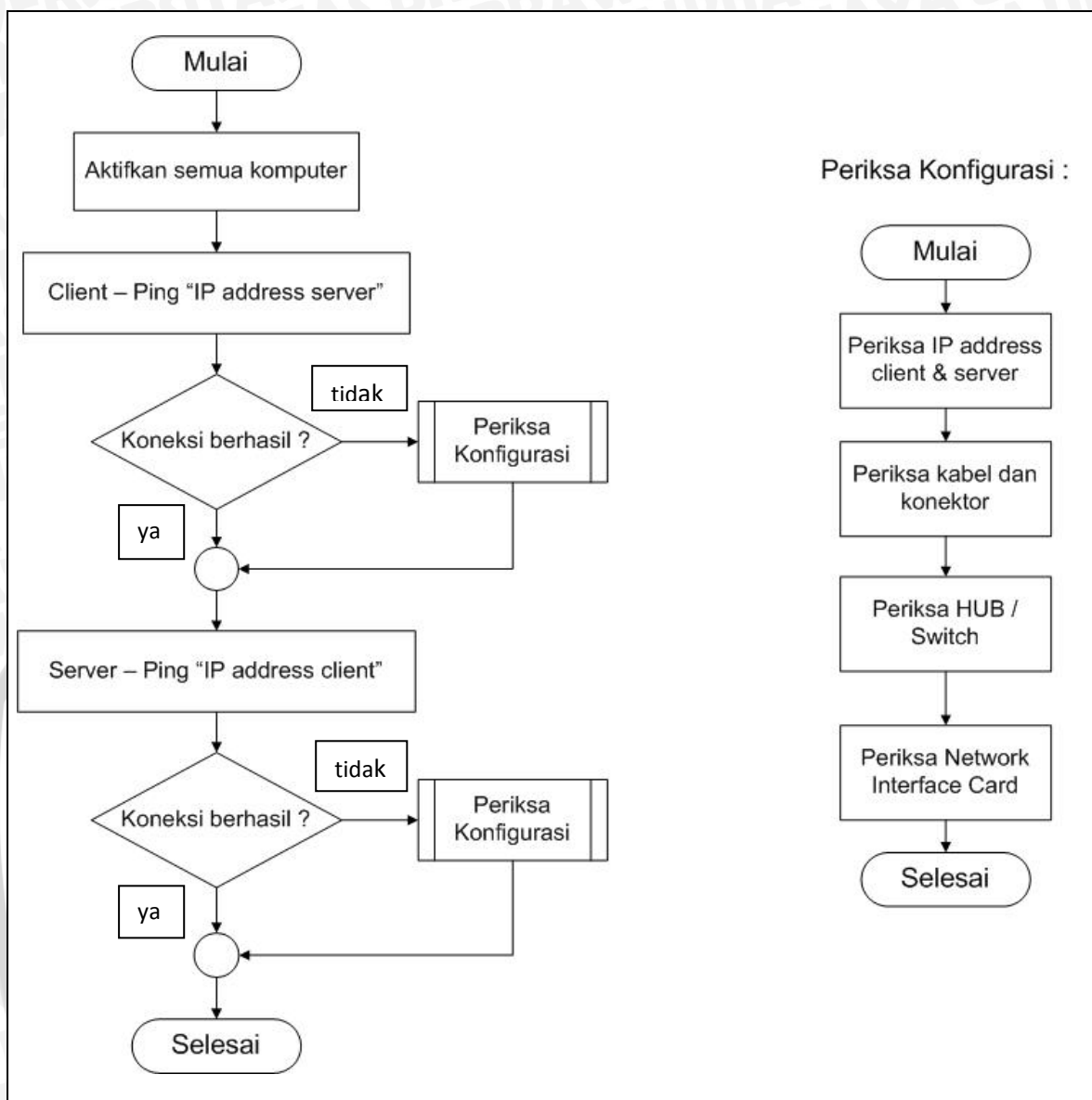
Sumber : Perancangan

3.3.2 Pengujian Jaringan

Pengujian pada skripsi ini bertujuan untuk mengetahui performansi jaringan *VoIP* berbasis *SIP* menggunakan *IPv6*. Pengujian ini meliputi : Pengujian Interkoneksi *client-server* dan Pengujian performansi jaringan *VoIP*.

- **Pengujian Interkoneksi *client-server***

Pengujian Interkoneksi ini dilakukan dengan tujuan mengetahui bahwa telah terjadi hubungan antara *client* dengan *server*, dan antara *client* dan *server* dapat melakukan proses komunikasi. Gambar 3.4 menunjukkan diagram alir pengujian interkoneksi *client-server*.



Gambar 3.4 : Diagram alir pengujian interkoneksi *client-server*
 Sumber : Perancangan

Berikut ini adalah penjelasan dari diagram alir di atas: Langkah pertama yang dilakukan adalah mengaktifkan seluruh komputer, setelah itu dilakukan test ping dari sisi Client menuju server, apabila proses ping berhasil maka dilanjutkan dengan test ping pada sisi server menuju client, apabila gagal maka dilakukan pemeriksaan terhadap konfigurasi jaringan antara lain : periksa IP address antara client dan server, periksa kabel dan konektor, periksa HUB atau switch, dan periksa network Interface card. Pada sisi server juga dilakukan hal yang sama seperti pada sisi client.



- **Pengujian Performansi Jaringan VoIP berbasis SIP menggunakan IPv6**

Pengujian ini bertujuan untuk mengetahui performansi jaringan VoIP berbasis SIP berdasarkan arsitektur jaringan IPv6 yang telah dirancang. Perhitungan performansi VoIP ini meliputi *Throughput*, *delay*, dan *packet loss* yang diukur menggunakan software jaringan Wireshark.

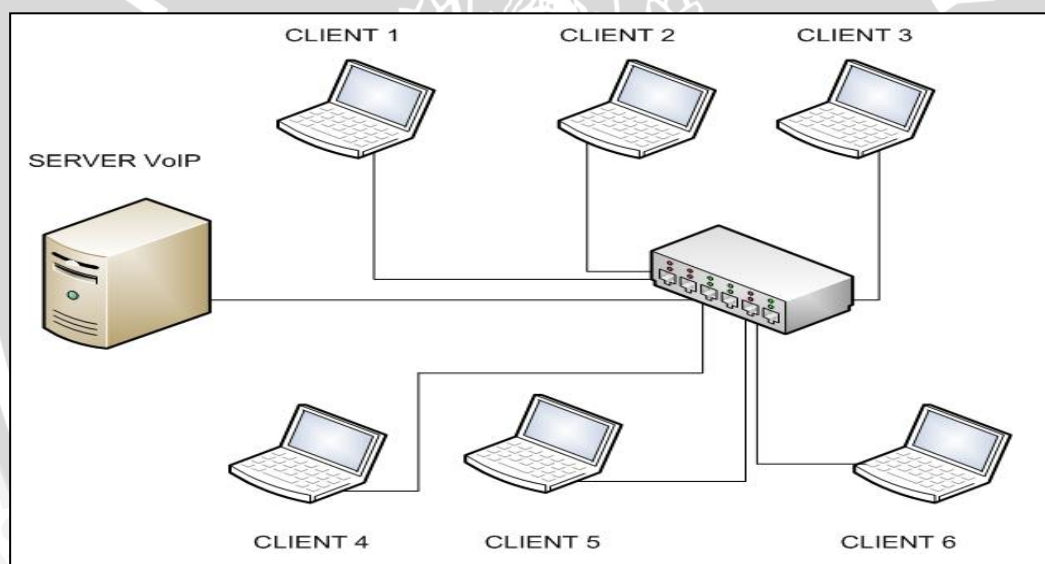


BAB IV PERANCANGAN JARINGAN

Bab IV ini menjelaskan tentang perancangan jaringan *VoIP* yang meliputi perancangan topologi jaringan, perancangan *server*, dan perancangan *client*. Secara lengkap uraian dari mekanisme perancangan ditunjukkan dalam sub bab dibawah ini.

4.1 Perancangan Topologi Jaringan

Topologi Jaringan *VoIP* yang dirancang pada penelitian ini berbentuk topologi *star* yang terdiri dari satu *server* dan enam *client*. Untuk menghubungkan *server* dan *client* digunakan *switch* yang berisi 8 *port* dan kabel UTP Category 5 yang dilengkapi konektor RJ-45. Gambar 4.1 menunjukkan topologi jaringan yang digunakan pada penelitian ini.



Gambar 4.1 : konfigurasi jaringan *VoIP*
Sumber : Perancangan

Untuk merancang jaringan *VoIP* sesuai dengan topologi yang telah dirancang, maka diperlukan beberapa komponen, komponen-komponen tersebut adalah :

- 7 unit *PC (Personal Computer)* / Laptop (1 unit digunakan sebagai *server*, 6 unit digunakan sebagai *client*).
- 1 unit *Switch* TP-LINK model TL-SF1008D yang berisi 8 port.
- 7 buah kabel UTP Category5 (Cat.5) sebanyak (masing-masing sepanjang 2 m).
- 14 buah konektor RJ-45.

Penjelasan komponen-komponen yang membentuk jaringan VoIP diatas adalah sebagai berikut :

PC (Personal Computer) / Laptop

Jaringan *VoIP* merupakan jaringan yang digunakan oleh banyak pengguna dalam waktu yang sama. Untuk mengambil sampel dari kondisi jaringan *VoIP* yang besar, maka perancangan jaringan *VoIP* dalam penelitian ini menggunakan satu *server* dan 6 *client*. Tabel 4.1 menunjukkan spesifikasi dari *PC/Laptop* yang digunakan untuk membangun jaringan *VoIP* pada penelitian ini.

Tabel 4.1 Spesifikasi PC/Laptop

Hardware	Server	Client1	Client2	Client3	Client4	Client5	Client6
Processor	Intel Pentium4 2.0 GHz	Intel Corei5 2.3GHz	Intel core2duo 2.4 GHz	Intel core2duo 2 GHz	Intel coreduo 2 GHz	Intel coreduo 2 GHz	Intel coreduo 2.2GHz
Memory	1 GB	2 GB	4 GB	2 GB	2 GB	2 GB	2 GB
Hardisk	80 GB	500 GB	120 GB	250 GB	120 GB	120 GB	120 GB
LAN card	I unit	I unit	I unit	I unit	I unit	I unit	I unit
Soundcard	I unit	I unit	I unit	I unit	I unit	I unit	I unit

Sumber : Perancangan

Switch

Untuk menghubungkan koneksi antara *server* dengan *client* sehingga menjadi suatu jaringan, diperlukan sebuah perangkat *Switch*. Pada perancangan jaringan, *switch* yang digunakan adalah TP-LINK model TL-SF1008D yang berisi 8 *port*. Berikut ini adalah gambar dan spesifikasi dari TP-LINK model TL-SF1008D.



Gambar 4.2 : TP-LINK model TL-SF1008D



Tabel 4.2 spesifikasi TP-LINK model TL-SF1008D

Standards	IEEE 802.3 10BASE-T Ethernet IEEE 802.3u 100BASE-TX Ethernet IEEE 802.3x Flow Control
Protocol	CSMA/CD
Data Transfer Rate	Ethernet : 10 Mbps (half-duplex), 20 Mbps (full-duplex) Fast Ethernet : 100 Mbps (half-duplex), 200 Mbps (full-duplex)
Topology	Star
Network Cables	10 BASE-T : 2 pair UTP Cat. 3,4,5 (100 m) 100 BASE-T : 2 pair UTP Cat.5 (100 m), EIA/TIA-568 100 ohm STP (100 m)
Number of Ports	8 X 10/100 Mbps auto-negotiation, auto MDI-X ports

Sumber : www.tp-link.com

Kabel UTP Cat. 5 dan Konektor RJ-45

Jaringan Voip yang dirancang pada penelitian ini diimplementasikan pada jaringan LAN. Untuk mengoneksikan client dan server melalui switch dibutuhkan kabel UTP Cat.5 yang masing-masing ujungnya dilengkapi dengan konektor RJ-45. Berikut ini adalah gambar dan spesifikasi dari kabel UTP Cat.5.



Gambar 4.3 : Kabel UTP Cat.5 dilengkapi dengan konektor RJ-45
Sumber : Telecommunications,1998:410

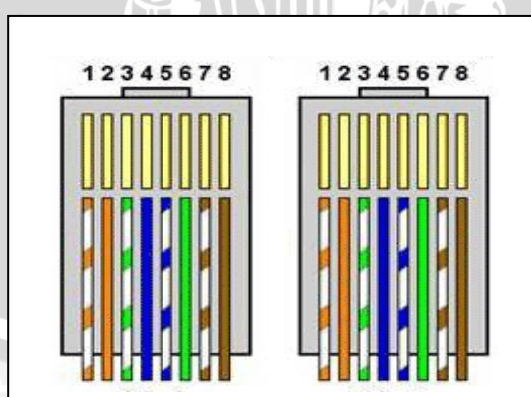


Tabel 4.3 Karakteristik kabel UTP Cat.5

Karakteristik	Frekuensi 10 MHz	Frekuensi 100 MHz
Atenuasi	20 dB / 1000 kaki	22 dB / 1000 kaki
Near-End Cross-Talk	47 dB / 1000 kaki	32.3 dB / 1000 kaki
Resistansi	28.6 Ω / 1000 kaki	28.6 Ω / 1000 kaki
Kapasitansi	18 pF / 1000 kaki	18 pF / 1000 kaki
Impedansi	100 Ω ($\pm 15\%$)	100 Ω ($\pm 15\%$)
Return Loss	16 dB	16 dB
Delay Propagasi	45 nano detik / 100 meter	45 nano detik / 100 meter

Sumber : <http://www.w3.org/1999/xhtml>

Ada dua macam metode pengkabelan, yaitu metode *straight* dan metode *cross*. Metode pengkabelan yang dilakukan pada perancangan jaringan VoIP ini adalah metode *straight*, yaitu dengan cara masing-masing ujung kabel UTP Cat. 5 dipasang konektor RJ-45 dengan konfigurasi *straight*. Metode pengkabelan konfigurasi *straight* dilakukan dengan tujuan untuk menghubungkan NIC pada masing-masing PC/Laptop dengan Switch, apabila menghubungkan antara PC dengan PC atau Switch dengan switch maka metode pengkabelan yang digunakan adalah metode *cross*. Pada pemasangan metode *straight* hanya digunakan 4 kabel saja, yaitu kabel urutan 1 dan 2 digunakan untuk transmit (putih oranye, oranye) dan urutan 3 dan 6 untuk receive (putih hijau, hijau). Walaupun demikian, kabel lainnya tetap ikut terpasang ke konektor RJ-45.



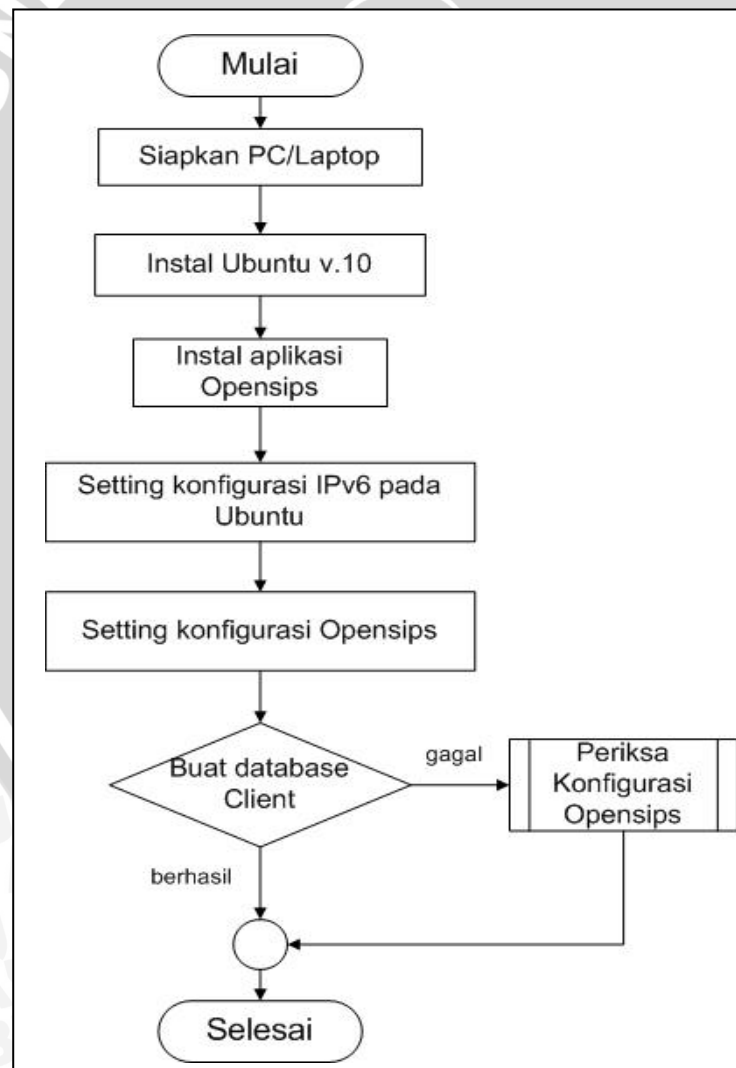
Gambar 4.4 : Kabel UTP dengan konfigurasi *straight*
 Sumber : Pengantar Pengkabelan dan Jaringan, 2003:12

4.2 Perancangan Server

Server VoIP yang digunakan pada jaringan ini adalah *Opensips* 1.6.0 yang diimplementasikan pada laptop dengan sistem operasi Linux Ubuntu versi 10, dengan spesifikasi sebagai berikut :

- Processor : Intel Pentium 4, 2.0 GHz
- Memori : 1 GB
- Hardisk : 80GB
- Operating System : Linux Ubuntu versi 10
- LAN Card : Broadcom Netlink (TM) Gigabit Ethernet
- IP Address : 2001:ffff:ffff:1

Secara keseluruhan diagram alir perancangan *server* dengan menggunakan *Opensips* 1.6.0 pada sistem jaringan *VoIP* ditunjukkan pada gambar 4.5.

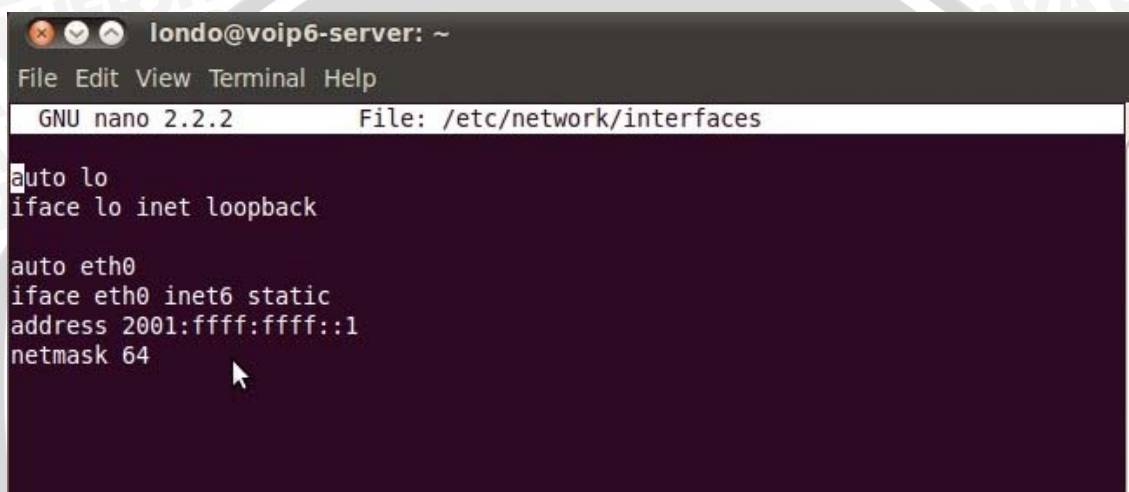


Gambar 4.5 : Diagram alir perancangan *server*
Sumber : Perancangan

Berikut ini adalah penjelasan dari diagram alir di atas : Hal pertama yang dilakukan adalah meng-instal sistem operasi Ubuntu versi 10, Setelah proses instalasi selesai, berikutnya adalah menginstal aplikasi *Opensips* pada Ubuntu versi 10. Setelah semua proses instalasi selesai, berikutnya adalah melakukan setting konfigurasi *IPv6* pada Ubuntu v.10 yaitu dengan cara :

Applications* → *Terminal* → ketik `sudo nano /etc/network/interfaces` → *enter

Gambar 4.6 menunjukkan tampilan setting konfigurasi *IPv6* pada ubuntu versi 10.



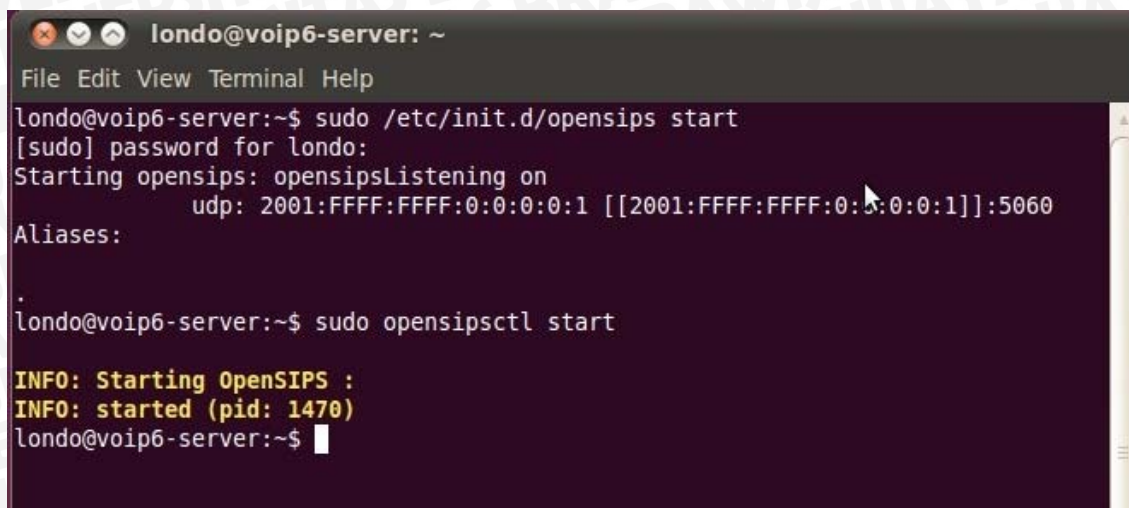
```
londo@voip6-server: ~
File Edit View Terminal Help
GNU nano 2.2.2 File: /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet6 static
address 2001:ffff:ffff::1
netmask 64
```

Gambar 4.6 : setting konfigurasi *IPv6*
Sumber : Perancangan

Setelah mengetikkan perintah di atas, selanjutnya adalah memasukkan alamat yang akan digunakan server yaitu `2001:ffff:ffff::1` dan secara otomatis netmask yang didapat adalah `64`.

Langkah berikutnya adalah menjalankan program *Opensips*, yaitu dengan mengetikkan perintah : ***sudo /etc/init.d/opensips start* → *enter*** setelah itu ketik lagi ***sudo opensipsctl start* → *enter***. Gambar 4.7 menunjukkan bahwa *Opensips* telah aktif dan siap untuk dioperasikan.



```

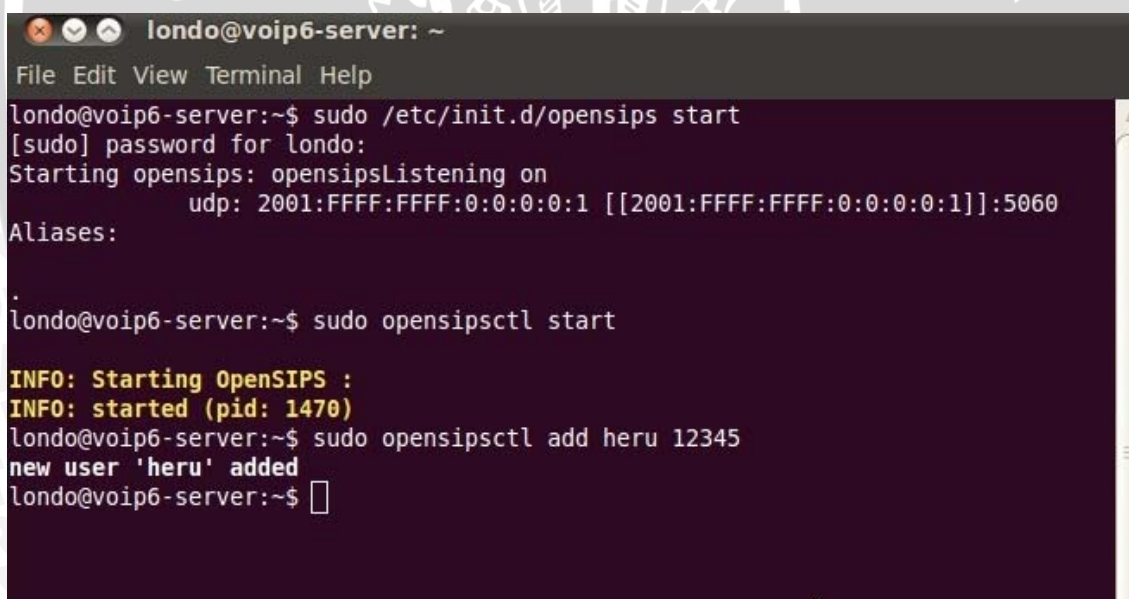
londo@voip6-server: ~
File Edit View Terminal Help
londo@voip6-server:~$ sudo /etc/init.d/opensips start
[sudo] password for londo:
Starting opensips: opensipsListening on
      udp: 2001:FFFF:FFFF:0:0:0:1 [[2001:FFFF:FFFF:0:0:0:1]]:5060
Aliases:
.
londo@voip6-server:~$ sudo opensipsctl start

INFO: Starting OpenSIPS :
INFO: started (pid: 1470)
londo@voip6-server:~$

```

Gambar 4.7 : running Opensips
Sumber : Perancangan

Setelah program *Opensips* diaktifkan, langkah berikutnya adalah memasukkan jumlah *client* yang ingin digunakan pada jaringan VoIP. Untuk menambahkan *client* dapat dilakukan dengan mengetikkan perintah : ***sudo opensipsctl add user password.***



```

londo@voip6-server: ~
File Edit View Terminal Help
londo@voip6-server:~$ sudo /etc/init.d/opensips start
[sudo] password for londo:
Starting opensips: opensipsListening on
      udp: 2001:FFFF:FFFF:0:0:0:1 [[2001:FFFF:FFFF:0:0:0:1]]:5060
Aliases:
.
londo@voip6-server:~$ sudo opensipsctl start

INFO: Starting OpenSIPS :
INFO: started (pid: 1470)
londo@voip6-server:~$ sudo opensipsctl add heru 12345
new user 'heru' added
londo@voip6-server:~$

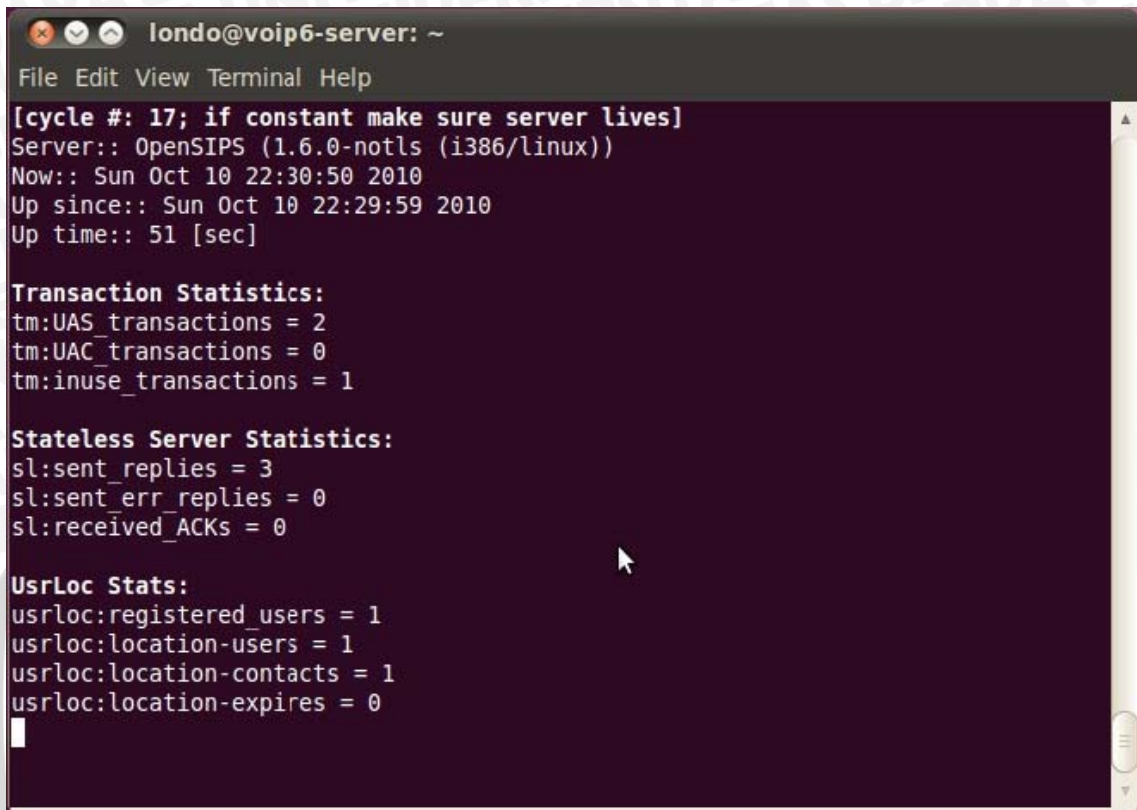
```

Gambar 4.8 : menambahkan *database client* pada Opensips
Sumber : Perancangan

Setelah muncul tampilan seperti gambar 4.8, maka selanjutnya adalah memasukkan database client dengan cara mengisi username dan password. Setelah itu muncul (new user 'username' added), hal ini berarti bahwa user telah berhasil ditambahkan pada database server.

Pada program *Opensips* ini juga dapat melakukan monitoring terhadap *client-client* yang terdaftar dalam *server Opensips*. Untuk melakukan monitoring terhadap

client-client tersebut dapat dilakukan dengan mengetik perintah : **sudo opensipsctl moni** → **enter**.



```
londo@voip6-server: ~
File Edit View Terminal Help
[cycle #: 17; if constant make sure server lives]
Server:: OpenSIPS (1.6.0-notls (i386/linux))
Now:: Sun Oct 10 22:30:50 2010
Up since:: Sun Oct 10 22:29:59 2010
Up time:: 51 [sec]

Transaction Statistics:
tm:UAS_transactions = 2
tm:UAC_transactions = 0
tm:inuse_transactions = 1

Stateless Server Statistics:
sl:sent_replies = 3
sl:sent_err_replies = 0
sl:received_ACKs = 0

UsrLoc Stats:
usrloc:registered_users = 1
usrloc:location-users = 1
usrloc:location-contacts = 1
usrloc:location-expires = 0
```

Gambar 4.9 : proses monitoring *client* pada *Opensips*
Sumber : Perancangan

Pada gambar 4.9 terlihat berapa banyak user yang telah teregristrasi pada server, berapa banyak user yang aktif, dan berapa banyak user yang sedang melakukan proses komunikasi pada jaringan VoIP.

4.3 Perancangan Client

Client pada jaringan VoIP ini menggunakan *software SIP-communicator*. *Client* berupa laptop/PC dengan spesifikasi sebagai berikut :

- a. Client 1
 - Processor : Intel Core i5 430M, 2.26 GHz
 - Memori : 2 GB
 - Operating System : Windows 7 Ultimate 32-bit
 - LAN Card : Broadcom Netlink Gigabit Ethernet
 - Sound Card : NVIDIA high devinition audio
 - IP Address : 2001::ffff:ffff::2
 - Subnet prefix length : 64

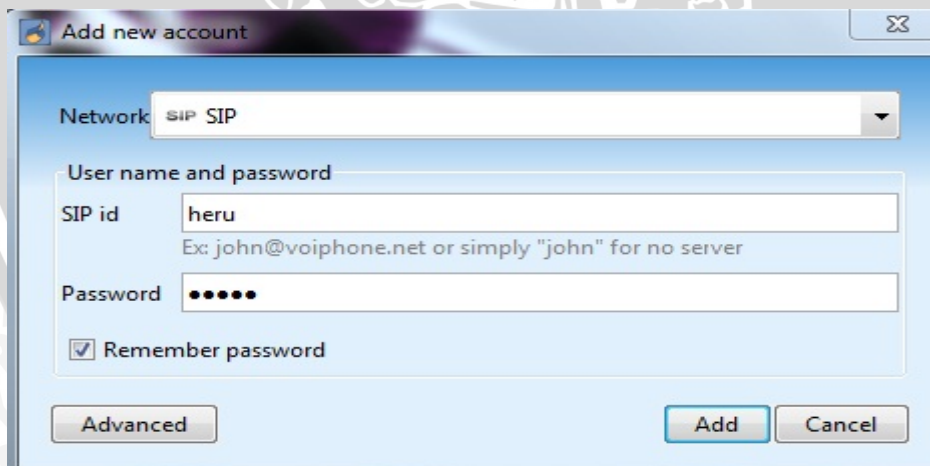
- Default Gateway : 2001:ffff:ffff::1
- b. Client 2
- Processor : Intel Core2duo, 2.4 GHz
 - Memori : 4 GB
 - Operating System : Windows 7 Ultimate 32-bit
 - LAN Card : Broadcom Netlink Gigabit Ethernet
 - Sound Card : Realtek high devinition audio
 - IP Address : 2001:ffff:ffff::3
 - Subnet prefix length : 64
 - Default Gateway : 2001:ffff:ffff::1
- c. Client 3
- Processor : Intel core2duo, 2 GHz
 - Memori : 2 GB
 - Operating System : Windows 7 Ultimate 32-bit
 - LAN Card : Realtek PCI-E fast ethernet NIC
 - Sound Card : Realtek high devinition audio
 - IP Address : 2001:ffff:ffff::4
 - Subnet prefix length : 64
 - Default Gateway : 2001:ffff:ffff::1
- d. Client 4
- Processor : Intel core duo, 2 GHz
 - Memori : 2 GB
 - Operating System : Windows Vista Home Basic
 - LAN Card : Marvell yukon PCI-E fast ethernet NIC
 - Sound Card : Connexant High devinition Audio
 - IP Address : 2001:ffff:ffff::5
 - Subnet prefix length : 64
 - Default Gateway : 2001:ffff:ffff::1
- e. Client 5
- Processor : Intel core duo, 2 GHz
 - Memori : 2 GB
 - Operating System : Windows Vista Home Basic

- LAN Card : Marvell yukon PCI-E fast ethernet NIC
- Sound Card : Connexant High devinition Audio
- IP Address : 2001:ffff:ffff::6
- Subnet prefix length : 64
- Default Gateway : 2001:ffff:ffff::1

f. Client 6

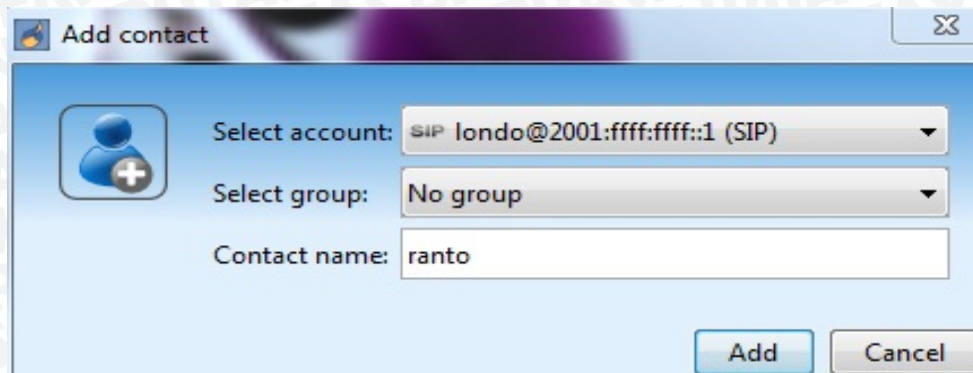
- Processor : Intel core duo, 2.2 GHz
- Memori : 2 GB
- Operating System : Windows 7 Ultimate 32-bit
- LAN Card : Broadcom Netlink Gigabit Ethernet
- Sound Card : NVIDIA high devinition audio
- IP Address : 2001:ffff:ffff::7
- Subnet prefix length : 64
- Default Gateway : 2001:ffff:ffff::1

Proses pertama yang dilakukan adalah meng-instal program *SIP-communicator* pada PC/laptop *client*. Setelah itu memasukkan *SIP ID* dan *password* yang telah ditambahkan dalam *database server VoIP* seperti ditunjukkan dalam gambar 4.10.



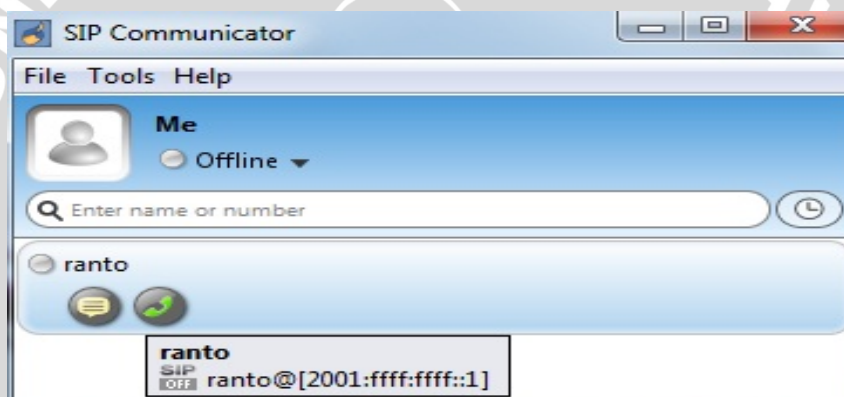
Gambar 4.10 : setting *Client* pada *SIP-communicator*
 Sumber : Perancangan

Setelah berhasil terhubung dengan *server* maka kita dapat menambahkan contact dengan cara : **File** → **add contact** lalu masukkan *SIP ID* yang ada pada *database server* yang ingin kita tambahkan sebagai daftar teman kita seperti ditunjukkan dalam gambar 4.11.



Gambar 4.11 : menambah *contact* pada *SIP-communicator*
Sumber : Perancangan

Setelah *contact* berhasil ditambahkan, kita dapat melakukan komunikasi dengan *contact* yang ada pada daftar teman kita dengan meng-klik tombol dial pada *SIP-communicator*, seperti yang ditunjukkan pada gambar 4.12.



Gambar 4.12 : *dialling contact* pada *SIP-communicator*
Sumber : Perancangan

Setelah semua proses perancangan jaringan yang meliputi perancangan topologi jaringan, perancangan server, perancangan client telah selesai dilakukan, maka jaringan VoIP berbasis SIP menggunakan alamat IPv6 telah siap untuk melakukan proses komunikasi.

BAB V

PENGUJIAN DAN ANALISIS

Pengujian pada penelitian ini bertujuan untuk mengetahui apakah jaringan yang telah dirancang sudah berjalan sesuai dengan yang diharapkan atau belum. Parameter-parameter yang diukur dalam pengujian ini adalah *trouput*, *delay end to end*, *packet loss* dan kapasitas jaringan.

5.1 Pengujian Interkoneksi *Client-Server*

Pengujian Interkoneksi *Client-Server* dilakukan dengan tujuan untuk memastikan bahwa semua *client* dapat melakukan hubungan komunikasi dengan *server* dan semua *client* dapat melakukan hubungan komunikasi antar *client*.

Alat-alat yang Digunakan

Peralatan yang digunakan dalam pengujian interkoneksi *client-server* adalah sebagai berikut :

- 7 unit *PC (Personal Computer)* / Laptop (1 unit digunakan sebagai *server*, 6 unit digunakan sebagai *client*).
- 1 unit *Switch TP-LINK* model TL-SF1008D yang berisi 8 port.
- 7 buah kabel UTP Category5 (Cat.5) sebanyak (masing-masing sepanjang 2 m).
- 14 buah konektor RJ-45.

Prosedur Pengujian

Prosedur pengujian dalam pengujian interkoneksi *client-server* pada jaringan *VoIP* adalah sebagai berikut :

- Semua alat dirangkai sesuai dengan topologi jaringan yang telah dirancang yang ditunjukkan pada gambar 4.1.
- Server* dan *client* dalam kondisi aktif
- Buka *command prompt display* pada setiap *client*, yaitu dengan cara klik **Start** → **Run** → *ketik cmd* → **enter**.
- Untuk melakukan pengujian interkoneksi kepada *server* maka semua *client* melakukan perintah Ping 2001:ffff:ffff::1
Untuk melakukan pengujian interkoneksi antar *client* Ping “alamat *client* yang akan dilakukan proses pengujian interkoneksi” .

Hasil Pengujian

Setelah melakukan semua prosedur di atas, apabila *server* dan *client* sudah terkoneksi dengan baik, maka *host* tujuan akan me-reply request dari *host* sumber. Gambar 5.1 menunjukkan hasil dari pengujian interkoneksi *client-server* pada jaringan VoIP.

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\londo>ping 2001:ffff:ffff::1

Pinging 2001:ffff:ffff::1 with 32 bytes of data:
Reply from 2001:ffff:ffff::1: time<1ms
Reply from 2001:ffff:ffff::1: time<1ms
Reply from 2001:ffff:ffff::1: time<1ms
Reply from 2001:ffff:ffff::1: time<1ms

Ping statistics for 2001:ffff:ffff::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\londo>
```

Gambar 5.1 : pengujian interkoneksi *client-server*
Sumber : Hasil Pengujian

Pada gambar 5.1 terlihat bahwa *host* tujuan me-reply request dari *host* sumber. Pada ping *statistic* terlihat bahwa paket yang dikirimkan sama dengan paket yang diterima, hal ini menunjukkan bahwa jaringan sudah terkoneksi dengan baik. Adapun hasil keseluruhan pengujian *client-server* ditunjukkan pada tabel 5.1.

Tabel 5.1 Hasil pengujian interkoneksi *client-server*

Client	Keterangan
1	Berhasil
2	Berhasil
3	Berhasil
4	Berhasil
5	Berhasil
6	Berhasil

Sumber : Hasil Pengujian

Dari hasil pengujian yang ditunjukkan tabel 5.1, bahwa semua *client* sudah terkoneksi dengan baik dengan *server*. Sehingga *server* dapat melayani permintaan komunikasi dari semua *client*.

5.2 Pengujian Jaringan VoIP Berbasis SIP Pada LAN Menggunakan IPv6

Pengujian dilakukan untuk mengetahui apakah sistem dapat berjalan dengan baik, serta dalam pengujian dilakukan pengambilan data sekunder yang kemudian dianalisis untuk mengetahui performansi dari sistem.

Alat-alat yang Digunakan

Peralatan yang digunakan dalam pengujian jaringan *VoIP* berbasis *SIP* pada *LAN* menggunakan *IPv6* adalah sebagai berikut :

- 7 unit *PC (Personal Computer)* / Laptop (1 unit digunakan sebagai *server*, 6 unit digunakan sebagai *client*).
- 1 unit *Switch TP-LINK* model TL-SF1008D yang berisi 8 port.
- 7 buah kabel UTP Category5 (Cat.5) sebanyak (masing-masing sepanjang 2 m).
- 14 buah konektor RJ-45.

Prosedur Pengujian

Prosedur pengujian dalam jaringan *VoIP* berbasis *SIP* pada *LAN* menggunakan *IPv6* adalah sebagai berikut :

- Server* dan keenam *client* dalam keadaan aktif.
- Aktifkan *software wireshark* pada semua *client*.
- Pengujian pertama adalah 2 *client* melakukan hubungan komunikasi, yaitu *client* 1 dengan alamat 2001:ffff:ffff::2 menghubungi *client* 3 dengan alamat 2001:ffff:ffff::4.
- Pengujian kedua adalah 4 *client* melakukan komunikasi, yaitu *client* 1 dengan alamat 2001:ffff:ffff::2 menghubungi *client* 3 dengan alamat 2001:ffff:ffff::4 dan *client* 2 dengan alamat 2001:ffff:ffff::3 menghubungi *client* 4 dengan alamat 2001:ffff:ffff::5.
- Pengujian kedua adalah 6 *client* melakukan komunikasi, yaitu *client* 1 dengan alamat 2001:ffff:ffff::2 menghubungi *client* 3 dengan alamat 2001:ffff:ffff::4 , *client* 2 dengan alamat 2001:ffff:ffff::3 menghubungi *client* 4 dengan alamat

2001:ffff:ffff::5 dan *client* 5 dengan alamat 2001:ffff:ffff::6 menghubungi *client* 6 dengan alamat 2001:ffff:ffff::7.

Hasil Pengujian

Dari pengujian yang dilakukan sesuai dengan prosedur di atas, maka dapat diketahui apakah *signal dialling* yang dikirimkan oleh *host* sumber dapat diterima atau tidak dapat diterima oleh destination client. Apabila signal dialling dapat diterima oleh *host* tujuan, maka pada *host* tujuan akan berdering dan terdapat *ringing message*.

Dari hasil ketiga skenario uji coba, diperoleh hasil seperti yang ditunjukkan pada tabel tabel 5.2.

Tabel 5.2 Hasil pengujian panggilan

Pengujian	Interkoneksi	keterangan
Pertama	<i>Client</i> 1 menghubungi <i>client</i> 3	<i>ringing</i>
Kedua	<i>Client</i> 1 menghubungi <i>client</i> 3 <i>Client</i> 2 menghubungi <i>client</i> 4	<i>ringing</i> <i>ringing</i>
Ketiga	<i>Client</i> 1 menghubungi <i>client</i> 3 <i>Client</i> 2 menghubungi <i>client</i> 4 <i>Client</i> 5 menghubungi <i>client</i> 6	<i>ringing</i> <i>ringing</i> <i>ringing</i>

Sumber : Hasil Pengujian

Berdasarkan tabel 5.2 terlihat bahwa pada setiap proses pengujian, host sumber berhasil melakukan komunikasi dengan host tujuan.

5.3 Analisis

Dari hasil pengujian didapatkan data-data yang selanjutnya dianalisis untuk mengetahui performansi dari jaringan *VoIP* yang telah dirancang. Data-data yang diperoleh melalui *software wireshark* pada pengujian jaringan *VoIP* berbasis *SIP* menggunakan *IPv6* ditunjukkan pada tabel 5.3

Tabel 5.3 Data hasil pengujian

Pengujian	Komunikasi antara	T _{total} (s)	N _{paket} (paket)	M _{data diterima} (byte)	N _{lost} (paket)
Pertama : 2 client	Client 1 (2001:ffff:ffff::2) dengan Client 3 (2001:ffff:ffff::4)	34,306	3210	377964	0
Kedua : 4 client	Client 1 (2001:ffff:ffff::2) dengan Client 3 (2001:ffff:ffff::4)	34,62	3160	370263	0
Ketiga : 6 client	Client 1 (2001:ffff:ffff::2) dengan Client 3 (2001:ffff:ffff::4)	33,69	3090	363940	0

Sumber : Hasil Pengujian

5.3.1 Analisis Throughput

Throughput adalah rata-rata paket data yang diterima benar yang dapat ditransmisikan pada satu waktu yang sama. *Throughput* pada jaringan komputer diukur dengan melihat jumlah paket data per detik. Pada pengujian ini, *throughput* dapat dihitung dengan menggunakan persamaan (1).

Pengujian 1 (2 client melakukan hubungan komunikasi)

Berdasarkan pengujian pada komunikasi antara *client* 1 (*IPv6 address* = 2001:ffff:ffff::2) dengan *client* 3 (*IPv6 address* = 2001:ffff:ffff::4), dapat diketahui :

- Jumlah paket data yang diterima (N_{paket}) = 3210 paket
- Total waktu transmisi (T_{total}) = 34,306 s

waktu rata-rata untuk mentransmisikan 1 paket yang benar (t_v) dapat dihitung dengan menggunakan persamaan (2).

$$\begin{aligned}
 t_v &= \frac{T_{total}}{N_{Paket}} \\
 &= \frac{34,306}{3210} \\
 &= 0.01068 \text{ s/paket}
 \end{aligned}$$

Dengan mensubstitusikan nilai t_v ke dalam persamaan (1), maka diperoleh nilai throughput sebagai berikut :

$$\begin{aligned}\lambda &= \frac{1}{t_v} \\ &= \frac{1}{0.01068} \\ &= 93.6329 \text{ paket/s}\end{aligned}$$

Pengujian 2 (4 *client* melakukan hubungan komunikasi)

Berdasarkan pengujian pada komunikasi antara *client* 1 (IPv6 address = 2001:ffff:ffff:2) dengan *client* 3 (IPv6 address = 2001:ffff:ffff:4), dapat diketahui :

- a. Jumlah paket data yang diterima (N_{paket}) = 3142 paket
- b. Total waktu transmisi (T_{total}) = 38,510 s

waktu rata-rata untuk mentransmisikan 1 paket yang benar (t_v) dapat dihitung dengan menggunakan persamaan (2).

$$\begin{aligned}t_v &= \frac{T_{\text{total}}}{N_{\text{Paket}}} \\ &= \frac{34,62}{3160} \\ &= 0.0109 \text{ s/paket}\end{aligned}$$

Dengan mensubstitusikan nilai t_v ke dalam persamaan (1), maka diperoleh nilai throughput sebagai berikut :

$$\begin{aligned}\lambda &= \frac{1}{t_v} \\ &= \frac{1}{0.0109} \\ &= 91,74 \text{ paket/s}\end{aligned}$$

Pengujian 3 (6 *client* melakukan hubungan komunikasi)

Berdasarkan pengujian pada komunikasi antara *client* 1 (IPv6 address = 2001:ffff:ffff:2) dengan *client* 3 (IPv6 address = 2001:ffff:ffff:4), dapat diketahui :

- a. Jumlah paket data yang diterima (N_{paket}) = 3225 paket
- b. Total waktu transmisi (T_{total}) = 55,576 s

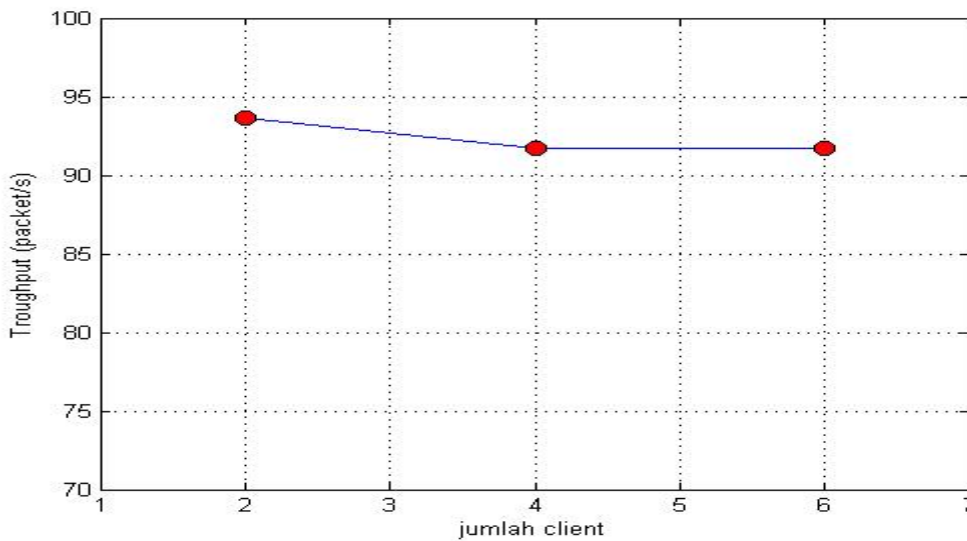
waktu rata-rata untuk mentransmisikan 1 paket yang benar (t_v) dapat dihitung dengan menggunakan persamaan (2).

$$\begin{aligned}
 t_v &= \frac{T_{total}}{N_{Paket}} \\
 &= \frac{33,69}{3090} \\
 &= 0.0109 \text{ s/paket}
 \end{aligned}$$

Dengan mensubstitusikan nilai t_v ke dalam persamaan (1), maka diperoleh nilai *troughput* sebagai berikut :

$$\begin{aligned}
 \lambda &= \frac{1}{t_v} \\
 &= \frac{1}{0.0109} \\
 &= 91,74 \text{ paket/s}
 \end{aligned}$$

Gambar 5.2 menunjukkan grafik analisa *troughput* pada pengujian pertama, pengujian kedua, dan pengujian ketiga.



Gambar 5.2 : Nilai *Troughput* terhadap jumlah *client*
 Sumber : Hasil Pengujian

Berdasarkan Gambar 5.2, dapat ditarik kesimpulan bahwa jumlah *client* tidak terlalu berpengaruh terhadap nilai dari *troughput*, hal ini disebabkan karena total waktu transmisi dan jumlah paket data yang diterima pada setiap pengujian hampir sama.



5.3.2 Analisis Packet loss

Packet loss adalah banyaknya paket data yang hilang ketika dikirimkan dari host sumber menuju host tujuan. Prosentase *packet loss* dapat dihitung dengan menggunakan persamaan (13).

Pengujian 1 (2 *client* melakukan hubungan komunikasi)

Berdasarkan pengujian pada komunikasi antara *client* 1 (*IPv6 address* = 2001:ffff:ffff::2) dengan *client* 3 (*IPv6 address* = 2001:ffff:ffff::4), dapat diketahui :

- Jumlah paket data yang diterima (N_{paket}) = 3210 paket
- Jumlah paket yang hilang (N_{lost}) = 0 paket

Dengan mensubstitusikan nilai t_v pada persamaan (13), maka diperoleh nilai prosentase packet loss sebagai berikut :

$$\begin{aligned} \text{packet loss}(\%) &= \frac{N_{\text{packet loss}}}{N_{\text{paket}} + N_{\text{packet loss}}} \times 100 \% \\ &= \frac{0}{3210} \times 100\% \\ &= 0 \% \end{aligned}$$

Pengujian 2 (4 *client* melakukan hubungan komunikasi)

Berdasarkan pengujian pada komunikasi antara *client* 1 (*IPv6 address* = 2001:ffff:ffff::2) dengan *client* 3 (*IPv6 address* = 2001:ffff:ffff::4), dapat diketahui :

- Jumlah paket data yang diterima (N_{paket}) = 3142 paket
- Jumlah paket yang hilang (N_{lost}) = 0 paket

Dengan mensubstitusikan nilai t_v pada persamaan (13), maka diperoleh nilai prosentase packet loss sebagai berikut :

$$\begin{aligned} \text{packet loss}(\%) &= \frac{N_{\text{packet loss}}}{N_{\text{paket}} + N_{\text{packet loss}}} \times 100 \% \\ &= \frac{0}{3160} \times 100\% \\ &= 0 \% \end{aligned}$$

Pengujian 3 (6 *client* melakukan hubungan komunikasi)

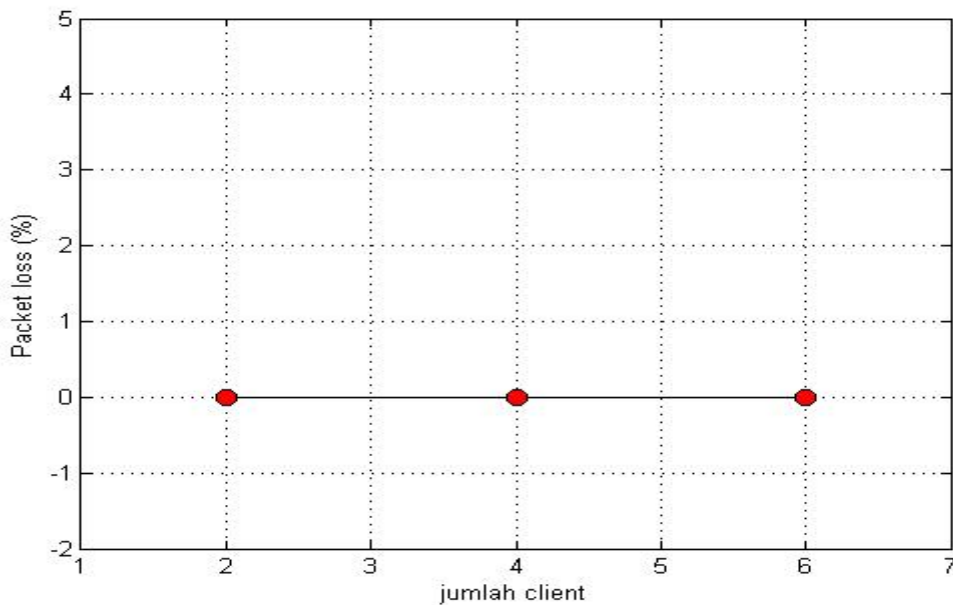
Berdasarkan pengujian pada komunikasi antara *client* 1 (*IPv6 address* = 2001:ffff:ffff:2) dengan *client* 3 (*IPv6 address* = 2001:ffff:ffff:4), dapat diketahui :

- a. Jumlah paket data yang diterima (N_{paket}) = 3225 paket
- b. Jumlah paket yang hilang (N_{lost}) = 0 paket

Dengan mensubstitusikan nilai t_v pada persamaan (13), maka diperoleh nilai prosentase *packet loss* sebagai berikut :

$$\begin{aligned} \text{packet loss}(\%) &= \frac{N_{\text{packet loss}}}{N_{\text{paket}} + N_{\text{packet loss}}} \times 100 \% \\ &= \frac{0}{3090} \times 100\% \\ &= 0 \% \end{aligned}$$

Gambar 5.3 menunjukkan grafik nilai dari prosentase *packet loss* pada semua pengujian.



Gambar 5.3 : Prosentase *Packet loss* terhadap jumlah *client*
Sumber : Hasil Pengujian

Berdasarkan Gambar 5.3 terlihat bahwa banyaknya jumlah *client* yang melakukan proses komunikasi tidak berpengaruh terhadap prosentase *packet loss* pada sistem jaringan VoIP yang telah dirancang. Hal ini terjadi karena pada setiap pengujian tidak ada data yang hilang ($N_{\text{lost}} = 0$ paket). Prosentase dari *packet loss* pada pengujian

jaringan VoIP pada penelitian ini tidak melebihi batas maksimum yang diperbolehkan ISO yaitu sebesar 5%.

5.3.3 Analisis Delay

Pengujian 1 (2 *client* melakukan hubungan komunikasi)

- **Delay Codec :**

Delay codec terdiri dari *delay* enkapsulasi (t_{enc}) dan *delay* paketisasi (t_{pack}).

Delay codec merupakan *delay* yang besarnya tetap, untuk *codec* G.711 dengan *payload* 160 byte.

$$(t_{enc}) = 20 \text{ ms}$$

$$(t_{pack}) = 1 \text{ ms}$$

Sehingga total *delay codec* pada sisi pengirim dan sisi penerima adalah :

$$\begin{aligned} T_{codec} &= 2 \times (t_{enc} + t_{pack}) \\ &= 2 \times (20 + 1) \\ &= 42 \text{ ms} \end{aligned}$$

- **Delay Transmisi :**

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket multimedia ke media transmisi. Semakin tinggi kecepatan transmisi, *delay* transmisi akan semakin kecil. Dengan menggunakan persamaan (6) maka besarnya *delay* transmisi adalah :

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

Diketahui :

- a. Panjang frame (W_{frame}) adalah 111 byte/paket
- b. Kapasitas media transmisi (C) adalah 100 Mbps, dimana 1Mb (Megabit) = (1024 x 1024)bit. Sehingga kapasitas media transmisi (C) adalah 104857600 bps

Dengan mensubstitusikan nilai W_{frame} dan C kedalam persamaan (6), maka diperoleh nilai *delay* transmisi

$$\begin{aligned} t_i &= \frac{W_{frame}}{C} \times 8 \\ &= \frac{111}{104857600} \times 8 \end{aligned}$$

$$= 0,000008468 \text{ s}$$

$$= 0,008468 \text{ ms}$$

- **Delay Propagasi :**

Delay propagasi tergantung pada jarak antara sumber dan tujuan. Besarnya delay propagasi dapat dihitung menggunakan standar propagasi pada setiap jenis saluran. Delay propagasi dapat dihitung dengan menggunakan persamaan (7).

$$t_p = t_{DTE} + t_{UTP} + t_{rou} + t_{HUB}$$

Berdasarkan topologi jaringan antara *client* 1 (*IPv6 address* = 2001:ffff:ffff:2) dengan *client* 3 (*IPv6 address* = 2001:ffff:ffff:4), dapat diketahui :

- Total NIC (N_{DTE}) yang digunakan adalah 3 unit.
- Panjang kabel UTP cat.5 (P_{UTP}) antara *client* 1 dengan *client* 3 adalah 4 meter.
- Total HUB (N_{HUB}) yang dilewati proses komunikasi antara *client* 1 dengan *client* 3 adalah 1 unit.
- Total router (N_{rou}) yang dilewati proses komunikasi antara *client* 1 dengan *client* 3 adalah 0 unit.
- Kapasitas media transmisi (C) adalah 100 Mbps = 104857600 bps

Dengan mensubstitusikan nilai N_{DTE} , N_{UTP} , N_{rou} , N_{HUB} dan C pada persamaan (8), maka diperoleh nilai delay propagasi sebagai berikut :

$$\begin{aligned} t_p &= [N_{DTE} \times 25 \times \frac{1}{C}] + [0,556 \times P_{utp} \times \frac{1}{C}] + [N_{hub} \times \mu s] + [N_{router} \times 5ms] \\ &= [\frac{3 \times 25}{104857600}] + [\frac{0,556 \times 4}{104857600}] + [1 \times 0,46 \times 10^{-6}] + 0 \\ &= 0,000000715 + 0,0000000212 + 0,00000046 + 0 \\ &= 0,00000119 \text{ s} \\ &= 0,00119 \text{ ms} \end{aligned}$$

- **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. Dari hasil pengujian komunikasi antara *client* 1 (*IPv6 address* = 2001:ffff:ffff:2) dengan *client* 3 (*IPv6 address* = 2001:ffff:ffff:4), dapat diketahui :

- Jumlah paket data yang diterima (N_{paket}) = 3210 paket

b. Panjang data yang diterima ($M_{\text{data diterima}}$) = 377964 byte

c. $Troughput$ = 93.6329 paket/s

Panjang paket rata-rata (l) dapat dihitung dengan cara :

$$\begin{aligned} l &= \frac{M_{\text{data diterima}}}{N_{\text{paket}}} \\ &= \frac{377964}{3210} \\ &= 117,75 \text{ byte/paket} \end{aligned}$$

Sehingga kecepatan pelayanan (μ) dapat dihitung

$$\begin{aligned} \mu &= \frac{C}{l \times 8} \\ &= \frac{104857600}{117,75 \times 8} \\ &= 111313,8 \text{ paket/s} \end{aligned}$$

Dengan mensubstitusikan nilai λ dan μ ke dalam persamaan (10), maka nilai *delay* antrian adalah :

$$\begin{aligned} t_w &= \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} \\ &= \frac{93.6329}{111313,8 (111313,8 - 93.6329)} + \frac{1}{111313,8} \\ &= 0.00000000756 + 0.00000898 \\ &= 0.00000898 \text{ s} \\ &= 0.00898 \text{ ms} \end{aligned}$$

- **Delay end to end**

Berdasarkan hasil perhitungan di atas, maka nilai dari *delay end to end* dapat dihitung dengan menggunakan persamaan (3).

$$\begin{aligned} t_{\text{end to end}} &= t_{\text{codec}} + t_{\text{trans}} + t_p + t_w \\ &= 42 \text{ ms} + 0,008468 \text{ ms} + 0,00119 \text{ ms} + 0.00898 \text{ ms} \\ &= 42.018 \text{ ms} \end{aligned}$$

Pengujian 2 (4 client melakukan hubungan komunikasi)

- **Delay Codec :**

Karena menggunakan *codec* yang sama maka nilai dari *delay codec* pada pengujian kedua sama dengan nilai dari *delay codec* pada pengujian pertama yaitu sebesar 42 ms.

- **Delay Transmisi :**

Delay transmisi adalah waktu yang dibutuhkan untuk meletakkan sebuah paket multimedia ke media transmisi. Semakin tinggi kecepatan transmisi, *delay* transmisi akan semakin kecil. Dengan menggunakan persamaan (6) maka besarnya *delay* transmisi adalah :

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

Diketahui :

c. Panjang frame (W_{frame}) adalah 115 byte/paket

d. Kapasitas media transmisi (C) adalah 100 Mbps, dimana 1Mb (Megabit) = (1024 x 1024)bit. Sehingga kapasitas media transmisi (C) adalah 104857600 bps

Dengan mensubstitusikan nilai W_{frame} dan C kedalam persamaan (6), maka diperoleh nilai *delay* transmisi

$$\begin{aligned} t_i &= \frac{W_{frame}}{C} \times 8 \\ &= \frac{115}{104857600} \times 8 \\ &= 0,00000877 \text{ s} \\ &= 0,00877 \text{ ms} \end{aligned}$$

- **Delay Propagasi :**

Karena menggunakan komponen-komponen yang sama dan panjang dari kabel UTP cat.5 sama panjang, maka nilai dari *delay* propagasi pada pengujian kedua ini sama dengan nilai *delay* propagasi pada pengujian pertama yaitu sebesar 0,00119 ms.

- **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. Dari hasil pengujian komunikasi antara *client* 1 (*IPv6 address* = 2001:ffff:ffff::2) dengan *client* 3 (*IPv6 address* = 2001:ffff:ffff::4), dapat diketahui :

- d. Jumlah paket data yang diterima (N_{paket}) = 3160 paket
- e. Panjang data yang diterima ($M_{\text{data diterima}}$) = 370263 byte
- f. *Troughput* = 91,74 paket/s

Panjang paket rata-rata (l) dapat dihitung dengan cara :

$$l = \frac{M_{\text{data diterima}}}{N_{\text{paket}}}$$

$$= \frac{370263}{3160}$$

$$= 117,17 \text{ byte/paket}$$

Sehingga kecepatan pelayanan (μ) dapat dihitung

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

$$= \frac{104857600}{117,17 \times 8}$$

$$= 111864,81 \text{ paket/s}$$

Dengan mensubstitusikan nilai λ dan μ ke dalam persamaan (10), maka nilai *delay* antrian adalah :

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

$$= \frac{91,74}{111864,81 (111864,81 - 91,74)} + \frac{1}{111864,81}$$

$$= 0.00000000734 + 0.00000895$$

$$= 0.00000895 \text{ s}$$

$$= 0.00895 \text{ ms}$$

- **Delay end to end**

Berdasarkan hasil perhitungan di atas, maka nilai dari *delay end to end* dapat dihitung dengan menggunakan persamaan (3).

$$t_{\text{end to end}} = t_{\text{codec}} + t_{\text{trans}} + t_p + t_w$$

$$= 42 \text{ ms} + 0,00877 \text{ ms} + 0,00119 \text{ ms} + 0.00895 \text{ ms}$$

$$= 42.018 \text{ ms}$$

Pengujian 3 (6 client melakukan hubungan komunikasi)

- **Delay Codec :**

Karena menggunakan *codec* yang sama maka nilai dari *delay codec* pada pengujian ketiga sama dengan nilai dari *delay codec* pada pengujian pertama yaitu sebesar 42 ms.

- **Delay Transmisi :**

Karena besar frame dan kapasitas kanal pada pengujian ketiga sama dengan besar frame dan kapasitas kanal pada pengujian kedua, maka nilai dari *delay transmisi* pada pengujian ketiga sama dengan nilai *delay transmisi* pada pengujian kedua yaitu sebesar 0,00877 ms.

- **Delay Propagasi :**

Karena menggunakan komponen-komponen yang sama dan panjang dari kabel UTP cat.5 sama panjang, maka nilai dari *delay propagasi* pada pengujian ketiga ini sama dengan nilai *delay propagasi* pada pengujian pertama yaitu sebesar 0,00119 ms.

- **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. Dari hasil pengujian komunikasi antara *client 1* (*IPv6 address* = 2001:ffff:ffff:2) dengan *client 3* (*IPv6 address* = 2001:ffff:ffff:4), dapat diketahui :

- g. Jumlah paket data yang diterima (N_{paket}) = 3090 paket
- h. Panjang data yang diterima ($M_{\text{data diterima}}$) = 363940 byte
- i. *Troughput* = 91,74 paket/s

Panjang paket rata-rata (l) dapat dihitung dengan cara :

$$\begin{aligned} l &= \frac{M_{\text{data diterima}}}{N_{\text{paket}}} \\ &= \frac{363940}{3090} \\ &= 117,78 \text{ byte/paket} \end{aligned}$$

Sehingga kecepatan pelayanan (μ) dapat dihitung

$$\begin{aligned}\mu &= \frac{C}{lx8} \\ &= \frac{104857600}{117,78 \times 8} \\ &= 111285,44 \text{ paket/s}\end{aligned}$$

Dengan mensubstitusikan nilai λ dan μ ke dalam persamaan (10), maka nilai *delay* antrian adalah :

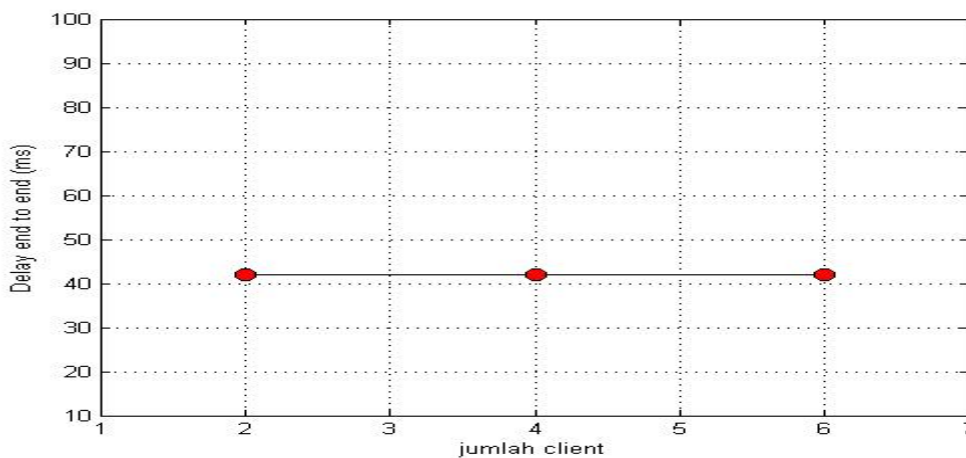
$$\begin{aligned}t_w &= \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} \\ &= \frac{91,74}{111285,44 (111285,44 - 91,74)} + \frac{1}{111285,44} \\ &= 0.00000000741 + 0.00000898 \\ &= 0.00000898 \text{ s} \\ &= 0.00898 \text{ ms}\end{aligned}$$

- **Delay end to end**

Berdasarkan hasil perhitungan di atas, maka nilai dari *delay end to end* dapat dihitung dengan menggunakan persamaan (3).

$$\begin{aligned}t_{end\ to\ end} &= t_{codec} + t_{trans} + t_p + t_w \\ &= 42 \text{ ms} + 0,00877 \text{ ms} + 0,00119 \text{ ms} + 0.00898 \text{ ms} \\ &= 42.019 \text{ ms}\end{aligned}$$

Gambar 5.4 menunjukkan grafik besarnya *delay end to end* pada setiap pengujian yang dilakukan.



Gambar 5.4 : *Delay end to end* terhadap jumlah *client*
Sumber : Hasil Pengujian

Berdasarkan gambar 5.4, terlihat bahwa besarnya *delay end to end* pada setiap pengujian mendekati sama. Hal ini disebabkan karena semua komponen atau parameter yang digunakan untuk menghitung nilai dari *delay end to end* pada setiap pengujian adalah sama. Komponen atau parameter tersebut adalah *codec* yang digunakan (G.711), panjang dari kabel *UTP cat.5*, jumlah *NIC* yang digunakan, Jumlah *HUB* yang dilalui, kapasitas media transmisi.

Besarnya *delay end to end* pada jaringan *VoIP* yang dirancang pada penelitian ini memenuhi standar *delay end to end* yang diperbolehkan oleh ITU. 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.

5.3.4 Kapasitas Jaringan

Kapasitas jaringan atau 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 (12).

$$K_T = \frac{BW_{st}}{BW_t}$$

Dimana pada perancangan jaringan didapatkan nilai :

- Kapasitas media transmisi (C) adalah 100 Mbps = 104857600 bps
- Bit rate codec* G.711 adalah 64 kbps
- Payload codec* G.711 adalah 160 byte

Banyaknya paket suara per detik (pps) dapat dihitung dengan menggunakan persamaan (14).

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

Bandwidth transmisi (BW_T) dapat dihitung dengan menggunakan persamaan (13)

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

Diketahui :

- Panjang paket rata-rata komunikasi pada pengujian pertama adalah 111 byte.

- b. Panjang paket rata-rata komunikasi pada pengujian kedua adalah 115 byte.
- c. Panjang paket rata-rata komunikasi pada pengujian ketiga adalah 115 byte.

$$\text{Panjang paket internet rata-rata } (L_{PI}) = \frac{111+115+115}{3}$$

$$= 113,67 \text{ byte}$$

Dengan mensubtitusikan panjang paket rata-rata (L_{PI}) dengan paket suara per detik (pps) ke dalam persamaan (13), maka

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

$$= (113,67 \times 8) \times 50$$

$$= 45466,67 \text{ bps}$$

Kapasitas transmisi dapat dihitung dengan menggunakan persamaan (12).

$$K_T = \frac{BW_{st}}{BW_t}$$

$$= \frac{104857600}{45466,67}$$

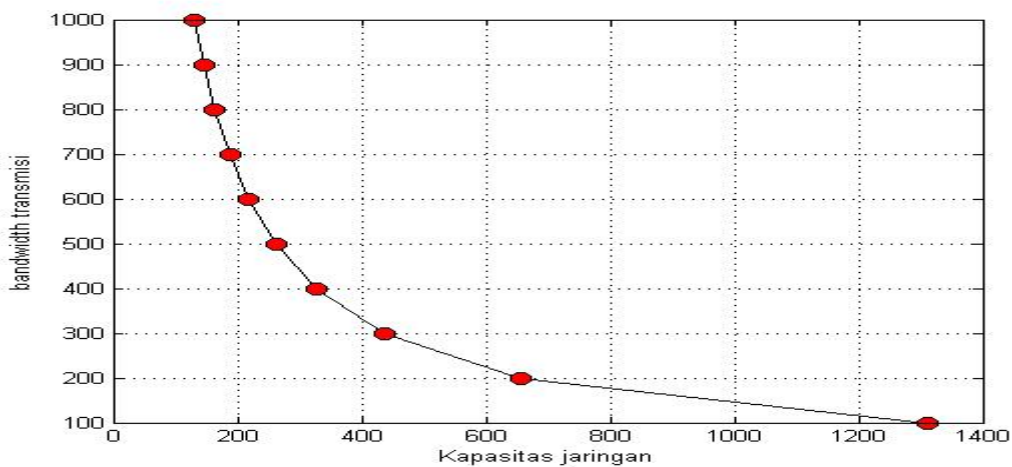
$$= 2306,25$$

Karena *VoIP* merupakan sistem komunikasi *duplex*, maka kapasitas transmisi jaringan adalah :

$$K_T = \frac{2306}{2}$$

$$= 1153 \text{ komunikasi}$$

Jaringan *VoIP* yang dirancang pada penelitian ini dapat melayani 1153 komunikasi dalam waktu yang bersamaan. Gambar 5.5 menunjukkan grafik perbandingan antara kapasitas jaringan dan *Bandwidth* Transmisinya.



Gambar 5.5 : grafik kapasitas jaringan terhadap *Bandwidth* Transmisi
 Sumber : Hasil Pengujian

Pada Gambar 5.5 menunjukkan bahwa terjadi perubahan yang linear antara *bandwidth* dengan kapasitas jaringan. Semakin kecil *bandwidth* transmisi maka semakin besar kapasitas jaringan yang bisa digunakan, sehingga semakin banyak komunikasi yang bisa digunakan dalam waktu yang bersamaan dan sebaliknya semakin besar *bandwidth* transmisinya maka semakin kecil kapasitas jaringan yang bisa digunakan, sehingga semakin sedikit komunikasi yang bisa digunakan dalam waktu yang bersamaan.

5.4 Mean Opinion Score

Salah satu cara menilai kualitas suara atau video itu baik atau buruk yaitu dengan metode *Mean Opinion Score (MOS)*. *MOS* merupakan standart dari *International Telecommunication Union - Telecommunication (ITU-T) P.800* berupa angka yang didapatkan berdasarkan *survey*. *Survey* tersebut dilakukan oleh jumlah pria dan wanita yang seimbang untuk menilai kualitas dari suara kita sebagai hasil dari pentransmisiannya. Parameter dari setiap nilai adalah sebagai berikut :

- 5 - Sangat baik, hampir menyerupai kualitas suara percakapan.
- 4 - baik, suara terdengar jelas seperti saat melakukan komunikasi dengan telepon.
- 3 - suara dapat didengar tapi diperlukan usaha yang lebih untuk memahaminya.
- 2 - buruk, suara hampir tidak bisa dilakukan berkomunikasi
- 1 - sangat buruk, sangat tidak mungkin dilakukan komunikasi

Tabel 5.4 menunjukkan hasil pengujian *MOS* pada sistem *VoIP* yang dirancang pada penelitian ini.

Tabel 5.3 Hasil Pengujian MOS

Nama	Nilai	
	Dengan Headset	Tanpa Headset
Ryandicka Anugerah	4	3
Yudi Yurianto	4	3
Reza Fadillah	4	2
Nuke Yunika	4	3
Novita Ika D	4	2
Herrah A.M.	4	3
	MOS = 4	MOS = 2.67

Sumber : Hasil Pengujian

Berdasarkan Tabel 5.4, dapat disimpulkan bahwa kualitas suara dengan menggunakan *headset* lebih baik daripada tanpa menggunakan *headset*. Hal ini disebabkan karena kualitas *microphone* pada *headset* lebih baik daripada kualitas *microphone* pada laptop, sehingga berpengaruh terhadap perbedaan kualitas suaranya. Nilai *MOS* dengan menggunakan *headset* adalah 4, yang berarti suara terdengar jelas seperti saat melakukan komunikasi dengan menggunakan telepon. Sedangkan nilai *MOS* tanpa menggunakan *headset* adalah 2.67, yang berarti suara dapat didengar tapi diperlukan usaha yang lebih untuk memahaminya.



BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan analisis dari perancangan jaringan *VoIP* berbasis *SIP* dengan menggunakan *IPv6*, dapat ditarik kesimpulan sebagai berikut :

1. Untuk merancang jaringan *VoIP* berbasis *SIP* dengan menggunakan alamat *IPv6*, langkah-langkah yang harus dilakukan adalah :
 - Menentukan topologi jaringan yang akan digunakan. Topologi yang digunakan pada penelitian ini adalah topologi *star* yang terdiri dari satu *server* dan enam *client*. Untuk menghubungkan *server* dan *client* digunakan *switch* yang berisi 8 *port* dan kabel *UTP Category 5* yang dilengkapi konektor *RJ-45* dan dipasang dengan konfigurasi *straight*.
 - Merancang *server* yang mampu menggunakan protokol *SIP* dan protokol *IPv6*. *Server VoIP* yang digunakan pada penelitian ini adalah *Opensips 1.6.0* yang diimplementasikan pada laptop dengan sistem operasi *Linux Ubuntu* versi 10.
 - Merancang *client* yang mampu menggunakan protokol *SIP* dan protokol *IPv6*. *client* pada jaringan *VoIP* ini menggunakan *software SIP-communicator*.
2. Analisis performansi jaringan *VoIP* berbasis *SIP* dengan menggunakan *IPv6* meliputi *throughput*, *delay end to end*, *packet loss*, dan kapasitas jaringan adalah sebagai berikut :
 - *Throughput*, nilai dari *throughput* pada masing-masing pengujian adalah sama, dan dapat ditarik kesimpulan bahwa pada analisis pengujian jaringan ini, jumlah *client* tidak terlalu berpengaruh terhadap nilai dari *throughput*, hal ini disebabkan karena total waktu transmisi dan jumlah paket data yang diterima pada setiap pengujian hampir sama.
 - *Delay end to end*, besarnya *delay end to end* pada setiap pengujian mendekati sama yaitu 42.019 ms. Hal ini disebabkan karena semua komponen atau parameter yang digunakan untuk menghitung nilai dari *delay end to end* pada setiap pengujian adalah sama. Komponen atau parameter tersebut adalah *codec* yang digunakan (*G.711*), panjang dari kabel *UTP*

cat.5, jumlah *NIC* yang digunakan, Jumlah *HUB* yang dilalui, kapasitas media transmisi. Besarnya *delay end to end* pada jaringan *VoIP* yang dirancang pada penelitian ini memenuhi standar *delay end to end* yang diperbolehkan oleh ITU. 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.

- *Packet loss*, prosentase *packet loss* pada setiap pengujian adalah sama, yaitu sebesar 0%. Banyaknya jumlah *client* yang melakukan proses komunikasi tidak berpengaruh terhadap prosentase *packet loss* pada sistem jaringan *VoIP* yang telah dirancang. Hal ini terjadi karena pada setiap pengujian tidak ada data yang hilang ($N_{\text{lost}} = 0$ paket). Prosentase dari *packet loss* pada pengujian jaringan *VoIP* pada penelitian ini tidak melebihi batas maksimum yang diperbolehkan ISO yaitu sebesar 5%.
 - Kapasitas jaringan, besarnya kapasitas jaringan *VoIP* yang dirancang pada penelitian ini adalah dapat melayani 1153 komunikasi dalam waktu yang bersamaan.
3. Kualitas suara pada jaringan *VoIP* yang dirancang pada penelitian ini berdasarkan *MOS* adalah sebagai berikut: Nilai *MOS* dengan menggunakan *headset* adalah 4, yang berarti suara terdengar jelas seperti saat melakukan komunikasi dengan menggunakan telepon. Sedangkan nilai *MOS* tanpa menggunakan *headset* adalah 2.67, yang berarti suara dapat didengar tapi diperlukan usaha yang lebih untuk memahaminya.

6.2 Saran

Beberapa hal yang disarankan untuk pengembangan lebih lanjut terhadap sistem yang dirancang pada penelitian ini antara lain :

1. Melakukan pengujian dengan jumlah *client* yang lebih banyak, agar karakteristik performansi dari jaringan *VoIP* yang dirancang lebih terukur.
2. Jaringan *VoIP* diaplikasikan ke jaringan yang lebih besar, yaitu *MAN* (*Metropolitan Area Network*) atau *WAN* (*Wide Area Network*).