Bab II

TINJAUAN PUSTAKA

2.1. Konsep Jaringan Komputer

Jaringan Komputer adalah sekelompok komputer otonom yang saling berhubungan antara satu dengan lainnya dengan menggunakan protokol komunikasi untuk komputer, sehingga dapat saling berbagi informasi, program-program, penggunaan bersama perangkat keras seperti printer, hardisk, dan lain sebagainya. Selain itu jaringan komputer bisa diartikan sebagai kumpulan sejumlah terminal komputer yang berada diberbagai lokasi yang terdiri dari lebih dari satu komputer yang saling berhubungan.

Pembangunan jaringan komputer dapat mencegah ketergantungan pada komputer pusat. Setiap proses data tidak harus dilakukan pada satu komputer saja, melainkan dapat didistribusikan ketempat lainnya. Sehingga dapat terbentuk data yang terintegrasi dengan demikian memudahkan pemakai untuk memperoleh dan mengolah informasi setiap saat.

Dengan adanya jaringan komputer ini, maka pengembangan peralatan dapat dilakukan dengan mudah dan menghemat biaya. Jaringan komputer dapat memudahkan pemakai dalam merawat hardisk dan peralatan lainnya, misalnya untuk memberikan perlindungan terhadap serangan virus maka pemakai cukup memusatkan perhatian pada hardisk yang ada di komputer pusat.

Sistem jaringan komputer memberikan perlindungan terhadap data. Jaminan keamanan data tersebut diberikan melalui pengaturan hak akses para

pemakai dan *password,* serta teknik perlindungan terhadap hardisk sehingga data mendapatkan perlindungan yang efektif.

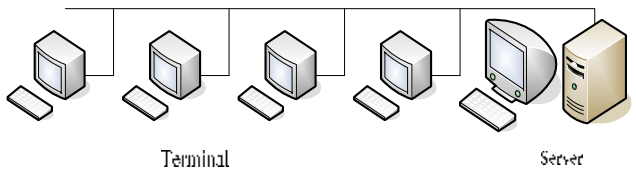
Dengan adanya pemakaian sumber daya secara bersama-sama, maka pemakai bisa mendapatkan hasil dengan maksimal dan kualitas yang tinggi. Selain itu data atau informasi yang diakses selalu terbaru, karena setiap ada perubahan yang terjadi dapat segera lengsung diketahui oleh setiap pemakai.

2.2. Topologi Jaringan Komputer

Topologi jaringan komputer adalah pola hubungan antar terminal dalam suatu jaringan komputer. Ada beberapa macam topologi yang dapat digunakan, tetapi bentuk topologi yang utama adalah Bus, Star, dan Ring.

2.2.1. Topologi Bus

Pada topologi Bus semua terminal terhubung ke jalur komunikasi. Informasi yang dikirim akan melewati semua terminal pada jalur tersebut. Jika alamat yang tercantum dalam data atau informasi yang dikirim sesuai dengan alamat terminal yang dilewati, maka data atau informasi tersebut akan diterima dan diproses. Jika alamat tersebut tidak sesuai, maka informasi tersebut akan diabaikan oleh terminal yang dilewati.

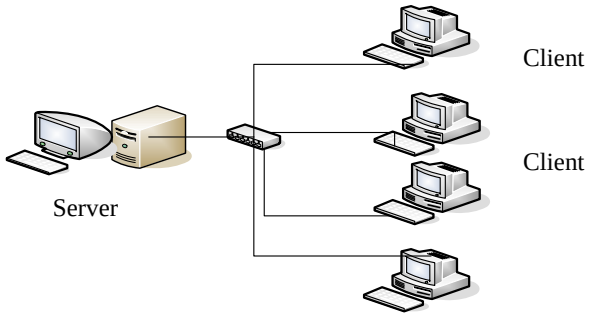


Gambar II.1 Topologi bus

Jumlah terminal dapat ditambahkan dan dikurangi secara fleksibel. Namun demikian, jumlah terminal hendaknya perlu dibatasi, karena pada topologi model ini, jika terminal yang terhubung sangat banyak, maka kinerja jaringan akan turun drastis. Kekurangan yang lain dari topologi ini, bila ada terminal yang mati, maka operasional jaringan akan terganggu.

2.2.2. Topologi *Star*

Dalam topologi *star*, sebuah terminal pusat bertindak sebagai pengatur dan pengendali semua komunikasi data yang terjadi. Terminal-terminal lain terhubung padanya dan pengiriman data dari satu terminal ke terminal lainnya melalui terminal pusat. Terminal pusat akan menyediakan jalur komunikasi khusus untuk dua terminal yang akan berkomunikasi. Sebagai salah satu contoh penggunaan topologi *Star* adalah jaringan telepon.



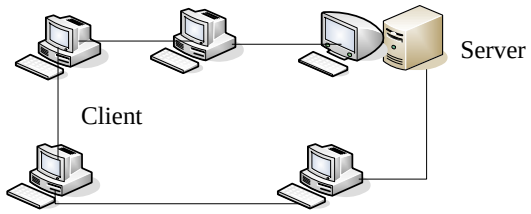
Gambar II.2 Topologi star

Topologi ini mudah untuk dikembangkan, baik untuk penambahan maupun pengurangan terminal. Banyak terminal yang dapat terhubung tergantung pada jumlah port yang tersedia pada hub yang digunakan. Pada topologi Star ini,

hub yang digunakan akan menjadi titik kritis, sehingga perlu adanya perhatian dan pemeliharaan terhadap hub tersebut.

2.2.3. Topologi Ring

Jaringan komputer lokal dengan topologi ini mirip dengan topologi Bus, tetapi kedua terminal yang berada diujung saling dihubungkan sehingga menyerupai lingkaran. Setiap informasi yang diperoleh diperiksa alamatnya oleh terminal yang dilewatinya. Jika bukan untuknya, informasi dilewatkan sampai menemukan alamat yang benar. Setiap terminal dalam jaringan komputer lokal saling tergantung sehingga jika terjadi kerusakan pada satu terminal maka seluruh jaringan akan terganggu.



Gambar II.3 Topologi ring

2.3. Elemen Dasar Jaringan

Jaringan komputer tersusun dari beberapa elemen dasar yang meliputi komponen *hardware* dan *software*, yaitu :

2.3.1. Komponen H*ardware*

a. *Personal computer* (PC) :

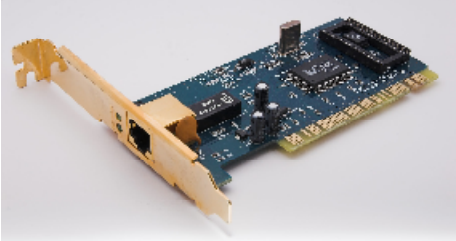


Gambar II.4 PC desktop

Tipe personal komputer yang digunakan di dalam jaringan akan sangat menentukan unjuk kerja dari jaringan tersebut. Komputer dengan unjuk kerja tinggi akan mampu mengirim dan mengakses data dalam jaringan dengan cepat. Di dalam jaringan tipe *Client-Server*, komputer yang difungsikan sebagai *server* mutlak harus memiliki unjuk kerja yang lebih tinggi dibandingkan komputer-komputer lain sebagai workstation-nya, karena *server* akan bertugas menyediakan fasilitas dan mengelola operasional jaringan tersebut.

b. *Network Interface Card* (NIC) :

Berdasarkan tipe bus, ada beberapa tipe *Network Interface Card* (NIC) atau *Network Card*, yaitu ISA dan PCI. Saat inin terdapat jenis *network card* yang banyak digunakan, yaitu PCI.



Gambar II.5 ***Network interface card***

c. *Ethernet* :

Dalam jaringan dengan *protocol* akses CSMA/CD atau *Carrier Sense Multiple Access with Collision Detection*, suatu node (A) yang akan mengirimkan data akan memeriksa dahulu kondisi jalur data. Bila tidak terdapat aliran data/kosong maka node tersebut akan mengirimkan datanya dan bila node lain (B) yang sedang menggunakan jalur data maka node (A) akan menunggu dan akan mencoba memeriksa kembali. Dalam protokol akses ini dimungkinkan pada suatu saat terjadi beberapa node mengirimkan datanya secara bersamaan sehingga mengakibatkan collision atau tabrakan. Dalam kondisi demikian node-node tersebut akan batal mengirimkan data dan akan mencobanya kembali bila jalur tidak sibuk.

d. *Network Adapter Card* :

Setiap *network card* akan memiliki driver atau program yang berfungsi untuk mengaktifkan dan mengkonfigurasi *network adapter* tersebut disesuaikan dengan lingkungan dimana *network card* tersebut dipasang agar dapat digunakan untuk melakukan komunikasi data.

2.3.2. Komponen *Software*

a. Sistem Operasi Jaringan

Untuk mengelola suatu jaringan diperlukan adanya sistem operasi jaringan. Sistem operasi jaringan dibedakan menjadi dua berdasarkan tipe jaringannya, yaitu sistem operasi client-*server* dan operasi jaringan peer to peer.

b. *Network Adapter* Driver

Agar perangkat keras *network interface card* (NIC) dapat berkomunikasi dengan sistem operasi, dibutuhkan *network adapter driver* sebagai jembatan komunikasi antara kedua poin berikut.

c. Protokol Jaringan

Suatu standarisasi yang mengatur bagaimana caranya agar antar komponen diatas dapat berkomunikasi.

2.3.3. Sharing *Resource*

Sharing resource bertujuan agar seluruh program, peralatan atau *peripheral* lainnya dapat dimanfaatkan oleh setiap orang yang ada pada jaringan komputer tanpa terpengaruh oleh lokasi maupun pengaruh dari pemakai. Dengan kata lain, seorang pemakai yang letaknya yang sangat jauh sekalipun dapat memanfaatkan data maupun informasi yang lainnya tanpa mengalami kesulitan. Jadi dengan adanya *sharing resources* ini dapat menekan biaya pembelian *peripheral* atau *software* karena adanya peningkatan sumber daya tersebut.

2.3.4. Media Komunikasi

Jaringan komputer memungkinkan terjadinya komunikasi antar pengguna, baik untuk suara maupun untuk mengirim pesan atau informasi yang penting lainnya. Dengan menggunakan jaringan komputer, dua orang atau lebih yang jaraknya sangat jauh akan lebih mudah bekerja sama.

2.3.5. Integrasi Data

Pembangunan jaringan komputer dapat mencegah ketergantungan pada komputer pusat. Setiap proses data tidak harus dilakukan pada satu komputer saja, melainkan dapat didistribusikan ketempat lainnya. Oleh sebab inilah maka dapat terbentuk data yang terintegrasi sehingga dengan demikian memudahkan pemakai untuk memperoleh dan mengolah informasi setiap saat.

2.3.6. Pengembangan dan Pemeliharaan

Dengan adanya jaringan komputer ini, maka pengembangan peralatan dapat dilakukan dengan mudah dan menghemat biaya. Jaringan komputer dapat memudahkan pemakai dalam merawat *hardisk* dan peralatan lainnya, misalnya untuk memberikan perlindungan terhadap serangan virus maka pemakai cukup memusatkan perhatian pada hardisk yang ada di komputer *server*/pusat.

2.3.7. Keamanan Data

Sistem jaringan komputer memberikan perlindungan terhadap data. Jaminan keamanan data tersebut diberikan melelui pengaturan hak akses para pemakai dan *password*, serta teknik perlindungan terhadap *hardisk* sehingga data mendapatkan perlindungan yang efektif.

2.3.8. Sumber Daya Lebih Efisien dan Informasi Terkini

Dengan adanya pemakaian sumber daya secara bersama-sama, maka pemakai bisa mendapatkan hasil dengan maksimal dan kualitas yang tinggi. Selain itu data atau informasi yang diakses selalu terbaru, karena setiap ada perubahan yang terjadi dapat segera lengsung diketahui oleh setiap pemakai.

2.4. Paket Data

Data link layer merupakan lapisan kedua dari *standard* OSI. Tugas utama *data link layer* adalah sebagai fasilitas transmisi *raw data* dan mentransformasi data tersebut ke saluran yang bebas dari kesalahan transmisi. Sebelum diteruskan ke *network layer*, *data link layer* melaksanakan tugas ini dengan memungkinkan pengirim memecah-mecah data input menjadi sejumlah data *frame* (biasanya berjumlah ratusan atau ribuan *byte*). Kemudian data link *layer* mentransmisikan *frame* tersebut secara berurutan dan memproses *acknowledgement frame* yang dikirim kembali oleh penerima. Karena *pysical layer* menerima dan mengirim aliran bit tanpa mengindahkan arti atau arsitektur *frame*, maka tergantung pada data *link layer*-lah untuk membuat dan mengenali batas-batas *frame* itu. Hal ini bisa dilakukan dengan cara membubuhkan bit khusus ke awal dan akhir *frame*.

Pada saat data akan ditransmisikan, maka data akan dibagi menjadi paket yang kecil-kecil. Hal ini dilakukan karena:

a. Jaringan tertentu hanya dapat menerima paket dengan panjang tertentu. (misal ARPANET hanya mampu mengirim paket sebanyak 8063).
b. Jenis *flow control* tertentu akan efisien jika berita yang dikirim dibagi dalam paket-paket yang kecil (misal pada *select repeat* ARQ bila terjadi

kerusakan data pada saat transmisi, maka *transmitter* hanya perlu mengirim kembali paket data tersebut).

c. Agar pengirim jaringan tidak didominasi oleh *user* tertentu, karena dengan paket data maka setiap *user* dapat dibatasi jumlah paket data yang akan dikirim sehingga dapat bergantian dengan *user* lainnya dalam memanfaatkan jaringan tersebut.

d. Paket data yang kecil hanya memerlukan *buffer* yang kecil pada bagian *receiver.*

Akan tetapi dalam melakukan pemotongan data menjadi paket data, harus memperhatikan bahwa data tidak boleh dipotong terlalu kecil, karena:

a. Setiap data memerlukan bit *overhead* (tiap paket harus disertai dengan SYN, ADDRESS, CONTROL FIELD, CRC, FLAG, dan lain-lain). Pengiriman paket akan effisien jika bagian data lebih besar dari bit *overhead.*

b. Bila paket data terlalu kecil maka waktu pemrosesan yang diperlukan untuk pengiriman paket lebih besar dari waktu untuk pemrosesan sebuah paket yang besar.

2.5. Konsep Dasar *Internet Protocol*

Pada dasarnya, komunikasi data merupakan proses mengirimkan data dari satu komputer ke komputer yang lain. Untuk dapat mengirimkan data, pada komputer harus ditambahkan alat khusus, yang dikenal dengan *network interface* (*interface* jaringan). Jenis *interface* jaringan ini bermacam-macam, tergantung pada media fisik yang digunakan untuk men*transfer* data tersebut.

Dalam proses pengiriman data ini terdapat beberapa masalah yang harus dipecahkan. Pertama, data harus dikirim ke komputer yang tepat, sesuai tujuannya. Hal ini akan menjadi rumit jika komputer tujuan *transfer* data ini tidak berada pada jaringan lokal, melainkan ditempat yang jauh. Jika lokasi komputer yang saling berkomunikasi ”jauh” (secara jaringan) maka terdapat kemungkinan data rusak atau hilang karenanya perlu ada mekanisme yang mencegah rusaknya data ini. Hal ini yang perlu diperhatikan ialah pada komputer tujuan *transfer* data mungkin terdapat lebih dari satu aplikasi yang menunggu datangnya data. Data yang dikirim harus sampai ke aplikasi yang tepat, pada komputer yang tepat, tanpa kesalahan.

Cara alamiah untuk menghadapi setiap masalah yang rumit ialah memecahkan masalah tersebut menjadi bagian yang lebih kecil. Dalam memecahkan masalah *transfer* data di atas, para ahli jaringan komputerpun melakukan hal yang sama. Untuk setiap problem komunikasi data, diciptakan solusi khusus berupa aturan-aturan yang menangani problem tersebut. Untuk menangani suatu masalah komunikasi data, keseluruhan aturan ini harus kerja sama satu dengan lainnya. Sekumpulan aturan untuk mengatur proses pengiriman data ini disebut sebagai *protokol komunikasi data*. Protokol ini di implementasikan dalam bentuk program komputer (*software*) yang terdapat pada komputer dan peralatan komunikasi data lainnya.

Dalam dunia komunikasi data komputer, protokol mengatur bagaimana sebuah komputer berkomunikasi dengan komputer lain. Dalam jaringan komputer kita dapat menggunkan banyak macam protokol tapi agar sebuah komputer dapat

berkomunikasi, keduanya harus mengunakan protokol yang sama. Protokol berfungsi mirip dengan bahasa. Agar dapat berkomunikasi seseorang perlu berbicara dan mengerti bahasa yang mereka gunakan.

2.5.1. *Transmision Control Protocol* (TCP)

TCP/IP (*Transmission Control Protocol / Internet Protocol*) adalah sekumpulan protokol yang mengatur komuniksai data komputer dengan komputer yang lain maupun komputer dengan jaringan *Internet*. Karena menggunakan bahasa yang sama, yaitu protokol TCP/IP, perbedaan jenis komputer dan sistem operasi tidak menjadi masalah.

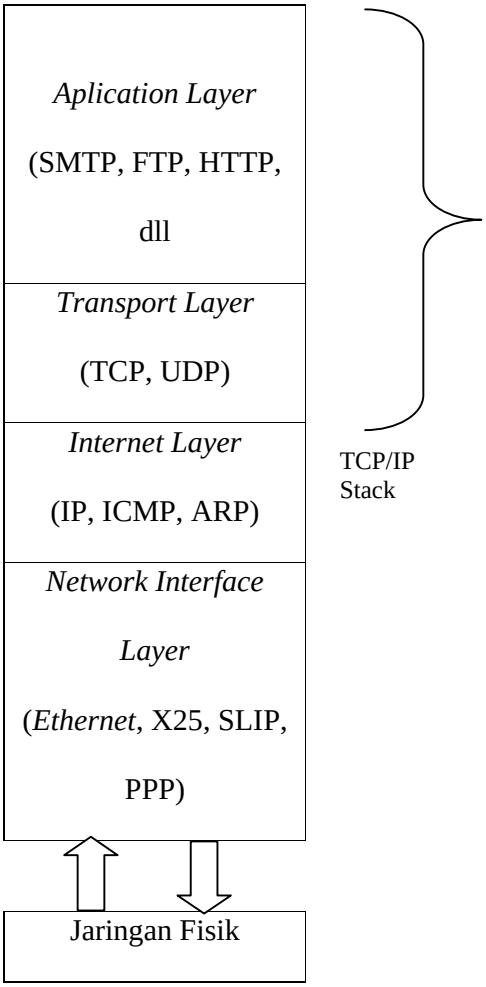
Komputer PC dengan menggunakan sistem operasi *Windows* dapat berkomunikasi dengan komputer *Macintos* atau dengan *Sun SPARC* yang menjalankan Solaris. Jadi, jika sebuah komputer menggunakan protokol TCP/IP dan terhubung langsung ke *Internet*, maka komputer tersebut dapat berhubungan dengan komputer di belahan dunia manapun yang juga terhubung ke *Internet*.

Perkembangan TCP/IP yang diterima luas dan praktis menjadi standar defacto jaringan komputer berkaitan dengan ciri-ciri yang terdapat pada protokol itu sendiri :

a. Protokol TCP/IP dikembangkan menggunakan standart protokol yang terbuka.
b. Standart protokol TCP/IP dalam bentuk *Request For Comment* (RFC) dapat diambil oleh siapun tanpa biaya.
c. TCP/IP dikembangkan dengan tidak tergantung pada sistem operasi atau perangkat keras tertentu.

d. Pengembangan TCP/IP dilakukan dengan konsensus dan tidak tergantung pada vendor tertentu.
e. TCP/IP independen terhadap perangkat keras jaringan dan dapat dijalankan pada jaringan *Ethernet*, *Token Ring*, Jalur telepon dial-up, jaringan X25, dan praktis jenis media transmisi apa pun..
f. Pengalamatan TCP/IP bersifat untuk dalam skala global. Dengan cara ini, komputer dapat saling terhubung walaupun jaringannya seluas *internet* sekarang ini.

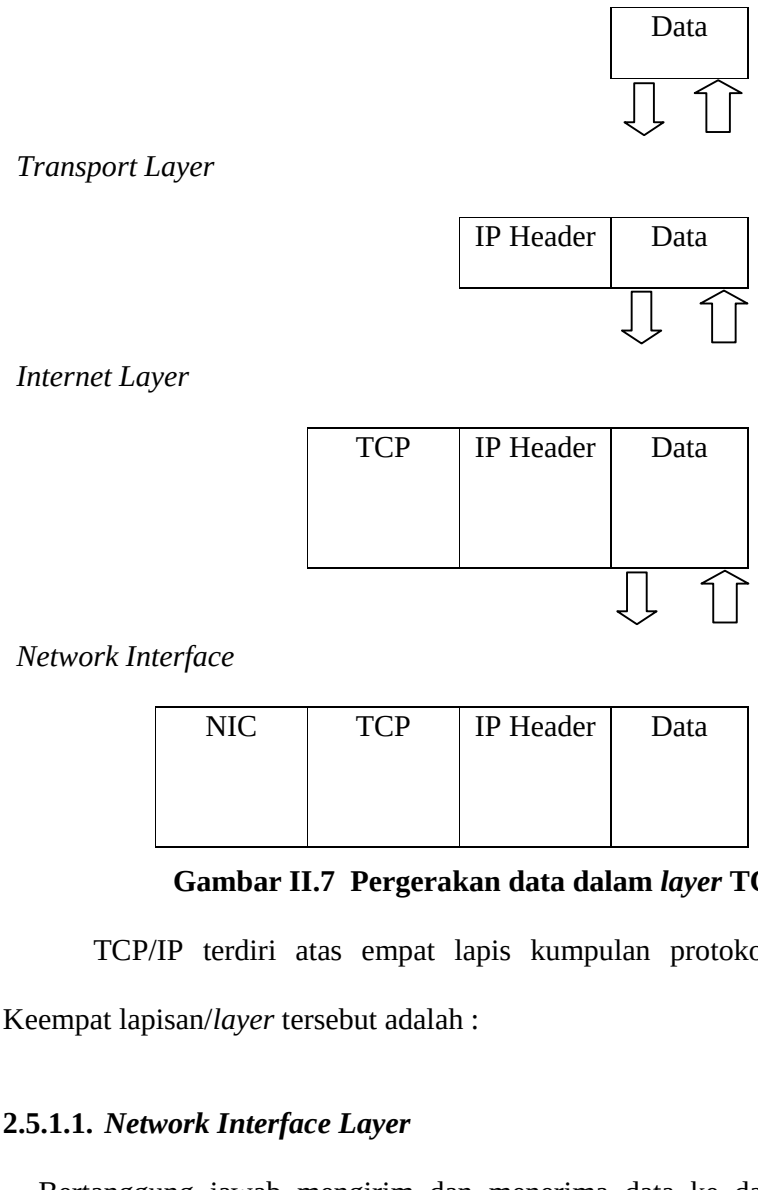
TCP/IP bukanlah sebuah protokol tunggal, tetapi satu kesatuan protokol. TCP/IP merupakan sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data pada *Wide Area Network* (WAN). TCP/IP terdiri dari protokol-protokol yang bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Berkat prinsip ini, tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui cara kerja protokol yang lain, sepanjang ia masih bisa mengirim dan menerima data. Berkat penggunaan prinsip ini, TCP/IP menjadi protokol komunikasi data yang fleksibel. Protokol TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan *interface* jaringan, karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Sekumpulan protokol TCP/IP ini dimodelkan dengan empat *layer* TCP/IP, sebagai mana terlihat pada gambar di bawah ini.



Gambar II.6 Model protokol TCP/IP

Jika suatu protokol menerima data dari protokol lain di *layer* atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut, informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu, data ini diteruskan lagi ke protokol pada *layer* dibawahnya. Hal ini yang sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada *layer* dibawahnya. Jika data ini dianggap *valid*, protokol akan melepas informasi tambahan tersebut, untuk kemudian meneruskan data itu ke protokol lain yang

berada pada *layer* di atasnya. Pergerakan data dalam *layer* TCP/IP seperti gambar dibawah ini:



Gambar II.7 Pergerakan data dalam *layer* TCP/IP

TCP/IP terdiri atas empat lapis kumpulan protokol yang bertingkat. Keempat lapisan/*layer* tersebut adalah :

2.5.1.1. *Network Interface Layer*

Bertanggung jawab mengirim dan menerima data ke dan dari media fisik (berupa kabel, serat optik atau gelombang radio), maka dari itu harus mampu menerjemahkan sinyal listrik menjadi digital yang dimengerti komputer.

2.5.1.2. *Internet Layer*

Bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada *layer* ini terdapat tiga macam protokol yaitu:

a. IP (*Internet Protocol*) berfungsi untuk menyampaikan paket data ke alamat yang tepat, IP memiliki sifat sebagai *unrealieble* (ketidakandalan: tidak menjamin *datagram* yang terkirim sampai tujuan), *connectionless*, *datagram delivery service*.
b. ARP (*Address Resolution Protocol*) Protokol yang digunakan untuk nenemukan alamat *Hardware* dari *host*/komputer yang terletak pada *network* yang sama.
c. ICMP (*Internet Control message protokol*) Protokol yang digunakan untuk mengirimkan pesan dan melapor kegagalan pengiriman data.

2.5.1.3. *Transport Layer*

Bertanggung jawab untuk mengadakan komunikasi antara dua host/komputer, yang terdiri dari protokol TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*). TCP merupakan protokol yang menyediakan service *connection oriented*, sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukann hubungan terlebih dahulu.

2.5.1.4. *Aplication Layer*

Terletak semua aplikasi yang menggunakan protokol TCP/IP.Fungsi masing-masing *layer* / lapisan protokol serta aliran data pada *layer* TCP / IP diatas,

dapat dicontohkan dengan mengggunakan analogi yang sangat sederhana. Seperti analogi pengiriman surat, seperti berikut ini:

a. Pertama, kita harus menulis dahulu isi surat tersebut. Maka kita harus mengambil selembar kertas dengan ballpoint untuk menulis berita tersebut.
b. Setelah langkah ini terselesaikan, maka kita harus mengambil amplop surat agar terlindung dari kerusakan.
c. Maka kita harus memilih amplop yang tertutup (TCP) atau amplop yang terbuka (UDP).
d. Barulah kita menulis alamat yang dituju dengan jelas, serta nama pengirim dan alamat pengirim.
e. Maka selesailah sudah pengiriman surat tersebut dengan menitipkan surat itu pada kantor pos.

Cara kerja TCP/IP dalam satu komputer adalah sangat mirip dengan cerita diatas. Mengirimkan e-mail dll, terlebih dahulu diolah di TCP. Saat diolah TCP memberi amplop untuk melindungi data-data yang hendak dikirim, yang berupa data tambahan (no.urut), 16 *bit source port* number (nama pengirim dan penerima).

Komputer dengan protokol TCP/IP dapat berhubungan dengan komputer lain dan jaringan lain karena bantuan peralatan jaringan komputer. Peralatan ini biasanya disebut *network interface*. Selain peralatan tersebut masih diperlukan peralatan lain yang disebut dengan *device* penghubung jaringan, yang secara umum dibagi menjadi beberapa kategori :

a. *Repeater*

Fungsinya adalah menerima sinyal dari satu segment kabel LAN dan memancarkan kembali dengan kekuatan yang sama dengan sinyal aslinya. Pada *segment* (satu atau lebih) kabel LAN lain, dengan adanya *repeater* ini jarak antara dua jaringan komputer bisa diperjauh.

b. *Bridge*

Bridge lebih cerdas dan fleksibel dibandingkan dengan *repeater*. *Bridge* bekerja dengan meneruskan paket *ethernet* dari satu jaringan ke jaringan yang lain. Beberapa *bridge* mempelajari alamat *ethernet* setiap device yang terhubung dengannya dan mengatur alur *frame* berdasarkan alamat tersebut. Dapat dihubungkan dalam jaringan dengan metode transmisi yang berbeda. LAN *ethernet* dengan LAN token ring dan mampu memisahkan sebagian trafik karena adanya pemfilteran *frame*.

c. *Router*

Melewatkan paket IP dari satu jaringan ke jaringan lain yang mungkin memiliki banyak jalur diantara keduanya.

2.5.2 *User Datagram Protocol* (UDP)

UDP yang merupakan salah satu *protocol* utama diatas IP merupakan *transport protocol* yang lebih sederhana dibandingkan dengan TCP. UDP digunakan untuk situasi yang tidak mementingkan mekanisme reliabilitas. *Header* UDP hanya berisi empat *field* yaitu *source port*, *destination port*, *length* dan UDP *checksum* dimana fungsinya hampir sama dengan TCP, namun fasilitas *checksum* pada UDP bersifat opsional.

UDP pada *VoIP* digunakan untuk mengirimkan *audio stream* yang dikirimkan secara terus menerus. UDP digunakan pada *VoIP* karena pada pengiriman *audio streaming* yang berlangsung terus menerus lebih mementingkan kecepatan pengiriman data agar tiba di tujuan tanpa memperhatikan adanya paket yang hilang walaupun mencapai 50% dari jumlah paket yang dikirimkan.

Karena UDP mampu mengirimkan data *streaming* dengan cepat, maka dalam teknologi *VoIP* UDP merupakan salah satu *protokol* penting yang digunakan sebagai *header* pada pengiriman data selain RTP dan IP. Untuk mengurangi jumlah paket yang hilang saat pengiriman data (karena tidak terdapat mekanisme pengiriman ulang) maka pada teknolgi *VoIP* pengiriman data banyak dilakukan pada *private network*.

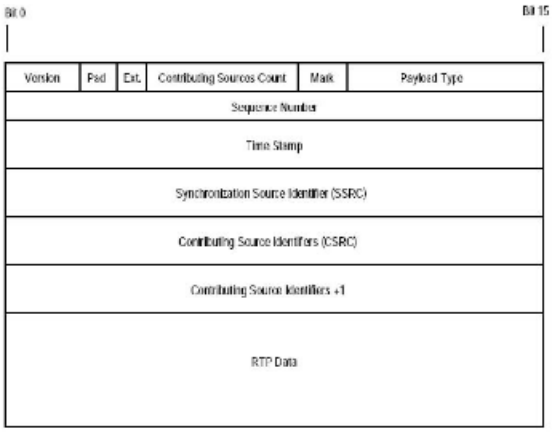
Dalam mendukung proses pengiriman data, UDP menggunakan beberapa protokol untuk sistem transmisinya yaitu :

2.5.2.1. RTP (*Real-time Transport Protocol*)

Protokol RTP menyediakan *transfer* media secara *real-time* pada jaringan paket. Protokol RTP mengunakan Protokol UDP dan *header* RTP mengandung informasi kode bit yang spesifik pada tiap paket yang dikirimkan; hal ini membantu penerima untuk melakukan antisipasi jika terjadi paket yang hilang.

RTP adalah *protocol* yang dibuat untuk mengkompensasi *jitte*r dan *desequencing* yang terjadi pada jaringan IP. RTP dapat digunakan untuk beberapa macam data stream yang *realtime* seperti data suara dan data video. RTP berisi informasi tipe data yang di kirim, *timestamps* yang digunakan untuk pengaturan waktu suara percakapan terdengar seperti sebagaimana diucapkan, dan *sequence*

numbers yang digunakan untuk pengurutan paket data dan mendeteksi adanya paket yang hilang.



Gambar II.8 Komponen RTP *header*[1]

RTP didesain untuk digunakan pada *tansport layer*, namun demikian RTP digunakan diatas UDP, bukan pada TCP karena TCP tidak dapat beradaptasi pada pengerimiman data yang *real-time* dengan keterlambatan yang relatif kecil seperti pada pengiriman data komunikasi suara.

Dengan menggunakan UDP yang dapat mengirimkan paket IP secara *multicast*, RTP *stream* yang di bentuk oleh satu terminal dapat dikirimkan ke beberapa terminal tujuan.

2.5.2.2. RTCP (*Real-time Control Transport Protocol*)

Protokol RTCP merupakan *protokol* yang mengendalikan *transfer* media. *Protokol* ini bekerja sama dengan *protokol* RTP (untuk memudahkan

[1] Stefan Brunner, Akhlaq A. Ali, "*Understanding VoIP Networks"*, Juniper *Network*s, Inc., http://www.Juniper.net

pemahaman). Dalam satu sesi komunikasi, *protokol* RTP mengirimkan paket RTCP secara periodik untuk memperoleh informasi *transfer* media dalam memperbaiki kualitas layanan.

Terdapat dua komponen penting pada paket RTCP, yang pertama adalah *sender report* yang berisikan informasi banyaknya data yang dikirimkan, pengecekan *timestamp* pada *header* RTP dan memastikan bahwa datanya tepat dengan *timestamp*-nya. Elemen yang kedua adalah *receiver report* yang dikirimkan oleh penerima panggilan. *Receiver report* berisi informasi mengenai jumlah paket yang hilang selama sesi percakapan, menampilkan *timestamp* terakhir dan *delay* sejak pengiriman *sender report* yang terakhir.

2.5.2.3. RSVP(*Resource Reservation Protocol*)

RSVP bekerja pada *layer transport*. Digunakan untuk menyediakan *bandwidth* agar data suara yang dikirimkan tidak mengalami *delay* ataupun kerusakan saat mencapai alamat tujuan *unicast* maupun *multicast*.

RSVP merupakan *signaling protocol* tambahan pada *VoIP* yang mempengaruhi QoS. RSVP bekerja dengan mengirimkan *request* pada setiap node dalam jaringan yang digunakan untuk pengiriman data *stream* dan pada setiap node RSVP membuat *resource reservation* untuk pengiriman data.

Resource reservation pada suatu node dilakukan dengan menjalankan dua modul yaitu *admission control* dan *policy control*.

Admission control digunakan untuk menentukan apakah suatu node tersebut memiliki *resource* yang cukup untuk memenuhi QoS yang dibutuhkan. *Policy control* digunakan untuk menentukan apakah *user* yang memiliki ijin

administratif (*administrative permission*) untuk melakukan reservasi. Bila terjadi kesalahan dalam aplikasi salah satu modul ini, akan terjadi RSVP error dimana *request* tidak akan dipenuhi. Bila kedua modul ini berjalan dengan baik, maka RSVP akan membentuk parameter *packet classifier* dan *packet scheduler*. *Packet Clasiffier* menentukan kelas QoS untuk setiap paket data yang digunakan untuk menentukan jalur yang digunakan untuk pengiriman paket data berdasarkan kelasnya dan *packet scheduler* berfungsi untuk menset antarmuka (*interface*) tiap node agar pengiriman paket sesuai dengan QoS yang dinginkan.

2.6. Konsep Dasar *Voice* Over *Internet Protocol* (*VoIP*)

Dalam perencanaan untuk mengimplementasikan *VoIP* pada *Local Area Network* (LAN), perlu memperhitungkan kebutuhan *bandwidth*, karena saat pengiriman video estimasi alokasi *bandwidth* menjadi sangat penting karena akan memakan sebagian besar *bandwidth* komunikasi yang ada. Sehingga teknik-teknik untuk melakukan kompresi data menjadi sangat strategis untuk memungkinkan penghematan *bandwidth* komunikasi.

Sebagai gambaran sebuah kanal suara (video) yang baik tanpa di kompresi akan mengambil *bandwidth* sekitar 9Mbps, sedangkan sebuah kanal suara (audio) yang baik dengan di kompresi akan mengambil *bandwidth* sekitar 64Kbps. Dari gambaran diatas dapat diasumsikan bahwa kebutuhan minimal *bandwidth* yang dibutuhkan untuk mengirimkan gambar dan suara adalah sebesar 9,064 Mbps, memang akan membutuhkan *bandwidth* yang sangat lebar. Namun dengan teknik kompresi yang ada, sebuah kanal suara dan gambar sebelum dilewatkan dalam jaringan TCP/IP akan terlebih dahulu melalui proses kompresi sehingga dapat

menghemat sebuah kanal video menjadi sekitar 30Kbps dan kanal suara menjadi 6Kbps (*half-duplex*), artinya sebuah saluran *Local Area Network* (LAN) yang memiliki *bandwidth* sebesar 10/100 Mbps sebetulnya dapat digunakan untuk menyalurkan video dan audio sekaligus. Tentunya untuk kebutuhkan konferensi yang *multiuser* akan dibutuhkan *multi bandwidth* pula, artinya minimal sekali kita harus menggunakan kanal 32-36Kbps dikalikan dengan berapa *user* konferensi dilakukan dalam jaringan.

Pertanyaan awal yang paling sering ditanyakan pada saat kita ingin mengimplemetasikan sistem *VoIP* adalah " apakah *VoIP* itu legal? " jawabannya adalah seperti apa yang tertuang dalam pasal 60 keputusan Menteri Perhubungan nomor KM 21 tahun 2001 tentang penyelengaraan jasa telekomunikasi yang berbunyi, " *penyelenggaraan jasa internet telephoni untuk keperluan publik sebagaimana dimaksud dalam pasal 46 ayat (1) huruf d merupakan penyelenggaraan internet telephoni yang bersifat komersial, dihubungkan ke jaringan telekomunikasi "*.

Sementara pasal 46 dari KM 21 tahun 2001 isinya adalah sebagai berikut :

(1) Penyelenggaraan jasa mul*time*dia sebagaimana dimaksud dalam Pasal 3 huruf c terdiri atas :

   a. Jasa televisi berbayar
   b. Jasa akses *internet*
   c. Jasa interkoneksi *internet*
   d. Jasa *internet* telephoni untuk keperluan publik

e. Jasa wireless *aplication protocol* (WAP)

f. Jasa portal

g. Jasa *small office home office* (SOHO)

h. Jasa transaksi *online*

i. Jasa aplikasi *packet switched* selain sebagaimana dimaksud dalam huruf a, b, c, d, e, f, g dan huruf h

(2) Penyelenggaraan jasa mul*time*dia selain sebagaimana dimaksud dalam ayat(1) ditetapkan oleh Direktur Jendral.

Jadi, secara sederhana sebetulnya yang diatur oleh pemerintah dan memerlukan ijin penyelenggraan dari menteri hanyalah :

a. Penyelenggara layanan telepon *internet* unutk keperluan publik

b. Bersifat komersial

c. Tersambung ke jaringan telekomunikasi (telkom)

Dari uraian diatas jika kita mengimplemetasikan sistem *VoIP* sebagai berikut :

a. Untuk keperluan pribadi

b. Tidak dikomersialkan

c. Tidak disambungkan ke jaringan telekomunikasi

Maka penggunaannya tidak melanggar keputusan menteri diatas sehingga sistem yang kita bangun mempunyai aspek legalitas.

2.6.1. Kompresi Suara

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

a. G.711 – Mengkonversi *voice* ke 64 kbps *voice stream*. *CODEC* ini digunakan pada traditional TDM T1 *voice*.

b. G.723.1 – Terdapat 2 type berbeda untuk compression G.723.1. Pertama menggunakan *Code-Excited Linear Prediction* (CELP) *compression algorithm* dan mempunyai bit rate 5.3 kbps. Type kedua menggunakan *Multi Pulse- Maximum Likelihood Quantization MP-MLQ algorithm* dan memiliki kualitas suara lebih bagus. Type ini mempunyai bit rate of 6.3 kbps.

c. G.726 – *CODEC* memiliki beberapa bit rate yang berbeda-beda, yaitu 40 kbps, 32 kbps, 24 kbps, dan 16 kbps. *CODEC* ini paling sesuai untuk interkoneksi ke PBX dengan bit rate 32 kbps.

d. G.728 – *CODEC* memiliki kualitas suara yang bagus dan spesifik di desain untuk *low latency applications*. *CODEC* ini mengkompress *voice* menjadi 16 kbps stream.

e. G.729 – *CODEC* ini adalah salah satu *codec* berkualitas lebih baik (*better voice quality CODEC*). *CODEC* ini mengkonversi *voice* menjadi 8 kbps. Terdapat 2 versi yaitu G.729 dan G.729a. G.729a memiliki algoritma yang lebih sederhana dan membutuhkan *processing power* lebih sedikit dibandingkan G.729.

Pada tabel terlampir daftar beberapa teknik kompresi suara yang sering digunakan dengan beberapa parameter yang mencerminkan kinerja dari teknik kompresi suara tersebut.

Tabel II.1 Standarisasi kompresi suara menurut ITU-T[2]

Kompresi	Kbps	MIPS	ms	MOS
G.711PCM	64	0.34	0.125	4.1
G.726 ADPCM	32	14	0.125	3.85
G.728 LD-CELP	16	33	0.625	3.61
G.729 CS-ACELP	8	20	10	3.92
G.729 x2 Encoding	8	20	10	3.27
G.729 x3 Encoding	8	20	10	2.68
G.729a CS-ACELP	8	10.5	10	3.7
G.723.1 MPMLQ	6.3	16	30	3.9

Kolom Kbps memperlihatkan berapa lebar *bandwidth* yang di ambil untuk mengirimkan suara yang di kompres menggunakan teknik kompresi tertentu. MIPS (*Mega Instruction Per Second*) memperlihatkan berapa kebutuhan waktu pemrosesan data pada saat melakukan kompresi suara dalam juta instruksi per detik. Mili-detik (ms) adalah waktu yang dibutuhkan untuk melakukan kompresi. *Mean Opinion Score* (MOS) adalah nilai opini pendengar di penerima.

[2] http://www.voiptroubleshooter.com/basics/mosr.html

2.6.2. Coding dan Decoding

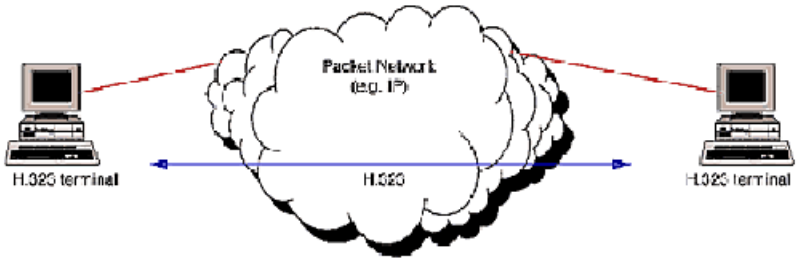
Salah satu komponen yang terpenting dalam *VoIP* adalah peralatan *codec* (*coder* dan *decoder*), *codec* menggunakan teknik penyamplingan sinyal analog untuk dirubah menjadi sinyal digital lalu mereduksi lebar pita sinyal sesuai dengan kebutuhan. Algoritma sebagai proses pengkodean sinyal-sinyal informasi sehingga lebar pita sinyal tersebut dapat direduksi, dipakai pada alat *codec* ini untuk pengkompresan data yang telah didapat dari hasil sampling.

2.6.3. Standar Komunikasi *VoIP*

VoIP dapat berkomunikasi dengan sistem lain yang beroperasi pada jaringan packet-switch. Untuk dapat berkomunikasi dibutuhkan suatu standar sistem komunikasi yang kompatibel satu sama lain. Ada dua standar komunikasi yang digunakan pada *VoIP* yaitu :

2.6.3.1. 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 mul*time*dia seperti komunikasi suara (IP *telephony*), komunikasi video dengan suara (video telephony), dan gabungan suara, video dan data.



Gambar II.9 Terminal pada jaringan paket

Tujuan desain dan pengembangan H.323 adalah untuk memungkinkan interoperabilitas dengan tipe terminal mul*time*dia lainnya. Terminal dengan standar H.323 dapat berkomunikasi dengan terminal H.320 pada N-ISDN, terminal H.321 pada ATM, dan terminal H.324 pada *Public Switched Telephone Network* (PSTN). Terminal H.323 memungkinkan komunikasi *real time* dua arah berupa suara, video dan data.

2.6.3.2. *Session Initiation protocol* (SIP)

SIP merupakan protokol persinyalan yang bertujuan untuk mengendalikan inisiasi, modifikasi, serta terminasi sesi-sesi mul*time*dia, termasuk sesi komunikasi audio atau video. SIP merupakan protokol berbasis teks yang mirip dengan protokol HTTP dan *Simple Mail Transfer Protocol* (SMTP).

SIP adalah protokol *peer-to-peer* yang mengandung arti bahwa fungsi-fungsi call routing dan *session* management didistribusikan ke semua node (termasuk *endpoint* dan *server*) di dalam jaringan SIP. Hal ini berbeda dengan sistem telepon konvensional di mana terminal-terminal telepon sangat bergantung kepada perangkat *switching* yang terpusat.

SIP memiliki fungsi-fungsi yang didefinisikan sebagai berikut:

a. *User location*

   SIP menyediakan kemampuan untuk menemukan lokasi pengguna akhir yang bermaksud akan membangun sebuah sesi atau mengirimkan sebuah permintaan.

b. *User capabilities*

   SIP memungkinkan determinasi kemampuan media dari perangkat yang terlibat di dalam sesi.

c. *User availability*

   SIP memungkinkan determinasi keinginan pengguna untuk melakukan komunikasi.

d. *Session setup*

   SIP memungkinkan modifikasi, *transfer*, dan terminasi dari sebuah sesi aktif.

Jaringan SIP terdiri dari elemen-elemen sebagai berikut:

a. *User Agent* (UA) berfungsi untuk menginisiasi atau merespon transaksi SIP. Sebuah UA dapat bertindak sebagai klien atau *server*.

b. *User Agent* Client (UAC) berfungsi untuk menginisiasi permintaan SIP dan menerima respon SIP.

c. *User Agent Server* (UAS) berfungsi untuk menerima permintaan SIP dan mengirimkan kembali respon SIP.

d. *SIP Proxy* adalah entitas yang berfungsi untuk proses routing dan meneruskan permintaan SIP kepada UAS atau *proxy* lain atas permintaan UAC.

e. *Redirect Server* adalah sebuah UAS yang membangkitkan respon SIP terhadap permintaan yang diterima, memungkinkan UAC secara langsung menghubungi *Uniform Resource Identifiers (URI).*

f. *Registrar Server* adalah sebuah UAS yang menerima permintaan registrasi SIP dan memperbaharui informasi dari pesan tersebut ke dalam database lokasi.

g. *Back*-to-*Back User Agent* (B2BUA) adalah entitas yang berfungsi untuk memproses permintaan SIP yang diterima di mana B2BUA akan bertindak sebagai UAC, membangkitkan kembali permintaan SIP dan mengirimkannya ke dalam jaringan.

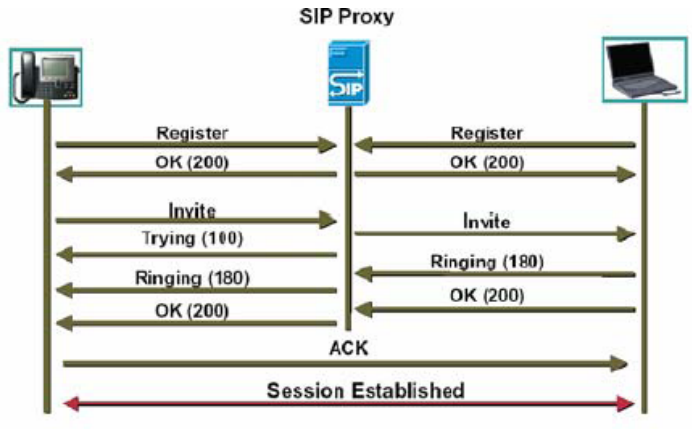
Pada jaringan SIP ada 2 komponen yang selalu ada untuk melakukan komunikasi yaitu *user agent client* dan *server*. Pada SIP terdiri dari 2 jenis *server* yaitu

a. *Proxy server*

   Proxy server adalah *server* yang menerima *request*, mengolahnya, serta meneruskan *request* yang diterimanya ke next hop *server* atau ke end *user* setelah mengubah beberapa header pada pesan *request*.

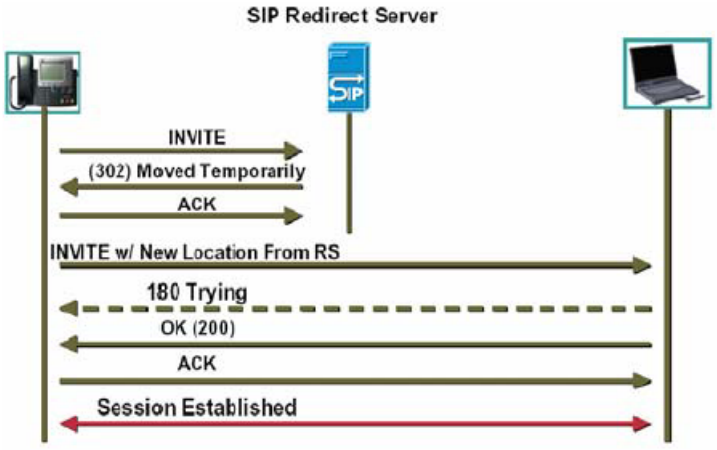
b. *Redirect server*

   Komponen ini merupakan *server* yang menerima pesan *request* serta memberikan respon terhadap *request* tersebut yang berisi alamat dari next hop *server*. Proses persinyalan Menggunakan *Proxy server* Secara umum proses pesan pada *proxy server* adalah sebagai berikut:



Gambar II.10 Proses koneksi menggunakan *proxy server*

Jika komunikasi akan berakhir maka akan dilakukan prosedur berikut 1. Pesan *request* BYE dikirimkan ke *proxy server* 2. Pesan *request* diteruskan *proxy server* ke called 3. Pesan respon 200 K dikirimkan dari called ke *proxy server* 4. Pesan respon diteruskan *proxy server* ke *caller* dan komunikasi berakhir Proses persinyalan menggunakan *redirect server* Berbeda dengan *proxy server* pada *redirect server* ini akan memberi tahukan alamat called untuk kemudian caller yang akan melakukan proses koneksi kepada *called*.



Gambar II.11 Proses koneksi menggunakan redirecr *server*

Prosedur pemutusan hubungan pada *redirect server*

1. Pesan *request* BYE dikirmkan ke *called*

2. Pesan respon 200 OK dikirimkan ke *caller* maka komunikasi berakhir

2.7. Quality of Service

QoS merupakan terminologi yang digunakan untuk mendefinisikan kemampuan suatu jaringan untuk menyediakan tingkat jaminan layanan yang berbeda-beda. Melalui Q0S, seorang *network administrator* dapat memberikan prioritas trafik tertentu. Suatu jaringan, mungkin saja terdiri dari satu atau beberapa teknologi data *link layer* yang mampu dimplementasikan QoS, misalnya; *Frame* Relay, *Ethernet, Token Ring, Point-to-Point Protocol* (PPP), HDLC, X.25, ATM, SONET. Setiap teknologi mempunyai karakteristik yang berbeda-beda yang harus dipertimbangkan ketika mengimplementasikan QoS. QoS dapat dimplementasikan pada situasi *congestion management* atau *congestion avoidance*. Teknik-teknik *congestion management* digunakan untuk mengatur dan memberikan prioritas trafik pada jaringan di mana aplikasi meminta lebih banyak lagi *bandwidth* daripada yang mampu disediakan oleh jaringan. Dengan menerapkan prioritas pada berbagai kelas dari trafik, teknik *congestion management* akan mengoptimalkan aplikasi bisnis yang kritis atau *delay sensitive* untuk dapat beroperasi sebagai mana mestinya pada lingkungan jaringan yang memiliki kongesti. Adapun teknik *collision avoidance* akan membuat mekanisme teknologi tersebut menghindari situasi kongesti. Melalui implementasi QoS di jaringan ini, *network administrator* akan memiliki fleksibilitas yang tinggi untuk mengontrol aliran dan kejadian-kejadian yang ada di trafik pada jaringan.

QoS merupakan peralatan-peralatan yang tersedia untuk menerapkan berbagai jaminan, dimana tingkat minimum layanan dapat disediakan. Banyak protokol dan aplikasi yang tidak begitu sensitif terhadap *network congestion*. *File Transfer Protocol* (FTP) contohnya, mempunyai toleransi yang besar untuk *network delay* dan terbatasnya *bandwidth*. Di sisi *user*, kejadian tersebut akan menyebabkan proses *transfer file* seperti *download* atau *upload* yang lambat, walaupun mengganggu *user*, namun kelambatan ini tidak akan menggagalkan operasi dari aplikasi tersebut. Lain halnya dengan aplikasi-aplikasi baru seperti*Voice* dan *Video*, yang pada umumnya sensitif terhadap *delay*. Jika paket dari *voice* mengalami proses yang lama untuk sampai ke tujuan, maka akan dapat merusak *Voice* yang didengarkan. Dalam hal ini QoS dapat digunakan untuk menyediakan jaminan layanan untuk aplikasi-aplikasi tersebut. SNA merupakan salah satu contoh protokol yang sangat sensitif dengan menggunakan protokol *handshake* dan biasanya akan melakukan terminasi dari *session* jika tidak memperoleh suatu *acknowledgement*, lain halnya dengan TCP/IP. Sehingga dalam kasus ini, memberikan prioritas pada trafik SNA di atas protokol lainnya akan memberikan QoS yang lebih baik.

2.7.1. Alasan Perlunya QoS

Ada beberapa alasan mengapa kita memerlukan QoS, yaitu:

a. Untuk memberikan prioritas untuk aplikasi-aplikasi yang kritis pada jaringan.
b. Untuk memaksimalkan penggunaan investasi jaringan yang sudah ada.

c. Untuk meningkatkan performansi untuk aplikasi-aplikasi yang sensitif terhadap *delay*, seperti *Voice* dan Video.
d. Untuk merespon terhadap adanya perubahan-perubahan pada aliran trafik di jaringan.

Point terakhir nampaknya terasa tidak penting, benarkah demikian? *Naptser*, *PointCast*, *World-Wide-Web* adalah contoh aplikasi-aplikasi "*self-deployed*" yang dapat menyebabkan mimpi buruk bagi *network* administrator. Tidak seorang pun pernah merencanakan jalannya *Web browsing* seperti sekarang ini, hampir seluruh trafik di *Internet* membawa *prefix* "http". Dengan adanya perubahan permintaan *bandwidth*, QoS dapat digunakan untuk menjamin kualitas layanan, jika beberapa *user* dalam suatu perusahaan sedang mendengarkan siaran radio lewat *Internet*, maka tidak akan memperlambat trafik penting yang ada ke perusahaan tersebut.

Metode paling sederhana yang sering digunakan untuk memperoleh performansi yang lebih baik pada jaringan adalah dengan meminta lebih banyak *bandwidth*. Saat ini *Gigabit Ethernet* dan *Optical Network*ing sudah tersedia.

Peningkatan *bandwidth* dapat menjadi solusi sementara untuk meningkatkan kualitas layanan, namun tidak akan dapat untuk menjamin kualitas layanan seterusnya. Aplikasi-aplikasi yang didukung oleh protokol-protokol yang ada akan terus meminta *bandwidth* lagi. Langkah tepat untuk kondisi demikian adalah menganalisa trafik yang lewat, mengidentifikasi urutan kepentingan dari protokol dan aplikasi di jaringan, dan menentukan strategi untuk memberikan prioritas untuk mengakses *bandwidth* yang tersedia. QoS akan membuat seorang *network administator* mengawasi *bandwidth*, *latency* dan *jitter*, serta

mem2nimisasi paket yang hilang pada suatu newtwork, dengan memberikan prioritas pada protokol. *Bandwidth* adalah ukuran kapasitas pada suatu jaringan atau link. *Latency* adalah *delay* dari suatu paket untuk melewati jaringan. *Jitter* adalah perubahan *latency* pada suatu periode waktu. Melalui penerapan teknik-teknik QoS, maka akan dapat dilakukan pengaturan dari ketiga parameter di atas.

Saat ini di kebanyakan jaringan di perkantoran tidak begitu memperhatikan QoS. Namun, dengan berkembangnya aplikasi-aplikasi, misalnya mulicast, *streaming* mul*time*dia, dan *Voice* over IP (*VoIP*) kebutuhan akan QoS akan semakin terasa. Terlebih lagi aplikasi-aplikasi tersebut terhadap *jitter* dan *delay* dan performansi yang buruk akan sangat terasa pada *end user*. Dalam hal ini seorang *network* administrator dapat melakukan tindakan manajemen proaktif untuk aplikasi-aplikasi sensitif yang baru dengan mengaplikasikan teknik-teknik QoS pada jaringan. Penting untuk diketahui, bahwa QoS bukanlah solusi yang ajaib untuk setiap masalah kongesti, karena dapat saja solusi terbaik untuk mengatasi *congested network* memang adalah melakukan upgrade pada *bandwidth*.

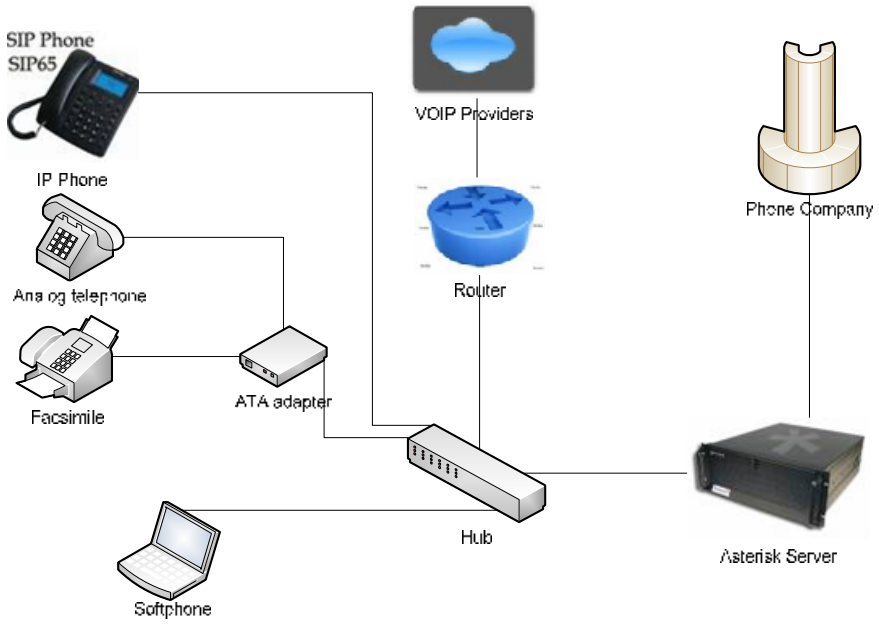
2.8. *Asterisk*

Asterisk adalah PBX dalam bentuk *software*. *Asterisk* dikembangkan pertama kali oleh *Mark Spencer* pada tahun 1999. *Asterisk* dapat dijalankan di berbagai sistem operasi dan menyediakan *feature-feature* yang tersedia di PBX pada umumnya. *Asterisk* juga dapat melakukan *VoIP* melalui berbagai protokol dan dapat berinteraksi dengan berbagai perangkat *telephony* yang harganya relatif murah.

Dengan melakukan instalasi *Asterisk* pada PC. maka PC berubah menjadi sebuah PBX dengan kemampuan untuk berkomunikasi dengan jaringan IP. PBX yang mempunyai kemampuan tersebut dikenal dengan istilah IP PBX.

Seperti halnya PBX, dalam IP PBX pun dikenal tiga komponen penting yang membentuk fungsi PBX, yaitu *extension, trunk,* dan *dial plan*.

a. *Extension* adalah komponen yang menangani registrasi dari pengguna, serta menyediakan *user*name dan password bagi *user* agar dapat terhubung dengan IP PBX.
b. *Trunk* adalah komponen yang menangani registrasi satu IP PBX ke IP PBX lainnya.
c. *Dial plan* adalah komponen yang mengatur penomoran dan *call routing*.



Gambar II.12 Contoh jaringan dengan *asterisk*

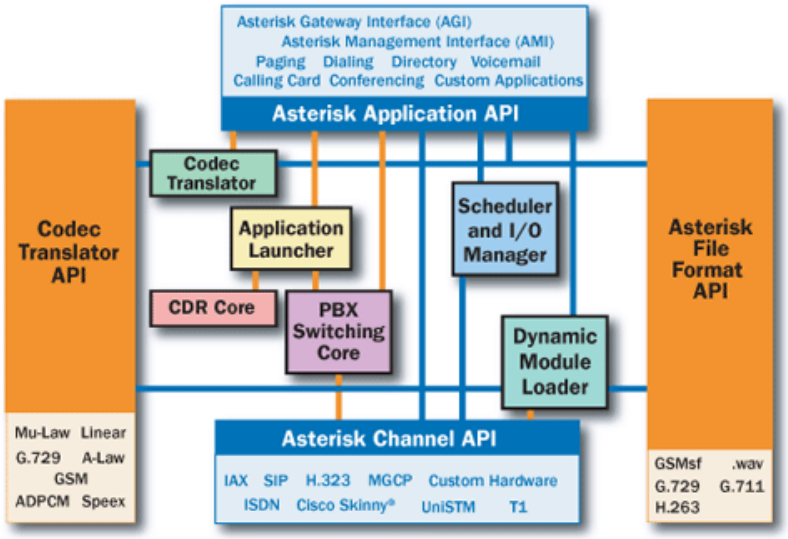
2.8.1. Arsitektur *Asterisk*

Pada dasarnya, arsitektur *Asterisk* sangatlah sederhana. Protokol yang d2mplementasikan oleh *Asterisk* antara lain SIP, H323, IAX, MGCP. Aplikasi yang didukung oleh *Asterisk* antara lain :

a. Mendukung bermacam-macam protokol *VoIP* gateway antara lain SIP, H323, IAX, MGCP.
b. IP PBX (*Internet Protocol* Private Branch eXchange).
c. *Interactive Voice* Response (IVR) *server.*
d. Conferencing server.
e. Translasi nomor telepon.
f. Aplikasi *calling card.*
g. Antrian pangilan.

Asterisk memiliki beberapa komponen inti yang memegang peranan penting. Ketika *Asterisk* pertama kali *start,* akan di-*load Dynamic Module Loader* yang menginisialisasi masing-masing *driver* untuk pengaturan *channel,* format file, *detail record call, codec,* dan aplikasi yang digunakan. Yang akan dilakukan berikutnya adalah *Asterisk* PBX *Switching Core* memulai menerima panggilan yang datang, dan ditangani menurut *dialplan* yang telah dikonfigurasi. Akan digunakan *Application Launcher* untuk bunyi dering, koneksi *voicemail, dialing* keluar *trunk,* dan lain-lain. *Asterisk* juga menyediakan standar *Scheduler and I/O Manager,* yang akan bermanfaaat dalam pelaksanaan aplikasi, terutama pengaturan jadwal-jadwal berkaitan dengan fungsi PBX. Komponen berikutnya

adalah *codec translator*, yang berfungsi untuk mengijinkan dua *codec* yang berbeda saling berkomunikasi.



Gambar II.13 Arsitektur *asterisk*

Sumber : Mark Spencer, dkk.. *The Asterisk Handbook, p.* 12. http:// www.digium.com/**handbook**-draft.pdf

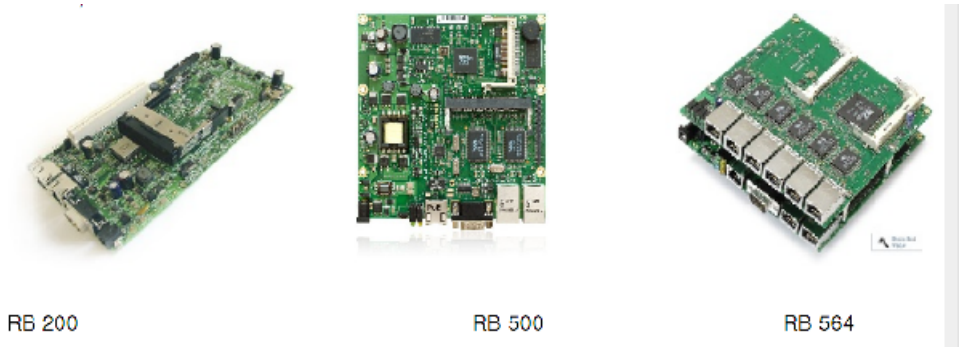
2.9. *Mikrotik*

Mikrotik RouterOS™ adalah sistem operasi dan yang dapat digunakan untuk menjadikan komputer manjadi *router network* yang handal, mencakup berbagai fitur lengkap untuk *network* dan *wireless*.



Gambar II.14 Negara asal *mikrotik*

Disamping *software router operating system*, *mikrotik* juga mengembangkan *hardware* dengan spesifikasi dan karakteristik unik serta mempunyai kehandalan sebagai mesin *router*. Dengan dipasarkannya *Mikrotik RouterBoard*TM dalam berbagai seri memudahkan kita untuk memilih produk sesuai dengan kebutuhan . fleksibilitas ini salah satu faktor yang menjadikan *mikrotik RouterBoard* booming dan menjadi pilihan ekonomis dengna kualitas prima disamping banyak keunggulan- keunggulan lainnya.



Gambar II.15 *Routerboard*

Mikrotik *Router*OS adalah sistem operasi untuk *Router* dari *Mikrotik* berbasis pada *Linux Software Programme*. Pada saat sekarang, *mikrotik* dapat dijalankan melalui program berbasis windows yang dinamakan winbox, sehingga *user* dapat dengan mudah mengakses dan mengkonfigurasi *router* sesuai dengan kebutuhan dengan mudah dan efektif dan efisien. Memperkecil kesalahan pada waktu konfigurasi, mudak dipahami dan *customize* sesuai yang kita inginkan. Fitur-fitur *mikrotik router* OS cukup banyak, bahkan lebih lengkap dibanding *router*OS yang lain, dan sangat membantu kita dalam berimprovisasi untuk mencari solusi dari kendala yang ada di lapangan. Berikut overview tentang karakter dan kemampuan *mikrotik router*OS, diantaranya adalah sebagai berikut :

2.9.1. Penanganan Protokol TCP/IP

Mikrotik dapat melakukan pengaturan – pengaturan pada protokol tcp/ip yang meliputi :

a. *Firewall dan NAT*

paket filtering yang stabil, filtering peer to peer, source and destination. NAT diklasifikasikan oleh source MAC, ip address, ports, *protocol*s, pilihan *protocol*s, *interface*s, internal marks, content, matching**.**

b. *Routing – static*

Static routing, multi path routing yang efisien, policy based routing, RIP v1, V2, OSPF v2, BGP v4.

c. *Data rate management*

Per ip / *protocol* / subnet / port / firewall mark , HTB, PCQ , RED, SFQ, byte limited queue, paket limited queue, pembatasan hirarkis berjenjang, CIR, MIR .

d. *Hotspot*

Hotspot gateway dengan RADIUS autentikasi dan akunting, pembatasan data rate, quota traffic, inmformasi status *real time*, walled garden, HTML login page customize, mendukung iPass, SSL secure authentication.

e. *Point to point tunneling protocol*

PPTP, PPPoE and L2TP access concentrator and clients, PAP, CHAP, MSCHAPv1, RADIUS authentication and accounting, MPPE encryption, PPPoE dial on demand.

f. *Simple tunnels*

IPIP tunnels, EoIP (*ethernet* over IP), IP sec – IP security AH and ESP *protocol*s.

g. *Web proxy*

FTP, HTTP and HTTPS caching *proxy server*, transparent HTTP caching *proxy*, mendukung SOCKS *protocol*, mendukung caching drive terpisah,access control list, caching list, parent *proxy* support dengan nama disesuaikan untuk pemakain alokal, dynamic DNS client, local DNS cache with static entries

h. DHCP

DHCP *server* per *interface, DHCP relay, DHCP client, multiple DHCP Network, static and dynamic DHCP leases*

i. Universal client

j. VRRP *protocol* untuk penggunaan tinggi

k. UPnP

l. NTP

m. *Monitoring* dan *accounting*

n. SNMP

o. M3p (*mikrotik packet packer protocol for wireless link and ethernet*)

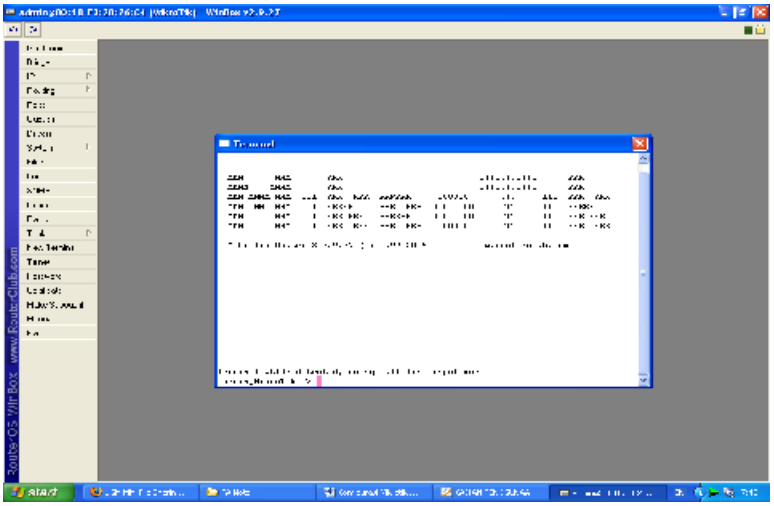
p. MNDP (*mikotik neighbor discovery protocol*)

q. Tools

ping, traceroute, bandwith test, ping flood, telnet, SSH, packet sniffer

2.9.2. Pilihan konfigurasi

*Router*OS menyediakan *interface* konfigurasi *command line interface* yang kuat. Anda juga bisa mengelola *router* melalui winbox. Winbox lebih mudah digunakan dan mendukung *remote programming* yang mudah digunakan dan dapat dijalankan pada komputer berbabasis *windows*. Sedangkan kalau kita menggunakan *Command line*, kita harus hafal perintah-perintah yang akan digunakan beserta opsi-opsi yang menyertainya.



Gambar II.16 Winbox

Feature utama:

a. *User interface* yang bersih dan konsisten
b. Konfigurasi dan monitoring yang sesuai waktu
c. Multiple connections
d. Mengikuti aturan pengguna
e. Action history, undo/redo actions
f. *Safe mode* operation
g. *Script* bisa dijadwalkan untuk eksekusi waktu tertentu, periodik atau insidentil. Semua perintah command line mendukung scripts.

Jika *router* tidak dikenal, hanya ada dua cara untuk mengkonfigurasikannya

a. local terminal console

AT, PS/2 or USB key*board* and VGA compatible video *card* dengan monitor.

b. Serial console

Menggunakan port serial yang pertama (COM1) yang diset deafault ke 9600bit/s, 8 data bits, 1 stop bits, no parity.

Setelah *router* dikonfigurasikan, harus dikelola dengan menggunakan *interface* berikut :

a. local terminal console

AT, PS/2 or USB key*board* and VGA compatible video *card* dengan monitor.

b. Serial console

Menggunakn port serial yang pertama (COM1) yang diset deafault ke 9600bit/s, 8 data bits, 1 stop bits, no parity.

c. Telnet

telnet server diposisikan pada port TCP 23 secara *default.*

d. SSH

Cara ini tersedia jika paket security d2nstalasi, menggunakan port TCP 22.

e. MAC telnet

Server mikrotik MAC Telnet *protocol* secara default berjalan di semua *interface ethernet* yang ada.

f. Winbox

Winbox adalah administrasi *router*OS remote GUI for windows, menggunakan port TCP 3986.

Karakteristik *mikrotik router*OS diatas cukup membuktikan bahwa dengan *system hardware, software, brainware* yang terintegrasi dengan seimbang,

mampu mempersembahkan kinerja optimal dan mendobrak berbagai kendala penghalang.

2.10. Commview

Commview merupakan perangkat lunak untuk melakukan analisa jaringan komputer dari vendor *tamosoft software*, karena dapat digunakan untuk mengcapture dengan lengkap semua data yang terjadi pada jaringan yang meliputi *bandwidth*, *delay*, dan *protocol* yang digunakan. Program ini merupakan program favorit para administrator jaringan untuk menganalisa jaringan karena kemudahan dan kelengkapan fitur yang menyertainya.*Software* commview dapat didownload versi trialnya secara gratis dari www.download.com.

Software commview ini mempunyai fitur yang lengkap dan memadai untuk melakukan analisa jaringan. Pada software ini terdapat fitur pengukuran parameter-paremeter yagnmenyangkut VoIP yaitu jitter, bandwith dan MOS Score sehingga software ini dapat memudahkan penelitian yang penulis lakukan. Paket-paket yang berjalan pada jaringan komputer akan difilter berdasarkan protokol yang digunakan. Hasil dari analisa dapat ditampilkan dalam *report* yang berbentuk grafik ataupun laporan berbasis teks.