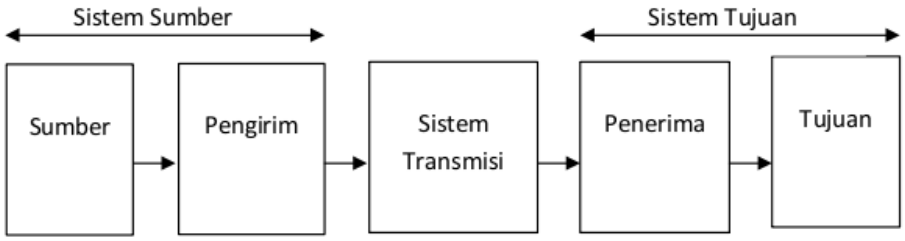
BAB II

LANDASAN TEORI

2.1 Komunikasi

Istilah komunikasi berasal dari kata berbahasa Inggris yaitu *communications*. Sedangkan kata *communications* berasal dari bahasa latin yaitu *communicare* yang berarti saling berbagi (*share*). Komunikasi dapat diartikan sebagai proses menampilkan, mengubah, menginterpretasikan, atau mengolah informasi antara manusia dan mesin.

Proses komunikasi secara umum melibatkan beberapa elemen yaitu pengirim (*transmitter*), penerima (*receiver*) dan sebuah medium transmisi untuk tempat mengalirnya informasi. Elemen-elemen tersebut dapat dilihat pada gambar di bawah ini :



Gambar 2.1 Sistem Komunikasi

Berikut keterangan detail setiap elemen dalam proses komunikasi pada gambar di atas :

1. Sumber *(source)*

   Membangkitkan data atau informasi yang akan ditransmisikan, contoh telepon dan PC.

2. Pengirim (*transmitter*)

   Data atau informasi yang dibangkitkan oleh sistem sumber tidak ditransmisikan secara langsung dalam bentuk aslinya. Sebuah *transmitter* cukup memindahkan dan menandai informasi dengan cara yang sama seperti menghasilkan sinyal-sinyal elektromagnetik yang dapat ditransmisikan melewati beberapa sistem transmisi berurutan.

3. Sistem Transmisi (*transmission system*)

   Merupakan jalur transmisi tunggal (*single transmission line*) atau merupakan jaringan kompleks yang menghubungkan sumber dan tujuan.

4. Penerima (*receiver*)

   Berfungsi menerima sinyal dari sistem transmisi dan menggabungkan ke dalam bentuk tertentu yang dapat ditangkap dan dimengerti oleh tujuan (*destination*).

5. Tujuan (*destination*)

   Menangkap data yang dihasilkan oleh *receiver*.

2.2 Internet

Internet adalah suatu jaringan komputer global yang terbentuk dari jaringan-jaringan komputer lokal dan regional yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut. Dengan menggunakan protokol *Transmission Control Protocol/Internet Protocol* (TCP/IP) dan didukung oleh media komunikasi seperti satelit dan paket radio, internet telah memungkinkan komunikasi antar komputer dengan jarak yang tidak terbatas.

Internet dapat menghubungkan komputer dan jaringan komputer yang berada di ratusan negara dan departemen atau instansi baik swasta maupun

pemerintah. Melalui internet, siapa saja dapat dengan leluasa mengakses berbagai macam informasi dari tempat mana saja. Informasi yang dapat diakses pun dapat berupa teks, grafik, suara maupun radio.

2.3 Jaringan Komputer

Jaringan komputer adalah sekumpulan peralatan komputer (*hardware* dan *software*) yang dihubungkan agar dapat saling berkomunikasi dengan tujuan komunikasi dan berbagi sumber daya. Jaringan bisa terbentuk dari minimal dua buah komputer sederhana dan kecil yang saling berbagi-pakai sumber daya seperti *printer* dan CD-ROM yang terpasang pada salah satu komputer. Namun jaringan juga bisa berbentuk jalinan saluran komputer terbesar di dunia yaitu *internet*.

2.3.1 Manfaat Jaringan Komputer

Perkembangan teknologi komputer saat ini berkembang sangat cepat. Banyak sekali manfaat yang dapat diperoleh dalam suatu jaringan komputer. Manfaat-manfaat tersebut diantaranya adalah :

a. Jaringan komputer memungkinkan seseorang untuk mengakses *file* yang dimilikinya atau *file* orang lain yang telah diizinkan untuk diakses, di manapun dan kapanpun.
b. Jaringan komputer memungkinkan proses pengiriman data berlangsung cepat dan efisien.
c. Jaringan komputer memungkinkan adanya *sharing hardware* antar *client*.
d. Jaringan komputer memungkinkan seseorang berhubungan dengan orang lain di berbagai negara dengan komunikasi lewat teks, gambar, *audio*, dan *video* secara *real-time*.

e. Jaringan komputer dapat menekan biaya operasional seperti pemakaian kertas, pengiriman surat atau berkas, telepon, serta pembelian *hardware*.

2.4 Klasifikasi Jaringan Komputer

2.4.1 Berdasarkan Skala

a. *Local Area Network* (LAN)

LAN digunakan untuk menghubungkan komputer yang berada di dalam suatu area yang kecil, misalnya di dalam suatu gedung perkantoran, kampus, atau warnet. Jarak antar komputer yang dihubungkan dapat dicapai 5 sampai 10 km. Suatu LAN biasanya bekerja pada kecepatan mulai 10 Mbps sampai 100 Mbps. LAN menjadi populer karena memungkinkan banyak pengguna untuk memakai sumber daya secara bersama-sama. Contoh dari sumber daya yang dapat digunakan itu misalnya suatu *mainframe*, *file server*, *printer*, dan sebagainya.

b. *Metropolitan Area Network* (MAN)

MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang lokasinya berjauhan. Jangkauan MAN bisa mencapai 10 km sampai ratusan kilometer. Suatu MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps.

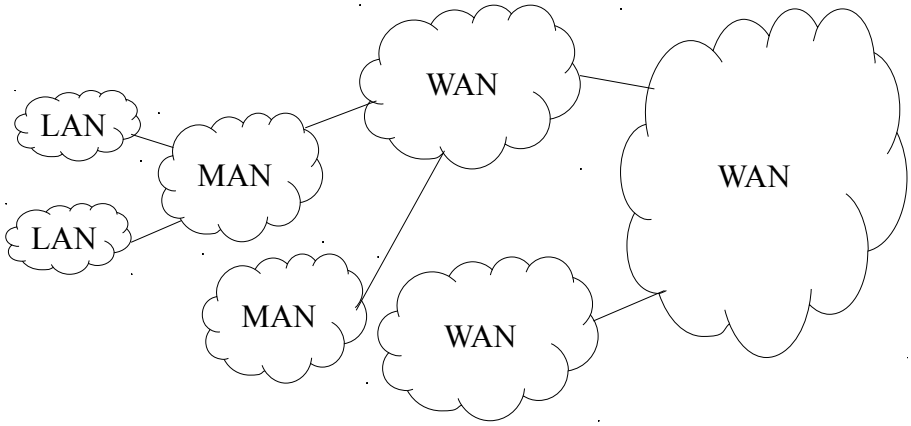
c. *Wide Area Network* (WAN)

WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari satu kota ke kota yang lain di dalam suatu negara. Cakupan WAN bisa meliputi 100 km sampai 1.000 km, dan kecepatan antar kota bervariasi antara 1,5 sampai 2,4 Gbps.

d. Global Area Network (GAN)

GAN merupakan suatu jaringan yang menghubungkan negara-negara di seluruh dunia. Kecepatan GAN bervariasi mulai dari 1,5 Mbps sampai dengan 100 Gbps dan mempunyai area cakupan mencapai ribuan kilometer.

LAN, MAN, WAN dan GAN dapat berinteraksi satu sama lain membentuk jaringan komputer besar yang disebut sebagai internet. Gambar 2.2 menunjukkan interaksi antara jaringan-jaringan tersebut.

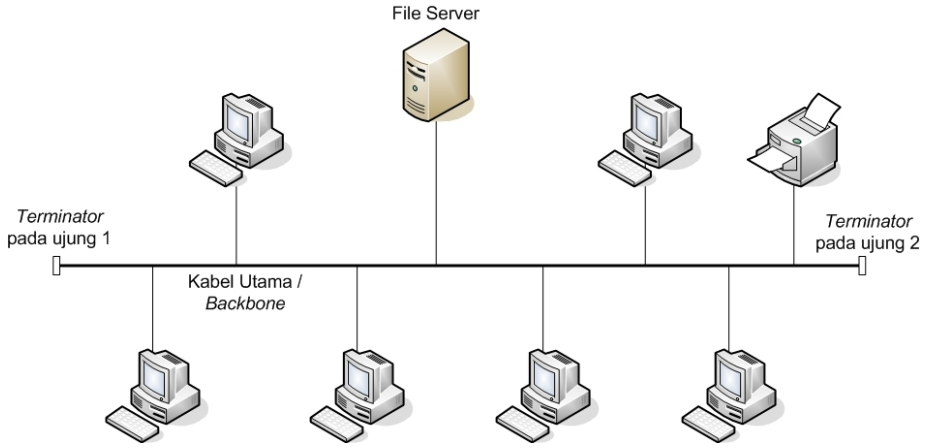


Gambar 2.2 Interaksi antara LAN, MAN, WAN dan GAN

2.4.2 Berdasarkan Topologi

a. Topologi *Linear Bus*

Topologi Jaringan adalah gambaran secara fisik dari pola hubungan antara komponen-komponen jaringan, yang meliputi *server*, *workstation*, *hub* dan pengkabelannnya. Secara umum jaringan komputer dapat dikategorikan menjadi empat macam sesuai dengan topologi jaringan yang digunakan, yaitu *Linear Bus*, *Star*, *Ring* dan *Tree*.



Gambar 2.3 Topologi Linear Bus

b. Topologi *Star*

Pada topologi *Star*, setiap *nodes* (*file server*, *workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*. Data yang dikirim ke jaringan lokal akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Concentrator* akan mengatur dan mengendalikan keseluruhan fungsi jaringan dan juga bertindak sebagai *repeater* (penguat aliran data).

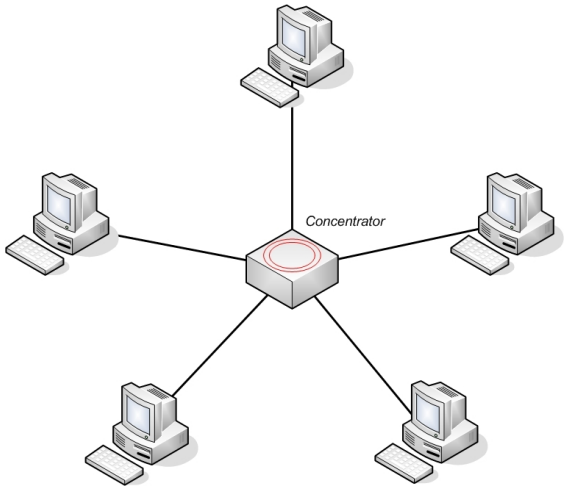
Beberapa kelebihan dari topologi *star* :

1. Pemasangan dan pengkabelan yang lebih mudah.
2. Tidak mengakibatkan gangguan pada jaringan ketika akan memasang atau memindahkan perangkat jaringan lainnya.
3. Mudah untuk mendeteksi kesalahan dan memindahkan perangkat-perangkat lainnya.

Adapun kekurangan yang terdapat pada topologi *star* adalah :

1. Membutuhkan lebih banyak kabel daripada topologi linear bus.
2. Membutuhkan *concentrator.* Bila *concentrator* rusak maka semua *node* yang terkoneksi tidak dapat terdeteksi.

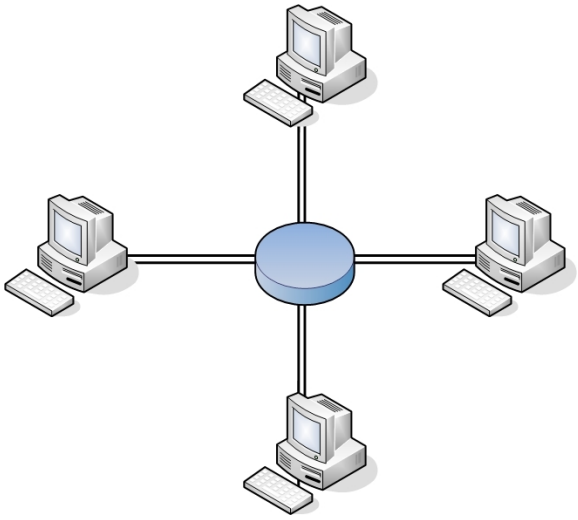
3. Lebih mahal dari pada topologi *linear bus*, karena penambahan biaya perangkat *concentrator* dan kabel yang lebih panjang.



Gambar 2.4 Topologi *Star*

c. Topologi *Ring*

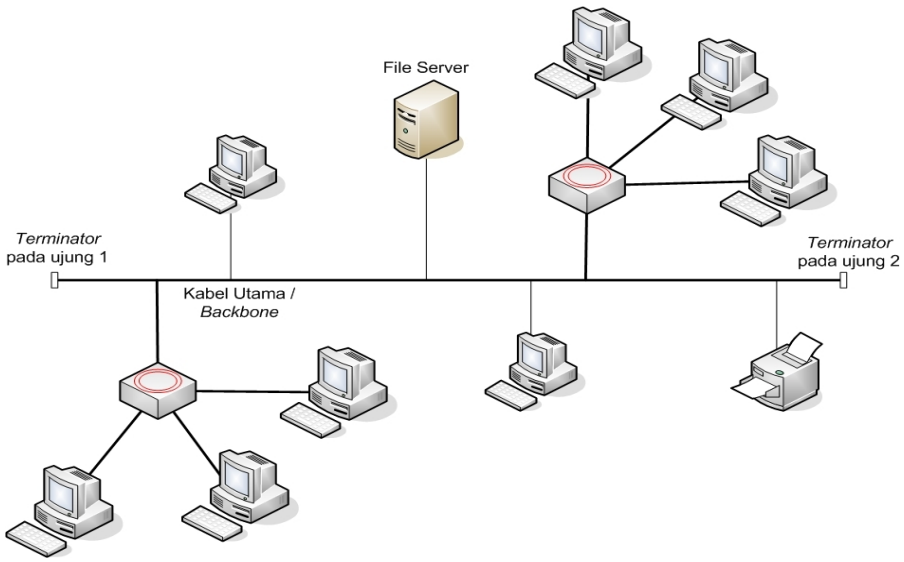
Dalam topologi r*ing* semua *workstasion* dan *server* dihubungkan sehingga terbentuk suatu pola lingkaran atau *ring*. Tiap *workstasion* atau pun *server* akan menerima dan melewatkan informasi dari satu komputer ke komputer lain, bila alamat-alamat yang dimaksud sesuai maka informasi diterima dan bila tidak informasi akan dilewatkan.



Gambar 2.5 Topologi *Ring*

d. Topologi *Tree*

Topologi model ini sebenarnya merupakan perpaduan antara topologi *Linear Bus* dan *Star*. Topologi *Tree* terdiri dari kelompok-kelompok *workstation* dengan konfigurasi *Star* yang terkoneksi ke kabel utama dengan menggunakan topologi *Linear Bus*. Topologi ini memungkinkan untuk pengembangan jaringan yang telah ada, dan memungkinkan untuk melakukan konfigurasi jaringan sesuai dengan kebutuhan.

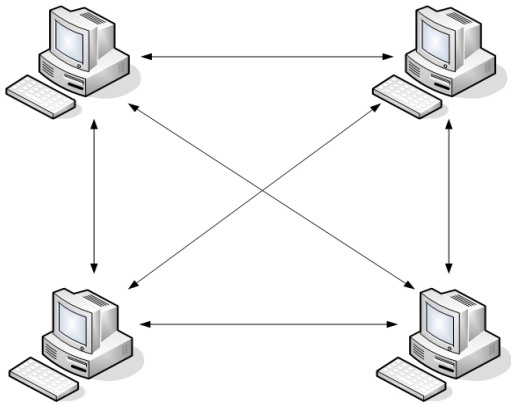


Gambar 2.6 Topologi *Tree*

2.4.3 Berdasarkan Arsitektur

a. *Peer-to-Peer*

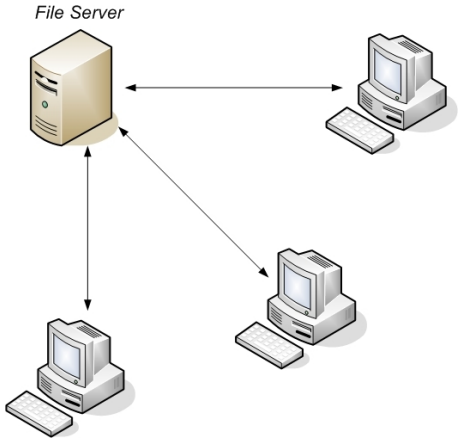
Model hubungan *peer to peer* memungkinkan user membagi sumber daya yang ada di komputernya baik berupa *file*, layanan *printer* dan lain-lain serta mengakses sumber daya yang terdapat pada komputer lain. Pada jaringan komputer model *peer to peer,* setiap *host* dapat menjadi *server* dan juga menjadi *client* secara bersamaan.



Gambar 2.7 Jaringan *Peer-to-Peer*

b. *Client*/*Server*

Model hubungan *Client Server* memungkinkan jaringan untuk memusatkan fungsi dan aplikasi kepada komputer *server*. Sebuah *server* menjadi jantung dari keseluruhan sistem, memungkinkan untuk mengakses sumber daya, dan menyediakan keamanan. Komputer server menangani berbagai multi *service* diantaranya yaitu *mail server, web server, file server, database server* dan lainnya.



Gambar 2.8 Jaringan *Client/Server*

2.5 Media Transmisi

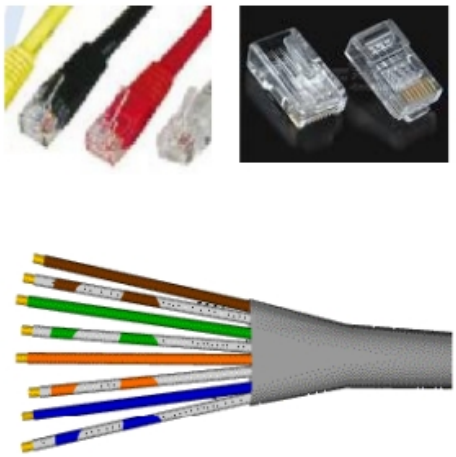
2.5.1 Media Transmisi pada LAN

Kabel merupakan media transmisi untuk mengirimkan informasi dari satu komputer ke komputer yang lain. Ada beberapa macam tipe kabel yang umumnya digunakan pada LAN, yaitu :

a. Kabel *Unshielded Twisted Pair* (UTP)

Kabel *twisted pair* terdiri dari dua tipe yaitu *Shielded* dan *Unshielded*. *Unshielded Twisted Pair* (UTP) adalah yang paling populer dan umumnya merupakan pilihan yang terbaik untuk jaringan sederhana.

Kualitas kabel UTP berbeda dengan kabel telepon, kabel jenis ini mempunyai empat pasangan kabel di dalamnya. Setiap pasangan adalah jenis kabel kembar. Jenis konektor untuk kabel jenis ini adalah konektor RJ-45.



Gambar 2.9 Kabel *Unshielded Twisted Pair* dan Konektor Rj-45

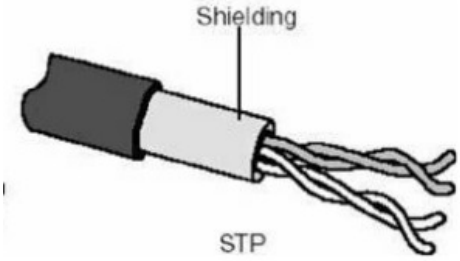
Kabel UTP memiliki kategori atau jenis yang berbeda sesuai dengan fungsi dan kebutuhan dalam jaringan yang dibangun. Berikut tabel kategori kabel dari UTP.

Tabel 2.1 Kategori Kabel *Unshielded Twisted Pair*

Kategori	Penggunaan
Kategori 1	Komunikasi suara
Kategori 2	Komunikasi data sampai dengan 4 Mbps
Kategori 3	Komunikasi data sampai dengan 10 Mbps
Kategori 4	Komunikasi data sampai dengan 20 Mbps
Kategori 5	Komunikasi data sampai dengan 100 Mbps

b. Kabel *Shielded Twisted Pair* (STP)

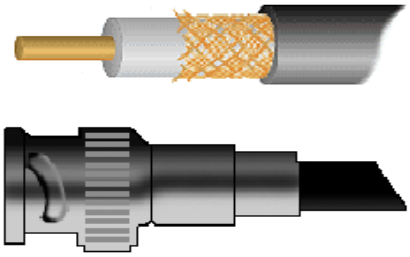
Kekurangan kabel jenis ini adalah, sangat sensitif terhadap sinyal radio dan listrik. Kabel seperti ini sangat baik digunakan pada lingkungan dengan intensitas pengaruh listrik yang kurang, serta biasanya digunakan pada jaringan yang menggunakan topologi *Token Ring*.



Gambar 2.10 Kabel *Shielded Twisted Pair (STP)*

c. Kabel *Coaxial*

Kabel *coaxial* adalah kabel yang memiliki satu *copper conductor* di bagian tengahnya. Sebuah lapisan plastik menutupi diantara konduktor dan lapisan pengaman serat besi. Lapisan serat besi tersebut membantu menutupi gangguan dari arus listrik, lalu lintas kendaraan atau mesin, dan komputer. Tipe konektor untuk kabel jenis ini adalah konektor *Bayone-Neill-Concelman* (BNC).



Gambar 2.11 Kabel *Coaxial* dan Konektor BNC

d. Kabel *Fibre Optic*

Kabel *fibre optic* mempunyai kemampuan mentransmisi sinyal melalui jarak yang relatif jauh dan mempunyai kecepatan yang baik dari pada kabel *coaxial* ataupun kabel *twisted*. Kabel ini sangat baik digunakan untuk fasilitas konferensi radio atau layanan interaktif.



Gambar 2.12 Kabel *Fibre optic*

2.5.2 Media Transmisi pada Wireless-LAN (WLAN)

Beberapa media transmisi data yang digunakan pada jaringan WLAN adalah *Infra Red* (IR) dan *Radio Frequency* (RF).

a. Infra Red (IR)

Infra red banyak digunakan pada komunikasi jarak dekat, contoh paling umum pemakaian IR adalah *remote control.* WLAN menggunakan IR sebagai media transmisi karena IR dapat menawarkan *data rate* tinggi (sekitar ratusan Mbps), konsumsi dayanya kecil dan harganya murah.

b. Radio Frequency (RF)

Penggunaan RF banyak digunakan pada beberapa perangkat seperti stasiun radio, stasiun TV, telepon *cordless* dan lain-lain. WLAN menggunakan RF sebagai media transmisi karena jangkauannya jauh, dapat menembus tembok, mendukung teknik *hand off*, mendukung mobilitas tinggi, dan dapat digunakan di luar ruangan.

c. Gelombang Mikro Teresterial

Sambungan gelombang mikro teresterial (*terrestrial microwave*) secara luas digunakan untuk menyediakan komunikasi praktis namun secara fisik terlalu mahal. Karena gelombang mikro berjalan sepanjang atmosfir bumi, maka gangguan akibat beberapa faktor rentan terjadi.

Walaupun kondisi rentan sangat komunikasi menggunakan gelombang mikro yang menyusuri atmosfir bumi ini dapat digunakan untuk hubungan yang andal (*reliable*) sampai sejauh 50 km.

d. Satelit

Saat ini data juga data juga dapat ditransmisikan menggunakan sistem satelit. Data yang telah dimodulasi diterima dan dikirim ulang *(relay)* ke tujuan yang dimaksud menggunakan rangkaian terpasang yang dikenal denagn sebuah *transponde.*

Sebuah satelit dapat menyediakan ratusan saluran data yang mempunyai bitrate tinggi dengan menggunakan teknik *multiplexing*. Satelit yang biasanya digunakan untuk tujuan komunikasi adalah *geosttionary* (stasiun bumi).

2.6 Protokol Jaringan Komputer

2.6.1 Protokol

Protokol adalah sekumpulan aturan yang mendefinisikan beberapa fungsi seperti pembuatan hubungan, mengirim pesan, data, informasi atau *file* yang harus dipenuhi oleh *transmitter* dan *receiver* agar suatu sesi komunikasi data dapat berlangsung dengan baik dan benar.

2.6.2 Model Referensi OSI

Protokol jaringan disusun dalam bentuk lapisan-lapisan (*layer*). Jumlah, nama isi dan fungsi setiap lapisan tersebut berbeda. Susunan lapisan ini menunjukkan tahapan dalam melakukan komunikasi. Salah satu standar dalam protokol jaringan yang dikembangkan oleh *International Standars Organization* (ISO) adalah model referensi *Open System Interconnection* atau disebut dengan Model Referensi OSI.

Model referensi OSI menjabarkan sebuah pendekatan berlapis (*layering*) terhadap jaringan. Setiap lapis (*layer*) dari model OSI mewakili sebuah porsi yang berbeda dari komunikasi. Model referensi OSI menjadi standar internasional untuk komunikasi jaringan pada tahun 1984.

Pada awalnya model OSI akan menjadi standar terakhir untuk komunikasi data, tetapi protokol TCP/IP yang sekarang ini menjadi arsitektur model lapisan dari protokol internet yang sangat dominan bahkan terus menerus diuji,

dikembangkan dan diperluas standarnya. OSI merupakan himpunan protokol yang memungkinkan terhubung dua sistem yang berbeda yang berasal dari perangkat keras jaringan komputer yang berbeda pula.

Model Referensi OSI terdiri dari tujuh lapisan fisik sampai dengan aplikasi. Berikut penjelasan setiap lapisan dari Model Referensi OSI, yaitu :

1. *Physical Layer*

   Lapisan pertama ini berfungsi untuk mengatur sinkronisasi pengiriman dan penerimaan data, spesifikasi mekanis dan elektris, menerapkan prosedur untuk membangun, mengirimkan data/informasi dalam bentuk digit *biner*, memelihara dan memutuskan hubungan komunikasi.

2. *Data Link Layer*

   Lapisan kedua ini akan memberikan transfer data/informasi yang meyakinkan kepada lapisan fisik dalam bentuk paket yang dilengkapi dengan SYNC, *error control* dan *flow control*. Pada lapisan ini juga dilakukan persiapan untuk mengaktifkan, memelihara dan memutuskan suatu hubungan komunikasi, pendeteksian kesalahan yang mungkin terjadi pada saat pengiriman data dan pengendalian.

3. *Network Layer*

   Lapisan ini berfungsi memberikan memberikan layanan data dengan menentukan rute pengiriman dan mengendalikannya sehingga tidak terjadi kemacetan dan data dapat sampai di tempat tujuan dengan baik.

4. *Transmission*/*Transport Layer*

   Lapisan keempat ini menjamin bahwa data yang diterima atau dikirimkan dari atau ke *session layer* dalam keadaan utuh, urut, tanpa duplikasi dan

bebas dari kesalahan. Data yang diterima dari *session layer* kemudian akan dikirimkan ke *network layer.* Lapisan ini juga akan memeriksa apakah data telah sampai di tempat tujuan dengan baik.

5. *Session Layer*

   Lapzisan ini bertugas untuk memberikan pengontrolan terhadap kerja sama antar komputer yang sedang berkomunikasi.

6. *Presentation Layer*

   Lapisan ini akan melakukan konversi agar data/informasi yang dikirimkan dimengerti oleh perngirim dan juga oleh penerima. Selain itu, dapat juga dilakukan kompresi dan enkripsi data agar keamanan data terjamin.

7. *Application*/*Proces Layer*

   Lapisan ketujuh merupakan lapisan yang bertugas untuk mengatur interaksi antara pengguna komputer dengan program aplikasi yang dipakai. Lapisan ini merupakan lapisan tertinggi pada model referensi OSI, biasanya berupa program atau aplikasi pada tingkatan layanan informasi. Sebagai tambahan untuk transfer informasi, lapisan ini menyediakan layanan-layanan seperti :

   a. Mengidentifikasi *partner* komunikasi dengan alamat atau nama.
   b. Membangun otoritas untuk komunikasi.
   c. Penanganan perbaikan kesalahan.
   d. Mengidentifikasi dalam penekanan *syntax* data.



Gambar 2.13 Susunan lapisan tujuh layer *OSI*

Dari ketujuh lapisan dari model OSI dapat dibagi ke dalam dua kategori, yaitu lapisan atas (*Application Set*) dan lapisan bawah (*Transpot Set*).

Lapisan paling atas menangani persoalan mengenai aplikasi dan pada umumnya diimplementasikan hanya pada *software*. Lapisan tertinggi (lapisan aplikasi) adalah lapisan penutup sebelum ke pengguna (*user*). Pengguna dan lapisan aplikasi saling berinterkasi proses dengan *software* aplikasi yang berisi sebuah komponen komunikasi.

Lapisan paling bawah dari model OSI mengendalikan persoalan mengenai *transport* data. Lapisan fisik dan lapisan data link diimplementasikan ke dalam hardware dan *software*. Lapisan-lapisan bawah yang lain pada umumnya hanya diimplementasikan dalam *software*. Lapisan terbawah, yaitu lapisan fisik adalah lapisan penutup bagi media jaringan fisik (misalnya jaringan kabel), dan sebagai penanggung jawab bagi penempatan informasi pada media jaringan.

Tabel berikut ini menampilkan pemisahan kedua lapisan tersebut pada lapisan-lapisan model OSI.

Tabel 2.2 Pemisahan Lapisan Atas dan Lapisan Bawah pada model OSI

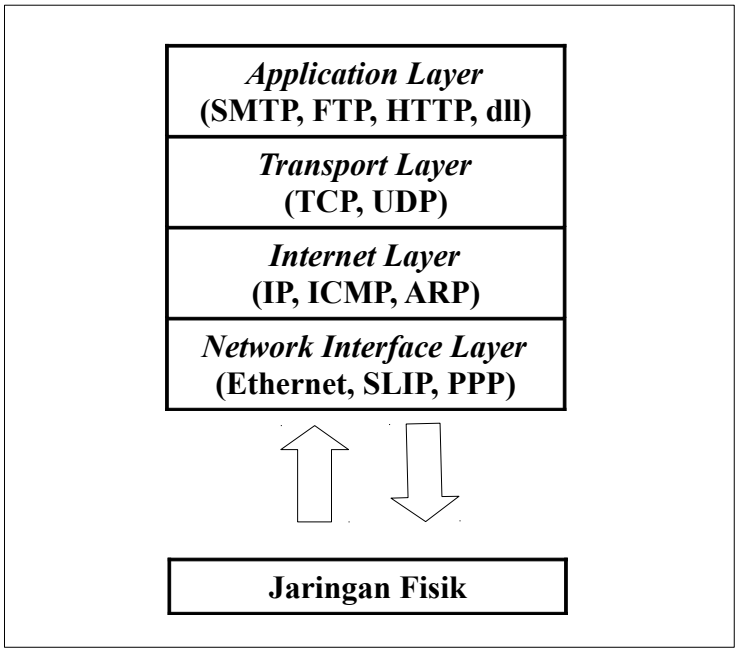
Layer	Jenis	Lapisan Atas/Bawah
Application	*Application*	Lapisan Atas
Presentation		
Session		
Transport	Data *Transport*	Lapisan Bawah
Network		
Data Link		
Physical		

2.6.3 *Transmission Control Protokol/Internet Protokol* (TCP/IP)

TCP/IP adalah protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data seperti pada LAN (*Local Area Network*) dan WAN (*Wide Area Network)*. TCP berfungsi sebagai pengontrol alur data (*Flow Control*) dan menangani pengiriman *packet* sedangkan IP berfungsi sebagai pengenal untuk setiap *host* (komputer).

Pada TCP terdapat *port*, *port* merupakan pintu masuk datagram dan data paket. *Port* data dibuat mulai dari port 0 sampai dengan port 65.536. *Port* 0 sampai dengan *port* 1024 disediakan untuk layanan standar, seperti FTP pada port 21, TELNET pada *port* 23, POP3 pada *port* 110. HTTP pada *port* 80 dan lainnya.

TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Pemodelan empat *layer* TCP/IP dapat terlihat pada Gambar 2.14.



Gambar 2.14 Layer TCP/IP

Berikut penjelasan di setiap lapisan atau *layer* pada TCP/IP :

a. *Network Interface Layer*

Network Interface Layer bertanggung jawab untuk mengirim dan menerima data dari media fisik yang dapat berupa kabel, serat optik atau gelombang radio. Karena tugasnya ini, protokol pada *layer* ini harus mampu menerjemahkan sinyal listrik menjadi data digital yang dimengerti komputer, yang berasal dari peralatan lain yang sejenis, misalnya *Ethernet*, SLIP, PPP, *repeater*, *brigde, router, hub.*

b. *Internet Layer*

Internet *Layer* bisa disebut juga i*nternetwork layer* atau *network layer*, bertanggung jawab dalam proses pengiriman ke alamat yang tepat. Internet Layera memberikan "*vitual network*" pada internet. *Internet Protocol* (IP) adalah protokol yang paling penting. IP memberikan fungsi routing pada jaringan dalam pengiriman data. Protokol lainnya antara lain : IP, ICMP, IGMP, ARP, RARP .

c. *Transport Layer*

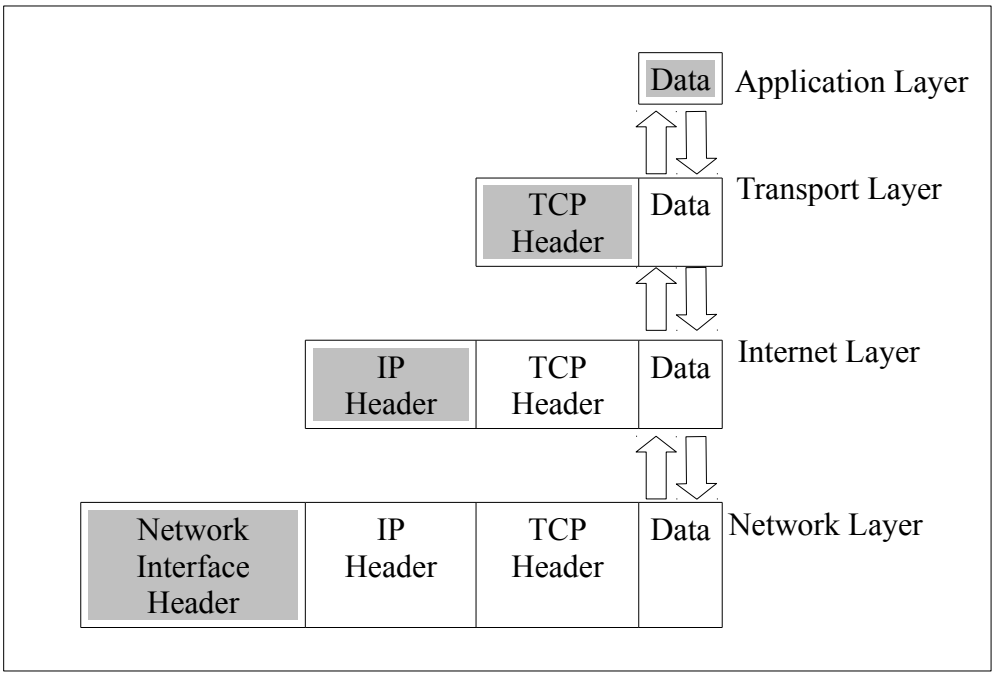
Transport *Layer* berisikan protokol-protokol yang bertanggung jawab dalam mengadakan komunikasi antar dua *host* atau komputer. Kedua protokol tersebut adalah TCP dan UDP .

d. *Application Layer*

Application Layer merupakan tempat aplikasi-aplikasi yang menggunakan TCP/IP stack berada, contohnya antara lain SMTP (*Simple Mail Transfer Protocol*) adalah suatu protokol aplikasi yang merupakan sistem pengiriman *message*/pesan atau *email*, HTTP (*HyperText Transfer Protocol*) adalah suatu protokol digunakan untuk transfer halaman *web* dan FTP (*File Transfer Protocol*) adalah layanan untuk melakukan *upload* dan *download file*.

Dalam TCP/IP, terjadi penyampaian data dari protokol yang berada di satu *layer* dengan protokol yang berada di *layer* yang lain. Setiap protokol memperlakukan semua informasi yang diterimanya dari protokol lain sebagai data. Jika suatu protokol menerima data dari protokol lain di *layer* atasnya, maka akan menambahkan informasi tambahan miliknya ke data tersebut. Informasi ini memiliki fungsi yang sesuai dengan fungsi dari protokol tersebut. Setelah itu, data ini diteruskan lagi ke protokol pada *layer* di bawahnya.

Hal yang lain juga terjadi jika suatu protokol menerima data dari protokol lain yang berada pada *layer* di bawahnya. 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* dapat dilihat pada Gambar 2.15.



Gambar 2.15 Pergerakan *data* dalam *layer* TCP/IP

2.7 *Internet Protocol* (IP)

Internet Protocol (IP) didesain untuk menghubungkan komunikasi komputer pada jaringan *packet-switched*. IP berfungsi menyampaikan paket data ke alamat yang tepat. Oleh karena itu IP memegang peranan yang sangat penting dari jaringan TCP/IP. IP merupakan protokol pada *network layer* yang bersifat :

a. ***Connectionless***, yakni setiap paket data yang dikimkan pada suatu saat akan melalui rute secara independen. Paket IP (*datagram*) akan melalui

rute yang ditentukan oleh setiap router yang dilalui oleh *datagram* tersebut. Hal ini memungkinkan keseluruhan *datagram* tiba di tempat tujuan dalam urutan yang berbeda karena menempuh yang berbeda pula.

b. ***Unreliable*** atau ketidakandalan, yakni protokol IP tidak menjamin *datagram* yang dikirim pasti sampai ke tempat tujuan. Ia hanya akan melakukan *best effort delivery* yakni melakukan usaha sebaik-baiknya agar paket yang dikirim tersebut sampai ke tujuan.

Paket-paket data dalam protokol IP dikirimkan dalam bentuk *datagram*. Sebuah *datagram* IP terdiri atas *header* IP dan muatan IP (*payload*). Penjelasan mengenai *header* IP dan muatan IP adalah sebagai berikut:

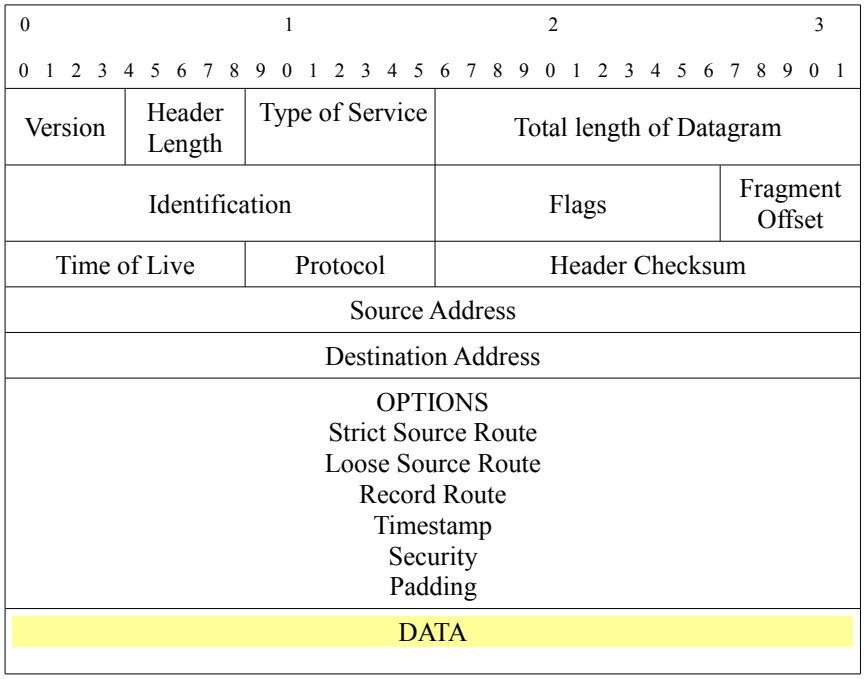
a. *Header* IP: Ukuran header IP bervariasi, yakni berukuran 20 hingga 60 byte, dalam penambahan 4-*byte. Header* IP menyediakan dukungan untuk memetakan jaringan (*routing*), identifikasi muatan IP, ukuran *header* IP dan *datagram* IP, dukungan *fragmentasi*, dan juga IP Options.
b. Muatan IP: Ukuran muatan IP juga bervariasi, yang berkisar dari 8 *byte* hingga 65515 *byte*.

Sebelum dikirimkan di dalam saluran jaringan, *datagram* IP akan dibungkus dengan *header* protokol lapisan antarmuka jaringan dan *trailer*-nya, untuk membuat sebuah *frame* jaringan. Pada saat proses transmisi data berlangsung, suatu *datagram* akan mungkin saja tidak bisa sampai dengan selamat ke tujuannya dikarenakan beberapa hal berikut, yaitu :

a. Adanya *bit error* pada saat pengiriman *datagram* pada suatu *medium.*
b. *Router* yang dilewati men-*dircard datagram* karena terjadinya *kongesti* dan kekurangan ruang memori *buffer*.

c. Putusnya *rute* ke tujuan untuk sementara waktu akibat adanya *router* yang *down*, terjadinya kekacauan *routing*, sehingga *datagram* mengalami *looping*.

Agar *datagram* IP dapat menemukan tujuannya, diperlukan informasi tambahan yang harus dicantumkan pada *header*. Struktur *header datagram* protokol IP dapat dilihat pada gambar berikut.



Gambar 2.16 Format *datagram* IP

Seperti yang terlihat pada gambar di atas, setiap paket IP membawa data yang terdiri dari beberapa komponen, yaitu :

a. Version, yaitu versi dari protokol yang dipakai.
b. *Header Length*, berisi panjang dari *header* paket IP dalam hitungan 32 *bit*.
c. *Type of Service*, berisi kualitas *service* yang dapat mempengaruhi cara penanganan paket IP.

d. *Total Length of Datagram*, panjang IP *datagram* total dalam ukuran *byte*.

e. *Identification*, *Flags*, dan *Fragment Offset*, berisi data yang berhubungan *fragmentasi* paket.

f. *Time to Live* (TTL), berisi jumlah maksimal *router* yang dilewati paket IP (*datagram*). Nilai maksimum *field* ini adalah 255. Setiap kali paket IP melewati satu *router*, isi dari *field* ini dikurangi satu. Jika TTL telah habis dan paket tetap belum sampai ke tujuan, paket ini akan dibuang dan *router* terakhir akan mengirimkan paket ICMP *time exeeded*. Hal ini dilakukan untuk mencegah paket IP terus menerus berada dalam *network*.

g. *Protocol*, mengandung angka yang mengidentifikasikan protokol *layer* atas pengguna isi data dari paket IP ini.

h. *Header Checksum*, berisi nilai checksum yang dihitung dari jumlah seluruh *field* dari *header* paket IP. Sebelum dikirimkan, protokol IP terlebih dahulu menghitung *checksum* dari *header* paket IP tersebut yang kemudian akan dihitung kembali di sisi penerima. Jika terjadi perbedaan, maka paket ini dianggap rusak dan dibuang.

i. *Source Address* dan *Destination Address*, berdasarkan penamaanya isi dari masing-masing *field* ini cukup jelas, yaitu berupa alamat pengirim dan alamat penerima dari *datagram*. Masing-masing *field* terdiri dari 32 *bit*, sesuai panjang IP *address* yang digunakan dalam internet. *Destination address* merupakan *field* yang akan dibaca oleh setiap *router* untuk menentukan tujuan paket IP tersebut akan diteruskan untuk mencapai *destination address* tersebut.

2.7.1 IP Address

IP a*ddress* merupakan kode (bilangan) *biner* 32 *bit* yang dibagi menjadi empat buah oktet berukuran 8-*bit.* Karena setiap oktet berukuran 8 *bit*, maka nilainya berkisar antara 0 hingga 255.

IP *address* yang dimiliki oleh sebuah *host* dapat dibagi dengan menggunakan *subnet mask* jaringan ke dalam dua buah bagian, yaitu :

1. *Network Identifier (Network* ID*)* atau *Network Address* (alamat jaringan) yang digunakan khusus untuk mengidentifikasikan alamat jaringan di mana *host* berada. Alamat *network* ID tidak boleh bernilai 0 atau 255. *network* ID menempati *Most Significiant Bit* (MSB, *bit-bit* paling kanan).
2. *Host Identifier*/Host ID atau *Host address* (alamat *host*) yang digunakan khusus untuk mengidentifikasikan alamat *host* (dapat berupa *workstation*, *server* atau sistem lainnya yang berbasis teknologi TCP/IP) di dalam jaringan. Nilai *host* ID tidak boleh bernilai 0 atau 255 dan harus bersifat unik di dalam n*etwork* ID jaringan di mana ia berada. *Host* ID berada pada *Least Significiant Bit* (LSB, *bit* paling kiri).

IP a*ddress* memiliki peranan penting dalam membangun jaringan komputer. Beberapa fungsi IP *address* yang digunakan dalam jaringan komputer adalah sebagai berikut :

1. Penunjuk alamat *interface* pada sebuah komputer.
2. Menentukan suatu rute jaringan yang dilalui oleh sebuah pengiriman data.
3. Pengalamatan dan meneruskan *packet* data ke tujuan.
4. Fragmentasi dan pengiriman *datagram* antar jaringan.

2.7.2 IP *Public* dan IP *Private*

IP *Public* adalah istilah untuk IP *address* yang digunakan untuk digunakan mengidentifikasi *host* di internet global, sedangkan IP *Private* adalah istilah untuk IP *address* yang hanya dapat digunakan untuk internet lokal dan tidak dapat digunakan untuk berkomunikasi dengan *host* di internet global.

2.7.3 Format IP *Address*

IP *address* terdiri dari sekelompok bilangan *biner* 32 bit yang dipisahkan oleh tanda pemisah berupa tanda titik (.) pada setiap 8 bit-nya. Tiap 8 *bit* disebut sebagai oktet, sehingga IP *address* memiliki 4 oktet pada satu buah alamat IP. Bentuk IP *address* adalah sebagai berikut :

xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

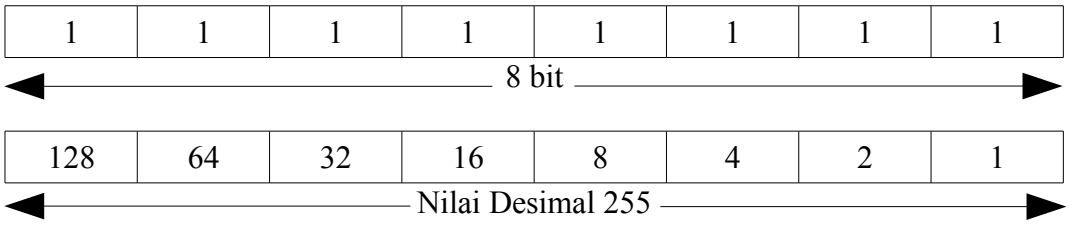
atau

WWW.XXX.YYY.ZZZ

Setiap simbol “x” dapat digantikan oleh angka 0 dan 1, karena IP *address* merupakan kumpulan bilangan *biner* (0 dan 1). Contoh format IP *address* dengan representasi kumpulan bilangan *biner* adalah sebagai berikut :

11000000.10101000.00000000.00000001

Bilangan *biner* di atas merupakan format IP *address* yang dapat ditulis dengan bilangan desimal yaitu 192.168.0.1. Gambar di bawah merupakan metode bagaimana bilangan *biner* dapat dikonversi ke bilangan desimal :



Gambar 2.17 Konversi *bit* pada sebuah oktet ke bilangan desimal

Konsep perhitungan bilangan *biner* ke bilangan desimal pada IP *address* adalah menghitung setiap *bit mask* yang bernilai satu (1) di setiap oktet. Berikut tabel yang menjelaskan bagaimana melakukan konversi perhitungan bilangan *biner* ke bilangan desimal.

Tabel 2.3 Konversi bit suatu oktet ke nilai desimal

Biner	Nilai Bit	Nilai Desimal
00000000	0+0+0+0+0+0+0+0	0
00000001	0+0+0+0+0+0+0+1	1
00000011	0+0+0+0+0+0+2+1	3
00000111	0+0+0+0+0+4+2+1	7
00001111	0+0+0+0+8+4+2+1	15
00011111	0+0+0+16+8+4+2+1	31
00111111	0+0+32+16+8+4+2+1	63
01111111	0+64+32+16+8+4+2+1	127
11111111	128+64+32+16+8+4+2+1	255
11111110	128+64+32+16+8+4+2+0	254
11111100	128+64+32+16+8+4+0+0	252
11111000	128+64+32+16+8+0+0+0	248
11110000	128+64+32+16+0+0+0+0	240
11100000	128+64+32+0+0+0+0+0	224
11000000	128+64+0+0+0+0+0+0	192
10000000	128+0+0+0+0+0+0+0	128

2.7.4 Kelas IP Address

Pada *teminologi* TCP/IP, “satu jaringan” adalah sekelompok host yang dapat berkomunikasi secara langsung tanpa *router*. *Host* dalam TCP/IP harus mempunyai sedikitnya satu IP *address*. Semua *host* TCP/IP yang menempati satu jaringan yang sama harus diberi *network* ID yang sama. *Host* yang mempunyai *network* ID berbeda dapat berkomunikasi dengan *host* lain dengan melalui *router*.

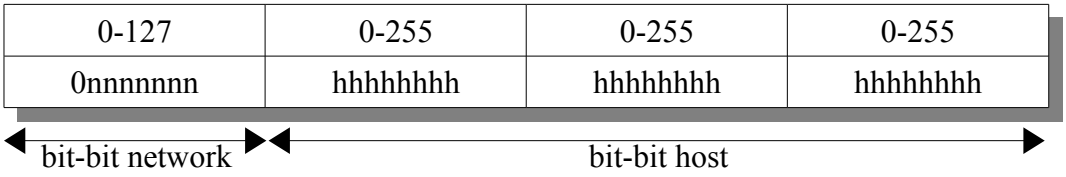
Dalam mengantisipasi besarnya jaringan komputer dan jumlah jaringan komputer, maka IP *address* dibagi atas beberapa kelas, yaitu kelas A, kelas B, kelas C, kelas D dan kelas E.

2.7.4.1 Kelas A

IP *address* kelas A diberikan untuk jaringan dengan jumlah *host* yang sangat besar. *Bit* pertama dari IP *address* kelas A selalu diset dengan nilai 0 (nol) sehingga *byte* terdepan dari IP *address* kelas A selalu bernilai antara 0 dan 127. Pada IP *address* kelas A, *network* ID berada pada 8 *bit* pertama atau pada oktet pertama, sedangkan *host* ID berada pada 24 *bit* atau pada 3 oktet terakhir. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme *Interprocess Communication* (IPC) di dalam mesin yang bersangkutan.

Karakteristik :

- Format : 0nnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh
- *Bit* pertama : 0
- *Byte* Pertama : 0-127 (*biner* 0=*0*000000;*biner* 127=*0*1111111)
- Standar : 8 *bit network* ID dan 24 *bit host* ID
- Jumlah *Network* : 126 Kelas A (0 dan 127 dicadangkan)
- IP *Host/Network* : 16.777.214 IP *address* pada setiap Kelas A
- *Range* IP : 1.xxx.xxx.xxx sampai 126.xxx.xxx.xxx



Gambar 2.18 IP *address* kelas A

2.7.4.2 Kelas B

IP *address* kelas B biasanya digunakan untuk jaringan berukuran sedang dan besar. Dua *bit* pertama dari IP *address* kelas B selalu diset dengan nilai 10 (satu nol) sehingga *byte* terdepan dari IP *address* kelas B selalu bernilai antara 128 hingga 191. Pada IP *address* kelas B, *network* ID berada pada 16 *bit* pertama, sedangkan *host* ID berada pada 16 *bit* berikutnya.

Karakteristik :

- Format : 10nnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh
- Bit awal : 10
- *Byte* Pertama : 128-191 (*biner* 128=*10*000000;*biner* 191=*10*111111)
- Standar : 16 *bit network* ID dan 16 *bit host* ID
- Jumlah *Network* : 16.384 Kelas B
- IP *Host/Network* : 65.534 IP *address* pada setiap Kelas B
- *Range* IP : 128.0.xxx.xxx sampai 191.255.xxx.xxx

128-191	0-255	0-255	0-255
10nnnnnn	nnnnnnnn	hhhhhhhh	hhhhhhhh

bit-bit network bit-bit host

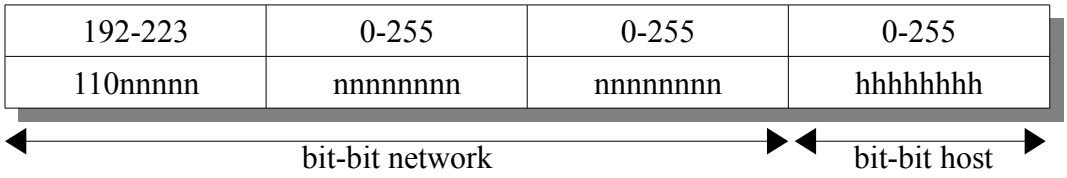
Gambar 2.19 IP *address* kelas B

2.7.4.3 Kelas C

IP *address* kelas C awalnya digunakan untuk jaringan berukuran kecil (misalnya LAN). Tiga *bit* pertama dari IP *address* kelas C selalu berisi *110*. Bersama 21 *bit* berikutnya, angka ini membentuk *network* ID 24 bit. *Host* ID terdapat pada 8 *bit* terakhir. Pada kelas C ini bisa dibentuk sekitar dua juta *network* dengan masing-masing *network* memiliki 254 IP *address*.

Karakteristik :

- Format : 110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh
- *Bit* awal : 110
- *Byte* Pertama : 192-223 (*biner* 192=*110*0000;*biner* 223=*110*11111)
- Standar : 24 bit *network* ID dan 8 bit *host* ID
- Jumlah *Network* : 2.097.152 Kelas C
- IP *Host/Network* : 254 IP *address* pada setiap Kelas C
- *Range* IP : 192.0.0.xxx sampai 223.255.255.xxx



Gambar 2.20 IP *address* kelas C

2.7.4.4 Kelas D

IP *address* kelas D digunakan untuk keperluan IP *multicasting*. Pada empat *bit* pertama diset 1110. Bit-bit berikutnya diatur sesuai keperluan *multicast group* yang menggunakan IP *address*. Dalam *multicasting* tidak dikenal *network bit* dan *host* bit.

Karakteristik :

- Format :1110nnnn.mmmmmmmm.mmmmmmmm. mmmmmmmm
- *Bit* awal : 1110
- *Byte* Pertama : 224-239 (*biner* 224=*1110*000;*biner* 239=*1110*1111)
- Bit *Multicast* : 28 *bit*
- Deskripsi : Kelas D adalah ruang alamat *multicast* (RFC 1112)

2.7.4.5 Kelas E

IP *address* kelas E adalah ruang alamat yang dicadangkan untuk keperluan eksperimental atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat *bit* pertama selalu diset dengan bilangan *biner* 1111, kemudian 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*. IP *address* kelas E tidak digunakan untuk umum.

Karakteristik :

- Format : 1111rrrr.rrrrrrrr.rrrrrrrr.rrrrrrrr
- *Bit* awal : 1111
- *Byte* Pertama : 240-247 (*biner* 240=*1111*0000;*biner* 247=*1111*0111)
- *Bit Multicast* : 28 *bit*

Dari uraian setiap kelas pada IP *address*, berikut tabel yang memperlihatkan perbandingan antara kelas-kelas pada IP *address*.

Tabel 2.4 Perbandingan kelas-kelas IP *address*

Kelas Alamat	Nilai oktet pertama	Bagian untuk Network ID	Bagian untuk Host ID	Jumlah jaringan maksimum	Jumlah host dalam satu jaringan maksimum
Kelas A	1-126	W	X.Y.Z	126	16.777.214
Kelas B	128-191	W.X	Y.Z	16.384	65.534
Kelas C	192-223	W.X.Y	Z	2.097.152	254
Kelas D	224-239	*Multicast IP Address*	*Multicast IP Address*	*Multicast IP Address*	*Multicast IP Address*
Kelas E	240-255	Dicadangkan; eksperimen	Dicadangkan; eksperimen	Dicadangkan; eksperimen	Dicadangkan; eksperimen

2.8 *Subnet Mask*

Suatu s*ubnet mask* didefinisikan dengan mengimplementasikan *masking bit (subnet mask)* kepada IP *address*. Struktur *subnet mask* sama dengan struktur IP *address* yakni terdiri dari 32 *bit* dan terbagi menjadi empat *segmen. Subnet mask* digunakan untuk memisahkan bagian IP *address* untuk membedakan *network* ID dari *host* ID dan menyatakan apakah IP *address host* tujuan terletak di jaringan lokal atau jaringan *remote* (luar).

2.8.1 Representasi *Subnet Mask*

Subnet mask memiliki cara penulisan tertentu yang umum digunakan oleh para seorang analis jaringan komputer. Ada dua metode yang dapat digunakan untuk merepresentasikan *subnet mask*, yaitu dengan Notasi Desimal Bertitik (*Dotted Decimal Notation*) dan Notasi Panjang Prefiks Jaringan (*Network Prefix Length Notation).*

a. Notasi Desimal Bertitik (*Dotted Decimal Notation*)

Sebuah *subnet mask* biasanya diekspresikan ke dalam notasi desimal bertitik (*dotted decimal notation*), seperti halnya IP *address*. Setelah semua *bit* diset sebagai bagian *network* ID dan *host* ID, hasil nilai 32 *bit* tersebut akan dikonversikan ke notasi desimal bertitik. Walaupun memiliki format yang sama dengan IP *address* yang menggunakan bilangan desimal bertitik, *subnet mask* bukan lah sebuah IP *address*. Format penulisan *subnet mask* dengan menggunakan notasi desimal bertitik adalah:

<IP *address*>**,** <*subnet mask* www.xxx.yyy.zzz>

Contoh :

192.168.0.1, 255.255.255.0

b. Notasi Panjang Prefix Jaringan (*Network Prefix Length Notation*)

Selain penulisan dengan notasi desimal bertitik, *subnet mask* dapat pula direpresentasikan dengan menggunakan *bit* yang mendefinisikan *network* ID sebagai sebuah *network prefix*. Karena menggunakan panjang dari bit *network* ID yang digunakan, maka notasi ini disebut sebagai *network prefix length* atau notasi panjang prefiks jaringan. Notasi *network prefix length* juga dikenal dengan istilah *Classless Inter-Domain Routing* (CIDR).

Format penulisan *subnet mask* dengan menggunakan notasi panjang prefiks jaringan adalah:

<IP address>/<jumlah bit yang digunakan sebagai *network* ID>

Contoh :

138.23.0.0/24

2.8.2 *Subnet Mask Default*

Subnet mask default dibuat berdasarkan kelas-kelas IP *address* dan digunakan di dalam jaringan TCP/IP yang tidak dibagi ke dalam beberapa *subnet*. Tabel di bawah ini menyebutkan beberapa *subnet mask default* dengan menggunakan notasi desimal bertitik dan notasi panjang prefix.

Tabel 2.5 *Subnet Mask default* untuk IP address kelas A, B, dan C

Kelas IP address	Bit yang digunakan untuk Subnet Mask	Notasi Desimal Bertitik	Notasi Panjang Prefix
Kelas A	11111111.00000000.00000000.00000000	255.0.0.0	<IP>/8
Kelas B	11111111.11111111.00000000.00000000	255.255.0.0	<IP>/16
Kelas C	11111111.11111111.11111111.00000000	255.255.255.0	<IP>/24

Subnet mask dapat menentukan alamat *network* ID suatu IP *addres*. Untuk menentukan *network* ID dari sebuah IP *address* dengan menggunakan sebuah *subnet mask* tertentu, dapat dilakukan dengan menggunakan sebuah operasi matematika, yaitu dengan menggunakan operasi logika perbandingan AND (*AND comparison*). Di dalam sebuah *AND comparison*, nilai dari dua hal yang diperbandingkan akan bernilai *true* hanya ketika dua item tersebut bernilai *true* dan menjadi *false* jika salah satunya *false*. Dengan mengaplikasikan prinsip ini ke dalam bit-bit, nilai 1 akan didapat jika kedua bit yang diperbandingkan bernilai 1, dan nilai 0 jika ada salah satu di antara nilai yang diperbandingkan bernilai 0.

Cara ini akan melakukan sebuah operasi logika AND comparison dengan menggunakan 32 bit alamat IP dan dengan 32 *bit subnet mask*, yang dikenal dengan operasi *bitwise logical AND comparison*. Hasil dari operasi *bitwise* alamat IP dengan *subnet mask* itulah yang disebut dengan *network* ID. Tabel di bawah merupakan operasi logika AND dan contoh operasi *bitwise* untuk menentukan *network* ID dari IP *address* dan *subnet mask default* kelas B.

Tabel 2.6 Operasi Logika AND

AND	**0**	**1**
0	0	0
1	0	1

Contoh perhitungan operasi *bitwise* dapat dilihat pada tabel 2.6. Bit *IP adress* 140.179.240.200 dengan bit sub*net mask* 255.255.000.000 setelah dilakukan operasi *bitwise* akan menghasilkan susunan bit yaitu 10001100.10110011.00000000.00000000, yang disebut sebagai *network ID*.

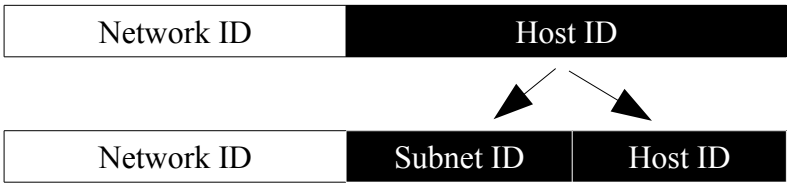
Tabel 2.7 Operasi *bitwise* pada IP *address* dan *Subnet mask* kelas B

Keterangan	Bit (biner)	Bit (desimal)
IP Address kelas B	10001100.10110011.11110000.11001000	140.179.240.200
Default mask kelas B	11111111.11111111.00000000.00000000	255.255.000.000
Network ID	10001100.10110011.00000000.00000000	140.179.000.000

2.8.3 *Subnetting*

Jumlah dari IP *address* yang ada sangatlah terbatas, apalagi jika harus memberikan alamat semua *host* di Internet. Oleh karena itu, perlu dilakukan efisiensi dalam penggunaan IP *address* supaya dapat mengalamati semaksimal mungkin *host* yang ada dalam satu jaringan.

Membagi jaringan menjadi *subnet* memerlukan *setup segmen* menggunakan *network* ID atau *subnet* ID yang berbeda. *Subnet* ID yang unik dibuat untuk setiap segmen dengan membagi bit di *host* ID menjadi dua bagian. Satu bagian digunakan untuk mengidentifikasikan segmen sebagai jaringan yang unik dan bagian lain digunakan untuk mengidentifikasi *host*. Hal ini lah yang disebut sebagai *subnetting* atau *subnetworking*.



Gambar 2.21 *Subnet* ID dan *Host* ID

Contoh pertama, pada sebuah kasus akan dilakukan *subnetting* terhadap sebuah jaringan lokal. Jaringan tersebut memiliki *network* ID 140.150.0.0 dan *subnet* 255.255.192.0.

Network ID	140.150.0.0
Subnet	140.150.0.0

Gambar 2.22 Contoh kasus 1 *subnetting*

Dengan melihat gambar 2.21 yang merupakan contoh kasus, dapat disimpulkan bahwa :

a. IP *address* yang digunakan adalah kelas B.
b. Rumus yang digunakan untuk menghitung besaran kelipatan setiap *subnet* yang akan terbentuk adalah 256 dikurangi angka oktet ketiga pada *subnet*.

256-192 = 64

Dari rumus di atas didapat kelompok *subnet* yang dapat digunakan dalam *network* ID yaitu kelipatan dari angka 64, yaitu 64 dan 128. Oleh karena itu, *subnet* yang terbentuk adalah :

140.150.64.0 dan 140.150.128.0

Setelah mendapat kumpulan *subnet* terbaru, maka selanjutnya akan didapat kelompok sebaran IP *address* yang dapat digunakan yaitu :

a. Kelompok *subnet* pertama : 140.150.64.1 sampai dengan 140.150.64.254
b. Kelompok *subnet* kedua : 140.150.128.1 sampai dengan 140.150.191.254

Contoh kedua, dengan *network* ID 140.200.0.0 dan *subnet* 255.255.224.0 dan dengan cara yang sama seperti pada kasus pertama, maka didapat kelipatan *subnet* yaitu :

256-224 = 32

Jadi, kelompok *subnet*-nya adalah kelipatan dari 32, yaitu 32, 64, 96, 128, 160, dan 192. Dari hasil angka kelipatan *subnet* yang didapat tersebut, maka sebaran kelompok IP *address* yang dapat dilihat pada tabel di bawah ini :

Tabel 2.8 Sebaran IP *address* pada *network* ID 140.200.0.0/255.255.224.0

Kelompok	Kelipatan	Sebaran IP Adrress
1	32	140.200.32.1 - 140.200.63.254
2	54	140.200.64.1 - 140.200.95.254
3	96	140.200.96.1 - 140.200.127.254
4	128	140.200.128.1 - 140.200.159.254
5	160	140.200.160.1 - 140.200.191.254
6	192	140.200.192.1 - 140.200.223.254

Selain dari kedua contoh kasus perhitungan *subnetting* di atas, dapat pula dihitung menggunakan cara yang lain yaitu dengan menggunakan rumus. Adapun rumus yang digunakan adalah :

1. Menghitung jumlah subnet dengan rumus = 2^n-2

   Variabel n adalah *bit mask* atau banyaknya angka *biner* satu (1) pada oktet terakhir dari *subnet*. Untuk kelas A adalah 3 oktet terakhir, dan untuk kelas B adalah 2 oktet terakhir.

2. Menghitung jumlah host per subnet dengan rumus = 2^N-2

   Variabel N adalah *bit mask* atau banyaknya angka *biner* nol (0) pada oktet terakhir *subnet*.

Untuk lebih jelas, dapat dilihat contoh dengan *subnet* yang terlihat pada tabel di bawah untuk melakukan perhitungan *subnet* berikut dengan menggunakan rumus.

Tabel 2.9 Contoh *subnet* untuk perhitungan jumlah *host* dan *subnet* dengan menggunakan rumus

Subnet	255	255	224	0
Biner	11111111	11111111	11100000	00000000
Oktet ke-	oktet 1	oktet 2	oktet 3	oktet 4

Dari nilai *subnet* di atas maka didapat jumlah *subnet* dan jumlah *host* per *subnet* dengan menggunakan rumus, yaitu :

1. Menghitung jumlah *subnet*

Dari contoh subnet pada tabel di atas maka didapat nilai n = 3, n merupakan banyaknya angka biner 1 pada 3 oktet terakhir (kelas B). Sehingga, rumus untuk menghitung jumlah subnet adalah :

$$2^n\text{-}2 = 2^3\text{-}2 = 6$$

Dengan demikian akan didapat 6 subnet pada jaringan yang menggunakan subnet 255.252.224.0.

2. Menghitung jumlah *host* per *subnet*

Dari contoh *subnet* pada tabel diatas maka didapat nilai N = 13, n merupakan banyaknya angka biner 0 pada oktet terakhir (kelas B). Sehingga, rumus untuk menghitung jumlah *host* per *subnet* adalah :

$$2^n\text{-}2 = 2^{13}\text{-}2 = 8192\text{-}2\text{=}8190$$

Dengan demikian akan didapat 8190 buah jumlah *host* per *subnet* dan 8190x6=49140 buah jumlah seluruh *host* pada jaringan yang menggunakan *subnet* 255.252.224.0.

2.9 *Dynamic Host Configuration Protocol* (DHCP)

DHCP merupakan suatu protokol yang mengatur mengenai pemberian alamat IP, *subnet mask*, *default router*, dan beberapa paramater lain pada komputer client. DHCP berfungsi untuk memberikan IP address secara otomatis pada komputer yang menggunakan protokol TCP/IP. DHCP bekerja dengan relasi *client*/*server* dimana DHCP *server* menyediakan suatu kelompok IP address yang dapat diberikan pada DHCP *client*. Dalam memberikan IP address ini, DHCP hanya meminjamkan IP *address* tersebut. Jadi pemberian IP address ini berlangsung secara dinamis.

Contoh penggunaannya dapat dilihat dalam jasa layananan penyedia akses internet broadband oleh *Internet Service Provider* (ISP) dan *wi-fi*/*hotspot.* Saat melakukan koneksi ke ISP atau *zona wi-fi*, setiap pelanggan akan dipinjamkan IP address unik secara otomatis oleh server. Dengan terdaftar dan memiliki nomor IP address, maka komputer client tersebut akan dapat terhubung pada jaringan lokal yang memberikan akses layanan internet seperti *browsing* dan lain sebagainya.

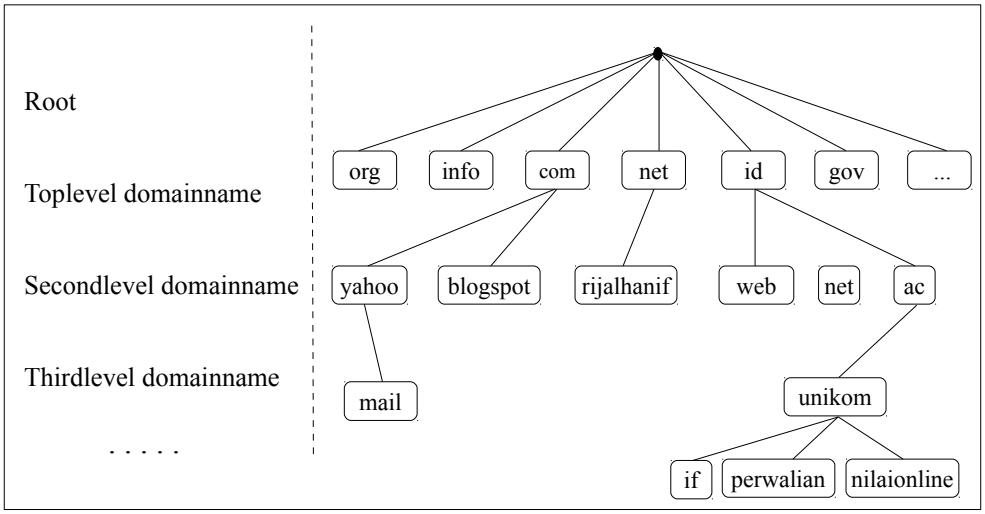
2.10 *Domain Name System* (DNS)

Domain Name System atau biasa disebut sebagai DNS, adalah suatu teknik untuk mengingat IP *address* yang sulit diingat akibat terdiri dari sederetan angka. *Routing* paket IP yang berbasis TCP/IP sebenarnya tidak memerlukan teknik DNS tersebut, cukup dengan IP *address*. Teknik DNS diperlukan karena yang

melakukan *routing* tidak lain adalah manusia, dan manusia pada umumnya lebih sulit menghafal sederetan angka. Untuk itu, perlu cara lain agar manusia mudah menghafalnya atau mengingatnya.

Cara yang ditempuh untuk mengatasi hal tersebut yaitu dengan melakukan pemetaan IP *address* menjadi *hostname*. *Hostname* atau nama *host*, seperti yahoo.com, gmail.com, facebook.com, ternyata lebih mudah dihapalkan dari pada angka-angka. Jadi, apabila seorang pengguna hendak mengakses *server web*, dia cukup menuliskan alamat situsnya saja, misal www.yahoo.com, tidak perlu mengetikkan IP *address*-nya.

DNS menggunakan prinsip penamaan hostname yang disebut nama *domain* atau *domain name*. Struktur DNS terbentuk seperti pohon terbalik (*tree*), bagian atas disebut *root* atau akar kemudian di bagian bawah *root* ada *toplevel domainname*, *second domainname*, dan seterusnya. Ilustrasinya dapat dilihat pada gambar berikut.



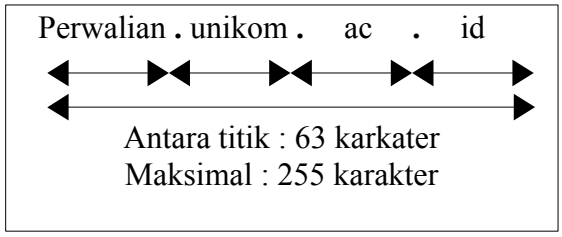
Gambar 2.23 Ilustrasi hirarki *domainname*

Dari gambar di atas dapat dilihat beberapa *domain name*, yaitu : yahoo.com, blogspot.com, unikom.ac.id, if.unikom.ac.id, perwalian.unikom.ac.id,

dan nilaionline.unikom.ac.id. *Toplevel domainname* berasal dari kode negara atau organisasi. Indonesia memiliki kode negara *id* dan instansi yang bersifat pendidikan seperti sekolah dan kampus memiliki kode *ac* (*academy*). Sehingga sebuah domain *unikom.ac.id* merupakan domain yang bergerak di bidang pendidikan / kampus bernama unikom yang berada di Indonesia.

Domain unikom.ac.id juga memiliki DNS *server* yang bertanggung jawab atas domain-domain di bawahnya. Domain tersebut dapat digunakan pada batasan area jurusan, layanan *online*, dan unit kegiatan mahasiswa yang ada di dalamnya. Contoh *domain* tersebut adalah if.unikom.ac.id dan perwalian.unikoma.ac.id

Jumlah karakter maksimal yang boleh digunakan sebuah *domain name* adalah 255 karakter (sudah termasuk karakter titik). Sedangkan jumlah karakter maksimal yang boleh digunakan di antara titik yaitu 63 karakter. Perhatikan contoh domainname *kampus.unikom.ac.id* berikut.



Gambar 2.24 Jumlah maksimum *domainname*

Saat ini ada 13 *server* DNS induk yang disebut *Root* NS. *Root* NS ini sebagian besar ada di Amerika Serikat. Selain *Root* NS tentu saja masih banyak *server-server* DNS yang tersebar di *domain-domain*. Sebagai contoh, domain unikom.ac.id memiliki sebuah *server* DNS yang khusus digunakan untuk mengelola *hostname* pada *domain*-nya saja.

Berikut daftar Root NS yang bertanggung jawab pada penggunaan nama *domain* yang ada di internet.

Tabel 2.10 Daftar Root NS

NS	IP address
A.ROOT-SERVER.NET	198.41.0.4
B.ROOT-SERVER.NET	192.228.79.201
C.ROOT-SERVER.NET	192.33.4.12
D.ROOT-SERVER.NET	128.8.10.90
E.ROOT-SERVER.NET	192.203.230.10
F.ROOT-SERVER.NET	192.5.5.241
G.ROOT-SERVER.NET	192.112.36.4
H.ROOT-SERVER.NET	128.63.2.53
I.ROOT-SERVER.NET	192.36.148.17
J.ROOT-SERVER.NET	192.58.128.30
K.ROOT-SERVER.NET	193.0.14.129
L.ROOT-SERVER.NET	198.32.64.12
M.ROOT-SERVER.NET	202.12.27.33

2.11 PABX (*Private Automatic Branch eXchange*)

PABX adalah suatu perangkat yang berfungsi sebagai sentral telepon, dalam suatu lokasi tertentu, misalnya : kantor, gedung, perumahan, dan lain-lain. Dalam skala kapasitas yang lebih besar, PABX dapat berupa Sentral Telepon Otomatis PSTN (*Public Switched Telephone Network*) yang digunakan oleh operator telepon besar untuk layanan ke rumah, kantor dan lain-lain, misalnya PT. Telkom, PT. Indosat, PT. Telkomsel, PT. Bakri dan lain-lain.

Perangkat PBAX akan mengatur panggilan yang masuk serta meneruskan panggilan ke nomor tujuannya, sehingga pengguna dapat dengan mudah melakukan panggilan ke nomor tujuan, cukup dengan menekan nomor tujuan nya.

Nomor tersebut merupakan nomor pelanggan yang digunakan untuk *men-dial* yang juga sering dipakai pada telepon rumah atau telepon seluler. Nomor ini disebut juga nomor *extension*.

Private Branch Exchange menggunakan teknologi telepon *analog* pada awalnya, tetapi sekarang PBX telah menggunakan teknologi telepon *digital* (sinyal digital diubah ke sinyal *analog*) untuk panggilan keluar pada *local loop* dengan menggunakan *Plain Old Telephone Service* (POTS). Tujuan utama dari penggunaan PBX adalah untuk menghemat biaya yang dibutuhkan untuk menarik kabel dari setiap pengguna ke *central office* (CO) perusahaan telepon.

Sebuah PBX kadang-kadang disebut *phone switch*, yaitu peralatan yang dapat menghubungkan antara telepon kantor dengan jaringan telepon umum (PSTN). PBX biasanya dimiliki dan dioperasikan suatu perusahaan atau organisasi dan bukan perusahaan telepon. Gambar 2.25 merupakan contoh jenis PABX merk Siemens HiPath 3800 yang memiliki kapasitas 48 line PSTN dan mampu membuat 500 extension dalam gedung.



Gambar 2.25 Contoh PABX Siemens HiPath 3800

Dalam PABX sedikitnya memiliki 2 bagian terpenting dalam membangun sentral komunikasi yaitu *Central Procesing Unit* (CPU) dan *Line Trunk*.

1. ***Central Procesing Unit* (CPU)**

Berfungsi sebagai pusat pengendali sistem dan mengontrol kerja sistem. Dalam CPU ini memiliki 8 bagian dasar pembentuk PABX seperti yang terlihat pada tabel di bawah ini.

Tabel 2.11 Komponen CPU PABX

No	Bagian	Keterangan
1	Interface RS-232	Digunakan untuk hubungan ke *Maintenance Operating Console* (MOC)
2	Peralatan Memori	Terdiri dari ROM dan RAM
3	*Time Division Switch* (TDSW)	Melakukan proses penyambungan kanal bicara antar pelanggan
4	*Digital Tone Generator* (DTG)	Pembangkitan sinyal
5	*Conference Trunk* (CFT)	Pembicaraan 3 pelanggan (*conference*)
6	*Public Branch Register* (PBR)	Register yang berisi nomor masing-masing pelanggan yang telah diprogram
7	*Public Branch Sender* (PBSD)	Menyimpan sementara nomor pelanggan pemanggil, sebelum terhubung ke tujuan
8	*Microprocessor* (MP)	Mengatur kerja masing-masing peralatam yang terdapat dalam CPU

2. *Line Trunk*

Line Trunk berfungsi mengatur line-line yang dapat digunakan untuk penyambungan serta mengontrol trunk dan menghitung dial pulsa. *Line Trunk* terdiri dari *Line Circuit* (LC), *Central Office Trunk* (COT) dan *Firmware Processor* (FP).

Tabel 2.12 Komponen *Line Trunk*

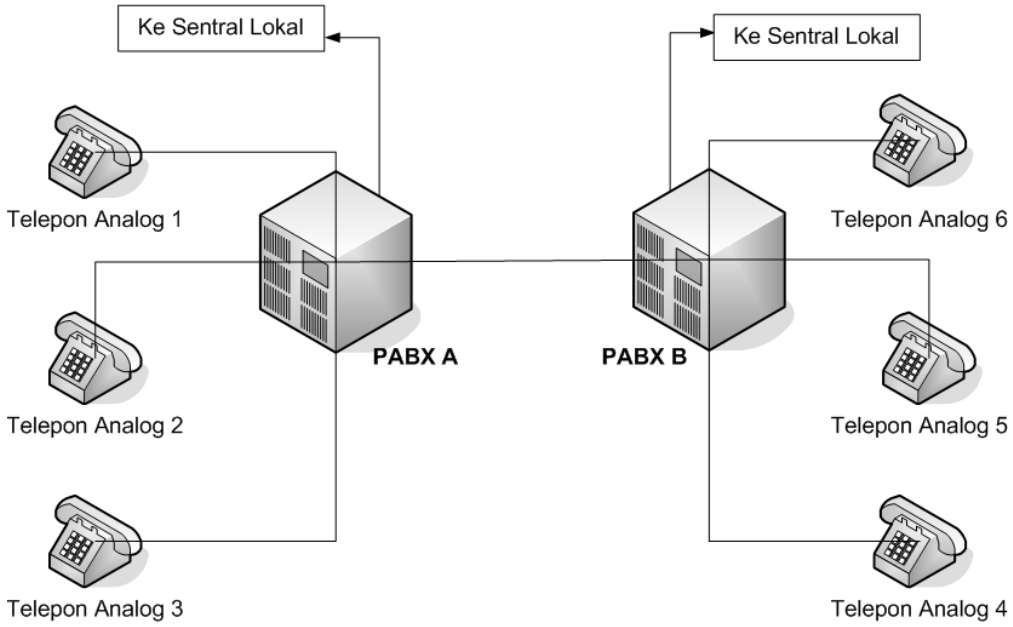
No	Bagian	Keterangan
1	Line Circuit (LC)	- penghubung antar sentral dengan pelanggan, setiap pelanggan memerlukan sebuah LC - mencatu arul loop DC
2	Central Office Trunk (COT)	Penghubung
3	Firmware Processor (FP)	Melakukan proses penyambungan kanal bicara antar pelanggan

Pada umunya sebuah PBX terdiri dari :

a. Saluran trunk telepon (*multiple phone*) yang berakhir pada PBX .
b. Sebuah komputer atau perangkat dengan *memory* yang mengatur switching panggilan keluar dan ke dalam pada PBX .
c. Jaringan saluran di dalam PBX .
d. Biasanya sebuah *console* atau *switchboard* untuk seorang operator .

Pemasangan PABX biasanya tersimpan dalam ruang khusus agar mempermudah dalam melakukan *maintenance* dan tentu dengan pertimbangan privasi keamanan.

Dua PABX atau lebih dapat dihubungkan satu sama lain dengan sistem *junction.* Pada Gambar 2.26 merupakan contoh topologi PABX yang dapat dihubungkan dengan PABX lainnya.



Gambar 2.26 Contoh Jaringan PABX

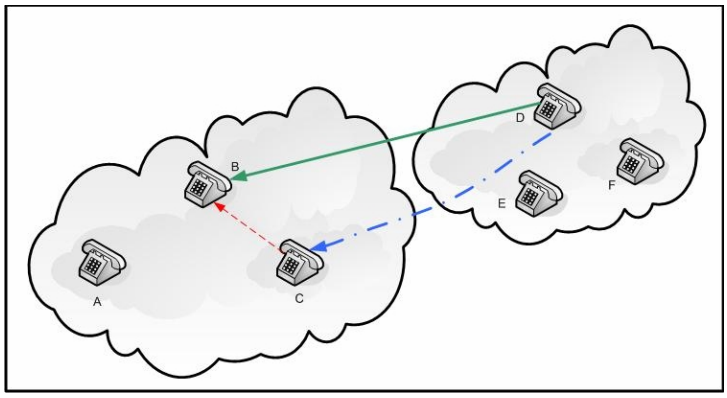
Cara kerja PABX dapat dijelaskan pada uraian berikut :

a. Ketika pelanggan pemanggil *off hook*, secara otomatis kita mengirim sinyal ke PABX yang dimengerti oleh PABX (dial tone).
b. Kemudian saat men-*dial* digit, PABX mengetahui apakah ini merupakan panggilan *internal* atau *eksternal*.
c. Proses routing dimulai. Jika *internal*, maka akan dikirmkan ke PABX tidak menggunakan "*trunk*" pada sisi luar. Namun jika *eksternal* dimulai dengan mencari nomor-nomor yang kita *dial* lalu mengrim informasi ke *Central Office* atau PABX lain.
d. Pada beberapa kasus, panggilan *eksternal* hanya dapat dilakukan dengan memasukkan kode (*password*) tertentu sebelum men-*dial* nomor *eksternal.*

PABX memiliki beberapa fitur untuk mendukung teknologi komunikasi yang baik. Fitur PABX tersebut diantaranya adalah sebagai berikut :

1. *Call Pickup*

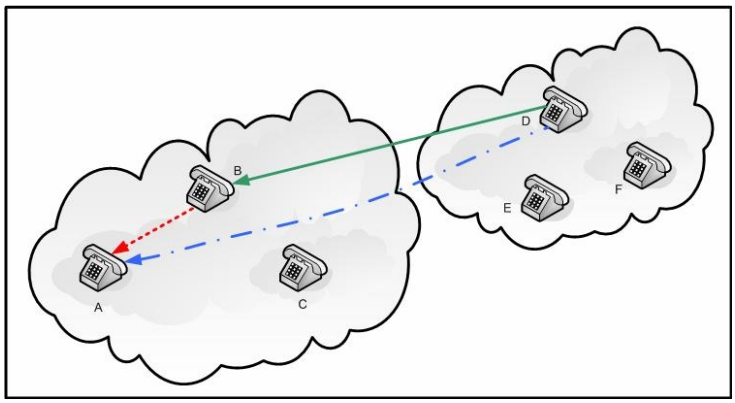
   Layanan ini disediakan untuk pengambilan nomor ekstensi lain oleh ekstensi yang terdekat (dalam satu grup), jika pengguna ekstensi lain tersebut tidak berada di tempat.



Gambar 2.27 *Call Pickup*

2. *Call Forward* / *Divert*

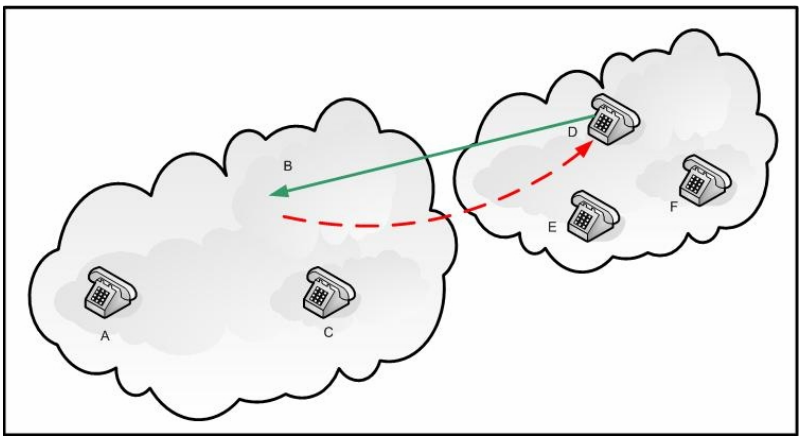
   Layanan ini meneruskan panggilan yang masuk jika si penerima tidak berada di tempat. Proses panggilan diteruskan ke ekstensi lain yang berada di satu jaringan.



Gambar 2.28 *Call Forward*

3. *Call Back*

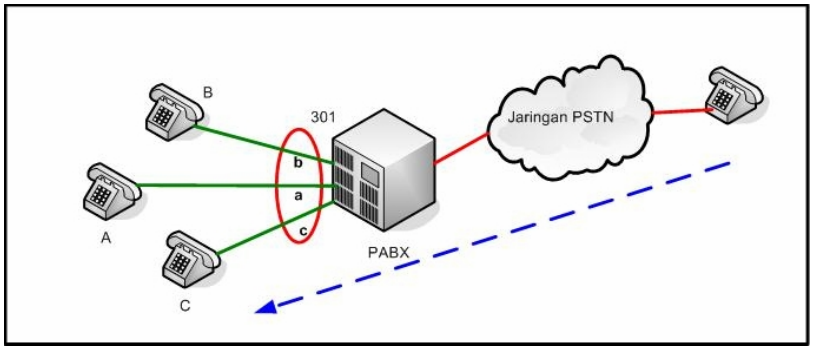
Saat ada panggilan ke sebuah nomor esktensi tertentu, kebetulan nomor ekstensi tersebut sedang berbicara, maka pemanggil hanya perlu menekan kode tertentu, kemudian *on hook*. Saat nomor yang dituju selesai bicara atau *on hook*, pemanggil mendengar nada panggil. Jika pemanggil *off hook*, lagsung tersambung ke tujuan (tanpa tekan nomor lagi).



Gambar 2.29 *Call Back*

4. *Hunting*

Proses pemberian sebuah nomor ekstensi kepada beberapa jalur pemakai. Pemberian sebuah nomor ini dimaksudkan untu memudahkan mengingat bagi pemakai.



Gambar 2.30 Hunting

PABX yang beredar di pasaran banyak jenisnya dengan kemampuan yang berbeda-beda. Berdasarkan sistem kerjanya PABX dapat dikelompokkan menjadi tiga, yaitu:

1. PABX *Analog*

PABX *analog* mempunyai ciri khas pada bagian *switching-nya*, yaitu menggunakan *electronic modular switch* dengan cara kerja *space division* dan memanfaatkan komponen IC *crosspoint* sebagai media penghubungnya.

2. PABX *Digital*

PABX *digital* identik dengan penggunaan *digital switching sistem time division* pada bagian penghubungnya, sehingga semua layanan komunikasi dilakukan oleh komponen *switching* yang digunakan oleh komponen IC *crosspoint.*

3. IP PABX

PABX ini merupakan pengembangan dari generasi sebelumnya, dengan penambahan pada bagian tertentu, terutama fitur *internet protocol* (IP). Dengan fitur tersebut, perangkat ini memiliki kemampuan yang sangat lengkap dibanding pendahulunya. Fitur IP tersebut terbagi menjadi dua, yaitu :

a. IP PABX / IP PBX

Untuk kebutuhan ini biasanya dibutuhkan sebuah modem berupa modem berupa IP *gateway card*, kemudian bagian ini disambungkan ke *Hub* atau *switch*, selnjutnya dihubungkan ke *router*. Kemudian dari *router* akan tersambung dengan *network connection* atau jaringan internet.

b. IP EXT

Aplikasi ini sangat tepat digunakan sebuah perusahaan untuk mengontrol cabangnya, di mana pada kantor cabang hanya terdapat sebuah *IP phone* berupa *Internet Protocol Proprietary Telephone* (IP IPT) yang langsung terkoneksi dengan jaringan internet.

2.12 IP PBX (*Internet Protocol Private Branch Exchange*)

IP PBX adalah PBX yang menggunakan teknologi IP. IP PBX adalah perangkat *switching* komunikasi telepon dan data berbasis teknologi *Internet Protocol* (IP) yang mengendalikan ekstensi telepon *analog* maupun ekstensi *IP Phone*. Fungsi yang dapat dilakukan dengan IP PBX antara lain: penyambungan, pengendalian, dan pemutusan hubungan telepon; translasi protokol komunikasi; translasi media komunikasi atau *transcoding*; serta pengendalian perangkat-perangkat pada *IP Telephony* seperti *VoIP Gateway*, *Access Gateway*, dan *Trunk Gateway*.

Solusi berbasis IP PBX merupakan konsep jaringan komunikasi generasi masa depan atau dikenal dengan istilah NGN (*Next Generation Network*) yang dapat mengintegrasikan jaringan telepon konvensional (PSTN/POTS), jaringan telepon bergerak (GSM/CDMA), jaringan telepon satelit, jaringan *Cordless* (DECT), dan jaringan berbasis paket (IP/ATM).

IP PBX awalnya hanyalah berupa seperangkat *hardware* saja, seiring dengan perkembangan teknologi saat ini, IP PBX dapat pula dibangun dengan melakukan instalasi aplikasi IP PBX pada *hardware* yang tepat.

Dalam IP PBX dikenal tiga komponen penting yang membentuk fungsi PBX, yaitu *extension*, *trunk*, dan *dial plan*.

a. *Extension*

   Extension adalah komponen yang menangani registrasi dari pengguna. Dalam implementasinya, *extension* akan digunakan berupa nomor-nomor telepon untuk masing-masing *user*.

b. *Trunk*

   Trunk adalah komponen yang menangani registrasi antara satu *server* IP PBX ke *server* IP PBX lainnya sehingga sebuah kelompok *server* IP PBX dapat saling berkomunikasi dan setiap user dapat memanggil user lain yang berada di lokasi *server* IP PBX yang lain.

c. *Dial Plan*

   Dial plan adalah komponen yang mengatur penomoran dan *call routing*. Dalam *dial plan* dilakukan perancangan bagaimana aturan *extension* dan *trunk* dilakukan.

2.13 *Voice over Internet Protocol* (VoIP)

Voice over Internet Protocol (disingkat VoIP) adalah teknologi yang memungkinkan percakapan suara jarak jauh melalui media internet. VoIP sering disebut juga dengan *IP Telephony, Internet Telephony* atau *Digital Phone*. VoIP memungkinkan seseorang untuk saling berkomunikasi (berbicara) melalui internet dengan biaya yang ekonomis jika dibandingkan dengan media telepon biasa.

2.13.1 Cara Kerja VoIP

Konsep cara kerja VoIP yaitu dengan melakukan pengiriman sebuah sinyal secara *digital.* Sebelum proses transmisi (pengiriman) dilakukan, data yang berupa sinyal *analog* akan dikonversikan terlebih dahulu dengan ADC (A*nalog to Digital Converter*) menjadi bentuk data digital. Setelah proses konversi dilakukan data digital akan ditransmisikan ke sumber tujuan. Setelah sampai, data sinyal digital tersebut akan dikonversi kembali menjadi data sinyal analog dengan DAC (D*igital to Analog Converter*) sehingga dapat diterima oleh sumber tujuan sesuai dengan data sinyal yang ditransmisikan.



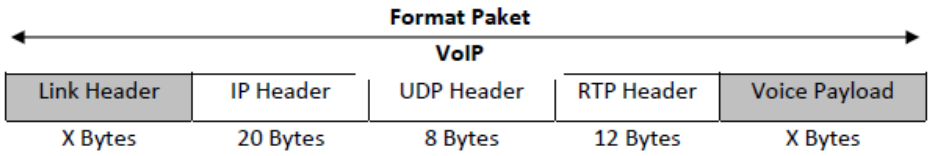
Gambar 2.31 Cara kerja VoIP

2.13.2 Format Paket VoIP

Tiap paket VoIP terdiri atas dua bagian, yakni *header* dan *payload* (beban). *Header* terdiri atas IP *header*, *Real-time Transport Protocol* (RTP) *header*, dan *User Datagram Protocol* (UDP) *header.*

IP header bertugas menyimpan informasi *routing* untuk mengirimkan paket-paket ke tujuan. Pada tiap *header* IP disertakan tipe layanan atau *type of service* (ToS) yang memungkinkan paket tertentu seperti paket suara diperlakukan berbeda dengan paket yang *non real time*.

UDP *header* memiliki ciri tertentu yaitu tidak menjamin paket akan mencapai tujuan sehingga UDP cocok digunakan pada aplikasi *real time* yang sangat peka terhadap *delay.* RTP header adalah *header* yang dapat dimanfaatkan untuk melakukan *framing* dan *segmentasi* data *real time*. Seperti UDP, RTP juga mendukung realibilitas paket untuk sampai di tujuan. RTP menggunakan protokol kendali yang mengendalikan RTCP (r*eal-time transport control protocol*) yang mengendalikan QoS dan sinkronisasi media *stream* yang berbeda.



Gambar 2.32 Format paket *VoIP*

2.13.3 Komponen VoIP

VoIP memiliki empat komponen utama, yaitu *User Agent*, *Proxy*, *Protokol*, dan *Codec*. Berikut penjelasan mengenai masing-masing komponen dalam membangun jaringan VoIP.

1. **User Agent**

User agent merupakan komponen yang digunakan oleh pengguna untuk memulai dan menerima sesi komunikasi. Dalam VoIP, *user agent* berupa komponen yang melakukan dial nomor telepon VoIP atau menerimanya.

User agent dapat berupa *software* atau biasa disebut dengan *softphone. Softphone* merupakan *user agent* yang paling poluler, hal ini dikarenakan banyak *softphone* dapat diperoleh secara gratis dan dapat langsung diunduh pada masing-masing *website* penyedia *softphone*.

Contoh *user agent* dengan jenis *softphone* adalah *Sjphone*, *X-Lite*, *QuteCom*, *Ekiga*, *ZoIPer*, *NetMeeting*, *VoIP Rakyat Communicator* dan masih banyak yang lainnya. Pada dasarnya semua fungsi *softphone* hampir sama, yaitu melakukan panggilan dan menerima panggilan serta memutuskan panggilan, layaknya melakukan sebuah percakapan dengan telepon biasa. *Softphone* harus terinstal pada komputer dan memerlukan sebuah sebuah *microphone* dan *speaker* sebagai alat tambahan dalam melakukan komunikasi. Gambar di bawah merupakan tampilan *X-Lite* yang digunakan sebagai *user agent*.



Gambar 2.33 X-Lite softphone yang digunakan sebagai user agent

Jenis *user agent* yang berupa *hardware* disebut *hardphone*. *Hardphone* saat ini memiliki ragam dan kemampuan serta fitur yang berbeda. Beberapa *hardphone* yang dapat digunakan sebagai *user agent* dalam jaringan VoIP adalah :

a. *IP-Phone*, bentuknya mirip telepon biasa pada umumnya. Cara penyambungannya dengan mengkoneksikan ke jaringan komputer berbasis TCP/IP, dapat juga melalui *switch* pada jaringan komputer yang telah memiliki fasilitas VoIP.



Gambar 2.34 IP Phone

b. *USB-Phone*, bentuknya seperti telepon *seluler*. Koneksi yang digunakan melalui USB *port* dari komputer. Umumnya digunakan berdampingan dengan softphone, oleh karena itu biasanya beberapa *softphone* memiliki driver tersendiri agar mengenali perangkat *USB-Phone* ini.



Gambar 2.35 USB-Phone

c. *Internet Telephony Gateway* (ITG), adalah *user agent* VoIP yang memiliki dua jenis *port*, yaitu *port* FXS yang terhubung ke telepon biasa dan FXO yang terhubung ke PSTN langsung atau bisa juga melalui PABX.



Gambar 2.36 *Internet Telephony Gateway* (ITG)

d. *Analog Telephony Adapter* (ATA), ini merupakan alat yang digunakan agar telepon rumah dapat digunakan menjadi *user agent* VoIP. ATA sama dengan ITG namun hanya memiliki satu *port* saja, yaitu FXS.



Gambar 2.37 Analog Telephony Adapter (ATA)

2. *Proxy*

Proxy dalam teknologi VoIP, sedikit berbeda dengan *proxy server* internet yang ada dalam sebuah jaringan komputer. *Proxy* yang dimaksud dalam teknologi VoIP merupakan aplikasi *server* yang mengatur jaringan VoIP. *Proxy* merupakan komponen yang menerima registrasi *user agent* dan bertugas mengatur penomoran dan *call routing*. *Proxy* juga dapat dikatakan sebagai IP PBX *Server*. *Proxy* yang saat ini digunakan mempunyai 2 jenis, yaitu berupa *hardware* mesin IPPBX dan berupa software yang disebut sebagai *softswitch* seperti *Asterisk* dan SER (*SIP Express Router*), dan *Yate*.

Beberapa *softswitch* yang dapat digunakan sebagai *Proxy* atau IP PBX server diantaranya adalah *Asterisk*, *Axon*, *MiniSIP Server*, *Trixbox*, dan *Briker*.

a. Asterisk

Asterisk adalah *software* yang mampu membuat komputer rumahan menjadi *server* untuk komunikasi suara. *Asterisk* bersifat *freeware* dan *open*

source, sehingga dapat dengan mudah digunakan tanpa perlu membayar lisensi dan bebas untuk digunakan dan dikembangkan. Sampai saat ini, *Asterisk* mendukung beberapa protokol untuk membangun IPPBX Server yaitu protokol SIP. H.323 dan IAX.

Asterisk sebenarnya merupakan *telepony toolkit open source* yang memungkinkan para pengembang untuk membuat beberapa macam aplikasi sebagai interface pada VoIP, dimana sebagian besar aplikasi menyerupai PBX (*Private Branch Exchaneg/Sentral*) yang bisa digunakan sebagai IVR (*Interface Voice Response*), *teleconference*, dan juga sebagai *voice mail system*. Oleh karena itu, semua fungsi tersebut disatukan dalam satu *server* dengan *software* yang dinamakan *Asterisk*.

b. Axon

Berbeda dengan *asterisk*, *Axon* merupakan *softswitch* yang khusus digunakan untuk *platform Windows*. *Axon* sejenis aplikasi kecil yang dapat diinstalasikan pada komputer. *Axon* memiliki fungsi sejenis IP PBX *Server.* Sedangkan dalam melakukan administrasi, A*xon* menggunakan layanan *web base* sehingga memudahkan proses konfigurasi.

c. MiniSIP Server

MiniSIP Server merupakan *softswitch* yang khusus digunakan untuk protokol SIP saja. Pada *website-nya MiniSIP Server* memiliki empat jenis *file* instalasi yang berbeda. Perbedaan *file* instalasi tersebut, memiliki perbedaan pada jumlah maksimum *user* yang akan digunakan.

Gambar 2.23 merupakan gambar halaman unduh *MiniSIP Server* dengan pilihan *file* instalasi berbeda sesuai dengan kebutuhan *user* yang akan digunakan sebagai IP PBX *server*.



Gambar 2.38 Halaman unduh *miniSIP Server*

d. Trixbox

Trixbox dibuat oleh Andrew Gillis pada bulan november 2004 dengan tujuan untuk membuat para pengguna komputer biasa dapat menggunakan secara maksimal *asterisk* PBX *system* tanpa dibutuhkannya pengajar atau pengetahuan lebih mengenai VoIP. Sebelumnya trixbox menggunakan nama *asterisk@home*, namun dikarenakan *asterisk* merupakan nama produk dari perusahaan Digium.Ltd dan *@home* tidak sesuai dengan fungsionalitas dari trixbox yang dapat melayani lebih dari sekedar pengguna rumahan atau bisnis sekala kecil dan menengah.

Cara instalasi server Trixbox ini tergolong mudah, berbeda dengan *server Asterisk* sebelumnya yang cara penginstalnya berupa paket-paket yang terpisah, maka pada *Trixbox* sudah di jadikan satu *bundle* dengan sistem operasi yang di usungnya yaitu CentOS sehingga kestabilan dari *server* ini dapat diandalkan.

Trixbox memiliki fitur-fitur yang mampu berfungsi sebagai IP PBX *Server*. Beberapa fitur *Trixbox* diantaranya adalah sebagai berikut :

1. AMP (*Asterisk Management Portal*), fitur ini adalah sebuah fitur yang sangat dapat melakukan konfigurasi melalui *interface* web tanpa harus mengedit *file* konfigurasi.
2. ARI (*Asterisk Recording Interface*), fitur ini berfungsi menyimpan percakapan, baik percakapan ke luar maupun ke dalam.
3. *Flash Operator Panel*, adalah sebuh fitur yang berguna untuk memonitor semua *extension* secara *real time* berbasis web.
4. *Cisco XML Service*.
5. *Music On Hold*, Trixbox menggunakan mpg123 untuk *music on hold*.
6. *Fax Support*, adalah suatu fitur yang untuk menerima *fax*.

e. Briker

Briker merupakan aplikasi berbasis *Open Source Software* (FOSS) yang dibangun untuk mengakomodir teknologi IP PBX. Briker akan merubah sebuah PC biasa menjadi mesin IP PBX. *Briker* dapat dikatakan sebagai distro, karena merupakan sistem operasi Linux yang khusus menangani masalah PBX. Konfigurasi yang dapat digunakan pada *Briker* sama seperti *Trixbox* yaitu menggunakan *web base* sehingga mempermudah proses administrasi hanya dengan memerlukan sebuah *web browser* saja.

Briker memiliki dukungan protokol SIP, H.3232 dan IAX. Dalam hal dukungan *codec,* Briker mendukung beberapa jenis *codec* seperti : ulaw, alaw, gsm, g723, dan g729.

Briker memiliki fitur yang sangat kompleks sesuai dengan fungsi utamanya yaitu sebagai IPPBX *Server*. Beberapa fitur yang dimiliki Briker dapat dilihat dari tabel di bawah ini.

Tabel 2.13 Fitur yang dimiliki IPPBX Briker

No	Kategori	Fitur
1	Briker IPPBX Core	Support VoIP protocol: SIP, H.323, and IAX2 Support analog and digital telephony device Support voice codec: ulaw, alaw, gsm, g723, g729 Support video codec: h264, h263p, h263, h261 Voice calling and conference Video calling and conference (limited) Unlimited registered accounts Up to max. 300 online accounts per server Up to max. 40 concurrent calls (transcoding) Up to max. 120 concurrent calls (no transcoding)
2	IPPBX Administration	Outbound and Inbound routing Interactive Voice Responce (IVR) system Automatic Call Distributin (ACD) Ring Group Call forwarding and follow me Voice recordings Voicemail configuration Direct Inward System Access (DISA) Music on hold Secure call termination by pin sets
3	Billing Administration	Prepaid and postpaid billing Auto refill balance, recurring service Multiple currency supported Call Detail Records (CDR) Detail call reports Least Cost Routing (LCR) Least Cost Dialing (LCD) Progressive billing Export report to PDF & CSV Generate invoices to PDF format
4	Server Administration	User and grups configuration Date/time configuration DHCP server configuration on web Web based Network configuration on web Reboot and shutdown from web

7. Protokol

Protokol adalah komponen berupa seperangkat aturan komunikasi antar *User Agent*, antar *Proxy* atau *User Agent* dengan *Proxy*. Protokol yang saat ini digunakan untuk membangun jaringan VoIP adalah H.323 dan SIP.

8. Codec

Codec merupakan kependekan dari *Compression/Decompression*. *Codec* merupakan teknologi yang memaketkan data *voice* ke dalam format lain dengan perhitungan matematis tertentu, sehingga menjadi lebih teratur dan mudah dipaketkan. *Codec* bertujuan untuk mengurangi penggunaan *bandwith* di dalam transmisi sinyal pada setiap pemanggilan dan sekaligus berfungsi untuk mengingkatkan jumlah panggilan. Dengan adanya *codec*, penggunaaan *bandwith* pada jaringan VoIP dapat dihemat.

Banyak sekali jenis protokol *vioce codec* yang tersedia untuk implementasi VoIP. *Voice Codec* yang umum dikenal adalah : G.711, G.723, G.726, G.728, dan G.729. Berikut gambaran mengenai masing-masing jenis *codec* tersebut.

a. G.711 – Mengkonversi *voice* ke 64 kbps voice stream. CODEC ini digunakan pada traditional TDM T1 *voice*. *The highest quality*.
b. G.723.1 – Terdapat 2 tipe 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. Tipe ini mempunyai *bit rate* 6.3 kbps.
c. G.726 – CODEC memiliki beberapa *bit rate* yang berbeda-beda, yaitu 40 kbps, 32 kbps, 24 kbps, dan 16 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.
f. G.729a memiliki algoritma yang lebih sederhana dan membutuhkan *processing power* lebih sedikit dibandingkan G.729.

2.14 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*. Penjelasan ketiga komponen tersebut adalah sebagai berikut :

a. *Extension* adalah komponen yang menangani registrasi dari pengguna, serta menyediakan *username* 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.*

2.14.1 Arsitektur asterisk

Pada dasarnya, arsitektur Asterisk sangatlah sederhana. Protokol yang dimplementasikan 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. 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.

2.15 Protokol H.323

H.323 adalah suatu standar yang menentukan komponen, protokol, dan prosedur yang menyediakan layanan komunikasi audio, video, dan data *real-time* (waktu nyata), melalui jaringan berbasis paket (*packed-base network*). Jaringan berbasis paket tersebut antara lain *Internet Protocol* (IP), *Internet Packet eXchange* (IPX), *Local Area Network* (LAN), *Enterprise Network* (EN), *Metropolitan Area Network* (MAN), dan *Wide Area Network* (WAN).

H.323 merupakan protokol yang dikembangkan oleh *International Telecommunications Union - Telecommunication* (ITU-T). H.323 dapat digunakan untuk layanan–layanan multimedia seperti komunikasi suara (*IP telephony*), komunikasi video dengan suara (*video telephony*), dan gabungan suara, video dan data.

2.15.1 Komponen H.323

Standar H.323 terdiri atas empat komponen penting yang terhubung. Keterhubungan komponen-komponen tersebut dalam suatu jaringan akan memberikan layanan komunikasi *point to point* dan *multipoint*. Ke empat komponen tersebut adalah *Terminal*, *Gateway*, *Gatekeeper*, dan *Multipoint Control Unit* (MCU).

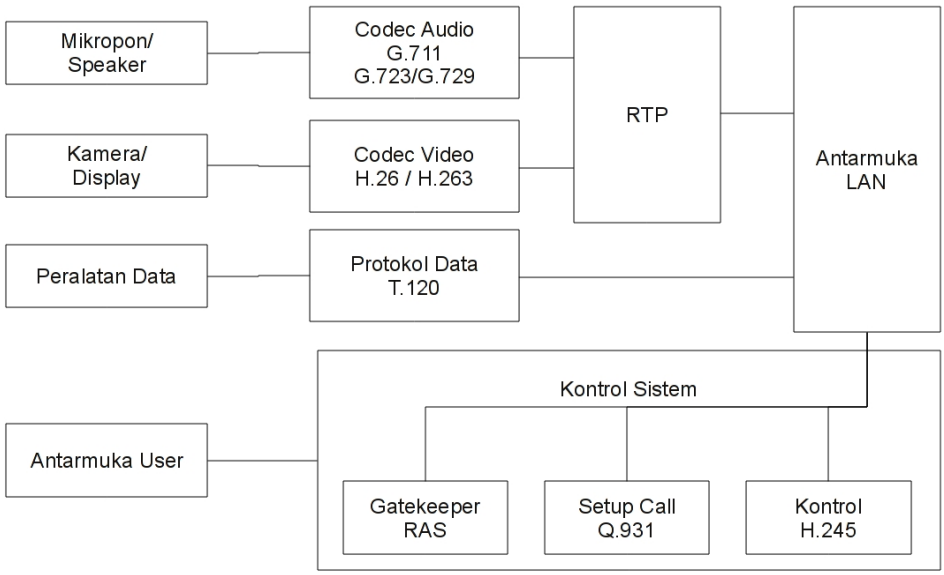
2.15.1.1 Terminal

Terminal adalah sebuah *end-point* pada LAN yang digunakan untuk komunikasi. *Terminal* H.323 harus mendukung komunikasi *audio* sedangkan

video dan data adalah pilihan. H323 menspesifikasikan mode-mode dan operasi yang dibutuhkan untuk *audio*, *video* atau data digunakan secara bersama.*Terminal* H.323 dapat berupa *personal computer* (PC) atau alat lain yang berdiri sendiri yang dapat menjalankan aplikasi multimedia. Karena pelayanan utama yang disediakan oleh H.323 adalah komunikasi *audio*, maka sebuah *terminal* H.323 harus bisa melakukan layanan *IP Telephony*.

Terminal H.323 harus mendukung H.245 yang digunakan untuk bernegosiasi dengan penggunaan kanal. Tiga komponen lain yang diperlukan adalah Q.931 untuk pensinyalan *(signalling) call setup* dan *terminasi call*, komponen yang disebut H.255.0 *Registration*/*Admission*/*Status* (RAS), yaitu protokol yang digunakan untuk berkomunikasi dengan *Gatekeeper* berhubungan dengan fungsi regstrasi, admisi dan status, serta *Real-Time Protocol/Real-Time Control Protocol* (RTP/RTCP) digunakan untuk mengatur sekuen dari paket *audio* video maupun data untuk ditransmisikan.

Komponen yang merupakan pilihan pada *terminal* H.323 adalah *video codec*, protokol data T.120 dan *Multi Control Unit*. Blok diagram *Terminal* H.323 secara lengkap ditunjukkan pada Gambar 2.39.



Gambar 2.39 *Terminal* H.323

Fungsi dan kemampuan terminal H.323 adalah sebagai berikut :

a. *Audio Codec*, mengodekan sinyal dari peralatan *audio* untuk transmisi dan menguraikan kode *audio* yang diterima. Fungsi-fungsi yang dibutuhkan antara lain mengodekan dan menguraikan kode pada G.711 dan mengirimkan dan menerima format a-law dan u-law. Sebagai tambahan audio *codec* ini juga dapat menkode dan menguraikan kode pada G.726, G.728, dan G.723.1.

b. *Video Codec*, merupakan fungsi tambahan pada terminal H.323

c. *Data Channel*, mendukung aplikasi-aplikasi perperti pengakses database, pengiriman file, dan *audiographics conferencing* (kemampuan untuk memodifikasi gambar untuk beberapa pengguna secara bersama-sama), dan direkomendasikan T.120.

d. *System Control Unit,* menyediakan H.255 dan *call control* H.245, pengiriman pesan, dan perintah-perintah pensinyalan.

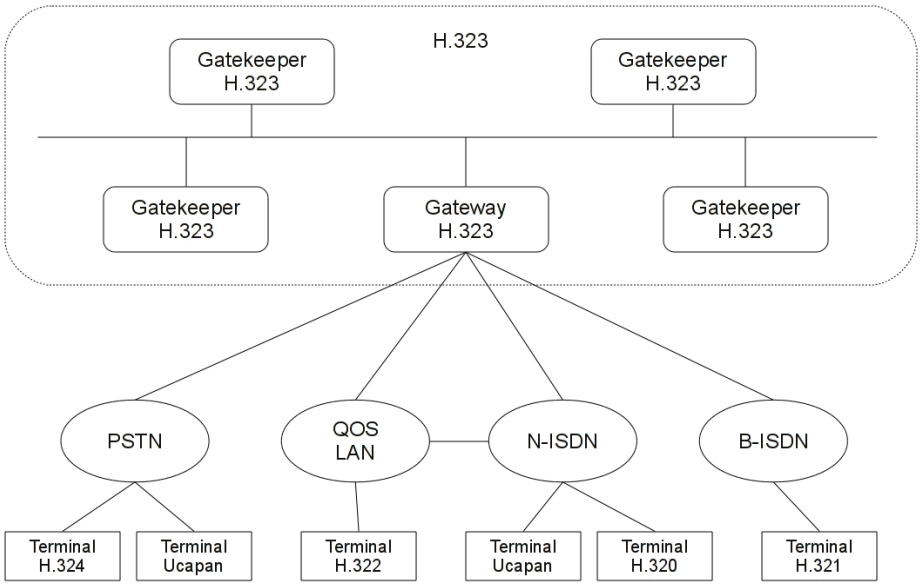
e. *Media Transmission*, membentuk format *audio*, *video*, data, *control stream*, dan *message* yang sesuai dengan antarmuka jaringan dan juga menerima dari antarmuka jaringan.

f. *Network Interface*, merupakan suatu antarmuka *packet-packet* untuk *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) pada layanan *unicast* maupun *multicast*.

2.15.1.2 Gateway

Gateway adalah elemen pilihan dalam sistem H.323. *Gateway* memberikan fungsi translasi antara terminal H.323 dengan *terminal* yang lain. Fungsi ini termasuk translasi antara format-format yang digunakan dalam transmisi (Misalnya H.255.0 ke H.221/H.223) dan antara prosedur-prosedur komunikasi (misalnya H.245 ke H.242). *Gateway* juga mentranslasikan antara *codec*-codec *audio* dan *video* dan membentuk *setup call* serta melakukan *clearing* untuk sisi jaringan LAN dan sisi jaringan *switched-circuit*.

Sebuah *gateway* menghubungkan dua buah jaringan yang berbeda. *Gateway* H.323 menghubungkan jaringan H.323 dengan jaringan *non*-H.323. *Gateway* memiliki kemampuan berkoneksi dengan jaringan yang berbeda dengan melakukan cara menerjemahkan protokol *call setup* dan mengubah format media antara jaringan yang berbeda, serta memindahkan informasi antar jaringan yang terhubung dengan *gateway*. Sebagai contoh, suatu *gateway* dapat menghubungkan dan menyediakan komunikasi antara terminal H.323 dengan *Switched Circuit Network* (SCN). Jaringan SCN mencakup semua jaringan telepon *switched*, misalnya *Public Switched Telephone Network (*PSTN).



Gambar 2.40 *Gateway* H.323

Gateway tidak diperlukan jika tidak berkomunikasi dengan jaringan yang lain, sehingga *terminal-terminal* dapat berkomunikasi secara langsung pada LAN yang sama. *Gateway* dapat mendukung *terminal-terminal* untuk standar H.320 konferensi *narrowband* ISDN, H.321 konferensi pada *broadband* ISDN (Jaringan ATM), H.322 konferensi pada LAN yang menjamin kualitas layanan, dan H.324 konferensi pada jaringan PSTN.

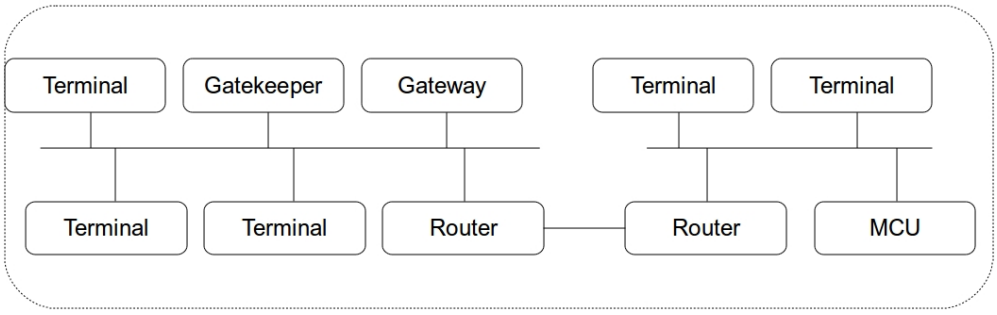
2.15.1.3 Gatekeeper

Gatekeeper merupakan komponen yang sangat penting yang memungkinkan H.323 bekerja. *Gatekeeper* berperilaku seperti *central point* untuk semua *call* dan memberikan layanan kontrol *call* ke *endpoin-endpoint* (*Terminal*, *Gateway*, atau MCU). Dalam beberapa hal, sebuah *gatekeeper* H.323 berfungsi sebagai sebuah *switch virtual.*

Gatekeeper membentuk dua fungsi kontrol *call* penting, yaitu membentuk translasi *Terminal* dan *Gateway* ke alamat-alamat IP/IPX. Fungsi yang kedua

adalah penanganan *bandwidth* yang dispesifikasikan dalam *RAS*. Sebagai contoh, jika seorang administrator jaringan telah menspesifikasikan sebuah batas ambang untuk sejumlah konferensi yang simultan pada LAN, *Gatekeeper* dapat menolak untuk membuat koneksi yang baru jika batas ambang sudah tercapai. Hal ini bertujuan untuk membatasi *bandwidth* yang digunakan untuk konferensi, sehingga *bandwidth* yang lain masih dapat digunakan untuk aplikasi internet yang lain.

Kumpulan dari semua *Terminal*, *Gateway*, dan *Multi Control Unit* ditangani oleh sebuah *Gatekeeper* yang disebut sebagai ***Zona H.323***, dapat dilihat pada gambar 2.41.



Gambar 2.41 Zona H.323

Gatekeeper mampu untuk melakukan routing call H.323, sehingga *call* dapat dikontrol secara efisien. Hall ini diperlukan oleh *service provider* untuk menghitung biaya untuk setiap *call* yang melalui jaringan mereka. Layanan ini juga dapat digunakan untuk melakukan *routing* kembali (*re-routing*) sebuah call. Kemampuan melakukan *re-routing* ini dapat membantu menentukan keputusan-keputusan yang berhubungan dengan keseimbangan di antara berbagai *gateway* (*multiple gateway).*

2.15.1.4 Multipoint Control Unit (MCU)

Multipoint Control Unit (MCU) sering disebut sebagai *server* konferensi yang mendukung konferensi di antara beberapa *endpoint/terminal*. MCU memberikan dukungan untuk konferensi tiga atau lebih *terminal* H.323. Semua terminal yang akan berpartisipasi dalam konferensi melakukan koneksi terlebih dahulu dengan *Multipoint Control Unit. Multipoint Control Unit* mengatur konferensi *resource*, negosiasi antar terminal untuk tujuan penentuan *audio* atau *video coder*/*decoder* (CODEC) yang digunakan.

MCU terdiri atas *Multipoint Controller* (MC) dan *Multipoint Processor* (MP). MC menangani negosiasi H.245 di antara semua *terminal* untuk menentukan kemampuan umum untuk pemrosesan video dan audio. MC juga mengontrol sumber daya konferensi dengan menentukan stream audio atau video yang akan dimulticast. Meskipun demikian MC tidak berhubungan langsung dengan *stream-stream* data. Berbeda dengan MP yang melakukan *mixing*, *switching*, dan pemrosesan audio, video, atau bit-bit data. MC dan MP dapat terintegrasi pada suatu terminal atau merupakan bagian lain dari komponen H.323.

Sebuah MC dapat diletakkan dalam sebuah *gatekeeper*, *gateway*, *terminal*, atau MCU. Sebuah MCU yang terpisah dapat digunakan untuk menangani fungsi-fungsi *audio*, *video*, data dan kontrol. Pada konfigurasi ini video dapat di *multicast* untuk penghematan *bandwidth*.

2.15.2 Alamat pada H.323

Alamat yang dimaksud disini merupakan alamat identitas yang digunakan untuk melakukan komunikasi dengan yang lainnya. Penggunaanya akan

dibutuhkan saat akan melakukan *call* kepada user lain. Alamat pada H.323 memiliki bentuk yang mirip dengan alamat sebuah e-mail. Dalam prakteknya Alamat pada H.323 terbagi menjadi 3 berdasarkan infrastruktur yang digunakan.

Tabel 2.14 Contoh pengalamatan H.323

No.	Alamat	Keterangan
1	H323:IP:192.168.1.12	Call dengan tujuan PC dengan IP tertentu (dapat pula menggunakan domainnya.)
2	H323:GK:H323ID:indra	Call menggunakan id yang terdaftar di Gatekeeper.
3	H323:GK:E164:135	Call menggunakan no (e164) yang terdaftar di Gatekeeper.
4	H323:GK:E164:95551234	Call dengan tujuan ke luar esktension (PSTN).
5	H323:GW:192.168.2.2:55512345	Call menggunakan Gateway.

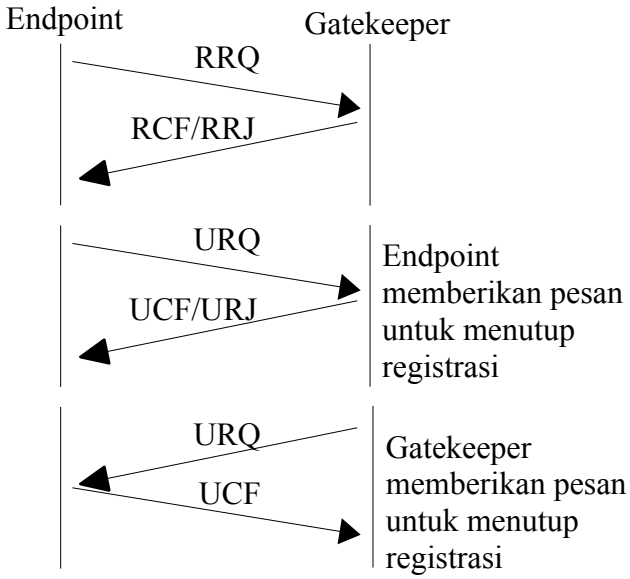
2.15.3 *Registration*

Registration merupakan proses pendaftaran (registrasi) agar *gateway*, *endpoint*, dan MCU tergabung dalam sebuah area atau daerah/zona H.323. Dalam melakukan proses registrasi ada enam pesan yang digunakan untuk menghubungkan dan menutup pesan registrasi tersebut.

a. *Registration Request* (RRQ), dikirim dari *endpoint* kepada *gatekeeper*.
b. *Registration Confirm* (RCF), dikirim oleh *gatekeeper* dan mengkonfirmasikan sebuah pesan registrasi dari *endpoint*.

c. *Registration Reject* (RRJ), dikirim oleh *gatekeeper* dan menolak sebuah pesan registrasi dari *endpoint*.
d. *Unregistration Request* (URQ), dikirim dari *endpoint* atau *gatekepeer* untuk membatalkan sebuah registrasi.
e. *Unregistration Confirm* (UCF), dikirim dari *endpoint* atau *gatekepeer* untuk mengkonfirmasi sebuah pesan pembatalan registrasi.
f. *Unregistration Reject* (URJ), indikasi pada sebuah *endpoint* yang sudah tidak lagi terdaftar pada *gatekeeper*.

Gambar di bawah merupakan pesan ilustrasi dan proses berurutan dalam proses *registration* yang dilakukan *endpoint* dan *gatekeeper.*



Gambar 2.42 Proses registration endpoint

Gambar di atas merupakan proses sebuah registrasi antara *endpoint* dengan *gatekeeper*. Pada mulanya, sebuah *endpoint* akan dapat melakukan komunikasi dengan mengirimkan pesan RRQ pada *gatekeeper*. Selanjutnya *gatekeeper* dapat membalas dari pesan RRQ dengan dua kemungkinan yaitu diterima (RCF) atau

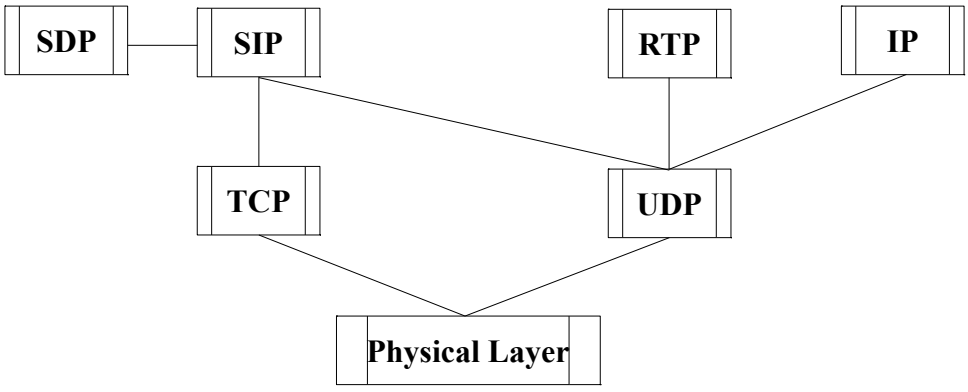
ditolak (RRJ). Proses penutupan registrasi dapat dilakukan oleh *endpoint* ataupun *gatekeeper* dengan mengirimkan pesan URQ.

2.16 *Session Initiation Protokol* (SIP)

Session Initiation Protocol (disingkat SIP) merupakan standar protokol multimedia yang dikeluarkan oleh grup yang tergabung dalam *Multyparty Multimedia Session Control* (MMUSIC) yang berada dalam organisasi *Internet Engineering Task Force* (IETF). SIP merupakan protokol yang berada pada layer aplikasi yang mendefinikan proses awal, pengubahan dan pengakhiran (pemutusan) suatu sesi komunikasi multimedia.

2.16.1 Susunan Protokol SIP

Protokol SIP didukung oleh beberapa protokol, antara lain RSVP untuk melakukan pemesanan pada jaringan, RTP dan RTCP untuk mentransmisikan media dan mengetahui kualitas layanan, serta SDP untuk mendeskripsikan sesi media. Secara *default*, SIP menggunakan protokol UDP tetapi beberapa kasus dapat juga menggunakan TCP sebagai protokol transport.



Gambar 2.43 Arsitektur protokol SIP

a. RTP (*Real-Time Transport Protocol*)

Protokol TRP menyediakan transfer media secara real-time pada jaringan paket. Protokol RTP menggunakan 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.

b. RTCP (*Real-Time Control Protocol*)

Protokol RTCP merupakan protokol yang mengendalikan transfer media. Protokol ini bekerja sama dengan protokol RTP. Dalam satu sesi komunikasi, protokol RTP mengirimkan paket RTCP secara periodik untuk memperoleh informasi transfer media dalam memperbaiki kualitas jaringan.

c. SDP (*Session Description Protocol*)

Protokol SDP merupakan protokol yang mendeskripsikan media dalam suatu komunikasi. Tujuan protokol SDP adalah untuk memberikan informasi aliran media dalam satu sesi komunikasi agar penerima yang menerima informasi tersebut dapat berkomunikasi. Hal-hal yang dicakup dalam protokol ini adalah :

a. Nama sesi komunikasi dan tujuannya.
b. Waktu sesi (jika) aktif.
c. Media dalam sesi komunikasi.
d. Informasi bagaimana cara menerima media (misalnya *port*, format, dan sebagainya).
e. Bandwidth yang digunakan dalam komunikasi.
f. Orang yang dapat dihubungi.

Spesifikasi SDP mempunyai bentuk standar, yaitu :

<type> = <value>

<type> adalah satu karakter yang mempunyai arti. Sedangkan <value> merupakan teks string yang terstruktur yang formatnya berdasarkan *type*. Di bawah ini diberikan *type* dan deskripsi yang dipakai dalam SDP, yang merupakan *session description* yaitu :

1. v=protokol version
2. o=creator/owner and session identifier
3. s=session name
4. i=*session information
5. u=*uniform resource identifier (URI)
6. e=*email address
7. p=*phone number
8. c=*connection information
9. b=*bandwidth information
10. z=*time zone adjusmet
11. k=*encryption key
12. a=*zero or more session atribute lines

Berikutnya tipe dan deskripsi yang termasuk *time description* adalah :

1. t=time the session is active
2. r=*zero or more repeat time

Sedangkan tipe dan deskripsi yag termasuk dalam *media description* adalah sebagai berikut:

1. m=media name
2. i=media title
3. c=*connection information
4. b=*bandwidth information
5. k=*encryption key
6. a=*zero more media attribute lines

Deskripsi yang bertanda bintang (*) bersifat opsional yang berarti bisa digunakan juga bisa tidak digunakan. Contoh informasi yang berkaitan dengan protokol SDP sesuai bentuk di atas yaitu :

a. v=0
b. o=bram 228139821 8219382198 IN IP4 132.97.1.32
c. s=testing
d. e=rijalhanif@gmail.com
e. a=recvonly
f. m=audio 49170 RTP/AVP 0
g. m=application 32416 udp wb

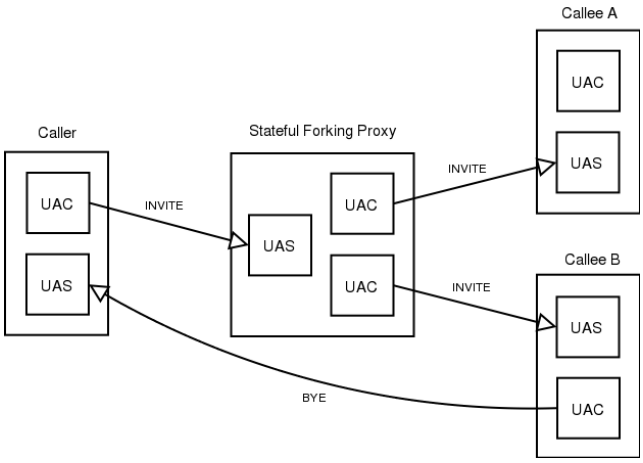
2.16.2 Komponen SIP

Dalam hubungannya dengan VoIP ada dua komponen yang ada dalam sistem SIP, yaitu *User Agent* dan *Network Server*.

2.16.2.1 *User Agent*

Sama seperti komponen VoIP pada umumnya, komponen terpenting dalam membangun VoIP berbasis protokol SIP *pun* membutuhkan *User Agent. User* agent merupakan sistem akhir (*end system*) yang digunakan untuk melakukan komunikasi. Pada SIP, *user Agent* terbagi atas dua bagian, yaitu :

1. *User Agent Client* (UAC), merupakan aplikasi pada *client* yang didesain untuk memulai SIP *request.*
2. *User Agent Server* (UAS), merupakan aplikasi *seirver* yang memberitahukan *user* jika menerima *request* dan memberikan respon terhadap *request* tersebut. Respon dapat berupa menerima atau menolak *request.*



Gambar 2.44 UAC dan UAS

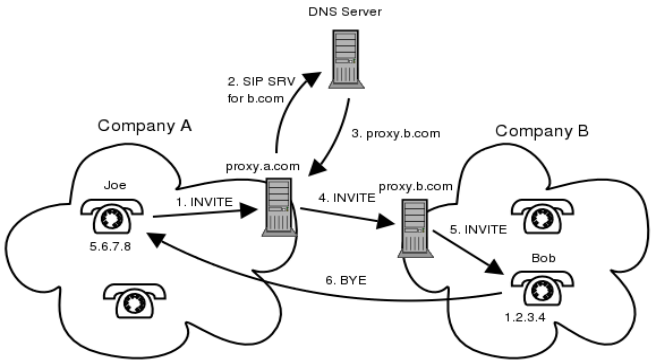
2.16.2.2 *Network Server*

Agar *user* pada jaringan SIP dapat memulai suatu panggilan dan dapat pula dipanggil, maka *user* terlebih dahulu harus melakukan registrasi agar lokasinya dapat diketahui. Registrasi dapat dilakukan dengan mengirimkan pesan REGISTER ke *server* SIP. Lokasi *user* dapat berbeda-beda sehingga untuk mendapatkan lokasi *user* yang aktual diperlukan *location server*. Pada jaringan SIP, ada tiga tipe *network server*, yaitu :

a. *Proxy Server*

Proxy server adalah komponen penengah antar *user agent. Proxy server* bertindak sebagai *server* dan *client* yang menerima *request message* dari *user agent* dan menyampaikan pada *user agent* lainnya. *Request* dapat dilayani sendiri atau disampaikan (*forward*) pada *proxy server* lain. *Proxy Server* bertugas menerjemahkan data dan/atau menulis ulang *request message* sebelum menyampaikan pada *user agent* tujuan atau *proxy* lain. Selain itu, *proxy server* bertugas menyimpan seluruh *state* sesi komunikasi antara UAC dan UAS. *Proxy*

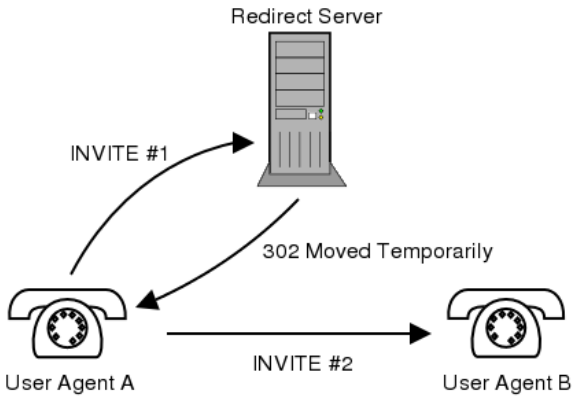
server dapat berfungsi sebagai *client* dan *server* karena *proxy server* dapat memberikan *request* dan respon.



Gambar 2.45 Proses sesi *invite* dengan *proxy server*

b. *Redirect Server*

Komponen ini merupakan server yang menerima request *message* dari user agent, memetakan alamat SIP *user agent* atau *proxy server* tujuan kemudian memberikan respon terhadap request tersebut dan menyampaikan hasil pemetaan kembali pada *user agent* pengirim (UAC). *Redirect Server* tidak menyimpan *state* sesi komunikasi antara UAC dan UAS setelah pemetaan disampaikan pada UAC. Berbeda dengan *Proxy Server*, *Redirect Server* tidak dapat memulai inisiasi *request message* dan tidak dapat menerima ataupun menutup sesi komunikasi.

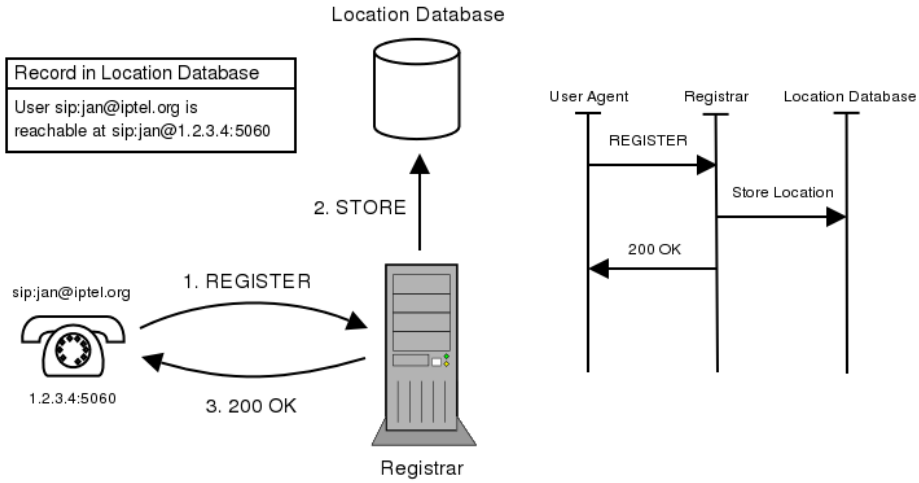


Gambar 2.46 Redirect Server

c. *Registrar Server*

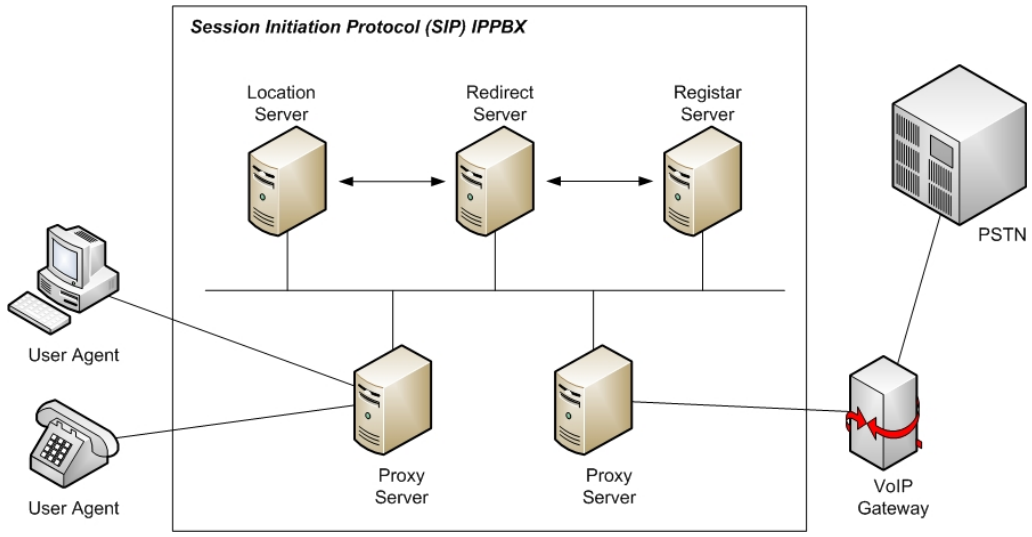
Registar Server adalah komponen yang menerima *request message* REGISTER. *Registrar Serve*r menyimpan databases user untuk otentikasi dan lokasi sebenarnya agar *user* dapat dihubungi oleh komponen SIP lainnya.

Pada Gambar 2.47 menunjukkan proses registrasi oleh user dengan alamat sip:jan@iptel.org. Alamat sip:jan@iptel.org atau sip:jan@1.2.3.4:5060 berada dalam database *server.* Proses yang dilakukan adalah user meregistrasikan dirinya ke *server* dengan mengirimkan pesan REGISTER ke *Registar.* Bila otentikasi yang diberikan *valid* dan ada dalam *database*, maka *Registrar* akan mengirimkan pesan respon 200 OK dan proses registasi pun selesai dilakukan.



Gambar 2.47 *Registrar Server*

Semua fungsi *network server* di atas, merupakan sekumpulan fungsi *server* yang telah dijadikan satu *bundle* pada sebuah fungsi IPPBX *Server* pada protokol SIP. Gambar 2.48 memperlihatkan keseluruhan hubungan antar komponen SIP.



Gambar 2.48 Arsitektur VoIP menggunakan SIP

2.16.3 Alamat pada SIP

Entitas pada jaringan SIP mempunyai alamat yang diberi atribut SIP URL (SIP *Uniform Resource Locator*) agar mudah dikenali. SIP URL yang digunakan pada jaringan SIP berbentuk seperti alamat *email* yaitu user@host dimana user dapat berupa nama *user*, nomor telepon, atau nama instansi. Host dapat berupa nama domain atau IP *address*. Contoh pengalamatan pada SIP dapat dilihat pada daftar tabel berikut :

Tabel 2.15 Contoh pengalamatan SIP

No	Alamat	Keterangan
1	sip: rijalhanif@voiprakyat.or.id	Merupakan host independent
2	sip: rijalhanif@voip.unikom.ac.id	Merupakan host spesific
3	sip: +62-22-2530153@voip.telkom.net	Merupakan nomor telepon user dengan domain gateway voip.telkom.net

2.16.4 Pesan pada SIP

Secara keseluruhan, pesan SIP terdiri atas dua bagian, yaitu *request* dan *respon*. Ketika client mengirimkan pesan request, *server* akan memberikan tanggapan terhadap pesan ini melalui pesan respon. SIP merupakan protokol yang berbasis *teks* dimana pesan *request* dan *respon* menggunakan *generic-message* yang didefinisikan pada standar pesan berbasis teks dalam komunikasi internet.

Pesan request dan respon terdiri atas start line, satu atau lebih *header field* atau biasanya disebut dengan *message header*, *empty line* yang menunjukkan akhir dari *header field*, serta *message body* yang mendefinisikan sesi komunikasi.

Format pesan SIP yaitu :

Generic-message = Start-Line (dalam pesan request)
Status-Line (dalam pesan respon)
Message header
Empty line
Message body

Contoh lengkap pesan yang dihasilkan saat komunikasi dengan menggunakan protokol SIP dapat dilihat pada Gambar 2.49.

```
INVITE sip:7170@iptel.org SIP/2.0
Via: SIP/2.0/UDP 195.37.77.100:5040;rport
Max-Forwards: 10
From: "jiri" <sip:jiri@iptel.org>;tag=76ff7a07-c091-4192-84a0-d56e91fe104f
To: <sip:jiri@bat.iptel.org>
Call-ID: d10815e0-bf17-4afa-8412-d9130a793d96@213.20.128.35
CSeq: 2 INVITE
Contact: <sip:213.20.128.35:9315>
User-Agent: Windows RTC/1.0
Proxy-Authorization: Digest username="jiri", realm="iptel.org",
algorithm="MD5", uri="sip:jiri@bat.iptel.org",
nonce="3cef753900000001771328f5ae1b8b7f0d742da1feb5753c",
response="53fe98db10e1074
b03b3e06438bda70f"
Content-Type: application/sdp
Content-Length: 451
v=0
o=jku2 0 0 IN IP4 213.20.128.35
s=session
c=IN IP4 213.20.128.35
b=CT:1000
t=0 0
m=audio 54742 RTP/AVP 97 111 112 6 0 8 4 5 3 101
a=rtpmap:97 red/8000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:6 DVI4/16000
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=rtpmap: 3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
```

Gambar 2.49 Contoh pesan dalam SIP

2.16.5 *Header Field*

Protokol SIP mempunyai 37 *header*, yaitu pesan-pesan yang terdapat dapat SIP menggunakan *header field* untuk mendefinisikan *caller*, *calle*, jalur pesan, tipe, dan panjang *message body*, dan sebagainya. *Header* SIP dikelompokkan ke dalam empat jenis *header*, yakni :

a. *General Header Field* (GHF)

GHF merupakan *header* yang dipakai pada pesan *request* dan respon. GHF umumnya digunakan pada pesan *request* dan respon, yakni :

1. Call-ID, *header* ini digunakan untuk mengidentifikasikan secara khusus suatu panggilan atau registrasi yang dilakukan oleh client. Call-ID mempunyai fungsi untuk mendeteksi adanya duplikasi dan mendeteksi suatu respondari request yang dikirmkan. Call-ID yang baru digunakan untuk setiap awal suatu panggilan baru. Contoh : 119687@voiprakyat.or.id.
2. From

   Header ini terdapat pada semua pesan *request* dan respon, berfungsi untuk menunjukkan tampilan nama dan alamat asal pesan tersebut.

   Contoh :

   From : sip:rijalhanif@voiprakyat.or.id

   Pesan di atas berasal dari Rijal Hanif. Dalam hal ini Rijal Hanif menggunakan loginnya di *domain voiprakyat.or.id* dengan *username rijalhanif.*
3. To

   Header ini terdapat pada semua pesan *request* dan respon. Berfungsi menunjukkan tujuan pesan tersebut.

   Contoh :

   To : sip:admin@192.168.20.248

   Pesan di atas ditujukan ke *admin* dengan IP *address* 192.168.20.248.

4. Via

   Header ini digunakan untuk mencatat *rute server* dari pesan *request* agar dapat mengirimkan balasan ke pesan *request* tersebut melalui *server* yang sama. Setiap *proxy server* yang dilalui pesan tersebut akan menambah *header via* yang berisi alamat *proxy server* itu sendiri.

   Contoh :

   INVITE
   To : sip:jaya@132.98.10.10
   Via: SIP/2.0/UDP 132.95.1.5

   Pesan di atas akan dikirimkan ke Wijaya yang menggunakan *username* jaya dengan IP *address* 132.98.10.10 melalui *proxy server* dengan IP *address* 132.95.1.5.

5. Cseq

   Setiap pesan *request* mempunyai *header* Cseq *(Command sequence)* yang berisikan *sequence number* dan *methode name*. Dalam setiap pangilan sequence number mengalami penambahan untuk setiap *request* yang baru (kecuali jika terjadi transmisi dari *request* yang sebelumnya). Pesan *request* ACK memiliki Cseq yang sama dalam *acknowledge reply* CANCEL memiliki Cseq yang sama terhadap *request* yag dibatalkan.

b. *Entity Header Field* (EHF)

EHF menunjukkan informasi *message body*. Jika *message body* tidak ada, *header* ini menunjukkan sumber yang diidentifikasi oleh *request*. Contoh *header* yang terdapat pada EHF, antara lain:

1. *Content Encoding*, *Header* ini menunjukkan panjang *message body* dalam satuan byte.

2. *Content Length*, *Header* ini menunjukkan tipe media dalam *message body.*
3. *Content Type*, *Header* ini dipakai untuk melakukan proses kompresi terhadap *message body* tanpa harus kehilangan identitas dari tipe media.

c. *Request Header Field* (RsHF)

RsHF adalah *header* dalam pesan request yang merupakan tambahan informasi tentang *client* dan pesan *request* itu sendiri. *Header* yang sering dipakai adalah *header Contact* yang menunjukkan informasi lokasi yang tergantung dari pesan tempat *header* itu berada.

d. *Response Header Field* (ReHF)

ReHF merupakan header yang dipakai oleh *sever* untuk menambahkan informasi tentang respon yang tidak dapat ditempatkan pada *start line request.*

Secara lengkap, *header* yang terdapat di dalam SIP dapat di lihat pada tabel 2.16 di bawah.

Tabel 2.16 Header SIP

General Header	Entity Header	Request Header	Response Header
Accept	Content-Encoding	User-Agent	Allow
Accept-Encoding	Content-Length	Contact	Proxy-Authenticate
Accept-Language	Content-Type	Hide	Retry-After
Call-ID		Max-Forwards	Server
Contact		Organization	Unsupported
CSeq		Priority	Warning
Date		Proxy-Authorization	WWW-Authenticate
Encryption		Proxy-Require	
Expires		Route	
From		Require	
Record-Route		Response-Key	
Timestamp		Subject	
To		User-Agent	
Via			

2.16.6 SIP *Request*

SIP *Request* merupakan sebuah pesan yang dikirimkan dari *client* ke *server*. Ada enam tipe pesan *request*, yaitu :

a. **INVITE,** Pesan ini digunakan untuk memulai suatu komunikasi. *Message body* pesan INVITE berisikan deskripsi media yang dapat digunakan dalam komunikasi.

b. **ACK,** Pesan ini berfungsi memberitahukan bahwa *client* telah menerima tanggapan terakhir terhadap INVITE. *Message body* pada pesan ACK dapat membaca deskripsi media yang digunakan oleh *user* yang dipanggil (disebut *calle*). Jika *message body* ini kosong berarti *calle* setuju dengan *message body* yang terdapat pada pesan INVITE.

c. **BYE,** Pesan ini dikirimkan oleh *client* untuk mengakhiri komunikasi.

d. **CANCEL,** Pesan CANCEL dikirimkan untuk membatalkan pesan *request* yang telah dikirimkan sebelum *server* mengirimkan pesan *final response*.

e. **OPTIONS,** Pesan ini dikirimkan oleh *client* ke *server* untuk mengetahui kapabilitasnya.

f. **REGISTER,** *Client* dapat melakukan registrasi lokasinya dengan mengirimkan pesan REGISTER ke *server* SIP dimana *server* yang menerima pesan REGISTER disebut SIP R*egister*.

2.16.7 SIP Response

SIP *Response* dikirimkan setelah menerima pesan *request* yang menunjukkan status keberhasilan *server*. Pesan respon dapat didefinisikan dengan tiga angka.

Tabel 2.17 merupakan kelas respon yang terdapat pada SIP. Angka pertama merupakan kelas respon sedangkan angka kedua dan ketiga menunjukkan arti dari respon tersebut.

Tabel 2.17 Kelas respon

Kelas Respon	Jenis Respon	Kategori Respon
1xx	Informational	Provisional
2xx	Succes	Final
3xx	Redirection	Final
4xx	Client error	Final
5xx	Server error	Final
6xx	Global error	Final

Pesan respon terbagi atas dua kategori, yakni :

a. *Provisional Response*

Respon ini merupakan respon yang dikirimkan oleh *server* untuk menunjukkan proses sedang berlangsung, tapi tidak mengakhiri transaksi SIP.

b. *Final Response*

Respon ini merupakan respon yang mengakhiri transaksi SIP.

Tabel di bawah berikut merupakan kumpulan kode respon SIP yang dikategorikan berdasarkan jenis responnya.

Tabel 2.18 Kode pesan respon SIP

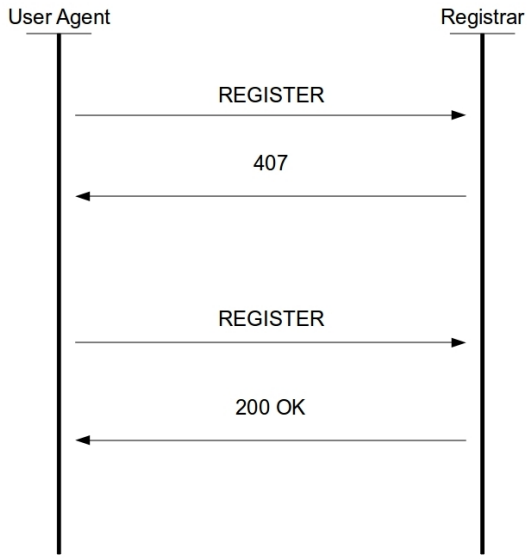
Kelas	Jenis Respon	Kode	Perintah
1xx	Informational Request diterima dan dilanjutkan dengan memproses request	100 180 181 182	Trying Ringin Call is being forwarded Queued
2xx	Succes Pesan telah diterima dan dimenegerti	200	OK
3xx	Redirection Perlu dilakukan tindakan seanjutnya untuk menyelesaikan request	300 301 302 380	Multiple choices Moved permanently Moved temporarily Alternative service
4xx	Client error Request tidak dapat diproses oleh server atau terdapat syntax eror pada request	400 401 402 403 404 405 406 407 408 409 410 411 413 414 415 420 480 481 482 483 484 485	Bad Request Unauthorized Payment Required Forbidden Not Found Method not Allowed Not Acceptable Proxy Auth. Required Request timeout Conflict Gone Length Required Request Message too Large Request URI too Large Unsupport media type Bad extension Not Available (temp.) Call leg Loop detected Too many hops Address incomplete Ambiguous
5xx	Server error Request tidak dapat diolah oleh server atau terdapat syntax error pada request	500 501 502 503 504 505	Internal server error Not implemented Bad gateway Service unavailable Gateway timeout SIP version not support
6xx	Global error Request invalid pada server	600 603 604 605	Busy everywhere Decline Doesn't exit anywhere Not Acceptable

2.16.8 Alur SIP

Alur SIP merupakan serangkaian perintah umum yang dilakukan dalam melakukan komunikasi di dalam SIP. Hal yang umum dilakukan pada sistem VoIP antara lain : *User Registration, Session Initiation,* dan *Session Termination.*

a. *User Registration*

User Registration merupakan sebuah skenario yang terdapat pada protokol SIP yang berfungsi untuk melakukan pendaftaran seorang *user* pada jaringan VoIP sehingga *user* tersebut dapat dipanggil oleh *user* lain. Pada saat registrasi, *user* harus memberikan informasi *username* dan *password* yang digunakan untuk melindungi *user* tersebut.



Gambar 2.50 Diagram pesan REGISTER

Gambar 2.50 di atas adalah alur proses *User Registration* pada protokol SIP. Berikut penjelasan tahapan demi tahapan pada saat proses *User Registration*:

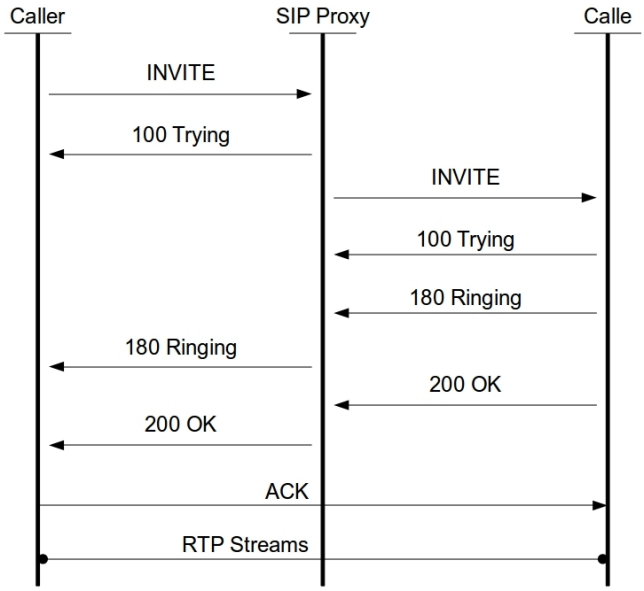
1. *User* mengirimkan *request* REGISTER kepada SIP *Server*.
2. Karena SIP *Server* tidak mengenali *user* tersebut maka SIP *Server* mengirimkan *response* 407 sehingga memaksa *user* untuk mengirimkan

ulang *request* REGISTER dan ditambahi informasi *password*.

3. User mengirimkan ulang *request* REGISTER dengan ditambahkan informasi *password*.
4. SIP *Server* melakukan pemrosesan terhadap informasi yang dikirimkan dan apabila informasi tersebut sesuai dengan data yang ada pada *database* maka SIP *Server* akan mengirimkan *response* OK dengan kode 200.

b. *Session Invitation*

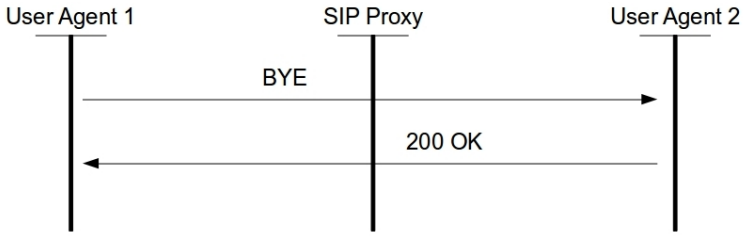
Pada protokol SIP, apabila seorang *user* ingin memulai sebuah percakapan maka *user agent* yang digunakan harus mengirimkan request INVITE. Apabila *user agent* tujuan memberikan persetujuan untuk melakukan pecakapan maka *user* agent tersebut akan mengirimkan *response* OK. Setelah mendapat *response* OK maka user agent pemanggil harus mengirimkan reques ACK untuk melakukan percakapan. Apabila *user* tujuan menerima request ACK tersebut maka sesi percakapan dengan menggunakan protokol RTP dapat dilakukan.



Gambar 2.51 Diagram pesan INVITE

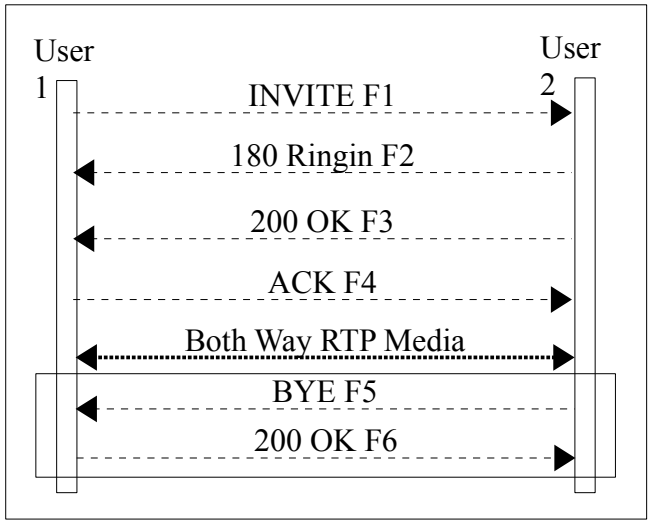
c. *Session Termination*

Session termination adalah suatu mekanisme yang terdapat pada protokol SIP yang berfungsi untuk memberhentikan sesi percakapan yang sedang berlangsung. Pada Gambar 2.52 memperlihatkan pemutusan sesi yang dilakukan *User Agent* 1 dengan mengirimkan pesan BYE.



Gambar 2.52 Diagram pesan BYE

Pemberhentian dapat dilakukan oleh user agent pemanggil (*caller*) atau penerima (*calle*). Pesan BYE dikirimkan untuk melakukan pemberhentian sesi dan apabila *user agent* akan mengirimkan pesan OK, setelah itu komunikasi yang dilakukan pun terputus atau selesai.



Gambar 2.53 Diagram pesan BYE dalam proses percakapan

2.16.9 Kualitas Layanan VoIP

Terdapat beberapa faktor yang mempengaruhi kualitas suara, yaitu waktu tunda (*delay*), variasi waktu tunda (*jitter*), dan tingkat paket hilang (*packet loss*). Ukuran dan pengalokasian kapasitas jaringan juga mempengaruhi kualitas VoIP secara keseluruhan.

Berikut penjelasan beberapa faktor yang mempengaruhi kualitas VoIP :

1. **Waktu Tunda (*delay*)**

Total waktu tunda merupakan penjumlahan dari waktu tunda pemrosesan, waktu tunda paketisasi, waktu tunda antrian, waktu tunda propagasi, dan waktu tunda akibat *jitter buffer* di sisi penerima. Waktu tunda sangat mempengaruhi kualitas layanan suara, karena pada dasarnya suara memiliki karakteristik ”*timing*”. Urutan pengucapan tiap suku kata yang ditransmisikan harus sampai ke sisi penerima dengan urutan yang sama pula sehingga dapat terdengar dengan baik secara *real-time.* ITU G.114 membagi karakteristik waktu tunda berdasarkan tingkat kenyamanan user, seperti pada Tabel 2.19.

Tabel 2.19 Pengelompokan waktu tunda berdasarkan ITU-T G.114

Waktu Tunda	Kualitas
0-150 ms	Baik
150-300 ms	Cukup
> 300 ms	Buruk

Ada beberapa komponen waktu tunda yang terjadi di jaringan. Komponen waktu tunda tersebut yaitu waktu tunda pemrosesan, waktu tunda paketisasi, waktu tunda propagasi, dan waktu tunda akibat adanya *jitter buffer* di *terminal* penerima.

Berikut ini penjelasan mengenai beberapa jenis waktu tunda yang dapat mempengaruhi kualitas layanan telepon internet:

1. Waktu Tunda Pemrosesan

Waktu tunda yang terjadi akibat proses pengumpulan dan pengkodean sampel analog menjadi digital. Waktu tunda ini tergantung pada jenis *codec* yang digunakan.

2. Waktu Tunda Paketisasi

Waktu tunda ini terjadi akibat proses paketisasi sinyal suara menjadi paket-paket yang siap ditransmisikan ke dalam jaringan.

3. Waktu Tunda Antrian

Waktu tunda yang disebabkan oleh antrian paket data akibat terjadinya *kongesti* jaringan.

4. Waktu Tunda Propagasi

Waktu tunda disebabkan oleh medium fisik jaringan dan jarak yang harus dilalui oleh sinyal suara pada media transmisi data antara pengirim dan penerima.

5. Waktu Tunda Akibat *Jitter Buffer*

Waktu tunda ini terjadi akibat adanya *jitter buffer* yang digunakan untuk meminimalisasi nilai *jitter* yang terjadi.

2. Variasi Waktu Tunda (Jitter)

Jitter merupakan perbedaan selang waktu kedatangan antar paket di terminal tujuan. *Jitter* dapat disebabkan oleh terjadinya *kongesti*, kurangnya kapasitas jaringan, variasi ukuran paket, serta ketidak urutan paket. Faktor ini perlu diperhitungkan karena karakteristik komunikasi *voice* adalah *sensitif* terhadap waktu tunda dan jitter.

Untuk meminimalisasi jitter dalam jaringan maka perlu diimplementasikan suatu *buffer* yang akan menahan beberapa urutan paket sepanjang waktu tertentu hingga paket terakhir datang. Namun adanya *buffer* tersebut akan mempengaruhi waktu tunda total sistem akibat adanya tambahan proses untuk mengompensasi jitter. Tabel 2.20 menjelaskan mengenai standar nilai *jitter* yang mempengaruhi kualitas layanan VoIP.

Tabel 2.20 Standar *Jitter*

Jitter	Kualitas
0-20 ms	Baik
20-50 ms	Cukup
> 50 ms	Buruk

3. Tingkat Paket Hilang (*Packet Loss*)

Sinyal suara pada telepon internet akan ditransmisikan dalam jaringan IP dalam bentuk paket-paket IP. Karena jaringan IP merupakan *best effort network* maka tidak ada jaminan pada pengiriman paket tersebut. Setiap paket dapat dirutekan pada jalur yang berbeda menuju penerima. Pada b*est effort network* tidak ada perbedaan antara paket data *voice* dengan paket-paket data lainnya yang mengalir di jaringan. Maka dari itu tentunya akan mempengaruhi kualitas layanan. Tabel 2.21 memperlihatkan standar tingkat paket hilang pada jaringan.

Tabel 2.21 Standar tingkat paket hilang

Tingkat Paket Hilang	Kualitas
0 - 1 %	Baik
1 -2 %	Cukup
> 2 %	Buruk

2.17 Pengukuran Kualitas VoIP

Ada dua pengujian yang biasa digunakan, yaitu uji *subyektif* dan uji *obyektif*. Uji *subyektif* dilakukan dengan cara melakukan *survey* terhadap sekelompok orang tentang bagaimana kualitas percakapan suara tersebut. Uji obyektif dilakukan dengan melakukan pengukuran-pengukuran seperti pengukuran waktu tunda. Namun hasil uji *obyektif* harus dibandingkan dengan hasil uji *subyektif*.

Uji *subyektif* dilakukan untuk mencari persepsi kualitas suara rata-rata dari suatu sistem. Uji ini dapat dilakukan dengan melakukan *survey* kepada sekelompok orang dan meminta pendapat mereka. Mereka diminta untuk menilai kualitas suara dengan memberikan suatu nilai misalnya antara 1 sampai 5. Kemudian dari hasil tersebut dapat dicari dari *Mean Opinion Score* (MOS). Hal yang membuat sulit dari pengujian ini adalah *subjektivitas* masing-masing orang berbeda menyebabkan sulit untuk menentukan kualitas sinyal suara.

Metode uji *obyektif* melakukan pengujian terhadap faktor-faktor kualitas layanan seperti yang telah disebutkan sebelumnya. Metode ini mudah dilakukan berulang-ulang, cepat, dan efisien sehingga cocok digunakan untuk pengujian dengan kombinasi parameter. Pada metode ini, aspek *fisiologi* dan persepsi manusia harus dimasukkan supaya menghasilkan hasil pengujian yang akurat. Sinyal masukan yang diberikan ke dalam pengujian ini harus memenuhi beberapa persyaratan. Pertama, sinyal masukan harus *di-filter* terlebih dahulu supaya sinyal tersebut sesuai dengan yang dibutuhkan oleh skema kompresi yang digunakan. Sinyal masukan yang berada di luar spesifikasi skema kompresi akan memberikan hasil yang tidak akurat. Kedua, sinyal tersebut harus memiliki panjang waktu

tertentu, tidak boleh terlalu singkat dan tidak boleh terlalu lama. Faktor terakhir adalah jenis suara yang digunakan. Jenis suara yang digunakan untuk pengujian haruslah sama.

2.17.1 *Mean Opinion Source* (MOS)

Merupakan sistem penilaian yang berhubungan dengan kualitas suara yang di dengar pada ujung pesawat penerima. Standar penilaian MOS dikeluarkan oleh ITU- T pada tahun 1996. Tabel 2.15 adalah tabel yang menunjukkan skala penilaian MOS.

MOS memberikan penilaian kualitas suara dengan skala 1 sampai 5, di mana satu mempresentasikan nilai kualitas suara yang paling buruk dan lima mempresentasikan kualitas suara yang paling baik. Penilaian dengan menggunakan MOS masih bersifat *subyektif* karena kualitas pendengaran dan pendapat dari masing-masing pendengar berbeda-beda.

Tabel 2.22 Skala penilaian MOS

Kualitas Percakapan	Nilai
Sangat Baik (*excellent*)	5
Baik (*good*)	4
Cukup Baik (*fair*)	3
Kurang Baik (*poor*)	2
Buruk (*bad*)	1