BAB 2

TINJAUAN PUSTAKA

2.1. Jaringan Komputer

Jaringan Komputer adalah kumpulan dua komputer atau lebih yang secara cerdas saling berbagi-pakai alat-alat *hardware* dan *software*. Jaringan bisa terbentuk dari minimal dua komputer sederhana dan kecil yang saling berbagi-pakai printer dan CD-ROM yang terpasang pada salah satu komputer. Namun jaringan bisa berbentuk jalinan saluran komputer terbesar di dunia yaitu internet.

2.2. Klasifikasi Jaringan Komputer

Dalam mempelajari macam-macam jaringan komputer terdapat dua klasifikasi yang sangat penting yaitu teknologi transmisi dan jarak. Secara garis besar, terdapat dua jenis teknologi transmisi yaitu jaringan broadcast dan jaringan point-to-point.

Jaringan broadcast memiliki saluran komunikasi tunggal yang dipakai bersama-sama oleh semua mesin yang ada pada jaringan.

Pesan-pesan berukuran kecil, disebut paket, yang dikirimkan oleh suatu mesin akan diterima oleh mesin-mesin lainnya. Field alamat pada sebuah paket berisi keterangan tentang kepada siapa paket tersebut ditujukan. Saat menerima paket, mesin akan mencek field alamat. Bila paket terserbut ditujukan untuk dirinya, maka mesin akan memproses paket itu , bila paket ditujukan untuk mesin lainnya, mesin terserbut akan mengabaikannya.

Jaringan point-to-point terdiri dari beberapa koneksi pasangan individu dari mesin-mesin. Untuk mengirim paket dari sumber ke suatu tujuan, sebuah paket pada jaringan jenis ini mungkin harus melalui satu atau lebih mesin-mesin perantara. Seringkali harus melalui banyak rute yang mungkin berbeda jaraknya. Karena itu algoritma rute memegang peranan penting pada jaringan *point-to-point.*

Pada umumnya jaringan yang lebih kecil dan terlokalisasi secara geografis cendurung memakai *broadcasting*, sedangkan jaringan yang lebih besar menggunakan *point-to-point*. Kriteria alternatif untuk mengklasifikasikan jaringan adalah didasarkan pada jaraknya. Tabel berikut ini menampilkan klasifikasi sistem *multiprosesor* berdasarkan ukuran-ukuran fisiknya.

Tabel 2.1 Klasifikasi prosesor interkoneksi berdasarkan jarak

Jarak antar prosesor	Prosesor di tempat yang sama	Contoh
0,1 m	Papan rangkaian	Data flow machine
1 m	Sistem	Multicomputer
10 m	Ruangan	
100 m	Gedung	Local Area Network
1 km	Kampus	

10 km	Kota	Metropolitan Area Network
100 km	Negara	Wide area Network
1.000 km	Benua	
10.000 km	Planet	The Internet

Dari tabel di atas terlihat pada bagian paling atas adalah *data flow machine*, komputer-komputer yang sangat paralel yang memiliki beberapa unit fungsi yang semuanya bekerja untuk program yang sama. Kemudian *multicomputer*, sistem yang berkomunikasi dengan cara mengirim pesan-pesannya melalui bus pendek dan sangat cepat. Setelah kelas *multicomputer* adalah jaringan sejati, komputer-komputer yang bekomunikasi dengan cara bertukar data/pesan melalui kabel yang lebih panjang. Jaringan seperti ini dapat dibagi *menjadi local area network* (LAN), *metropolitan area network* (MAN), dan *wide area network* (WAN). Akhirnya, koneksi antara dua jaringan atau lebih disebut *internetwork*. *Internet* merupakan salah satu contoh yang terkenal dari suatu *internetwork*.

A. Berdasarkan Skala

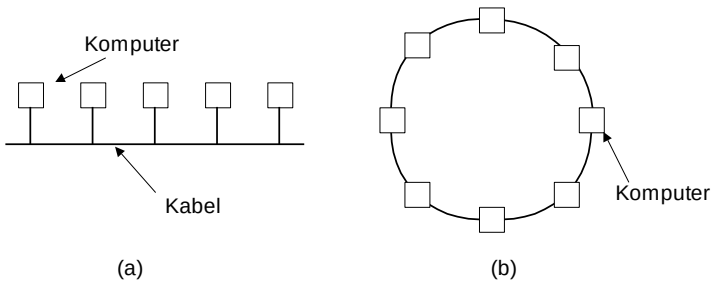
1. *Local Area Network (LAN)*

LAN (*Local Area Network*) adalah jaringan komputer yang jaringannya hanya mencakup wilayah kecil. Seperti jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil. Saat ini, kebanyakan LAN berbasis pada teknologi *IEEE 802.3 Ethernet* menggunakan perangkat *switch*, yang mempunyai

kecepatan transfer data 10, 100, atau 1000 Mbps. Selain teknologi *Ethernet*, saat ini teknologi 802.11b (atau biasa disebut *Wi-fi*) juga sering digunakan untuk membentuk LAN.

LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor perusahaan atau pabrik-pabrik untuk memakai bersama *resource* (misalnya, printer, scanner) dan saling bertukar informasi. LAN dapat dibedakan dari jenis jaringan lainnya berdasarkan tiga karakteristik: ukuran, teknologi transmisi dan topologinya.

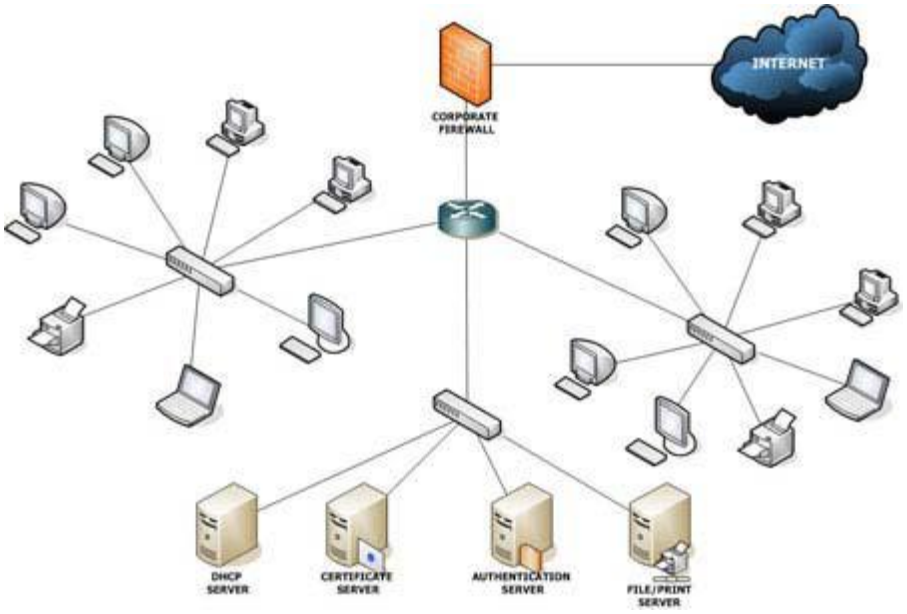
LAN mempunyai ukuran yang terbatas, yang berarti bahwa waktu transmisi pada keadaan terburuknya terbatas dan dapat diketahui sebelumnya. Dengan mengetahui keterbatasannya, menyebabkan adanya kemungkinan untuk menggunakan jenis desain tertentu. Hal ini juga memudahkan manajemen jaringan.



Gambar 2.1 Dua jenis jaringan *broadcast. (a) Bus. (b) Ring*

LAN seringkali menggunakan teknologi transmisi kabel tunggal. LAN tradisional beroperasi pada kecepatan mulai 10 sampai 100 Mbps (mega bit/detik) dengan delay rendah (puluhan mikro second) dan mempunyai faktor kesalahan yang

kecil. LAN-LAN modern dapat beroperasi pada kecepatan yang lebih tinggi, sampai ratusan megabit/detik.



Gambar 2.2 *Local Area Network*

Terdapat beberapa macam topologi yang dapat digunakan pada LAN *broadcast*. Pada jaringan bus (yaitu kabel liner), pada suatu saat sebuah mesin bertindak sebagai master dan dijinkan untuk mengirim paket. Mesin-mesin lainnya perlu menahan diri untuk tidak mengirimkan apapun. Maka untuk mencegah terjadinya konflik, ketika dua mesin atau lebih ingin mengirikan secara bersamaan, maka mekanisme pengatur diperlukan. Mekanisme pengatur dapat berbentuk tersentralisasi atau terdistribusi. IEEE 802.3 yang populer disebut Ethernet merupakan jaringan *broadcast bus* dengan pengendali terdesentralisasi yang beroperasi pada kecepatan 10 s.d. 100 Mbps. Komputer-komputer pada Ethernet dapat mengirim kapan saja mereka inginkan, bila dua buah paket atau lebih

bertabrakan, maka masing-masing komputer cukup menunggu dengan waktu tunggu yang acak sebelum mengulangi lagi pengiriman.

Sistem *broadcast* yang lain adalah *ring*, pada topologi ini setiap bit dikirim ke daerah sekitarnya tanpa menunggu paket lengkap diterima. Biasanya setiap bit mengelilingi ring dalam waktu yang dibutuhkan untuk mengirimkan beberapa bit, bahkan seringkali sebelum paket lengkap dikirim seluruhnya. Seperti sistem *broadcast* lainnya, beberapa aturan harus dipenuhi untuk mengendalikan *access* simultan ke ring. IEEE 802.5 (token ring) merupakan LAN ring yang populer yang beroperasi pada kecepatan antara 4 s.d 16 Mbps.

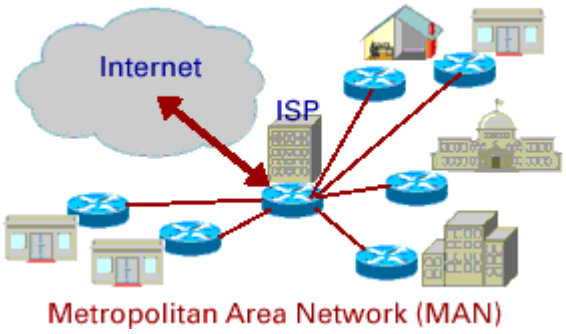
Berdasarkan alokasi *channel*nya, jaringan *broadcast* dapat dibagi menjadi dua, yaitu statik dan dinamik. Jenis alokasi statik dapat dibagi berdasarkan waktu interval-interval diskrit dan algoritma round robin, yang mengijinkan setiap mesin untuk melakukan *broadcast* hanya bila slot waktunya sudah diterima. Alokasi statik sering menyia-nyiakan kapasitas channel bila sebuah mesin tidak punya lagi yang perlu dikerjakan pada saat slot alokasinya diterima. Karena itu sebagian besar sistem cenderung mengalokasi *channel*-nya secara dinamik (yaitu berdasarkan kebutuhan).

Metoda alokasi dinamik bagi suatu *channel* dapat tersentralisasi ataupun terdesentralisasi. Pada metoda alokasi *channel* tersentralisasi terdapat sebuah *entity* tunggal, misalnya unit bus pengatur, yang menentukan siapa giliran berikutnya. Pengiriman paket ini bisa dilakukan setelah menerima giliran dan membuat keputusan yang berkaitan dengan algoritma internal. Pada metoda alokasi *channel*

terdesentralisasi, tidak terdapat *entity* sentral, setiap mesin harus dapat menentukan dirinya sendiri kapan bisa atau tidaknya mengirim.

2. *Metropolitan Area Network (MAN)*

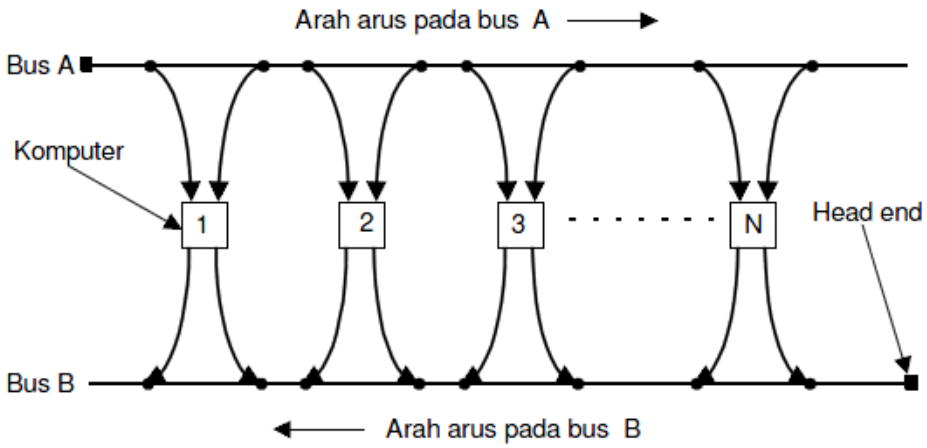
Metropolitan Area Network (MAN) pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya memakai teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang berdekatan dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. MAN biasanya mampu menunjang data dan suara, dan bahkan dapat berhubungan dengan jaringan televisi kabel. MAN hanya memiliki sebuah atau dua buah kabel dan tidak mempunyai elemen *switching*, yang berfungsi untuk mengatur paket melalui beberapa output kabel. Adanya elemen *switching* membuat rancangan menjadi lebih sederhana.



Gambar 2.3 Jaringan MAN

Alasan utama memisahkan MAN sebagai kategori khusus adalah telah ditentukannya *standart* untuk MAN, dan *standart* ini sekarang sedang diimplementasikan. *Standart* tersebut disebut DQDB (*Distributed Queue Dual Bus*) atau 802.6 menurut standart IEEE. DQDB terdiri dari dua buah kabel *unidirectional*

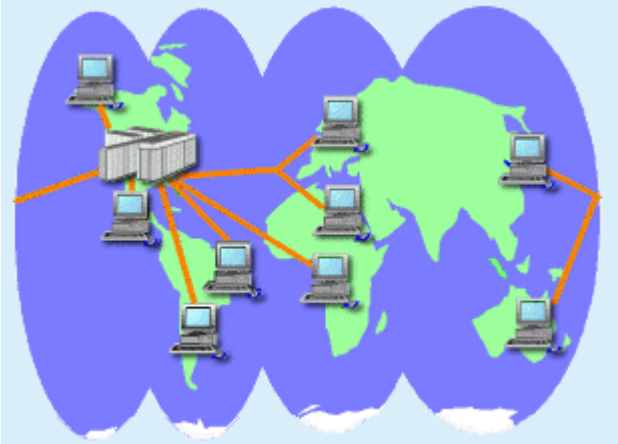
dimana semua komputer dihubungkan. Setiap *bus* mempunyai sebuah *head–end*, perangkat untuk memulai aktivitas *transmisi*. Lalu-lintas yang menuju komputer yang berada di sebelah kanan pengirim menggunakan *bus* bagian atas. Lalu-lintas ke arah kiri menggunakan *bus* yang berada di bawah.



Gambar 2.4 Arsitektur MAN DQDB

3. *Wide Area Network (WAN)*

WAN (*Wide Area Network)* merupakan jaringan komputer yang mencakup area yang besar (*wide). Wide Area Network* (WAN) mencakup daerah geografis yang luas, sertingkali mencakup sebuah negara atau benua. WAN terdiri dari kumpulan mesin yang bertujuan untuk mejalankan program-program aplikasi. Sebagai contoh yaitu jaringan komputer antar wilayah, kota atau bahkan negara, atau dapat didefinisikan juga sebagai jaringan komputer yang membutuhkan *router* dan saluran komunikasi publik. WAN digunakan untuk menghubungkan jaringan lokal yang satu dengan jaringan lokal yang lain, sehingga pengguna atau komputer di lokasi yang satu dapat berkomunikasi dengan pengguna dan komputer di lokasi yang lain.



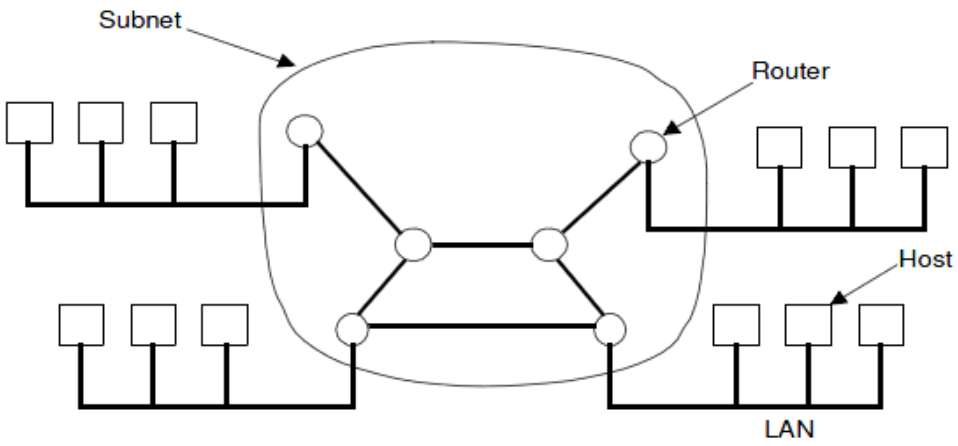
Gambar 2.5 Jaringan WAN

Kita akan mengikuti penggunaan tradisional dan menyebut mesin-mesin ini sebagai *host*. Istilah *End System* kadang-kadang juga digunakan dalam literatur. *Host* dihubungkan dengan sebuah subnet komunikasi, atau cukup disebut subnet. Tugas subnet adalah membawa pesan dari *host* ke *host* lainnya, seperti halnya sistem telepon yang membawa isi pembicaraan dari pembicara ke pendengar. Dengan memisahkan aspek komunikasi murni sebuah jaringan (*subnet*) dari aspek-aspek aplikasi (*host*), rancangan jaringan lengkap menjadi jauh lebih sederhana.

Pada sebagian besar WAN, subnet terdiri dari dua komponen, yaitu kabel transmisi dan elemen *switching.* Kabel transmisi (disebut juga sirkuit, *channel*, atau *trunk*) memindahkan bit-bit dari satu mesin ke mesin lainnya.

Element *switching* adalah komputer khusus yang dipakai untuk menghubungkan dua kabel transmisi atau lebih. Saat data sampai ke kabel penerima, element *switching* harus memilih kabel pengirim untuk meneruskan pesan-pesan tersebut. Sayangnya tidak ada terminologi *standart* dalam menamakan komputer

seperti ini. Namanya sangat bervariasi disebut paket *switching node*, intermidiate system, data *switching exchange* dan sebagainya.

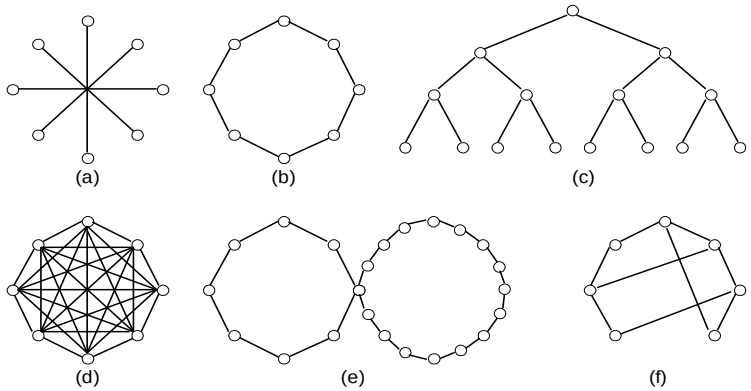


Gambar 2.6 Hubungan antara *host-host* dengan *subnet*

Sebagai istilah generik bagi komputer *switching*, kita akan menggunakan istilah *router*. Tapi perlu diketahui terlebih dahulu bahwa tidak ada konsensus dalam penggunaan terminologi ini. Dalam model ini, seperti ditunjukkan oleh Gambar 2.6 setiap *host* dihubungkan ke LAN tempat dimana terdapat sebuah *router*, walaupun dalam beberapa keadaan tertentu sebuah *host* dapat dihubungkan langsung ke sebuah *router*. Kumpulan saluran komunikasi dan *router* (tapi bukan *host*) akan membentuk *subnet.*

Istilah subnet sangat penting, tadinya subnet berarti kumpulan kumpulan routerrouter dan saluran-saluran komunikasi yang memindahkan paket dari *host- host* tujuan. Akan tatapi, beberapa tahun kemudian *subnet* mendapatkan arti lainnya sehubungan dengan pengalamatan jaringan.

Pada sebagian besar WAN, jaringan terdiri dari sejumlah banyak kabel atau saluran telepon yang menghubungkan sepasang *router*. Bila dua router yang tidak mengandung kabel yang sama akan melakukan komunikasi, keduanya harus berkomunikasi secara tak langsung melalui *router* lainnya. Ketika sebuah paket dikirimkan dari sebuah router ke *router* lainnya melalui *router* perantara atau lebih, maka paket akan diterima *router* dalam keadaan lengkap, disimpan sampai saluran output menjadi bebas, dan kemudian baru diteruskan.



Gambar 2.7 bebarapa topologi *subnet untuk poin-to-point* .

(a)Bintang (b)Cincin (c)Pohon (d)Lengkap (e) Cincin berinteraksi (f)Sembarang.

Subnet yang mengandung prinsip seperti ini disebut *subnet point-to-point*, *store-and-forward*, atau *packet-switched*. Hampir semua WAN (kecuali yang menggunakan satelit) memiliki *subnet store-and-forward*.

Di dalam menggunakan *subnet point-to-point*, masalah rancangan yang penting adalah pemilihan jenis topologi *interkoneksi router*. Gambar 2.7 menjelaskan beberapa kemungkinan topologi. LAN biasanya berbentuk topologi simetris, sebaliknya WAN umumnya bertopologi tak menentu.

2.2.1. Jaringan Tanpa Kabel

Komputer *mobile* seperti komputer *notebook* dan personal *digital assistant* (PDA), merupakan cabang industri komputer yang paling cepat pertumbuhannya. Banyak pemilik jenis komputer tersebut yang sebenarnya telah memiliki mesin-mesin desktop yang terpasang pada LAN atau WAN tetapi karena koneksi kabel tidaklah mungkin dibuat di dalam mobil atau pesawat terbang, maka banyak yang tertarik untuk memiliki komputer dengan jaringan tanpa kabel ini.

Jaringan tanpa kabel mempunyai berbagai manfaat, yang telah umum dikenal adalah kantor *portable*. Orang yang sedang dalam perjalanan seringkali ingin menggunakan peralatan elektronik portable-nya untuk mengirim atau menerima telepon, fax, e-mail, membaca *fail* jarak jauh login ke mesin jarak jauh, dan sebagainya dan juga ingin melakukan hal-hal tersebut dimana saja, darat, laut, udara. Jaringan tanpa kabel sangat bermanfaat untuk mengatasi masalah-masalah di atas.

Wireless	*Mobile*	Aplikasi
Tidak	Tidak	Worksation tetap di kantor
Tidak	Ya	Komputer portable terhubung ke len telepon
Ya	Tidak	LAN dengan komunikasi wireless
Ya	Ya	Kantor portable, PDA untuk persediaan

Tabel 2.2 Kombinasi jaringan tanpa kabel dan komputasi mobile

Walaupun jaringan tanpa kabel dan sistem komputasi yang dapat berpindah-pindah sering kali berkaitan erat, sebenarnya tidaklah sama, seperti yang tampak pada

tabel 2.2. Komputer *portabel* kadang-kadang menggunakan kabel juga, yaitu disaat seseorang yang sedang dalam perjalanan menyambungkan komputer *portable*-nya ke jack telepon di sebuah hotel, maka kita mempunyai mobilitas yang bukan jaringan tanpa kabel. Sebaliknya, ada juga komputer-komputer yang menggunakan jaringan tanpa kabel tetapi bukan *portabel*, hal ini dapat terjadi disaat komputer-komputer tersebut terhubung pada LAN yang menggunakan fasilitas komunikasi *wireless* (radio).

Meskipun jaringan tanpa kabel ini cukup mudah untuk di pasang, tetapi jaringan macam ini memiliki banyak kekurangan. Biasanya jaringan tanpa kabel mempunyai kemampuan 1-2 Mbps, yang mana jauh lebih rendah dibandingkan dengan jaringan berkabel. Laju kesalahan juga sering kali lebih besar, dan transmisi dari komputer yang berbeda dapat mengganggu satu sama lain.

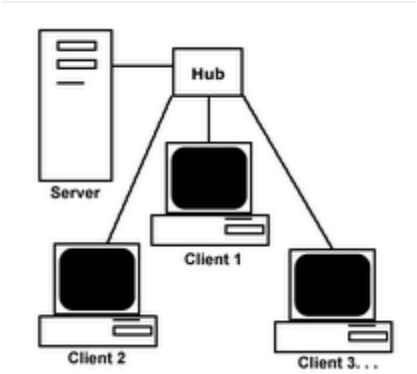
B. Berdasarkan fungsi

Pada dasarnya setiap jaringan komputer ada yang berfungsi sebagai *client* dan juga *server*. Tetapi ada jaringan yang memiliki komputer yang khusus didedikasikan sebagai *server* sedangkan yang lain sebagai *client*. Ada juga yang tidak memiliki komputer yang khusus berfungsi sebagai *server* saja. Karena itu berdasarkan fungsinya maka ada dua jenis jaringan komputer :

1. *Client-server*

Yaitu jaringan komputer dengan komputer yang didedikasikan khusus sebagai *server*. Sebuah *service* atau layanan bisa diberikan oleh sebuah komputer atau lebih. Seperti satu komputer sebagai *server* yang menangani berbagai *multi service* yaitu

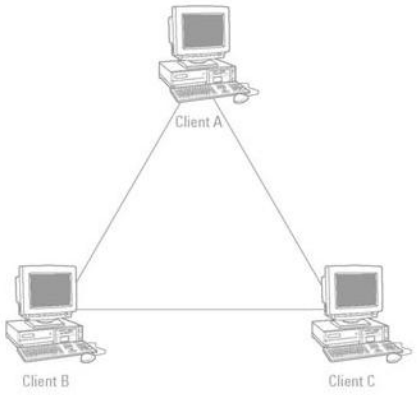
mail server, web server, file server, database server dan lainnya. Jaringan komputer *Client Server* ditunjukkan oleh Gambar 2.8.



Gambar 2.8 Jaringan *Client-server*

2. *Peer-to-peer*

Yaitu jaringan komputer dimana setiap *host* dapat menjadi *server* dan juga menjadi *client* secara bersamaan. *Peer to peer* bisa dilihat pada Gambar 2.9.



Gambar 2.9 Jaringan Peer-to-Peer

C. Berdasarkan topologi jaringan

Topologi Jaringan adalah gambaran secara fisik dari pola hubungan antara komponenkomponen jaringan, yang meliputi *server*, *workstation*, *hub* dan

pengkabelannnya. Terdapat tiga macam topologi jaringan umum digunakan, yaitu *Bus, Star* dan *Ring*.

1. *Topologi Bus*

Pada topologi *Bus* digunakan sebuah kabel tunggal atau kabel pusat di mana seluruh *workstation* dan *server* dihubungkan. Keunggulan topologi *Bus* adalah pengembangan jaringan atau penambahan *workstation* baru dapat dilakukan dengan mudah tanpa mengganggu *workstation* lain. Kelemahan dari topologi ini adalah bila terdapat gangguan di sepanjang kabel pusat maka keseluruhan jaringan akan mengalami gangguan.

2. *Topologi Star*

Pada topologi *Star*, masing-masing *workstation* ataupun *server* dihubungkan secara langsung ke *konsentrator*. Keunggulan dari topologi tipe *star* ini adalah bahwa dengan adanya kabel tersendiri untuk setiap *workstation* ke *server*, maka *bandwidth* atau lebar jalur komunikasi dalam kabel akan semakin lebar sehingga akan meningkatkan unjuk kerja jaringan secara keseluruhan. Bila terdapat gangguan di suatu jalur kabel maka gangguan hanya akan terjadi dalam komunikasi antara *workstation* yang bersangkutan dengan *server*, jaringan secara keseluruhan tidak mengalami gangguan. Kelemahan dari topologi *Star* adalah kebutuhan kabel yang lebih besar dibandingkan dengan topologi lainnya.

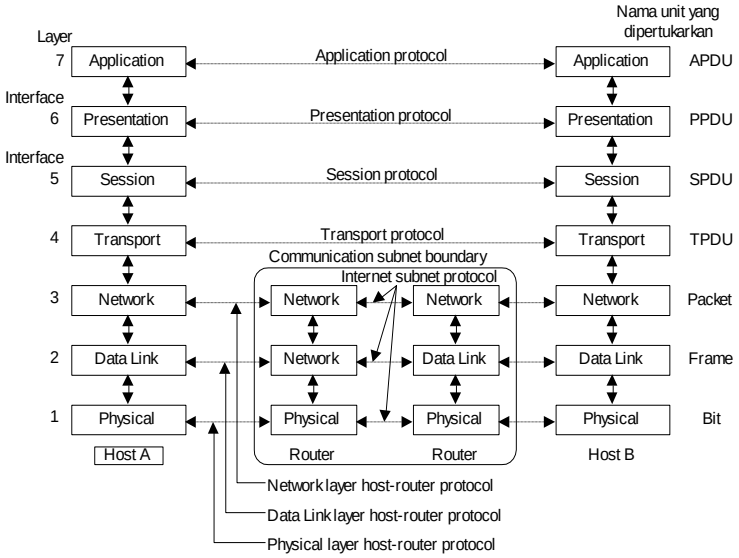
3. Topologi Ring

Di dalam topologi Ring semua *workstation* dan *server* dihubungkan sehingga terbentuk suatu pola lingkaran atau *ring*. Tiap *workstation* ataupun *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.

Kelemahan dari topologi ini adalah setiap *node* dalam jaringan akan selalu ikut serta mengelola informasi yang dilewatkan dalam jaringan, sehingga bila terdapat gangguan di suatu *node* maka seluruh jaringan akan terganggu. Keunggulan topologi Ring adalah tidak terjadinya *collision* atau tabrakan pengiriman data seperti pada topologi *Bus*, karena hanya satu *node* dapat mengirimkan data pada suatu saat.

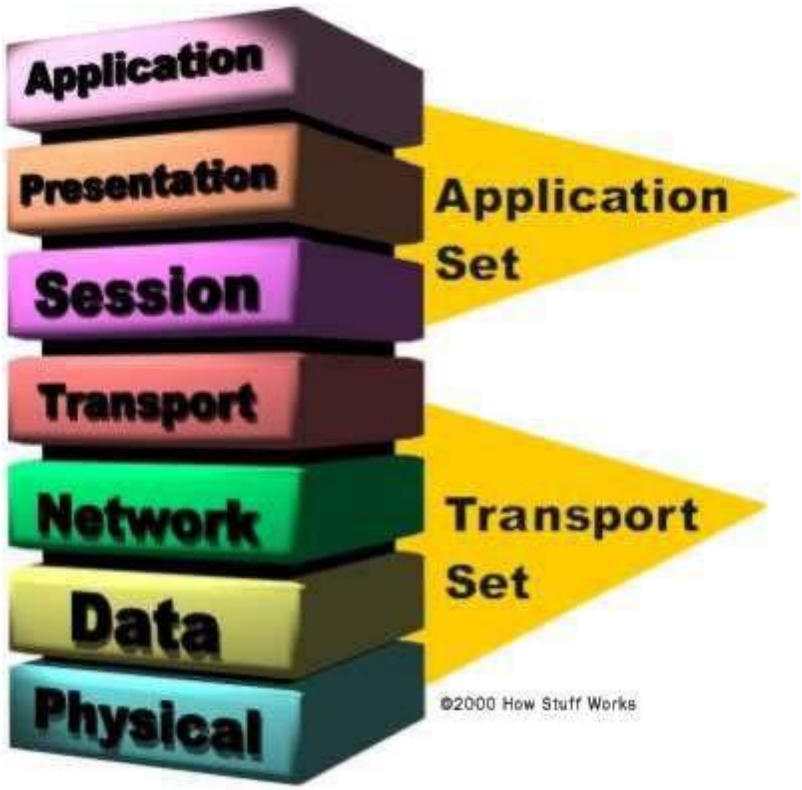
2.3. Protokol Jaringan Komputer

2.3.1. Model Referensi OSI



Gambar 2.10 Model Referensi OSI

Model Referensi OSI merupakan salah satu aturan standar yang dikeluarkan oleh badan pembuat aturan dan standar untuk komunikasi komputer bernama *International Standard Organization* (ISO). 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 terhubungnya dua sistem yang berbeda yang berasal dari perangkat keras jaringan komputer yang berbeda pula. Jadi tujuan OSI ini adalah untuk memfasilitasi bagaimana suatu komunikasi dapat terjalin dari sistem yang berbeda tanpa memerlukan perubahan yang signifikan pada perangkat keras dan perangkat lunak. Model referensi OSI bisa dilihat pada Gambar 2.11.



Gambar 2.10 Lapisan OSI Tujuh Layer

Model OSI disusun atas 7 lapisan :

1. *Physical Layer (Lapisan Fisik).*

Lapisan Fisik merupakan lapisan atau level yang paling rendah dari model OSI yang berhubungan dengan media fisik atau peralatan fisik sebagai prosedur standar dalam jaringan komunikasi data. Jadi di sini berisi perangkat keras yang digunakan dalam koneksi antara komputer dengan *network*. Termasuk di dalamnya adalah kabel, karakteristik tegangan listrik dan arus listrik.

Physical Layer berfungsi dalam pengiriman *raw bit* ke *channel* komunikasi. Masalah desain yang harus diperhatikan disini adalah memastikan bahwa bila satu sisi mengirim data 1 bit, data tersebut harus diterima oleh sisi lainnya sebagai 1 bit

pula, dan bukan 0 bit. Pertanyaan yang timbul dalam hal ini adalah : berapa volt yang perlu digunakan untuk menyatakan nilai 1? Dan berapa *volt* pula yang diperlukan untuk angka 0?. Diperlukan berapa mikrosekon suatu bit akan habis? Apakah transmisi dapat diproses secara simultan pada kedua arahnya? Berapa jumlah pin yang dimiliki jaringan dan apa kegunaan masing-masing pin? Secara umum masalah-masalah desain yang ditemukan di sini berhubungan secara mekanik, elektrik dan interface prosedural, dan media fisik yang berada di bawah physical layer.

2. *Data Link Layer (Lapisan Data Link)*

Lapisan ini melayani transmisi pada lapisan fisik dan bertanggung jawab akan data yang dikirim dari *host* ke *network.* Lapisan *data link* ini menjamin agar data yang dikirimkan sampai ke tujuan dalam keadaan baik. Karena di layer ini protokol harus mampu mendeteksi kesalahan pada pengiriman data. Sehingga pengiriman data yang *unreliable* harus tertera serta informasi akan suksesnya pengiriman data juga harus terdapat pada *layer* ini.

Tugas utama *data link layer* adalah sebagai fasilitas transmisi *raw data* dan mentransformasi data tersebut ke saluran yang bebas dari kesalahan transmisi. Sebelum diteruskan kenetwork layer, data link layer melaksanakan tugas ini dengan memungkinkan pengirim memecag-mecah data input menjadi sejumlah data frame (biasanya berjumlah ratusan atau ribuan byte). Kemudian *data link layer* mentransmisikan *frame* tersebut secara berurutan, dan memproses *acknowledgement frame* yang dikirim kembali oleh penerima. Karena *physical layer* menerima dan mengirim aliran bit tanpa mengindahkan arti atau arsitektur *frame,* maka tergantung

pada *data link layer*-lah untuk membuat dan mengenali batas-batas frame itu. Hal ini bisa dilakukan dengan cara membubuhkan bit khusus ke awal dan akhir *frame*. Bila secara insidental pola-pola bit ini bisa ditemui pada data, maka diperlukan perhatian khusus untuk menyakinkan bahwa pola tersebut tidak secara salah dianggap sebagai batas-batas *frame.*

Terjadinya *noise* pada saluran dapat merusak *frame.* Dalam hal ini, perangkat lunak *data link layer* pada mesin sumber dapat mengirim kembali *frame* yang rusak tersebut. Akan tetapi transmisi *frame* sama secara berulang-ulang bisa menimbulkan duplikasi *frame*. *Frame* duplikat perlu dikirim apabila *acknowledgement frame* dari penerima yang dikembalikan ke pengirim telah hilang. Tergantung pada *layer* inilah untuk mengatasi masalah-masalah yang disebabkan rusaknya, hilangnya dan duplikasi *frame*. *Data link layer* menyediakan beberapa kelas layanan bagi *network layer*. Kelas layanan ini dapat dibedakan dalam hal kualitas dan harganya.

Masalah-masalah lainnya yang timbul pada *data link layer* (dan juga sebagian besar layer-layer di atasnya) adalah mengusahakan kelancaran proses pengiriman data dari pengirim yang cepat ke penerima yang lambat. Mekanisme pengaturan lalu-lintas data harus memungkinkan pengirim mengetahui jumlah ruang *buffer* yang dimiliki penerima pada suatu saat tertentu. Seringkali pengaturan aliran dan penanganan *error* ini dilakukan secara terintegrasi.

Saluran yang dapat mengirim data pada kedua arahnya juga bisa menimbulkan masalah. Sehingga dengan demikian perlu dijadikan bahan pertimbangan bagi *software data link layer*. Masalah yang dapat timbul di sini adalah bahwa *frame*-

frame acknoeledgement yang mengalir dari A ke B bersaing saling mendahului dengan aliran dari B ke A. Penyelesaian yang terbaik (*piggy backing*) telah bisa digunakan.

Jaringan *broadcast* memiliki masalah tambahan pada data *link layer*. Masalah tersebut adalah dalam hal mengontrol akses ke saluran yang dipakai bersama. Untuk mengatasinya dapat digunakan sublayer khusus *data link layer*, yang disebut *medium access sublayer*.

3. *Network Layer (Lapisan Network)*

Lapisan *Network* bertanggung jawab membuat paket data yang dikirimkan dan *data* akan diberi informasi mengenai *address* dan *routing* tujuan. Serta melakukan pengontrolan aliran data pada komputer ke *interface* jaringan. Pada lapisan *Network* ini juga harus dapat menangani cara-cara pengalamatan oleh sebuah jaringan yang berbeda-beda serta mengatur paket-paket data yang memiliki ukuran yang berbeda. Selain itu *layer* ini juga harus merespon akan *network congestion problems*.

Network layer berfungsi untuk pengendalian operasi *subnet*. Masalah desain yang penting adalah bagaimana caranya menentukan route pengiriman paket dari sumber ke tujuannya. Route dapat didasarkan pada table statik yang “dihubungkan ke” network. Rute juga dapat ditentukan pada saat awal percakapan misalnya *session terminal*. Terakhir, rute dapat juga sangat dinamik, dapat berbeda bagi setiap paketnya. Oleh karena itu, rute pengiriman sebuah paket tergantung beban jaringan saat itu.

Bila pada saat yang sama dalam sebuah *subnet* terdapat terlalu banyak paket, maka ada kemungkinan paket-paket tersebut tiba pada saat yang bersamaan. Hal ini dapat menyebabkan terjadinya *bottleneck*. Pengendalian kemacetan seperti itu juga merupakan tugas *network layer*.

Karena operator *subnet* mengharap bayaran yang baik atas tugas pekerjaannya. Seringkali terdapat beberapa fungsi *accounting* yang dibuat pada *network layer*. Untuk membuat informasi tagihan, setidaknya *software* mesti menghitung jumlah paket atau karakter atau bit yang dikirimkan oleh setiap pelanggannya. *Accounting* menjadi lebih rumit, bilamana sebuah paket melintasi batas negara yang memiliki tarip yang berbeda.

Perpindahan paket dari satu jaringan ke jaringan lainnya juga dapat menimbulkan masalah yang tidak sedikit. Cara pengalamatan yang digunakan oleh sebuah jaringan dapat berbeda dengan cara yang dipakai oleh jaringan lainnya. Suatu jaringan mungkin tidak dapat menerima paket sama sekali karena ukuran paket yang terlalu besar. Protokolnya pun bisa berbeda pula, demikian juga dengan yang lainnya. *Network layer* telah mendapat tugas untuk mengatasi semua masalah seperti ini, sehingga memungkinkan jaringan-jaringan yang berbeda untuk saling terinterkoneksi.

4. *Transport Layer (Lapisan Transport)*

Fungsi dasar dari lapisan *transport* adalah menerima data dari lapisan *sessio*n, memisahkan menjadi bagian atau unit yang kecil dan meneruskan ke lapisan jaringan dan menjamin unit-unit data tersebut sampai dengan benar dan tentu saja data itu

tidak rusak. Lapisan transport juga menentukan bentuk layanan apa yang disediakan lapisan *session*.

Fungsi dasar transport layer adalah menerima data dari session layer, memecah data menjadi bagian-bagian yang lebih kecil bila perlu, meneruskan data ke network layer, dan menjamin bahwa semua potongan data tersebut bisa tiba di sisi lainnya dengan benar. Selain itu, semua hal tersebut harus dilaksanakan secara efisien, dan bertujuan dapat melindungi layer-layer bagian atas dari perubahan teknologi *hardware* yang tidak dapat dihindari.

Dalam keadaan normal, transport layer membuat koneksi jaringan yang berbeda bagi setiap koneksi transport yang diperlukan oleh session layer. Bila koneksi transport memerlukan throughput yang tinggi, maka transport layer dapat membuat koneksi jaringan yang banyak. Transport layer membagi-bagi pengiriman data ke sejumlah jaringan untuk meningkatkan *throughput*. Di lain pihak, bila pembuatan atau pemeliharaan koneksi jaringan cukup mahal, transport layer dapat menggabungkan beberapa koneksi transport ke koneksi jaringan yang sama. Hal tersebut dilakukan untuk membuat penggabungan ini tidak terlihat oleh session layer.

Transport layer juga menentukan jenis layanan untuk *session layer*, dan pada gilirannya jenis layanan bagi para pengguna jaringan. Jenis transport layer yang paling populer adalah saluran *error-free point to point* yang meneruskan pesan atau *byte* sesuai dengan urutan pengirimannya. Akan tetapi, terdapat pula jenis layanan *transport* lainnya. Layanan tersebut adalah *transport* pesan terisolasi yang tidak

menjamin urutan pengiriman, dan membroadcast pesan-pesan ke sejumlah tujuan. Jenis layanan ditentukan pada saat koneksi dimulai.

Transport layer merupakan *layer end to end* sebenarnya, dari sumber ke tujuan. Dengan kata lain, sebuah program pada mesin sumber membawa percakapan dengan program yang sama dengan pada mesin yang dituju. Pada layer-layer bawah, protokol terdapat di antara kedua mesin dan mesin-mesin lain yang berada didekatnya. Protokol tidak terdapat pada mesin sumber terluar atau mesin tujuan terluar, yang mungkin dipisahkan oleh sejumlah router. Perbedaan antara layer 1 sampai 3 yang terjalin, dan layer 4 sampai 7 yang *end to end*. Hal ini dapat dijelaskan seperti pada gambar 2-1.

Sebagai tambahan bagi penggabungan beberapa aliran pesan ke satu *channel*, *transport layer* harus hati-hati dalam menetapkan dan memutuskan koneksi pada jaringan. Proses ini memerlukan mekanisma penamaan, sehingga suatu proses pada sebuah mesin mempunyai cara untuk menerangkan dengan siapa mesin itu ingin bercakap-cakap. Juga harus ada mekanisme untuk mengatur arus informasi, sehingga arus informasi dari host yang cepat tidak membanjiri host yang lambat. Mekanisme seperti itu disebut pengendalian aliran dan memainkan peranan penting pada *transport layer* (juga pada layer-layer lainnya). Pengendalian aliran antara *host* dengan host berbeda dengan pengendalian aliran *router* dengan *router*. Kita akan mengetahui nanti bahwa prinsip-prinsip yang sama digunakan untuk kedua jenis pengendalian tersebut.

5. *Session Layer (Lapisan Session)*

Lapisan session menyediakan fasilitas pada *user* untuk melakukan percakapan / dialog dari satu komputer ke komputer yang lainnya.

Session layer mengijinkan para pengguna untuk menetapkan session dengan pengguna lainnya. Sebuah session selain memungkinkan *transport* data biasa, seperti yang dilakukan oleh *transport layer*, juga menyediakan layanan yang istimewa untuk aplikasi-aplikasi tertentu. Sebuah session digunakan untuk memungkinkan seseorang pengguna *log* ke *remote timesharing system* atau untuk memindahkan *file* dari satu mesin kemesin lainnya.

Sebuah layanan *session layer* adalah untuk melaksanakan pengendalian dialog. *Session* dapat memungkinkan lalu lintas bergerak dalam bentuk dua arah pada suatu saat, atau hanya satu arah saja. Jika pada satu saat lalu lintas hanya satu arah saja (analog dengan rel kereta api tunggal), session layer membantu untuk menentukan giliran yang berhak menggunakan saluran pada suatu saat.

Layanan session di atas disebut manajemen *token.* Untuk sebagian protokol, adalah penting untuk memastikan bahwa kedua pihak yang bersangkutan tidak melakukan operasi pada saat yang sama. Untuk mengatur aktivitas ini, session layer menyediakan token-token yang dapat digilirkan. Hanya pihak yang memegang token yang dijinkan melakukan operasi kritis.

Layanan *session* lainnya adalah sinkronisasi. Ambil contoh yang dapat terjadi ketika mencoba transfer file yang berdurasi 2 jam dari mesin yang satu ke mesin lainnya dengan kemungkinan mempunyai selang waktu 1 jam antara dua crash yang dapat terjadi. Setelah masing-masing transfer dibatalkan, seluruh transfer mungkin

perlu diulangi lagi dari awal, dan mungkin saja mengalami kegagalan lain. Untuk mengurangi kemungkinan terjadinya masalah ini, session layer dapat menyisipkan tanda tertentu ke aliran data. Karena itu bila terjadi crash, hanya data yang berada sesudah tanda tersebut yang akan ditransfer ulang.

6. *Presentation Layer (Lapisan Presentasi)*

Pada lapisan presentasi terjadi perubahan data, misalnya mode grafik akan diubah menjadi *bit stream*. Salah satu contoh layanan presentasi adalah *encoding* data, dimana data-data seperti nama, tanggal dan lain-lain dinyatakan dalam bentuk *string* karakter atau struktur data dalam bentuk yang sederhana.

Pressentation layer melakukan fungsi-fungsi tertentu yang diminta untuk menjamin penemuan sebuah penyelesaian umum bagi masalah tertentu. Pressentation Layer tidak mengijinkan pengguna untuk menyelesaikan sendiri suatu masalah. Tidak seperti layer-layer di bawahnya yang hanya melakukan pemindahan bit dari satu tempat ke tempat lainnya, presentation layer memperhatikan syntax dan semantik informasi yang dikirimkan.

Satu contoh layanan pressentation adalah encoding data. Kebanyakan pengguna tidak memindahkan string bit biner yang random. Para pengguna saling bertukar data sperti nama orang, tanggal, jumlah uang, dan tagihan. Item-item tersebut dinyatakan dalam bentuk string karakter, bilangan interger, bilangan floating point, struktur data yang dibentuk dari beberapa item yang lebih sederhana. Terdapat perbedaan antara satu komputer dengan komputer lainnya dalam memberi kode untuk menyatakan string karakter (misalnya, ASC2 dan Unicode), integer (misalnya komplemen satu

dan komplemen dua), dan sebagainya. Untuk memungkinkan dua buah komputer yang memiliki presentation yang berbeda untuk dapat berkomunikasi, struktur data yang akan dipertukarkan dapat dinyatakan dengan cara abstrak, sesuai dengan encoding standard yang akan digunakan "pada saluran". Presentation layer mengatur data-struktur abstrak ini dan mengkonversi dari representation yang digunakan pada sebuah komputer menjadi representation standard jaringan, dan sebaliknya.

7. *Application Layer (Lapisan Aplikasi)*

Lapisan aplikasi ini merupakan lapisan tertinggi pada model referensi OSI, biasanya berupa program atau aplikasi pada tingkatan layanan informasi. Sebagai tambahan untuk transfer informasi, lapisan aplikasi menyediakan layanan – layanan seperti :

i) Mengidentifikasi partner komunikasi dengan alamat atau nama.
i) Membangun otoritas untuk komunikasi.
ii) Penanganan perbaikan kesalahan.
ii) Mengindentifikasi dalam penekanan *syntax* data.

Application layer terdiri dari bermacam-macam protokol. Misalnya terdapat ratusan jenis terminal yang tidak kompatibel di seluruh dunia. Ambil keadaan dimana editor layar penuh yang diharapkan bekerja pada jaringan dengan bermacam-macam terminal, yang masing-masing memiliki layout layar yang berlainan, mempunyai cara urutan penekanan tombol yang berbeda untuk penyisipan dan penghapusan teks, memindahkan sensor dan sebagainya.

Suatu cara untuk mengatasi masalah seperti di ata, adalah dengan menentukan terminal virtual jaringan abstrak, serhingga editor dan program-program lainnya

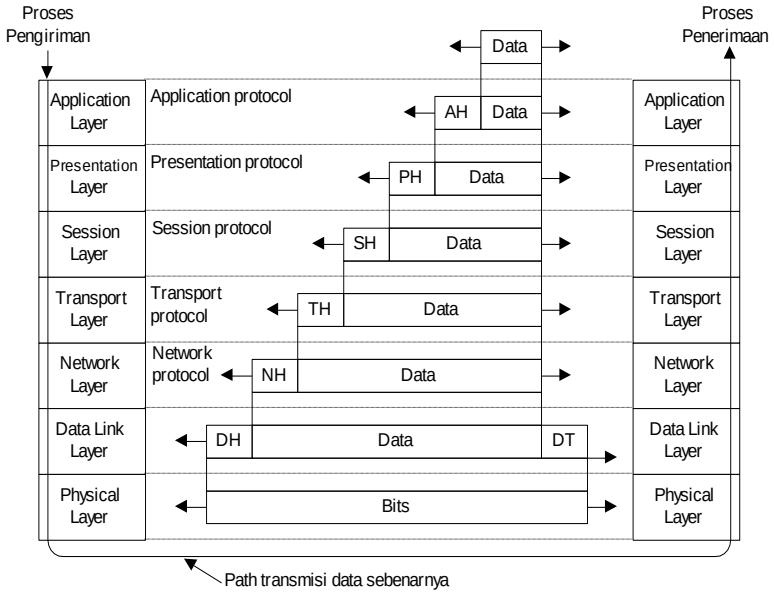
dapat ditulis agar saling bersesuaian. Untuk menangani setiap jenis terminal, satu bagian software harus ditulis untuk memetakan fungsi terminal virtual jaringan ke terminal sebenarnya. Misalnya, saat editor menggerakkan cursor terminal virtual ke sudut layar kiri, software tersebut harus mengeluarkan urutan perintah yang sesuai untuk mencapai cursor tersebut. Seluruh software terminal virtual berada pada application layer.

Fungsi application layer lainnya adalah pemindahan file. Sistem file yang satu dengan yang lainnya memiliki konvensi penamaan yang berbeda, cara menyatakan baris-baris teks yang berbeda, dan sebagainya. Perpindahan file dari sebuah sistem ke sistem lainnya yang berbeda memerlukan penanganan untuk mengatasi adanya ketidak-kompatibelan ini. Tugas tersebut juga merupakan pekerjaan appication layer, seperti pada surat elektronik, remote job entry, directory lookup, dan berbagai fasilitas bertujuan umum dan fasilitas bertujuan khusus lainnya.

Proses pengiriman memiliki data yang akan dikirimkan ke proses penerima. Proses pengirim menyerahkan data ke application layer, yang kemudian menambahkan aplication header, AH (yang mungkin juga kosong), ke ujung depannya dan menyerahkan hasilnya ke presentation layer.

Pressentation layer dapat membentuk data ini dalam berbagai cara dan mungkin saja menambahkan sebuah header di ujung depannya, yang diberikan oleh session layer. Penting untuk d2ngat bahwa presentation layer tidak menyadari tentang bagian data yang mana yang diberi tanda AH oleh application layer yang merupakan data pengguna yang sebenarnya.

Proses pemberian header ini berulang terus sampai data tersebut mencapai physical layer, dimana data akan ditransmisikan ke mesin lainnya. Pada mesin tersebut, semua header tadi dicopoti satu per satu sampai mencapai proses penerimaan.

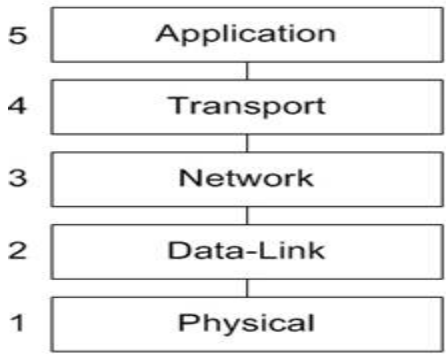


Gambar 2.12 Contoh tentang bagaimana model OSI digunakan

Yang menjadi kunci di sini adalah bahwa walaupun transmisi data aktual berbentuk vertikal seperti pada Gambar 2.12, setiap layer diprogram seolah-olah sebagai transmisi yang bersangkutan berlangsung secara horizontal. Misalnya, saat transport layer pengiriman mendapatkan pesan dari session layer, maka transport layer akan membubuhkan header transport layer dan mengirimkannya ke transport layer penerima.

2.3.2. TCP/IP (Transmission Control Protocol / Internet Protocol)

TCP/IP (Transmission Control Protocol/ Internet Protocol) adalah sekumpulan protokol yang didesain untuk melakukan fungsi-fungsi komunikasi data seperti pada LAN(*Local Area Network*) dan WAN(*Wide Area Network*). TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dari komunikasi data. Pemodelan empat layer TCP/IP, sebagaimana terlihat pada Gambar 2.11.

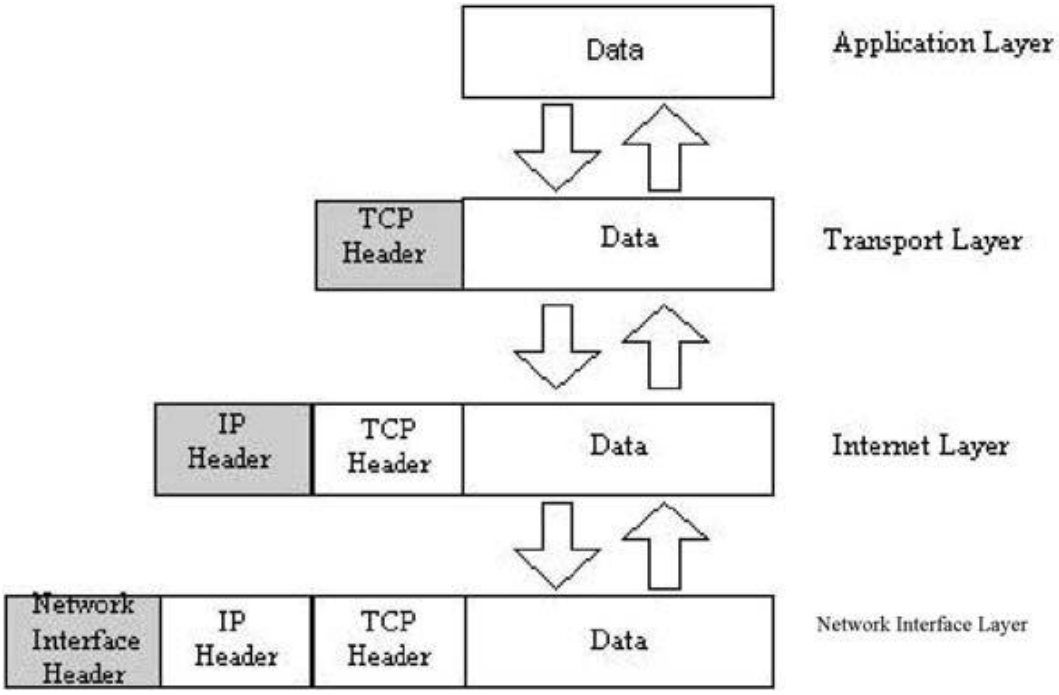


Gambar 2.11 Lapisan TCP/IP

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* dibawahnya.

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 bisa dilihat pada Gambar 2.14.



Gambar 2.12 Pergerakan data dalam layer TCP/IP

TCP/IP terdiri dari 4 lapisan (*layer*) yang berupa sekumpulan protokol yang bertingkat. Keempat lapis / *layer* tersebut adalah :

1. *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*.

2. *Internet Layer*

Internet Layer bertanggung jawab dalam proses pengiriman ke alamat yang tepat.

3. *Transport Layer*

Transport Layer berisikan protokol-protokol yang bertangung jawab dalam mengadakan komunikasi antar dua *host* atau komputer. Kedua protokol tersebut adalah TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*).

4. *Application Layer*

Application Layer merupakan tempat aplikasi-aplikasi yang menggunakan TCP/IP *stack* berada, contohnya antara lain SMTP (*Simple Mail Transfer Protoco*l) adalah suatu protokol aplikasi yang merupakan sistem pengiriman *messag*e/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.*

2.4. IP (*Internet Protocol*)

Internet Protocol didesain untuk menghubungkan komunikasi komputer pada jaringan *packet-switched*. IP menyediakan pengiriman data yang bersifat *connectionless* dan *best effort*. *Connectionless* berarti tidak ada pembentukan hubungan antara satu titik dengan titik lain sebelum proses pengiriman data. *Best*

effort berarti sedapat mungkin IP akan mengirimkan data ketujuan, tetapi IP idak menjamin data akan benar-benar sampai ke tujuan. (Alberto Leon Garcia,2000:548)

2.4.1 IP address

IP address merupakan pemberian identitas yang universal bagi setiap *interface* komputer lain. *IP address* berfungsi sebagai penunjuk alamat *interface* pada sebuah komputer dan juga untuk menentukan suatu rute jaringan yang dilalui oleh sebuah pengiriman data.

(a) *Format IP Address*

IP *Address* terdiri dari sekelompok bilangan biner 32 bit yang dipisahkan oleh tanda pemisah berupa tanda titik (**.**) setiap 8 bitnya. Tiap 8 bit disebut sebagai octet. Bentuk *IP address* adalah sebagai berikut :

xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

setiap simbol “x” dapat digantikan oleh angka 0 dan 1, misalnya sebagai berikut:

11000000 . 10101000 . 00001010 . 00000001

192.168.10.1

(b) Pembagian Kelas IP

Pembagian kelas-kelas *IP address* didasarkan pada dua hal antara lain :

1. Network ID, merupakan bagian dari IP address yang digunakan untuk menunjukkan jaringan tempat komputer ini berada.

2. Host ID, adalah bagian dari IP address yang digunakan untuk menunjukkan workstation, server router, dan semua host TCP / IP lainnya dalam jaringan tersebut. Dalam satu jaringan, host ID ini tidak boleh ada yang sama.
3. Pengalamatan IP diorganisasikan ke dalam kelas-kelas dengan pemeriksaan octet pertama sebagai berikut :
    a. Kelas A

        Format : 0nnnnnnn hhhhhhhh hhhhhhhh hhhhhhhh

        Bit pertama : 0

        Network ID : 8 bit host ID:

        24 bit Byte pertama : 0 – 127

        Jumlah jaringan : 126 kelas

        Range IP : 1.xxx.xxx.xxx sampai 126.xxx.xxx.xxx

        Jumlah IP : 16.777.214 IP address pada tiap kelas A

    b. Kelas B

        Format : 10nnnnnn nnnnnnnn hhhhhhhh hhhhhhhh

        Dua bit pertama : 10

        Network ID : 16 bit

        Host ID : 16 bit

        Byte pertama : 128 – 191

        Jumlah jaringan : 16.384 Kelas B

        Range IP : 128.0.xxx.xxx sampai 191.255.xxx.xxx

        Jumlah IP : 62.532 Host

c. Kelas C

Format : 110nnnnn nnnnnnnn nnnnnnnn hhhhhhhh

Tiga bit pertama : 110

Network ID : 24 bit

Host ID : 8 bit

Byte pertama : 192 – 233

Jumlah jaringan : 2.097.152 Kelas C

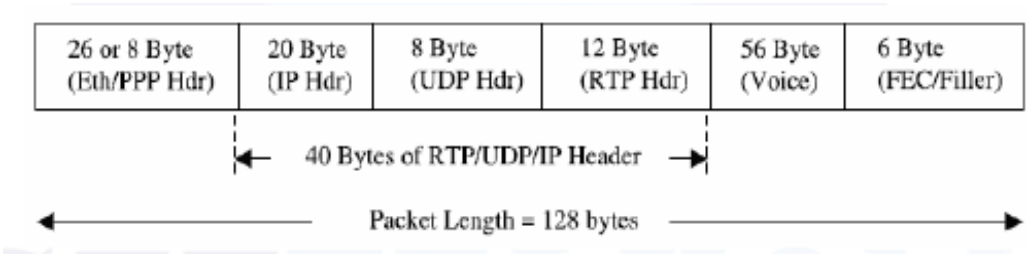
Range IP : 192.0.0.xxx sampai 223.255.255.xxx

Jumlah IP : 254 Host

2.5. VOIP (*Voice over Internet Protocol*)

VoIP dedefinisikan sebagai suatu sistem yang menggunakan jaringan internet untuk mengirimkan paket data suara dari suatu tempat ke tempat lainnya menggunakan perantara protokol IP. VoIP mentransmisikan sinyal suara dengan mengubahnya ke dalam bentuk digital, dan dikelompokkan menjadi paket–paket data yang dikirim dengan menggunakan *platform* IP (*Internet Protocol*). Ada berapa komponen yang dibutuhkan dalam membuat jaringan VoIP, yaitu:

- *Codecs*
- TCP/IP dan VoIP *protocols*
- IP *telephony servers* dan PBXs
- VoIP *gateways* dan *routers*
- IP *phones* atau *softphones*



Gambar 2.13 Contoh protocol stack VoIP

2.5.1. Kualitas VoIP

VoIP merupakan salah satu jenis layanan *realtime* yang membutuhkan QoS (*Quality of Service*). Untuk menjamin terjaganya kualitas VoIP maka ada beberapa faktor yang mempengaruhi diantaranya :

1. *Codec*

Sebuah *codec* (yang merupakan kepanjangan dari *compressor/decompresser* atau *coder/decoder*) adalah suatu *hardware* atau *software* yang melakukan *sampling* terhadap sinyal suara *analog*, kemudian mengkonversi ke dalam bit – bit digital dan mengeluarkannya. Beberapa jenis *codec* melakukan kompresi agar dapat menghemat *bandwidth*. Dibawah ini terdapat beberapa jenis *codec* yang digunakan secara umum, bersama dengan data *rate* dan *delay* paketisasi.

Nama *Codec*	*Data Rate*	*Delay Paketisasi*
G.711u	64.0 kbps	1.0 ms
G.711a	64.0 kbps	1.0 ms
G.726-32	32.0 kbps	1.0 ms
G.729	8.0 kbps	25.0 ms
G.723.1 MPMLQ	6.3 kbps	67.5 ms
G.723.1 ACELP	5.3 kbps	67.5 ms

Tabel 2.3 Jenis Codec

2. *Frame Loss*

Dalam jaringan *internet*, setiap paket yang dikirim ke penerima belum tentu sampai, hal ini disebabkan karena sifat jaringan *internet* yaitu *best effort*, dimana jaringan hanya berusaha menjaga agar paket sampai di penerima. Jika paket lama berputar di jaringan, oleh *router* paket itu akan dibuang. Begitu pula dengan QoS. Jaringan *internet* tidak menyediakan QoS bagi penggunanya. Hal itu yang menyebabkan VoIP menggunakan protokol RTP. Salah satu kelemahan protokol ini adalah tidak adanya transmisi ulang jika paket yang dikirim hilang.

3. *Delay*

Delay merupakan faktor yang penting dalam menentukan kualitas VoIP. Semakin besar *delay* yang terjadi maka akan semakin rendah kualitas VoIP yang dihasilkan. Adapun beberapa sumber *delay* antara lain:

• *Delay Algoritmic* merupakan delay yang dihasilkan oleh Codec dalam melakukan pengkodean.

• *Delay Paketisasi* merupakan delay yang terjadi antara satu paket RTP dengan paket lainnya. RFC 1890 menspesifikasikan bahwa *delay* paketisasi sebaiknya 20 ms.

• *Delay Serialisasi* merupakan waktu yang dibutuhkan untuk mengirimkan paket dari pengirim. *Delay* serialisasi juga terjadi pada router atau switch.

• *Delay Propagasi* merupakan waktu yang diperlukan untuk sinyal listrik atau sinyal *optic* untuk melewati media transmisi dari pengirim ke penerima. *Delay* ini tergantung dari media transmisi dan jarak yang harus ditempuh sinyal.

• *Delay* komponen merupakan *delay* yang terjadi yang disebabkan proses yang terjadi pada komponen contohnya *router*. Dalam *router* terdapat *delay* yang terjadi jika paket masuk dan diproses kemudian dikeluarkan ke *port output*.

Agar tidak terjadi suara *overlap* dalam komunikasi VoIP, G.114 menyediakan pedoman untuk batasan *delay* satu arah :

0 s/d 150 ms delay ini masih dapat diterima

150 s/d 400 ms masih dapat diterima jika administrator waspada terhadap waktu yang dapat mempengaruhi kualitas transmisi.

Diatas 400 ms tidak dapat diterima untuk layanan VoIP.

4. *Jitter*

Jitter adalah variasi yang ditimbulkan oleh *delay*, terjadi karena adanya perubahan terhadap karakteristik dari suatu sinyal sehingga menyebabkan terjadinya masalah terhadap data yang dibawa oleh sinyal tersebut.

5. *Packet Loss*

Packet loss adalah hilangnya paket data yang sedang dikirimkan disebabkan karena Jitter atau karena adanya permasalahan di perangkat-perangkat jaringan seperti router atau jalur komunikasi yang terlalu padat penggunanya.

6. *Keamanan*

VoIP berjalan pada jaringan intranet maupun internet kemungkinan data suara tersebut disadap oleh pihak-pihak yang tidak bertanggung jawab tetaplah ada, inilah yang mendasari penulis untuk fokus terhadap keamanan pada data suara voip tersebut.

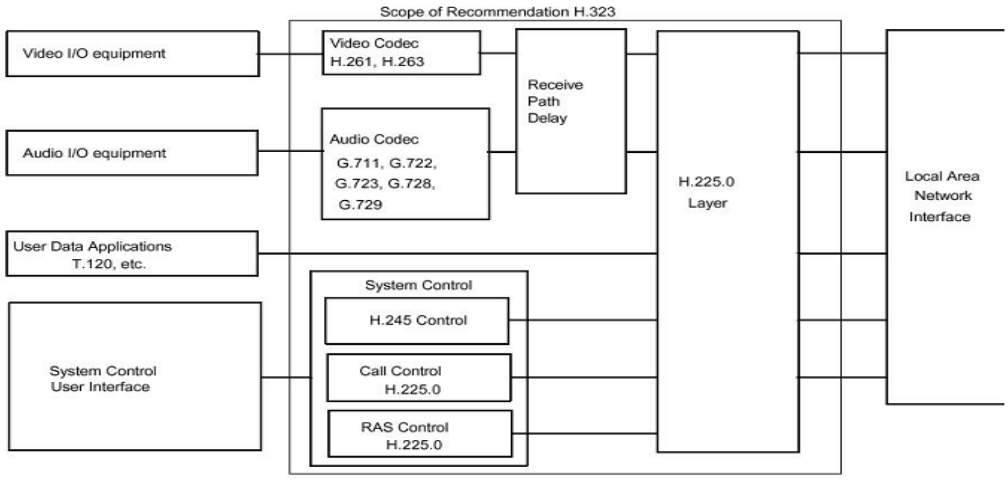
2.5.2. Protocol Untuk VoIP

Secara umum, terdapat dua teknologi yang digunakan untuk VoIP, yaitu H.323 dan SIP. H323 merupakan teknologi yang dikembangkan oleh ITU *(International Telecommunication Union*). SIP (*Session Initiation Protocol)* merupakan teknologi yang dikembangkan IETF (*Internet Enggineering Task Force*).

1. *H.323*

H.323 adalah salah satu dari rekomendasi ITU-t (*International Telecommunications Union – Telecommunications*). H.323 merupakan standar yang menentukan komponen, protokol, dan prosedur yang menyediakan layanan

komunikasi *multimedia*. Layanan tersebut adalah komunikasi *audio, video*, dan *data real-time*, melalui jaringan berbasis paket (*packet-based network*). (Tabratas Tharom, 2001;64). H.323 berjalan pada jaringan *intranet* dan jaringan *packet-switched* tanpa mengatur media jaringan yang di gunakan sebagai sarana transportasi maupun protokol *networ layer*. Karakteristik terminal H.323 dapat dilihat pada Gambar 2.14.



Gambar 2.14 Diagram blok terminal berbasis H.323

(Tabratas Tharom,2001:73)

Standar H.323 mengatur hal-hal sebagai berikut :

1. *Video Codec* (H.261 dan H.263). *Video Codec* bertugas mengkodekan data dari sumber *video* untuk dikirimkan dan mendekodekan sinyal kode yang diterima untuk di tampilkan di layar penerima.

2. *Audio Codec* (G.711, G.722, G723, G728 dan G.729). *Audio codec* betugas mengkodekan data dari sumber suara untuk dikirimkan dan mendekodekan sinyal kode yang diterima untuk didengarkan oleh penerima.

3. Data *channel* mendukung aplikasi-aplikasi seperti *electronic whiteboard*, dan kolaborasi aplikasi. Sttandar untuk aplikasi-aplikasi seperti ini adalah standar T.120 . Aplikasi dan protokol yang berbeda tetap dapat dijalankan dengan negosiasi menggunakan standar H.245

4. Sistem control unit (H.245 dan H.225.0) menyediakan signalling yang berkaitan dengan komunikasi antar terminal H.323.

5. H.225.0 layer memformat data video, suara, data , dan informasi kontrol lain sehingga dapat dikirimkan melalui LAN *Interface* sekaligus menerima data yang telah diformat melalui LAN Interface. Sebagai tambahan, layer ini juga bertugas melakukan *error detection, error correction* , dan *frame sequencing* agar data dapat mencapai tujuan sesuai denagn kondisi saat data dikirimkan. LAN *interface* harus menyediakan koneksi yang handal. Untuk *flow control* dan *unreliable* data *channel connection* (misal: UDP) dapat digunakan untuk pengiriman *audio* dan *video channel*.

2. *Session Initiation Protokol (SIP)*

Secara umum, terdapat dua teknologi yang digunakan untuk VoIP, yaitu H.323 dan SIP. H323 merupakan teknologi yang dikembangkan oleh ITU (International Telecommunication Union). SIP (Session Initiation Protocol) merupakan teknologi yang dikembangkan IETF (Internet Enggineering Task Force).

Session Initiation Protocol atau SIP merupakan standar IETF untuk suara atau layanan multimedia melalui jaringan internet. SIP (RFC 2543) di ajukan pada tahun 1999. Pencipta standar ini adalah Henning Schulzrinne. SIP merupakan protokol

layer aplikasi yang digunakan untuk manajemen pengaturan panggilan dan pemutusan panggilan. SIP digunakan bersamaan dengan protokol IETF lain seperti SAP,SDP,MGCP (MEGACO) untk menyediakan layanan VoIP yang lebih luas. Arsitektur SIP mirip dengan arsitektur HTTP (protocol client – server). Arsitekturnya terdiri dari request yang dikirim dari user SIP ke server SIP. Server itu memproses request yang masuk dan memberikan respon kepada client. Permintaan request itu, bersama dengan komponen respon pesan yang lain membuat suatu komunikasi SIP.

Komponen SIP Arsitektur SIP terdiri dari dua buah komponen seperti tertera dibawah ini.

1. *User Agent*

SIP User Agent merupakan akhir dari sistem (terminal akhir) yang bertindak berdasarkan kehendak dari pemakai. Terdiri dari dua bagian yaitu:

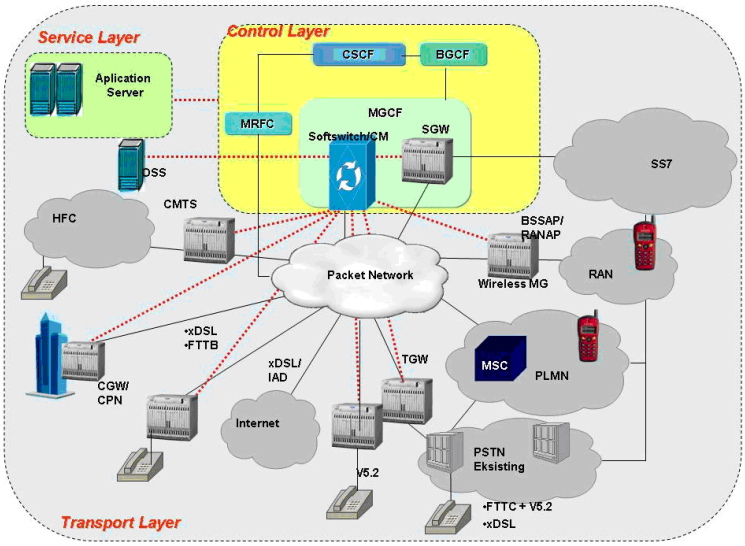
1. User Agent Client (UAC) : bagian ini terdapat pada pemakai (client), yang digunakan untuk melakukan inisiasi request dari server SIP ke UAS
2. User Agent Server (UAS) : bagian ini berfungsi untuk mendengar dan merespon terhadap request SIP

2. *SIP Server*

Arsitektur SIP sendiri menjelaskan jenis – jenis *server* pada jaringan untuk membantu layanan dan pengaturan panggilan SIP.

1. *Registration Server* : *server* ini menerima request registrasi dari user SIP dan melakukan update terhadap lokasi user dengan server ini.

2. *Proxy Server* : *server* ini menerima *request* SIP dan meneruskan ke server yang dituju yang memiliki informasi tentang user yang dipanggil
3. *Redirect Server* : *server* ini setelah menerima request SIP , menentukan server yang dituju selanjutnya dan mengembalikan alamat server yang dituju selanjutnya kepada client daripada meneruskan request ke server yang dituju tersebut.



Gambar 2.15 Operasi SIP

SIP menggabungkan beberapa macam protokol baik dari standar yang dikeluarkan oleh IETF maupun ITU – T

1. IETF Session Description Protocol (SDP) yang mendefinisikan suatu metode standar dalam menggambarkan karakteristik dari suatu sesi multimedia.
2. IETF *Session Announcement Protocol* (SAP) yang berjalan setiap periode waktu tertentu untuk mengumumkan parameter dari suatu sesi konferensi.

3. IETF *Realtime Transport Protocol* (RTP) dan *Realtime Transport Control* Protocol (RTCP) yang menyediakan informasi tentang manajemen transport dan session
4. Algoritma pengkodean. Yang direkomendasikan ITU – T seperti G.723.1, G.729,G728.

Pembangunan suatu komunikasi multimedia dengan SIP dilakukan melalui beberapa tahap :

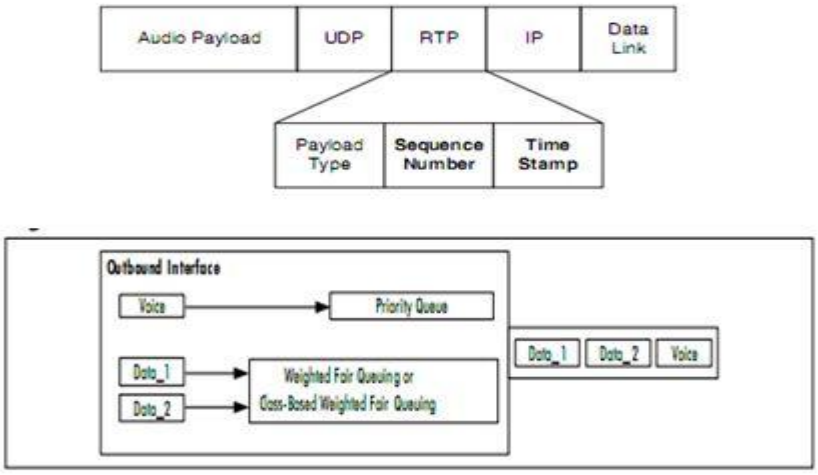
i) *User Location* adalah menentukan lokasi pengguna yang akan berkomunikasi.
ii) *User Availability* adalah menentukan tingkat keinginan pihak yang dipanggil untuk terlibat dalam komunikasi.
iii) *User Capability* adalah menentukan media maupun parameter yang berhubungan dengan media yang digunakan untuk komunikasi.
iv) *Session Setup* adalah pembentukan hubungan antara pihak pemanggil dengan pihak yang dipanggil.
v) *Session management* yaitu meliputi *transfer*, modifikasi, dan pemutusan sesi.

Secara garis besar SIP merupakan protokol yang digunakan dalam untuk membangun, memodifikasi, dan mengakhiri suatu sesi. Penggunaan protokol *codec video , audio* dan *Real-time Protocol* dengan H.323 tetap sama, hanya berbeda dalam sesi *signalling* sambungan VoIP.

Protokol VoIP

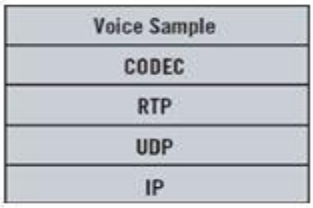
1. *Real – time Transport Protocol (RTP)*[1]

RTP digunakan untuk mengirim paket *real-time,* seperti *audio dan video*, melalui jaringan paket *switch*. Protokol ini digunakan oleh kedua *protocol* SIP dan H.323. protokol menyediakan informasi waktu kepada penerima sehingga dapat memberbaiki *delay* karena *jitter*. RTP juga memungkinkan penerima untuk mendeteksi paket yang hilang. *Header* dari RTP berisikan informasi yang membantu penerima agar dapat merekonstruksi ulang paket dan juga informasi mengenai bagaimana bit – bit dibagi menjadi paket oleh *codec.* RTP menyediakan informasi yang cukup kepada penerima sehingga sehingga dapat dipulihkan kembali bahkan jika terjadi kehilangan paket atau *jitter.*



Gambar 2.16 Komponen RTP *header*

Tiap-tiap packet RTP berisi potongan paket dari percakapan suara. Besarnya ukuran *packet* suara bergantung pada *CODEC* yang digunakan. Susunan *Protocol* RTP Diagram berikut memperlihatkan susunan *protocol* RTP.



Gambar 2.17 Susunan *protocol* RTP

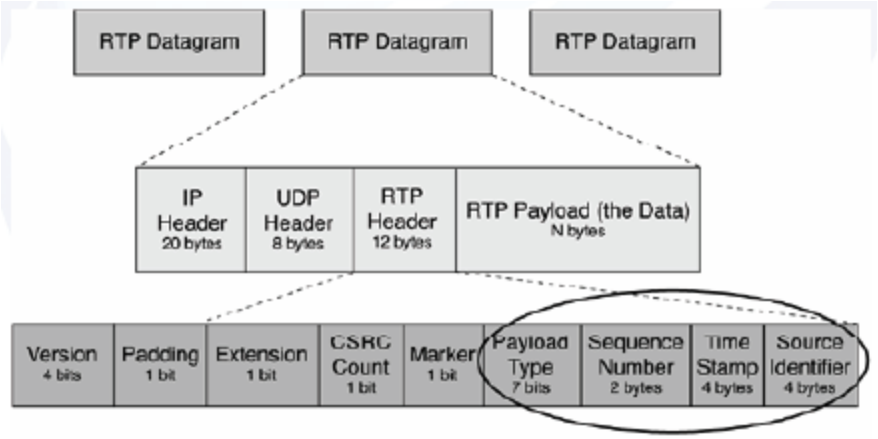
RTP merupakan standar format paket untuk mengirimkan *file audio* dan *video* pada *internet*. RTP dikembangkan oleh *Audio-Video Transport Working Group* pada *IETF* dan pertama kali keluar pada tahun 1996 sebagai RFC 1889 dan diperbaharui lagi pada tahun 2003 sebagai RFC 3550. RTP berfungsi untuk transmisi *real-time* seperti *audio, video, multimedia* secara *end-to-end*. RTP mendukung transmisi *unicast, broadcast* dan *multicast*. RTP hanyalah sebagai *protocol transport* dan hal ini tidak menjamin *QoS* untuk layanannya. Biasanya RTP memiliki *port* tertentu untuk melakukan pengiriman *payload* data. Walaupun tidak ada standar yang menyatakan secara pasti, RTP biasanya dikenali pada *port* 16384-32767. Menurut RFC 1889, layanan RTP termasuk:

1. *Payload-tipe Identification* – Indikasi yang menyatakan jenis *content* / isi yang dibawa.
2. Nomor sekuensial – Nomor Sekuensial PDU (*Protocol Data Unit*).
3. *Time stamp* – Waktu yang digunakan *content* yang dibawa oleh PDU
4. *Delivery monitoring*.

Streaming server memotong media data untuk masing-masing *track* menjadi paket RTP yang kemudian ditransmisikan melalui UDP. Masing-masing paket RTP ditransmisikan dengan beberapa informasi yaitu:

1. *RTP header*
2. *RTP payload header*
3. *RTP payload data*

Pada RTP *header* memuat informasi tentang RTP *version, sequence number, timestamp, Synchronization Source Identifier* (SSRC). *Sequence number* digunakan untuk inisialisasi paket yang dikirimkan dan bisa dipakai untuk perhitungan *paket loss*. *Timestamp* mununjukkan waktu paket yang dipakai untuk sinkronisasi dan perhitungan *jitter*. SSRC mengindikasikan keunikan dan keaslian paket dalam suatu sesi. Kunci keunggulan RTP adalah kemampuan interoperabilitas: jika dua aplikasi *internet phone* menggunakan RTP, maka tidak akan terjadi masalah.



Gambar 2.20 RTP *Header* dengan informasi *Payload* dan *Sequence*

RTP didesain untuk digunakan pada *tansport layer*, namun demikian RTP digunakan diatas UDP, bukan pada TCP karena TCP tidak dapat beradaptasi pada pengiriman data yang *real-time* dengan keterlambatan yang relatif kecil seperti pada pengiriman data komunikasi suara. Dengan menggunakan UDP yang dapat mengirimkan paket IP secara *multicast*, RTP *stream* yang di bentuk oleh satu terminal dapat dikirimkan ke beberapa terminal tujuan. Selain itu, oleh karena informasi RTP dienkapsulasi dalam *packet* UDP. Jika *packet* RTP hilang *(lost*) atau di*drop* di jaringan, maka RTP tidak akan melakukan *retransmission* (sesuai *standard protocol* UDP). Hal ini dilakukan agar *user* tidak terlalu lama menunggu *(long pause*) atau *delay*, dikarenakan permintaan *retransmission*. Oleh karena itu jaringan harus di*design* sebaik mungkin agar *lost packet* tidak terjadi.

2. *Real Time Control Protocol (RTCP)*[1]

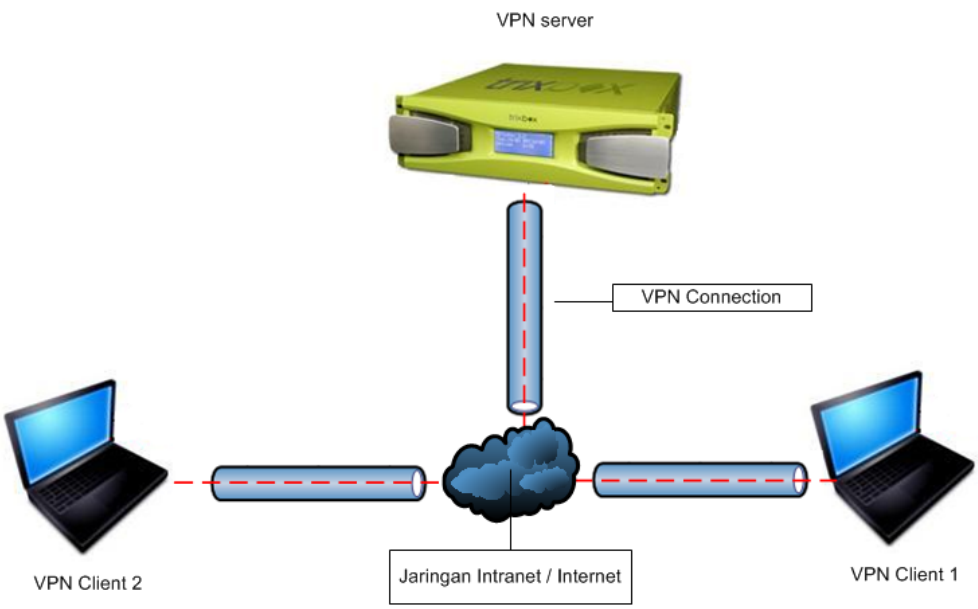
RTCP adalah suatu *protocol control* yang bekerja berhubungan dengan RTP. Dalam sebuah sesi RTP, terminal akhir *(endpoint*) secara periodik mengirimkan paket RTCP untuk menyebarkan informasi yang bermanfaat mengenai QoS. Terminal akhir ini dapat mengetahui takaran yang tepat unuk mengirimkan paket dengan lebih efisien melalui sesi RTP. Secara umum fungsi dari RTCP adalah untuk menyediakan *feedback* QoS, kontrol sesi, identifikasi *user*, dan sinkronisasi *inter media*, yaitu mensinkronkan antara *audio* dan *video* seperti mensinkronkan antara gerakan bibir pada *video* dengan suara yang keluar pada audio.

3. *Real – time Streaming Protocol (RTSP)*[1]

IETF telah mendefinisikan RTSP sebagai protokol server/client yang menyediakan kendali atas pengiriman aliran data *real-time*. Fungsi dari RTSP diantara media *server* dan *client*nya adalah membangun dan mengendalikan hubungan aliran data antara audio dan video.

2.6. *Virtual Private Network*(VPN)

VPN (*Virtual Private Network*) adalah teknik pengamanan jaringan dengan cara membuat suatu *tunnel* misalkan pada jaringan publik atau *internet* sehingga jaringan bersifat *private* dan aman. VPN dikatakan bersifat *private* karena ketika akan terbentuknya sebuah koneksi VPN dibutuhkannya *authentikasi* untuk memastikan bahwa kedua ujung dalam koneksi adalah *user* yang sesuai yang diberi kewenangan untuk mengakses suatu *server*. Setelah terbentuknya suatu koneksi VPN maka data akan di *enkripsi* untuk menjaga kerahasiaan paket yang dikirimkan. Penggunaan *authentikasi* dan *enkripsi* yang dipadukan dalam VPN membuat suatu sistem keamanan yang lebih baik seperti pada Gambar 2.21.

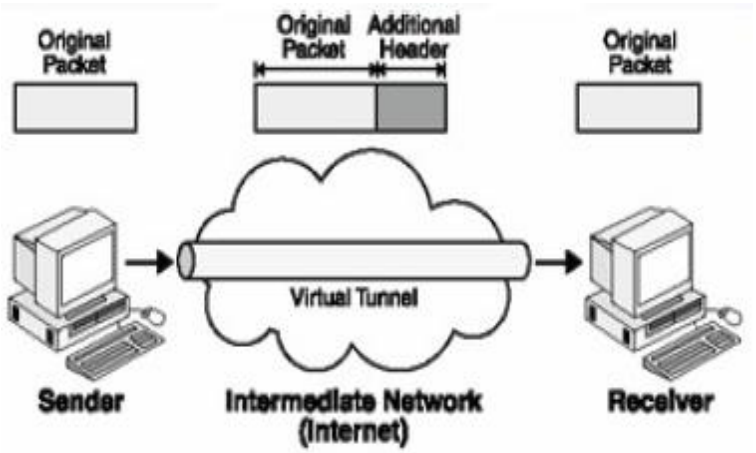


Gambar 2.21 Pelewatan data dengan VPN

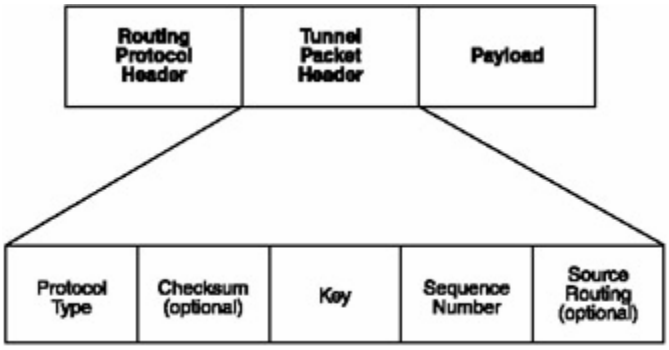
2.6.1. Tunneling

Tunneling merupakan inti dari teknologi VPN. Tunneling merupakan suatu teknik untung melakukan *enkapsulasi* terhadap seluruh data pada suatu paket yang menggunakan suatu *format* protokol tertentu. Dengan kata lain, header dari suatu protokol *tunneling* ditambahkan pada *header* paket yang asli. Kemudian barulah paket tersebut dikirimkan ke dalam jaringan paket data.

Ketika paket yang telah "*ditunnel*" dirutekan ke terminal tujuan. Maka paket – paket tersebut akan melewati suatu jalur logika yang dikenal dengan nama kanal. Ketika penerima menerima paket tersebut, maka akan dibuka dan dikembalikan lagi ke dalam *format* aslinya.



Gambar 2.18 Tunneling

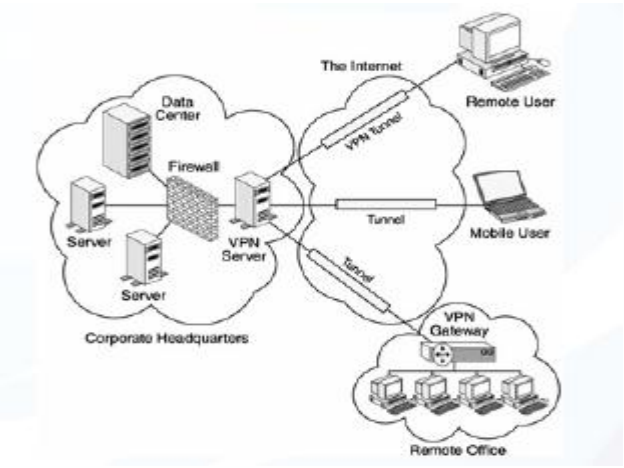


Gambar 2.22 Protocol stack untuk VPN tunneling

2.6.2. Jenis Jaringan VPN

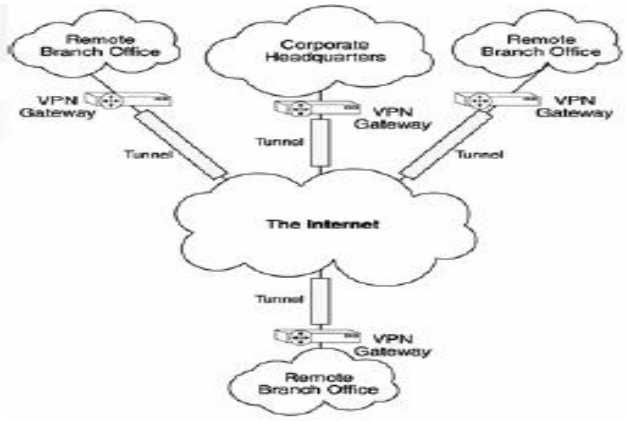
Ada 3 jenis jaringan VPN antara lain :

1. *Access* VPN – menyediakan *remote access* ke jaringan *intranet* atau *extranet* perusahaan yang memiliki kebijakan yang sama sebagai jaringan privat. *Access* VPN memungkinkan *user* untuk dapat mengakses data perusahaan kapanpun dimanapun perusahaan membutuhkan.



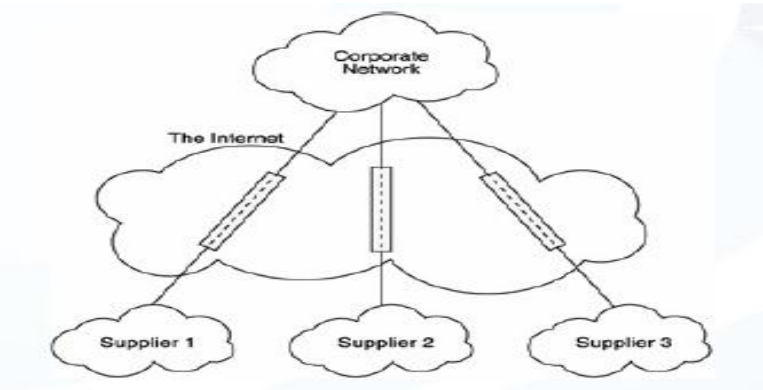
Gambar 2.19 Access VPN

2. *Intranet* VPN – menghubungkan antara kantor pusat suatu perusahaan dengan kantor cabang, kantor pembantu melalui *shared network* menggunakan koneksi yang permanent (*dedicated*).



Gambar 2.20 Intranet VPN

3. Extranet VPN – menghubungkan konsumen, *suppliers*, mitra bisnis atau berberapa komunitas dengan kepentingan yang sama ke jaringan *intranet* perusahaan melalui *infrastruktur* yang terbagi menggunakan koneksi *permanent* (*dedicated*).



Gambar 2.21 Extranet VPN

2.7. Keamanan Jaringan

Keamanan jaringan komputer adalah proses untuk menghindari atau melindungi dari gangguan baik untuk pencurian data, manipulasi data. Penggunaan sistem keamanan yang baik maka akan menjadikan komunikasi data pada komputer akan semakin aman.

1. Tipe Proteksi Jaringan Komputer

Pada lapisan jaringan komputer memiliki fungsi yang berbeda, maka perlindungan sesuai dengan lapisan pada jaringan tersebut . Disini dijelaskan mengenai perlindungan terhadap jaringan komputer sesuai pada setiap lapisan jaringan komputer.

i) *Data Link Layer* (Lapisan *Data Link*)

Pengamanan jaringan komputer, tahap paling mendasar adalah menjaga titik akses yang dapat digunakan seseorang untuk terhubung ke dalam jaringan. Pada umumnya, titik akses jaringan komputer adalah berupa *hub* atau *switch.* Mekanisme

umum yang biasa digunakan dalam mengamankan titik akses ke jaringan komputer, yaitu :

1. *Mac Address*

Mac Address Authentication adalah sebuah mekanisme di mana sebuah peralatan yang akan melakukan akses pada sebuah titik-akses sudah terdaftar terlebih dahulu. Metode ini untuk memastikan apakah peralatan yang akan melakukan akses adalah peralatan yang berhak untuk akses tanpa mempedulikan siapa yang mempergunakannya. Pada setiap peralatan jaringan komputer terdapat sebuah identitas yang unik. Berdasarkan identitas tersebutlah metode ini melakukan *otentikasi*. Pada setiap paket data yang dikirimkan sebuah peralatan akan mengandung informasi mengenai identitas peralatan tersebut, yang akan dibandingkan dengan daftar akses yang dimiliki setiap titik-akses, apabila ternyata identitas peralatan terdapat dalam daftar, paket yang dikirimkannya akan diteruskan apabila tidak, maka paket yang dikirimkannya tidak akan diteruskan. Keuntungan *metode* ini adalah *metode* ini sudah banyak diimplementasikan pada *switch/hub* yang sering digunakan sebagai titik akses.

Kelemahan utama dari *metode* ini adalah seseorang dapat dengan mudah memanipulasi identitas unik pada peralatan yang digunakannya, sehingga peralatan tersebut dapat melakukan akses ke sebuah jaringan komputer. Oleh karena itu sangat penting untuk menjaga integritas daftar identitas peralatan yang dapat melakukan akses ke jaringan.

2. *Network Layer (Lapisan Network)*

Pada lapisan ini, metode perlindungan jaringan komputer akan berdasarkan pada alamat IP dan *Port*. Pada setiap paket data yang dikirimkan oleh sebuah peralatan jaringan komputer ke peralatan lainnya akan mengandung alamat IP dan *Port* yang digunakan oleh pengirim serta alamat IP dan *Port* dari tujuan paket tersebut. Sebuah sistem pengamanan yang biasanya dikenal dengan nama *firewall* dapat melakukan *filtering* berdasarkan kedua hal tersebut. *Firewall* diletakkan pada gerbang masuk maupun keluar sebuah sistem jaringan komputer.

3. *Session Layer dan Transport Layer*

Pada lapisan ini, metode pengamanan lebih difokuskan dalam mengamankan data yang dikirimkan. Metode pengamanan yang banyak digunakan adalah VPN (*Virtual Private Network*). Kelemahan utama dari VPN adalah tidak adanya sebuah *standard* baku yang dapat dikuti oleh semua pihak yang berkepentingan. Akibatnya ada banyak implementasi VPN yang dapat digunakan, tapi antara satu implementasi dengan implementasi lainnya tidak dapat saling berhubungan. Selain itu, karena harus melalui proses *enkripsi* dan *dekripsi*, sehingga waktu yang dibutuhkan untuk melakukan transmisi bertambah.

4. *Application Layer*

Lapisan paling atas dari jaringan komputer adalah lapisan aplikasi. Oleh karena itu, keamanan sebuah sistem jaringan komputer tidak terlepas dari keamanan aplikasi yang menggunakan jaringan komputer tersebut, baik itu keamanan data yang dikirimkan dan diterima oleh sebuah aplikasi, maupun keamanan terhadap aplikasi jaringan komputer tersebut.

2.7.1. Tipe Keamanan

Pada sistem jaringan komputer terdapat dua jenis tipe pengamanan. Yaitu pengamanan yang bersifat pencegahan (*preventif*) seperti agar sistem tidak memiliki lubang keamanan dan pengamanan yang bersifat pengobatan (*recovery*) ketika suatu lubang keamanan sudah diexploitasi.

1. Mengatur Akses (*Access Control*)

Salah satu cara yang digunakan untuk mengamankan sistem adalah dengan mengatur akses ke informasi melalui mekanisme "*authentication*" dan "*access control*". Implementasi dari mekanisme ini antara lain dengan menggunakan "*password*". Untuk menggunakan sebuah sistem atau komputer disini pemakai diharuskan melalui proses *authentication* dengan menuliskan "*userid*" dan "*password*" yang akan dibandingkan dengan *userid* dan *password* yang berada di sistem. Apabila keduanya *valid*, pemakai yang bersangkutan diperbolehkan menggunakan sistem. Apabila ada yang salah, pemakai tidak dapat menggunakan sistem serta biasanya dicatat dalam berkas *log*.

Setelah proses *authentication*, pemakai diberikan akses sesuai dengan level yang dimilikinya melalui sebuah *access control*. *Access control* ini biasanya dilakukan dengan mengelompokkan pemakai dalam "*group*". Ada *group* yang berstatus pemakai biasa, ada tamu, dan ada juga *administrator* atau *super user* yang memiliki kemampuan lebih dari *group* lainnya. Pengelompokan ini disesuaikan dengan kebutuhan dari penggunaan sistem.

2. Pembatasan *Service*

Secara garis besar pada sistem diberikan dengan beberapa *service* yang dijalankan sebagai *default* dengan tujuan untuk mempermudah dalam menkonfigurasi sebuah sistem. Sebagai contoh, pada sistem UNIX *service* seperti: *finger, telnet, ftp, smtp, pop, echo*, dan sebagainya yang kesemuanya tidak dibutuhkan. Untuk mengamankan sistem, *service* yang tidak diperlukan di server (komputer) dapat dimatikan untuk menghindari kesalahan dari *service* tersebut serta kemungkinan adanya celah keamanan yang ditimbulkan oleh dalam *service* tersebut.

3. *Firewall*

Firewall merupakan sebuah perangkat yang dapat diletakkan antara *Internet* dengan jaringan internal atau antara *server* dengan jaringan luar. Informasi yang keluar atau masuk harus melalui *firewall.*

Tujuan utama dari *firewall* adalah untuk menjaga (*prevent*) agar akses (ke dalam maupun ke luar) dari orang yang tidak berwenang (*unauthorized access*) tidak dapat dilakukan. Konfigurasi dari *firewall* bergantung kepada kebijaksanaan (*policy*) dari sistem kebutuhan, yang dapat dibagi menjadi dua jenis:

1. Semua yang tidak diperbolehkan secara eksplisit dianggap tidak diperbolehkan (*prohibitted*)
2. Semua yang tidak dilarang secara eksplisit dianggap diperbolehkan (*permitted*)

   Firewall bekerja dengan cara mengamati paket IP (*Internet Protocol*) yang melewatinya yang dapat diatur berdasarkan *IP address*, *port*, dan arah informasi bergantung kepada masing-masing *firewall*.

Firewall dapat berupa sebuah perangkat keras yang sudah dilengkapi dengan perangkat lunak tertentu, sehingga pemakai (*administrator*) tinggal melakukan konfigurasi dari *firewall* tersebut. *Firewall* juga dapat berupa perangkat lunak yang ditambahkan kepada sebuah server (baik UNIX maupun Windows NT), yang dikonfigurasi menjadi *firewall*.

Dalam hal ini, sebetulnya perangkat komputer dengan prosesor Intel 80486 sudah cukup untuk menjadi *firewall* yang sederhana.

Firewall biasanya melakukan dua fungsi yaitu *filtering* dan fungsi *proxy*. Keduanya dapat dilakukan pada sebuah perangkat komputer (*device*) atau dilakukan secara terpisah. Beberapa perangkat lunak berbasis UNIX yang dapat digunakan untuk melakukan IP filtering antara lain:

1. *ipfwadm*: merupakan standar dari sistem Linux yang dapat diaktifkan pada level *kernel*
2. *ipchains*: versi baru dari Linux kernel *packet filtering* yang diharapkan dapat menggantikan fungsi *ipfwadm*

Fungsi *proxy* dapat dilakukan oleh berbagai *software* tergantung kepada jenis *proxy* yang dibutuhkan, misalnya *web proxy*, *rlogin proxy*, *ftp proxy* dan seterusnya. Untuk menggunakan *firewall* perangkat lunak berbasis UNIX untuk *proxy* adalah *Squid web proxy server*.

Secara garis besar penggunaan *firewall* dapat digunakan untuk pengamanan sistem dengan karakteristik sebagai berikut :

1. Seluruh hubungan/kegiatan dari dalam ke luar , harus melalui *firewall*. Hal ini dapat dilakukan dengan cara memblok/membatasi baik secara fisik semua akses terhadap jaringan Lokal, kecuali melewati *firewall*.
2. Hanya kegiatan yang terdaftar/dikenal yang dapat melewati/melakukan hubungan, hal ini dapat dilakukan dengan mengatur *policy* pada konfigurasi keamanan lokal.
3. *Firewall* itu sendiri haruslah kebal atau relatif kuat terhadap serangan/kelemahan. Hal ini berarti penggunaan sistem yang dapat dipercaya dan dengan *system* yang relatif aman.

2.8. *Enkripsi*

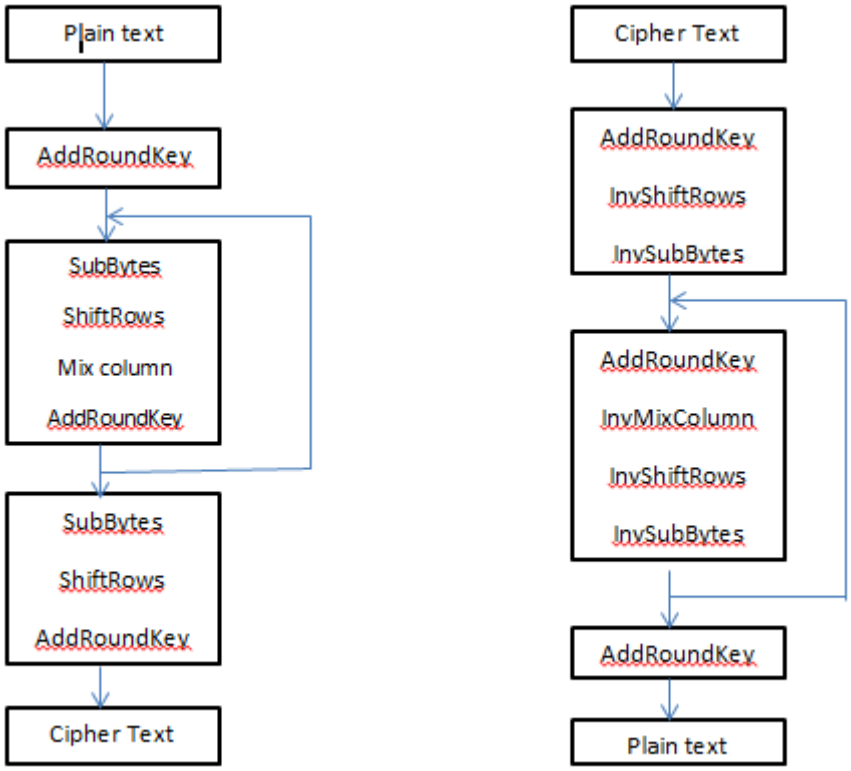
Salah satu keuntungan menggunakan VPN sebagai metode keamanan adalah kemampuan VPN menjaga kerahasiaan data. Untuk menjaganya VPN melakukan *metode enkripsi* terhadap data pesan (*Message*). Enkripsi merupakan proses perubahan pesan asli (*plain text*) menjadi pesan yang tidak dapat dibaca (*chipper text*) dengan menggunakan kunci *(key*). Sedangkan untuk dapat melakukan *enkripsi* digunakan algoritma *kriptografi* (*chipper*). *Chipper* biasanya berupa suatu fungsi matematis. Beberapa jenis *chipper* yang digunakan dalam tugas akhir ini adalah AES,Blwofish dan 3DES.

2. *Advance Encryption Standard (AES)*

AES memiliki ukuran block 128 bit dan ukuran *key* bervariasi mulai dari 128,192 atau 256 bit. , satu satunya serangan yang berhasil membongkar AES merupakan *side channel attack*. NSA (*National security Agency*), sebuah badan

keamanan Amerika Serikat telah menghargai AES sebagai *chipper* yang dapat digunakan untuk mengenkripsi data – data rahasia.

Proses *enkripsi* pada algoritma AES terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan A*ddRoundKey*. Pada awal proses enkripsi, *input* yang telah dikopikan ke dalam *state* akan mengalami transformasi byte *AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang-ulang sebanyak *Nr*. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns.*



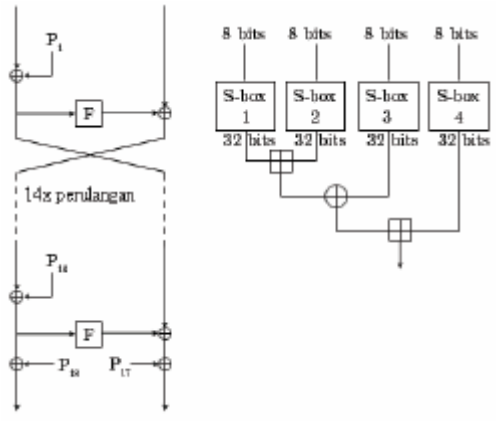
Gambar 2.22 Proses *Enkripsi* dan *Dekripsi pada AES*

*T*ransformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Transformasi *byte* yang digunakan pada *invers cipher* adalah *InvShiftRows, InvSubBytes, InvMixColumns,* dan *AddRoundKey.*

Cara yang paling umum untuk menyerang *block chipper* adalah dengan mencoba berbagai versi dari *chipper* dengan jumlah pengulangan yang dikurangi. AES memiliki 10 kali pengulangan untuk kunci 128 bit, 12 kali pengulangan untuk 192 bit dan 14 kali untuk 236 bit. Sedangkan pada tahun 2006 serangan yang paling jauh yang diketahui oleh khayalak umum adalah 7 kali pengulangan untuk 128 bit, 8

kali untuk 192 bit dan 9 kali untuk 256 bit. Meskipun hampir mendekati tetapi secari prakteknya untuk dapat melakukan serangan ke AES 128 bit memerlukan setidaknya 2^{120} operasi. Dimana hal ini sangat sulit dan memerlukan sumber yang cukup besar.

3. *Blowfish*



Gambar 2.23 Algoritma Blowfish dan Fungsi F

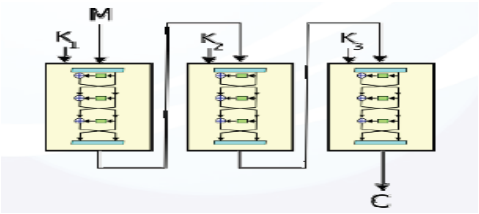
Blowfish memiliki 64 bit *block* dan panjang *key* antara 32 – 448 bit. Memiliki 16x perulangan *Feistel Chiper* dan menggunakan *S-boxes* yang tetap. Blok diagram metode Blowfish dapat dilihat pada Gambar 2.26.

Gambar diatas menunjukkan struktur Feisel dari Blowfish dan bagaimana cara kerja algoritma Blowfish. Setiap baris merepresentasikan 32 bit. Algoritma blowfish menggunakan dua buah array sub kunci. 18 array –entry P – dan 4 buah kotak 256 – entry S. S-box menggunakan input 8 bit dan menghasilkan *output* 32 bit. Satu buah entri dari array – P digunakan setiap *round* (perulangan) dan setelah perulangan terakhir, setiap setengah dari data blok di XOR dengan satu dari 2 array – P yang tidak digunakan.

Diagram di atas juga menunjukkan fungsi F dari Blowfish. Fungsi ini membagi 32 bit input ke dalam empat 8 bit S–box dan menggunakan 8 bit tersebut sebagai input ke dalam S-box. *Output* adalah modulo 2^{32} dan di XOR kan untung menghasilkan 32 bit terakhir.

Key schedule Blowfish dimulai dengan menginisialisasi array P dan S-box dengan nilai yang diambil dari digit hexadecimal oleh pi, yang berisi pola – pola yang mudah diketahui. *Key* kemudian di XORkan dengan nilai dari P. Blok 64 bit dengan nilai nol kemudian dienkripsi dengan algoritma tersebut. Hasil chipertext kemudian menggantikan P1 dan P2. Chipertext kemudian dienkripsi kembali dengan kunci baru. Dan hasil chipertextnya menggantikan P3 dan P4. Hal ini terus berlangsung, menggantikan semua array P dan semua masukan dari S-Box. Algoritma *enkripsi* Blowfish akan berjalan sebanyak 521 kali untuk dapat men-*generate* semua *subkey*.

4. *3DES*



Gambar 2.24 Enkripsi dengan menggunakan 3DES

Triple DES merupakan *chipper block* yang terbentuk dari DES *chipper* dengan menggunakannya tiga kali.

Ketika diketahui bahwa kunci 56 bit DES tidak cukup untuk bertahan melawan *brute force*, TDES dipilih sebagai cara yang mudah untuk memperbesar lebar kunci

tanpa perlu untuk mengganti algoritma. Varian dari TDES yang paling sederhana beroperasi sebagai berikut : DES (k3;DES(k2;DES(k1;M))), dimana M adalah pesan yang akan dienkripsi sedangkan k1,k2 dan k3 adalah kunci DES. Varian ini umumnya diketahui sebagai EEE karena ketiga operasi DES key *length* 168 bit. Tiga 56 bit kunci DES (dengan *parity bit* memiliki total panjang 192 bit). Tetapi karena the *meet-in-the-middle attack*, keamanan efektif hanya menyediakan 112 bit. TDES menggunakan 48 kali pengulangan

Serangan terhadap keamanan TDES membutuhkan sekitar 2^{32} *plaintext* yang diketahui, 2^{113} langkah, 2^{90} enkripsi single DES dan 2^{88} memory melakukan proses enkripsi pada TDES yang menggunakan tiga kunci yang berbeda (3TDES) memiliki key length 168 bit. Tiga 56 bit kunci DES (dengan parity bit memiliki total panjang 192 bit). Tetapi karena the *meet-in-the-middle attack*, keamanan efektif hanya menyediakan 112 bit. TDES menggunakan 48 kali pengulangan.

Serangan terhadap keamanan TDES membutuhkan sekitar 2^{32} *plaintext* yang diketahui, 2^{113} langkah, 2^{90} enkripsi single DES dan 2^{88} memory

2.9. Linux

Linux adalah sebuah sistem operasi komputer bertipe *Unix*. Linux merupakan salah satu contoh perangkat lunak bebas dan *open source* karena sebagian besar kode sumbernya (*source code*) dapat secara bebas dimodifikasi digunakan dan didistribusikan kembali oleh semua orang. Penggunaan nama “Linux” berasal dari nama *kernel*nya (*kernel* Linux), yang dibuat tahun 1991 oleh Linus Torvalds.

Sistemnya, peralatan sistem dan pustakanya umumnya berasal dari sistem operasi GNU(*General Public License*), yang diumumkan tahun 1983 oleh Richard Stallman. Kontribusi GNU adalah dasar dari munculnya nama alternatif GNU/Linux (http://id.wikipedia.org/wiki/Linux). Istilah Linux atau GNU/Linux digunakan sebagai rujukan kepada keseluruhan *distro* Linux (*Linux distribution*), yang di dalamnya disertakan program-program lain pendukung sistem operasi. Contoh program tersebut adalah *server* web, bahasa pemrograman, basisdata, tampilan layar (*Desktop Environment*) , dan aplikasi perkantoran (*office suite*) seperti . *Distro* Linux telah mengalami pertumbuhan yang pesat dari segi popularitas, sehingga lebih populer dari versi UNIX yang menggunakan sistem lisensi dan berbayar (*proprietary)* maupun versi UNIX bebas lain yang pada awalnya.

2.10. Kelebihan Linux

Di sini akan dijelaskan beberapa kelebihan dari sistem operasi Linux/UNIX sebagai *server*:

1. Dapat berjalan pada spesifikasi *Hardware* yang minimal.
2. *Multi User*, Linux dapat digunakan oleh satu atau lebih orang untuk menggunakan program yang sama atau berbeda dalam suatu mesin yang sama pada saat bersamaan, di terminal yang sama atau berbeda.
3. *Multiconsole*, Dalam satu komputer, pengguna dapat melakukan *login* dengan nama *user* yang sama atau berbeda lebih dari satu kali, tanpa perlu menutup sesi sebelumnya. *Multiconsole* dapat dilakukan pada Linux karena Linux merupakan *Non-Dedicated Server* yaitu *user* dapat bekerja seperti halnya

melalui klien menggunakan komputer *server* selagi *server* bekerja melayani klien-klien yang ada.

4. *Linux* yang di khususkan untuk *server* menggunakan perintah (CLI) *Command Line Interface*, sehingga tidak memakan *resource* yang besar untuk menjalankan perintah, dikarenakan pada *server resource* merupakan salah satu aspek yang penting.

2.11. Trixbox CE

Trixbox CE (*Community Edition*) adalah sebuah *VoIP Phone System* berbasiskan sistem *open source* paling populer didunia karena mengkombinasikan paket-paket *Open Source* Telepon terbaik yang disertakan didalam sistem operasi tersebut. Sebelumnya kebanyakan *user* yang ingin mengimplementasikan teknologi VoIP harus dengan usaha yang besar dikarenakan harus menjadi seorang *programer* untuk menghadapi *user interface* yang tidak *friendly* , oleh karena itu untuk mengatasi masalah seperti itu diluncukannya *trixbox* dengan penggunaan yang sangat mudah karena menu utama yang berbasiskan *web* untuk menkonfigurasi dan mengatur sistem, serta paket-paket untuk VoIP *Server* dijadikan dalam satu *bundle* dengan *operating system CentOS* sehingga menjadi sistem *TrixboxCE.*

2.12. Sejarah 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 dagang 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.

2.13. Komponen Trixbox

Pada sistem operasi *trixbox* semua packet yang digunakan merupakan *open source license* yang dapat dikembangkan ditambahkan secara bebas, berikut ini adalah komponen utama dalam sistem operasi *trixbox* :

1. *CentOS* (*community ENTerprice Operating System*)

CentOS adalah sistem operasi yang dikembangkan oleh komunitas kontributor dan pengguna (*user*). Sistem operasi linux *CentOS* adalah 100% *rebuild* kompatibel dengan *RedHat Enterprise Linux (RHEL)*, dan *full compliance* dengan persyaratan redistribusi *RedHat*. *CentOS* ditargetkan untuk siapa saja yang membutuhkan stabilitas (*enterprise class operating system stabilit*) tanpa biaya lisensi dan dukungan dari *RedHat.*

2. *Asterisk*

Asterisk adalah *telepony toolkit open source* yang memungkinkan pengembang untuk membuat beberapa macam aplkasi 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.*

3. *Free PBX*

Free PBX*(Private Branch Exchange)* adalah aplikasi yang digunakan untuk melakukan pengontrolan terhadap jaringan IP telepon *private* dengan konfigurasi *web base*, sehingga untuk melakukan konfigurasi terhadap *asterisk* tidak diperlukan kemampuan *programing* karena *user interface* yang mudah di konfigurasi.

4. *Flash Operator Panel*

Flash Operator Panel (FOP) adalah aplikasi yang digunakan untuk melihat status dari semua *extensions* dan line telepon secara *real time*.

5. *Automated Installation Tools*

Automated Installation Tools digunakan untuk sistem operasi, *script* konfigurasi yang secara otomatis ter-*install* dan ter-*configure* ketika menjalankan *trixbox setup script*

6. *Digium Card auto-config*

Digium Card auto-config untuk penambahan *Digium Hardware* akan secara otomatis ter-*configure*.