BAB 2

LANDASAN TEORI

2.1 Pengertian Jaringan Komputer

Jaringan komputer adalah beberapa komputer yang saling berhubungan dapat melakukan komunikasi dan *share resources* antara satu dengan yang lainnya menggunakan perangkat keras jaringan, seperti *ethernet card, bridge, modem,* dan lain-lain (ANDI dan Wahana Komputer, 2005, p.1). Tiap komputer, *printer* atau peralatan lainnya yang terhubung dengan jaringan disebut *node*. Sebuah jaringan dapat memiliki dua, belasan, puluhan, ribuan, bahkan jutaan *node*.

2.2 Klasifikasi Jaringan

1. *Local Area Network* (LAN)

   LAN merupakan suatu jaringan komunikasi yang saling menghubungkan berbagai jenis perangkat dan menyediakan pertukaran data diantara perangkat-perangkat tersebut dalam lingkup area terbatas dalam LAN adalah *ethernet, token ring, dan* FDDI.

   LAN dirancang untuk tujuan berikut :

   - Beroperasi dalam area geografis yang terbatas.

- Memungkinkan *multi-access* terhadap media dengan *bandwidth* yang besar.
- Mengatur jaringan secara *private* dalam kendali administrasi lokal.
- Menyediakan konektifitas *fulltime* pada *service* lokal.
- Secara fisik menghubungkan *device* yang berdekatan.

2. *Metropolitan Area Network* (MAN)

   MAN merupakan jaringan yang menghubungkan beberapa jaringan komputer dalam wilayah yang lebih luas (Rizky, 2006, p.12). Area cakupan dari MAN lebih besar dari pada LAN, namun lebih kecil dari WAN.

3. *Wide Area Network* (WAN)

   WAN merupakan sistem jaringan yang menghubungkan antar jaringan LAN (Arifin, 2003, p.149). Ruang lingkup pada WAN sangat luas, sudah terpisahkan oleh batas geografis, yang mana memberikan akses ke komputer atau file server yang terletak di lokasi berbeda.

   WAN dirancang dengan tujuan sebagai berikut :

   - Beroperasi di area yang luas dan secara geografis terpisah.
   - Memungkinkan *user* untuk berkomunikasi secara *real-time* dengan *user* lain.
   - Menyediakan servis berupa *e-mail*, internet, *transfer file* dan *e-commerce*.

2.3 Topologi Jaringan

Topologi jaringan (LAN) adalah pola koneksi dari *node-node* sebuah jaringan. Topologi jaringan hanya ditentukan oleh cara menghubungkan antar *node*. Jarak antar *node*, koneksi fisik, tingkat transmisi, dan atau tipe sinyal yang digunakan tidak diperhatikan dalam topologi jaringan, walaupun mereka dapat mempengaruhi pada jaringan yang sesungguhnya.

2.3.1 *Physical Topology*

Physical Topology adalah gambaran secara fisik dari pola hubungan antara komponen-komponen jaringan, yang meliputi *server, workstation, hub, switch,* pengkabelan, dll. Bentuk umum yang biasa digunakan adalah *Bus, Star, dan Ring.*

1. Topologi *Bus*

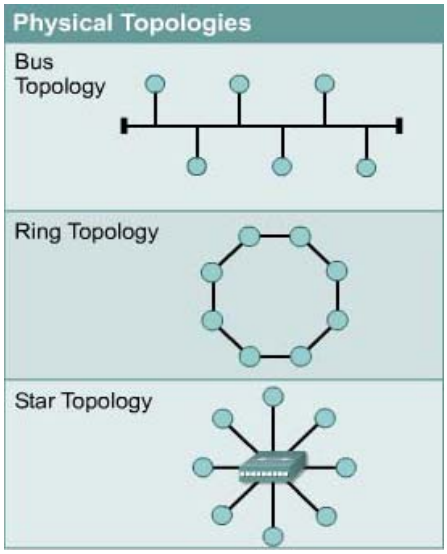
   Pada topologi ini, terdapat sebuah kabel tunggal atau kabel pusat dimana seluruh komputer dan server dihubungkan (Rizky, Microsoft Windows Server 2003, p.18).

2. Topologi *Star*

   Pada topologi ini, setiap komputer pada jaringan terhubung secara langsung dengan server atau *hub* (Rizky, Microsoft Windows Server 2003, p.19).

3. Topologi Ring

Pada topologi ini, semua komputer dan server dihubungkan sehingga terbentuk pola cincin atau lingkaran (Rizky, Microsoft Windows Server 2003,p.20).



Gambar 2.1 : Topologi-topologi yang sering digunakan.

Tabel 2.1 : Perbandingan Topologi *Bus, Star,* dan *Ring.*

Topologi	Kelebihan	Kekurangan
Topologi *Bus*	• Pengembangan jaringan atau penambahan komputer baru tidak mempengaruhi komputer lain.	• Bila terjadi gangguan di sepanjang kabel pusat, maka akan menggangu jaringan secara keseluruhan.

Topologi *Star*	• Bandwidth dapat digunakan secara optimal, karena setiap komputer mempunyai kabel tersendiri yang terhubung dengan server. • Bila terjadi gangguan di suatu jalur kabel, tidak akan menggangu jaringan secara menyeluruh.	• Dibutuhkan banyak kabel
Topologi *Ring*	• Tidak akan terjadi *collision* atau tabrakan pengiriman data.	• Bila suatu komputer atau kabel mengalami gangguan,maka seluruh jaringan akan terganggu.

2.3.2 *Logical Topology*

Logical Topology adalah gambaran secara maya bagaimana sebuah *host* dapat berkomunikasi melalui medium. Bentuk umum yang biasa digunakan adalah *broadcast* dan *Token Passing.*

1. *Broadcast Topology*

   Pada topologi ini, setiap *host* yang mengirim paket data akan mengirimkan paket tersebut ke semua *host* (*broadcast*) pada media komunikasi jaringan.

2. *Token Passing Topology*

   Pada topologi ini, setiap *host* mempunyai kemampuan mengendalikan akses jaringan dengan mem-*pass*-kan sebuah *token* elektronik yang sekuensial akan melalui masing-masing *host* dari jaringan tersebut. Ketika sebuah *host* mendapatkan *token* tersebut, berarti *host* tersebut diperbolehkan untuk mengirimkan data pada jaringan. Jika *host* tersebut tidak memiliki data yang akan dikirim, maka *token* akan dilewatkan ke *host* berikutnya. Kejadian ini akan terus berulang.

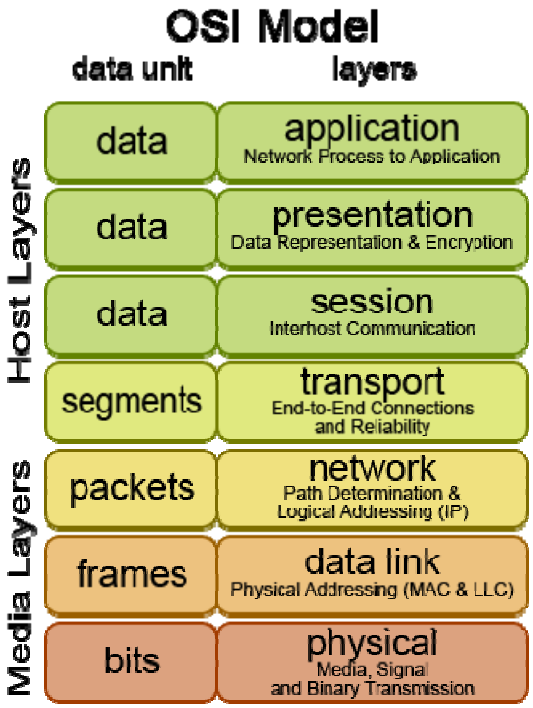
2.4 Protokol Jaringan

Protokol jaringan adalah suatu aturan yang mengatur cara-cara dalam suatu jaringan bertukar informasi. Model yang umum dijadikan referensi untuk mempelajari protokol jaringan adalah model referensi lapisan *Open System Interconnection* (OSI *Layers*). Sedangkan *Internet Protocol Suite* (TCP/IP) merupakan protokol jaringan yang saat ini sangat umum digunakan untuk *internetworking*.

2.4.1 Model Referensi OSI

Model referensi jaringan terbuka OSI atau *OSI Reference Model for open networking* adalah sebuah model arsitektural jaringan yang dikembangkan oleh badan *International Organization for Standardization* (ISO) di Eropa pada tahun 1977. OSI sendiri merupakan singkatan dari

Open System Interconnection. Model ini disebut juga dengan model "Model tujuh lapis OSI" (*OSI seven layer model*) *(www.wikipedia.com*).



Gambar 2.2 : *Seven OSI Layer*

Model OSI membagi fungsi – fungsi dari suatu protokol menjadi beberapa *layer*. Setiap *layer* mempunyai properti yang menggunakan fungsi *layer* di bawahnya, memproses data pada *layer* tersebut, lalu mengirim pada *layer* selanjutnya.

1. *Layer 1 – Physical Layer*

   Physical Layer merupakan lapisan terbawah pada model OSI. *Layer* ini berhubungan langsung dengan *hardware. Physical Layer* mendefinisikan semua spesifikasi fisik dan elektris untuk semua peralatan meliputi level tegangan, spesifikasi kabel, tipe konektor dan

timing. Fungsi utama dari *layer* ini adalah bertanggung jawab untuk mengaktifkan dan mengatur *physical interface* dari jaringan komputer, memodulasi data digital antara peralatan yang digunakan *user* dengan sinyal yang berhubungan. Peralatan yang merupakan *physical layer* antara lain *hub* dan *repeater*.

2. *Layer 2 – Data Link Layer*

   Berfungsi untuk menentukan bagaimana *bit-bit* data dikelompokkan menjadi format yang disebut sebagai *frame*. Selain itu, pada level ini terjadi koreksi kesalahan (*error notification*), pemesanan pengiriman data (*flow control*), pengalamatan perangkat keras (seperti halnya *Media Access Control Address* (MAC Address)), dan menentukan bagaimana perangkat-perangkat jaringan seperti *hub*, *bridge*, *repeater*, dan *switch layer 2* beroperasi. Spesifikasi IEEE 802, membagi level ini menjadi dua level anak, yaitu lapisan *Logical Link Control* (LLC) dan lapisan *Media Access Control* (MAC). *Switch* dan *bridge* merupakan peralatan yang bekerja pada layer ini.

3. *Layer 3 – Network Layer*

   Layer ini menyediakan koneksi dan pemilihan jalur antar dua sistem. Berfungsi untuk mendefinisikan alamat-alamat IP (*addressing*), membuat *header* untuk paket-paket (*logical protocol*), dan kemudian melakukan *routing* (*network routing*) melalui *internetworking* dengan menggunakan *router* dan *switch layer-3*.

4. *Layer 4 – Transport Layer*

   Berfungsi untuk memecah data kedalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Selain itu, pada level ini juga membuat sebuah tanda bahwa paket diterima dengan sukses (*acknowledgement*), dan mentransmisikan ulang terhadap paket-paket yang hilang di tengah jalan. Dalam menyediakan layanan yang *reliabel* pada *layer* ini menyediakan *error detection* dan *recovery* serta *flow control*.

5. *Layer 5 – Session Layer*

   Berfungsi untuk mendefinisikan bagaimana koneksi dapat dibuat, dipelihara, atau dihancurkan. Selain itu, di *level* ini juga dilakukan resolusi nama. *Session Layer* menyediakan servis kepada *Layer Presentation*. *Layer* ini juga mensinkronisasi dialog diantara dua *host layer presentation* dan mengatur pertukaran data.

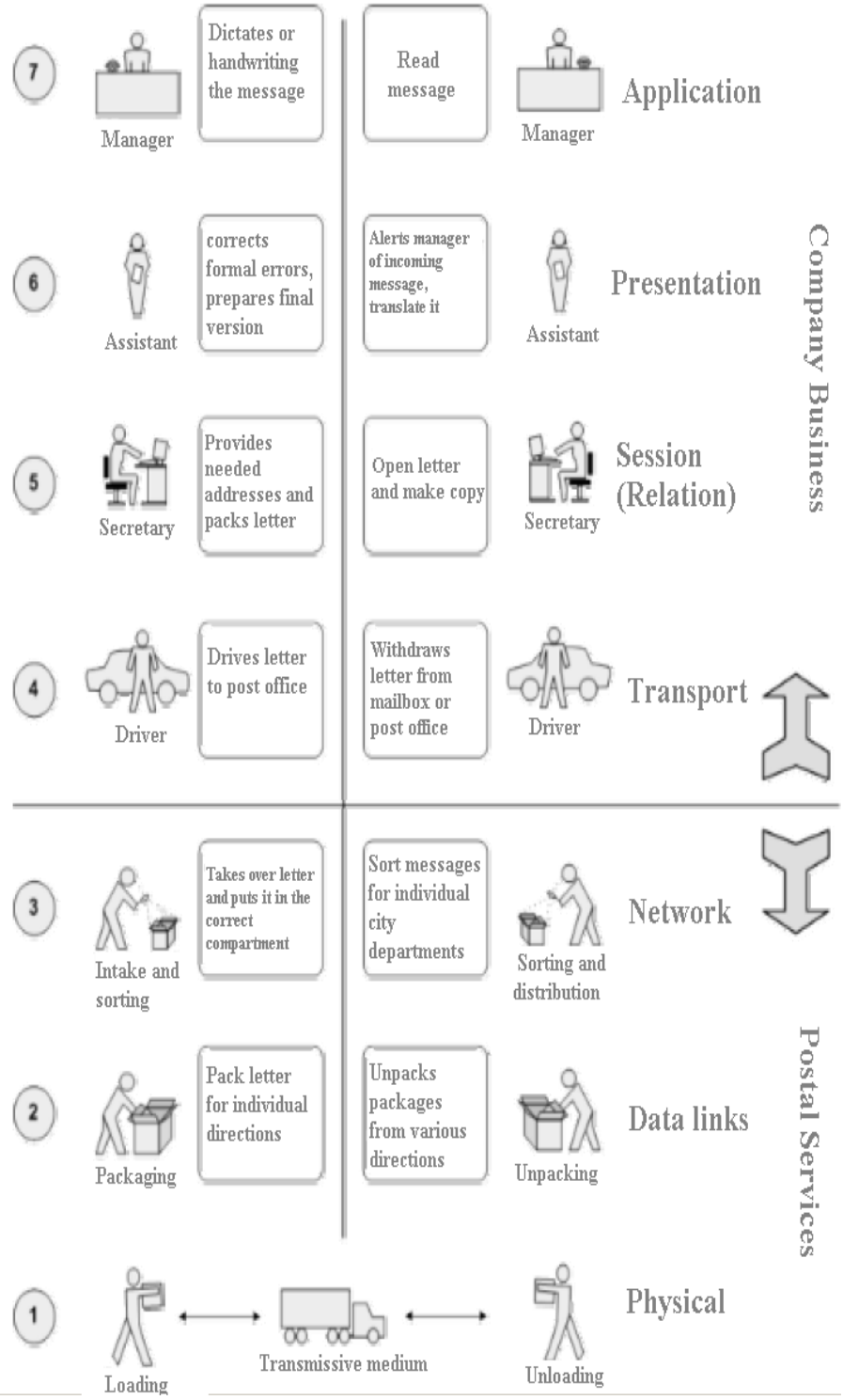
6. *Layer 6 – Presentation Layer*

   Berfungsi untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi (*application layer*) ke dalam format yang dapat ditransmisikan melalui jaringan agar dimengerti oleh aplikasi (*application layer*) di sistem lain. Jika diperlukan, pada *layer* ini dapat menerjemahkan beberapa data format yang berbeda, kompresi dan enkripsi. Protokol yang berada dalam level ini adalah perangkat lunak redirektor (*redirector software*), seperti layanan *workstation*

(dalam Windows NT) dan juga *network shel*l (semacam *Virtual Network Computing* (VNC) atau *Remote Desktop Protocol* (RDP)).

7. *Layer 7 – Application Layer*

   Application layer merupakan *layer* teratas dari model OSI. *Layer* ini adalah *layer* yang paling dekat dengan pengguna (*user*). Berfungsi sebagai antarmuka dengan aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan kesalahan. Protokol yang berada dalam lapisan ini adalah HTTP, FTP, SMTP, dan NFS. *Layer* ini merupakan tempat dimana *user* atau pengguna berinteraksi dengan komputer. *Layer* ini sebenarnya hanya berperan ketika dibutuhkan akses ke *network*.



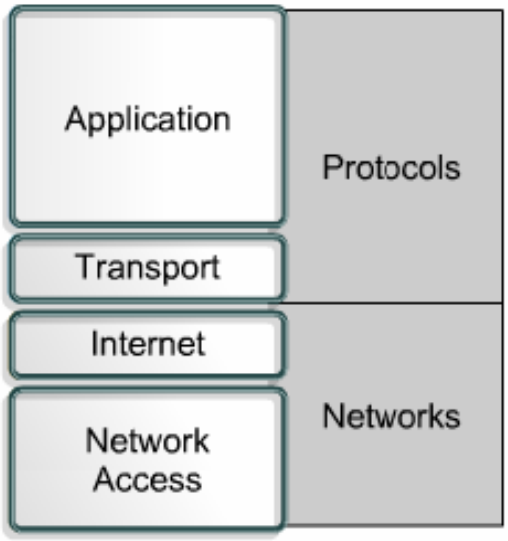
Gambar 2.3 : Analogi kerja tiap *layer* pada model *OSI*

2.4.2 *Internet Protocol Suite*

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah TCP/IP *stack*.

Protokol TCP/IP dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). TCP/IP merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme transport jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat IP (*IP Address*) yang mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di internet. Protokol ini juga bersifat ***routable*** yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti Microsoft Windows dan keluarga UNIX) untuk membentuk jaringan yang heterogen.

Model TCP/IP ini mempunyai 4 layer, yaitu : *application layer*, *transport layer*, *internet layer*, dan *network access layer*. Beberapa layer pada model TCP/IP mempunyai nama yang sama dengan model OSI.



Gambar 2.4 : Model TCP/IP

2.4.2.1 *Application Layer*

Application layer pada model TCP/IP menangani protokol tingkat tinggi yang berhubungan dengan representasi, *encoding* dan *dialog control*. Protokol TCP/IP menggabungkan seluruh hal yang berhubungan dengan aplikasi kedalam satu *layer* dan menjamin data dipaketkan dengan benar sebelum masuk ke *layer* berikutnya. Beberapa program berjalan pada layer ini, menyediakan layanan langsung kepada *user*. Program-program ini dan protokol yang berhubungan meliputi HTTP (*The World*

Wide Web), FTP, TFTP (*File Transport*), SMTP (*Email*), Telnet, SSH (*Secure remote login*), dan DNS (*Name management*).

2.4.2.2 *Transport Layer*

Layer transport menyediakan layanan transportasi dari *host* sumber ke *host* tujuan. *Layer transport* merupakan suatu koneksi logikal diantara *endpoints* dari suatu jaringan, yaitu *sending host* dan *receiving host*. *Transport protocol* membuat segmen dan mengumpulkan kembali aplikasi *layer* diatasnya menjadi *data stream* yang sama diantara *endpoints*. *Data stream* layer transport menyediakan layanan transportasi *end-to-end*. Protokol-protokol yang berfungsi pada layer ini adalah :

- *Transmission Control Protocol* (TCP)

  TCP berfungsi untuk mengubah suatu blok data yang besar menjadi segmen-segmen yang diberi nomor dan disusun secara berurutan agar si penerima dapat menyusun kembali segmen-segmen tersebut seperti pada waktu pengiriman. TCP ini adalah jenis protokol *connection oriented* yang memberikan layanan bergaransi.

- *User Datagram Protokol* (UDP)

  UDP adalah jenis protokol *connectionless oriented*. UDP bergantung pada lapisan atas untuk mengontrol kebutuhan data. Oleh karena penggunaan *bandwidth* yang efektif,

UDP banyak dipergunakan untuk aplikasi-aplikasi yang tidak peka terhadap gangguan jaringan seperti SNMP dan TFTP.

2.4.2.3 *Internet Layer*

Tujuan dari layer internet adalah untuk memilih jalur/*path* terbaik bagi paket-paket data di dalam jaringan. Protokol utama yang berfungsi pada *layer* ini adalah *Internet Protocol* (IP). Penentuan jalur terbaik dan *packet switching* terjadi pada layer ini. Protokol-protokol yang berfungsi pada *layer* ini antara lain adalah IP, ARP, RARP, BOOTP, DHCP, dan ICMP.

- *Internet Protocol*

  IP merupakan protokol yang memberikan alamat atau identitas logika untuk peralatan di jaringan komputer. IP mempunyai tiga fungsi utama, yaitu servis yang tidak bergaransi (*connectionless oriented*), pemecahan (*fragmentation*) dan penyatuan paket-paket, fungsi meneruskan paket (*routing*).

- *Address Resolution Protocol* (ARP)

  Protokol yang mengadakan translasi dari IP address yang diketahui menjadi alamat *hardware* atau *MAC address*. ARP ini termasuk jenis protokol *broadcast*.

- *Reverse Address Resolution Protocol* (RARP

  Protokol yang berguna mengadakan translasi *MAC address* yang diketahui menjadi *IP address*. Router menggunakan protokol RARP ini untuk mendapatkan *IP address* dari suatu *MAC address* yang diketahuinya.

- *Bootstrap Protocol* (BOOTP)

  Protokol yang digunakan untuk proses *boot diskless workstation*. Dengan protokol ini, suatu IP address dapat diberikan ke suatu peralatan di jaringan berdasarkan *MAC address*-nya.

- *Dynamic Host Configuration Protocol* (DHCP)

  DHCP merupakan kelanjutan protokol *bootstrap* yang dapat memberikan *IP address* secara otomatis ke suatu *workstation* yang menggunakan protokol TCP/IP. DHCP bekerja dengan relasi *client-server*.

- *Internet Control Message Protocol* (ICMP)

  ICMP adalah protokol yang berguna untuk melaporkan jika terjadi suatu masalah dalam pengiriman data.

2.4.2.4 *Network Access Layer*

Network Access Layer adalah metode yang digunakan untuk mengirim paket dari dua *host* yang berbeda. Proses ini dapat dikendalikan baik oleh *software device driver* dari kartu jaringan,

maupun pada *firmware* atau spesialis *chipset*. Hal ini dapat melaksanakan fungsi *data link* seperti penambahan *packet header*, menyiapkan paket tersebut untuk transmisi, lalu mengirim *frame* melalui media fisik. Persamaan dari *Data Link layer* dan *Physical Layer* dari model OSI yaitu *Network Access Layer* mengawasi pengalamatan secara *hardware* dan mendefinisikan protokol untuk transmisi fisik data.

2.5 Pengalamatan IP

Alamat IP (*Internet Protocol Address* atau sering disingkat IP) adalah deretan angka biner antar 32-bit sampai 128-bit yang dipakai sebagai alamat identifikasi untuk tiap komputer host dalam jaringan Internet. Panjang dari angka ini adalah 32-bit (untuk IPv4 atau IP versi 4), dan 128-bit (untuk IPv6 atau IP versi 6) yang menunjukkan alamat dari komputer tersebut pada jaringan Internet berbasis TCP/IP (*http://id.wikipedia.org/wiki/Alamat_IP*).

Anda bisa menggambarkan pengalamatan IP dengan tiga metode :

- *Dotted-decimal*, sepeti 172.16.30.56
- *Binner*, seperti 10101100.00010000.00011110.00111000
- *Hexa-decimal*, seperti AC.10.1E.38

2.5.1 Kelas-kelas Pengalamatan IP

Perancang internet membuat *class* dari jaringan berdasarkan ukuran jaringan.

- Class A address

  Class A didesain untuk mendukung *network* yang besar, dengan jumlah lebih dari 16 juta *host address* yang tersedia. IP *address Class* A hanya menggunakan oktet yang pertama untuk menunjukkan *network address*, dan tiga oktet sisanya tersedia untuk *host address*.

- Class B address

  Class B *address* didesain untuk mendukung kebutuhan jaringan dengan ukuran menengah sampai dengan ukuran besar. Sebuah IP *address Class* B menggunakan dua oktet pertama dari empat oktet untuk menunjukkan *network address*, dan sisanya menunjukkan *host address*.

- Class C address

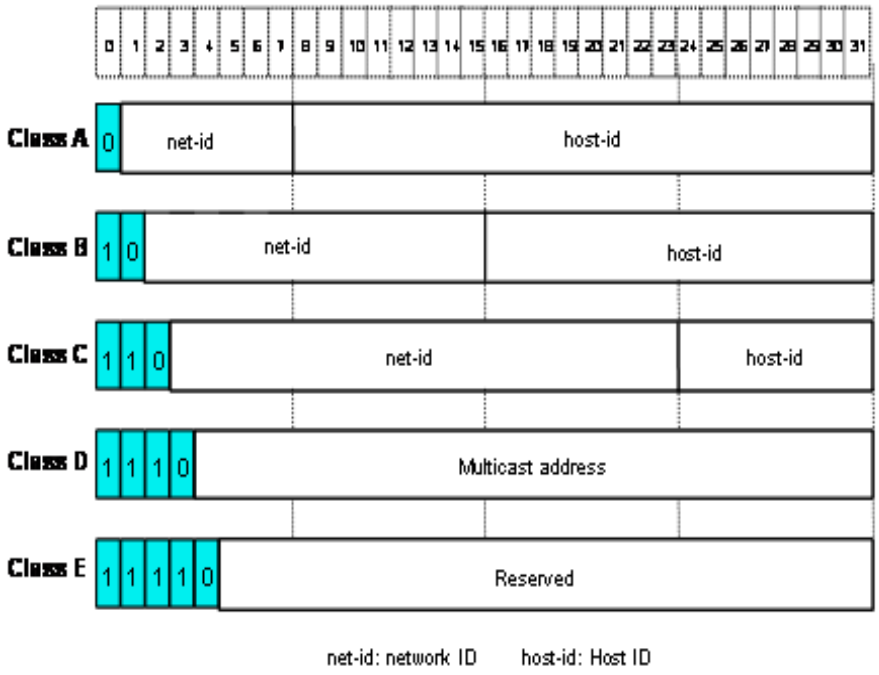
  Class C *address* adalah kebanyakan yang dipakai untuk alamat *address* yang sebenarnya. Alamat ini dimaksudkan untuk mendukung jaringan kecil dengan jumlah maksimum 254 *host*.

- Class D address

  Class D *address* diciptakan untuk memungkinkan *multicasting* di dalam suatu IP *address*. *Multicast address* adalah *network address* unik yang menunjukkan paket dengan *address* tujuan ke *group predefined* dari sebuah IP *address*, oleh karena itu *single unit* dapat mentransmisikan aliran tunggal dari data secara *simultan* ke penerima lebih dari satu.

- Class E address

  Class E *address* telah ditetapkan, namun *Internet Engineering Task Force* (IETF) menetapkan *address* ini untuk keperluan riset, oleh karena itu tidak ada IP di *Class* E address yang dikeluarkan untuk digunakan dalam *internet*.



Gambar 2.5 : Kelas Pengalamatan IP

Tabel 2.2 : Range alamat IP tiap kelas

	Range
Class A	0.x.x.x – 127.x.x.x
Class B	128.x.x.x – 191.x.x.x
Class C	192.x.x.x – 223.x.x.x
Class D	224.x.x.x – 239.x.x.x
Class E	240.x.x.x – 255.x.x.x

Tabel 2.3 : Alamat IP Khusus

Alamat	Fungsi
Alamat *network* semuanya 0	Diartikan ”jaringan atau segmen ini”.
Alamat *network* semuanya 1	Diartikan ”semua jaringan”.
Network 127.0.0.1	Dicadangkan untuk *local node* dan memungkinkan *node* tersebut mengirimkan paket tes ke dirinya sendiri tanpa menimbulkan lalu lintas jaringan.
Alamat *node* semuanya 0	Diartikan "alamat jaringan" atau semua *host* pada jaringan spesifik.
Alamat *node* semuanya 1	Diartikan "semua *node*" pada jaringan spesifik, sebagai contoh, 128.2.255.255

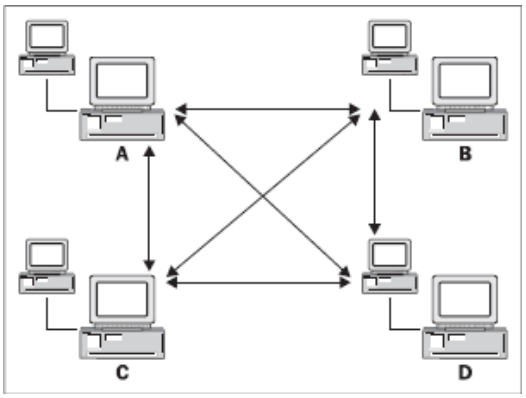
	artinya "semua *node*" pada jaringan 128.2 (alamat *Class B*).
Seluruh alamat IP di-set 0	Digunakan oleh *router* Cisco untuk menunjukan rute *default*. Bisa juga berarti "semua *network*".
Seluruh alamat IP di-set 1 (255.255.255.255)	*Broadcast* ke semua *node* pada *current network* (network yang sedang aktif). Terkadang disebut "all 1st *broadcast*" atau *broadcast* terbatas.

2.6 *Virtual Private Network* (VPN)

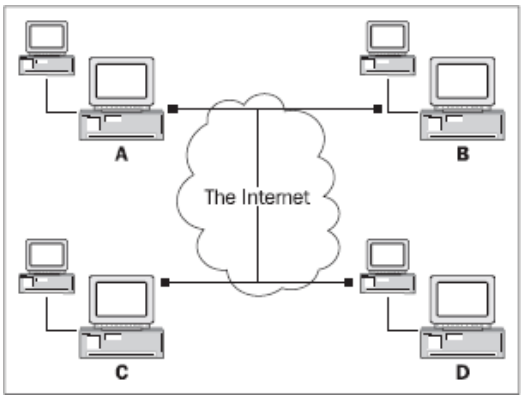
Menurut Stallings (2003) *Virtual Private Network* (VPN) adalah sebuah jaringan *private* yang dibuat di jaringan *public* dengan menggunakan *internet* sebagai media komunikasinya. Jika kita jabarkan berdasarkan suku katanya maka pengertian VPN adalah :

Virtual, karena tidak ada koneksi jaringan secara langsung antara dua atau lebih komputer, melainkan hanya ada koneksi virtual yang disediakan oleh VPN Software, biasanya melalui koneksi internet.

Private, karena hanya anggota dari badan/organisasi/perusahaan yang menggunakan VPN tersebut yang dapat mengakses dan melakukan transfer data.



Gambar 2.6: Jaringan biasa



Gambar 2.7 : *Virtual Private Network*

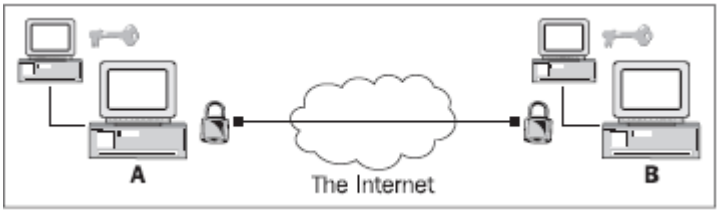
2.6.1 Cara Kerja VPN

Hal terutama yang dibutuhkan oleh sebuah VPN untuk bekerja adalah adanya koneksi internet yang baik. Kemudian juga diperlukan internet *gateway router* untuk melakukan *setting* akses internet bagi para staf. *Router* ini dikonfigurasikan untuk melindungi jaringan lokal perusahaan atau organisasi dari orang yang tidak berhak mengaksesnya

melalui internet. Dapat juga dikatakan *router* ini berfungsi sebagai *firewall*.

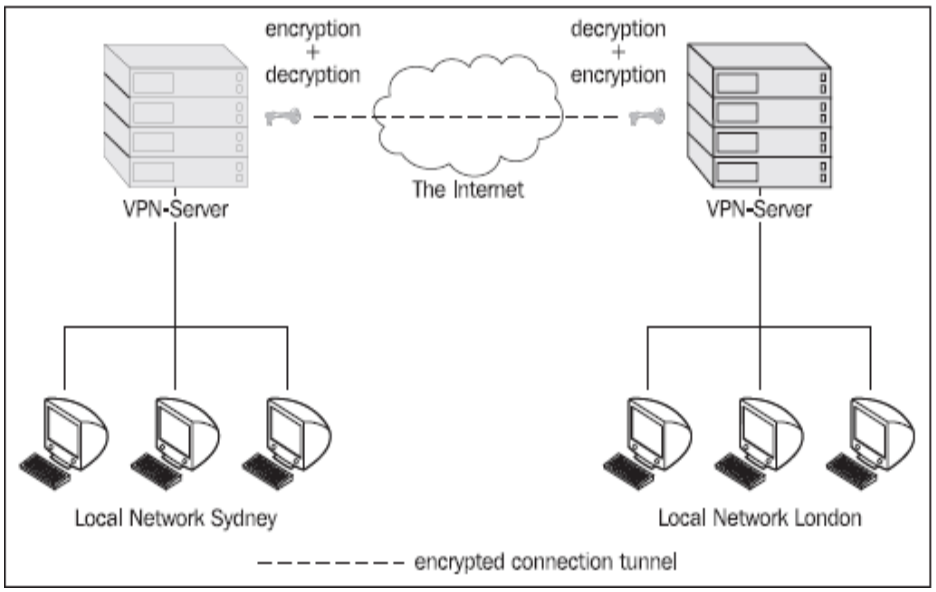
Kemudian software VPN di *install* pada *router* yang berfungsi sebagai *firewall* ini. Kemudian di konfigurasikan agar dapat tersambung dan tercipta sebuah koneksi *virtual*. Jika tahap ini sukses maka dua atau lebih jaringan perusahaan atau kantor sudah dapat terhubung melalui jaringan *virtual* (internet) layaknya jaringan nyata. Sudah dapat saling mengirim data dan saling mengakses jaringan, namun belum menjadi jaringan *private* karena belum terlindungi, sehingga orang lain yang memakai internet juga dapat mengambil data yang dikirim melalui jaringan ini.

Untuk menjadikan jaringan ini menjadi sebuah jaringan yang *private,* maka solusinya adalah dengan menggunakan enkripsi. *Traffic* VPN antara dua atau lebih perusahaan/kantor yang menggunakan VPN di kunci dengan enkripsi, dan hanya komputer atau orang yang berhak saja yang dapat membuka kunci dan melihat data yang dikirim dengan enkripsi tersebut. Data yang dikirim akan di enkripsi terlebih dahulu lalu setelah sampai pada tujuan akan di dekripsi. Enkripsi menjaga data tetap aman dalam jaringan internet yang begitu luas. Seperti terowongan kereta yang melewati gunung atau bawah tanah. Enkripsi menjaga transfer data tetap aman melalui media internet yang luas. Menciptakan terowongan *virtual,* jalur *private,* atau yang lebih dikenal dengan teknologi *tunnelling*.



Gambar 2.8 : *Tunnelling Technology*

Jadi VPN adalah jaringan virtual yang menggunakan internet sebagai media perantara (pengganti kabel ataupun *wireless hardware*) yang dibangun di antara dua *internet access router* yang dilengkapi *firewall* dan *software* VPN. *Software* harus di-*install* di masing-masing *router* yang berfungsi sebagai penghubung, *firewall* harus di-*setting* untuk pemberian akses dan pertukaran data melalui VPN harus di enkripsi. Enkripsi harus diberikan pada semua partner yang menggunakan VPN, sehingga pertukaran data hanya dapat dilakukan dan diterima oleh partner yang berhak saja.

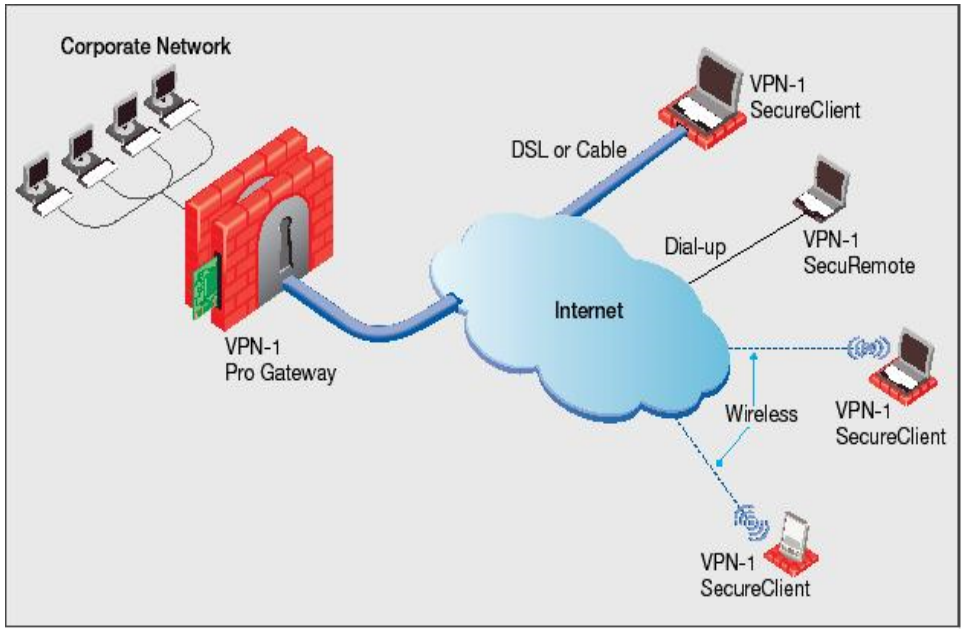


Gambar 2.9 : *Encrypt-Decrypt* VPN

2.6.2 Jenis-jenis VPN

2.6.2.1 *Remote Access* VPN

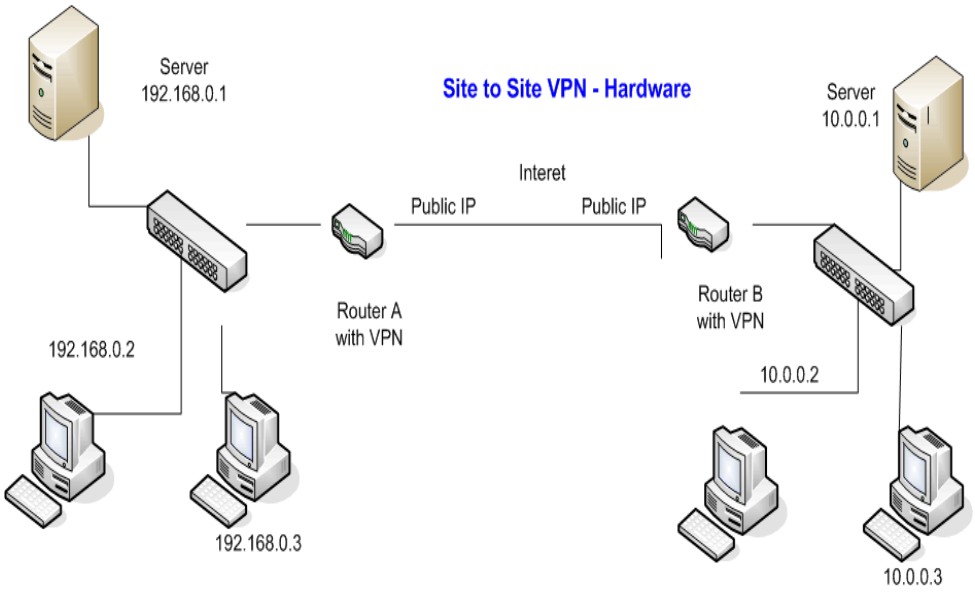
Remote Access VPN memungkinkan akses kapan saja dan dimana saja ke jaringan perusahaan/kantor. Jaringan ini biasa digunakan atau diminta oleh pegawai perusahaan yang berpergian jauh tetapi ingin selalu terhubung dengan jaringan perusahaannya.



Gambar 2.10 : *Remote Access* VPN

2.6.2.2 *Site-to-Site* VPN

Site-to-Site VPN disebut juga *router-to-router* VPN merupakan salah satu alternatif infrastruktur WAN yang biasa digunakan. VPN jenis ini menghubungkan dua atau lebih kantor cabang, kantor pusat, ataupun partner bisnis ke seluruh jaringan perusahaan.



Gambar 2.11 : *Site-to-site VPN*

Site-to-Site VPN terbagi menjadi dua, yaitu:

1. *Intranet* VPN

   Intranet VPN digunakan untuk menghubungkan antara kantor pusat dengan kantor cabang.

2. *Extranet* VPN

   Extranet VPN digunakan untuk menghubungkan suatu perusahaan dengan perusahaan lainnya (contohnya mitra kerja, pelanggan, atau *supplier*).

2.6.3 VPN *Security*

Ada tiga hal dalam pengamanan IT dan juga berlaku dalam VPN yang harus selalu dimiliki :

1. *Privacy* (*Confidentiality*) : Data yang dikirimkan hanya dapat dibuka/diakses oleh yang berhak.
2. *Reliability* (*Integrity*) : Data yang dikirimkan tidak boleh mengalami perubahan dari pengirim data ke penerima data.
3. *Availability* : Data yang dikirimkan harus tersedia ketika dibutuhkan.

Semua tujuan ini harus dicapai dengan menggunakan software, hardware, ISP, dan kebijakan keamanan yang tepat. Keamanan VPN itu sendiri dapat dicapai dengan menjaga lalu lintas (*traffic*), metode enkripsi yang kuat, teknik otentikasi yang aman, dan *firewall* yang mengatur *traffic* ke dan dari *tunnel*.

2.6.3.1 Enkripsi

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti. Dengan enkripsi, kita mengubah isi dari data yang kita kirim sehingga data tersebut tidak dapat dibaca oleh orang yang tidak berhak mendapatkannya. Informasi yang tidak acak disebut *clear-text* sedangkan yang sudah diacak disebut *cipher-text*. Di

setiap *tunnel VPN* terdapat *VPN gateway*. *Gateway* tempat pengiriman data mengenkripsi atau mengubah informasi *cleartext* menjadi *cipher-text* sebelum dikirim melalui *tunnel* ke internet. *VPN gateway* di tempat penerima mendekripsi atau mengubah *cipher-text* tersebut kembali menjadi *clear-text*.

Enkripsi terdiri dari dua jenis, yaitu *symmetric encryption* dan *asymmetric encryption*. *Asymmetric encryption* menggunakan *public* dan *private key* dalam proses enkripsi dan dekripsi sedangkan *symmetric encryption* menggunakan *key* yang sama dalam proses enkripsi dan dekripsi. Berikut merupakan metode-metode *encryption* :

1. *Symmetric Encryption*

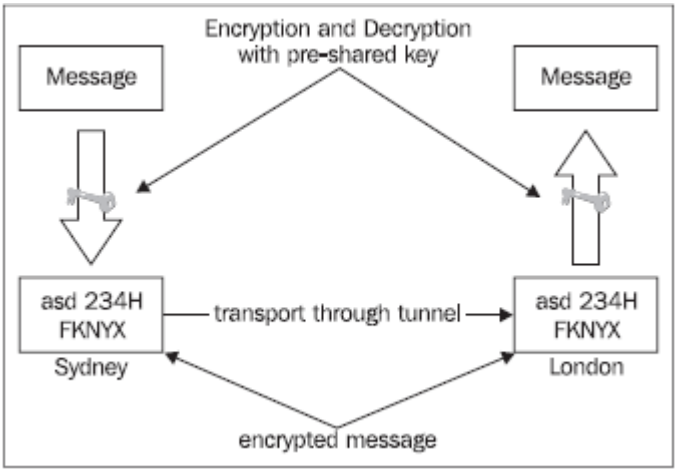
   Symmetrical key encryption menggunakan *private key* berarti komputer pengirim dan penerima sama-sama menggunakan kunci yang sama untuk mengenkripsi dan mendekripsi informasi. Karena satu *key* digunakan bersama-sama untuk enkripsi dan dekripsi, maka harus ada pengertian antara kedua pihak untuk menjaga kerahasiaan *key* tersebut.

   Semua yang mempunyai kunci enkripsi dapat mendekripsi data apa saja yang ada dalam lalu lintas VPN. Jika orang yang tak berwenang memiliki kunci enkripsi , ia dapat mendekripsi data yang ada dan masuk ke setiap jaringan yang terhubung melalui VPN. Selain itu kunci

enkripsi juga dapat dibuka dengan melakukan *brute force attack*. Hanya masalah waktu sampai sang *attacker* dapat membuka kunci enkripsi.

Oleh karena itu, *software* VPN seperti IPsec mengganti kunci enkripsi secara berkala dalam suatu *interval* waktu. Setiap kunci enkripsi hanya berlaku dalam jangka waktu tertentu.

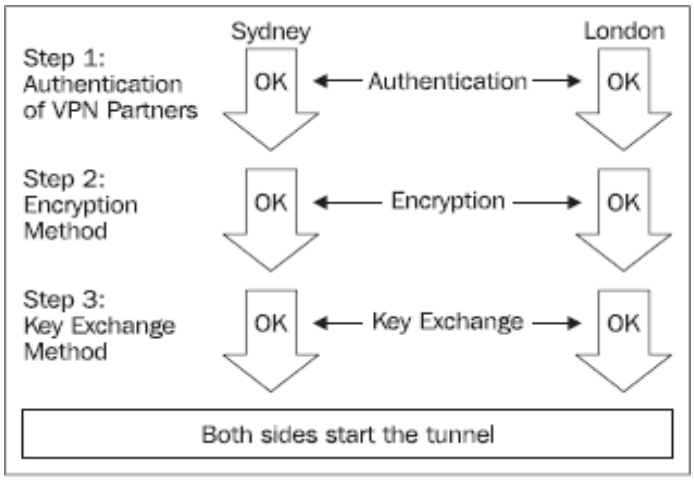
IPsec, teknologi VPN yang paling sering digunakan mempunyai protokol penggantian kunci enkripsi sendiri. Protokol ini diberi nama *Internet Key Exchange* (IKE).



Gambar 2.12 : *Symmetric Encryption*

Dalam klasik VPN yang menggunakan *symmetric key*, ada beberapa lapis otentikasi, pergantian kunci, dan enkripsi/dekripsi. Dibawah ini adalah tiga langkah dari VPN yang menggunakan *symmetric encryption*.

1. Pengirim dan penerima harus saling melakukan otentikasi satu sama lain.
2. Mereka harus saling setuju dalam metode pengenkripsian.
3. Mereka harus saling setuju dalam metode penggantian kunci.



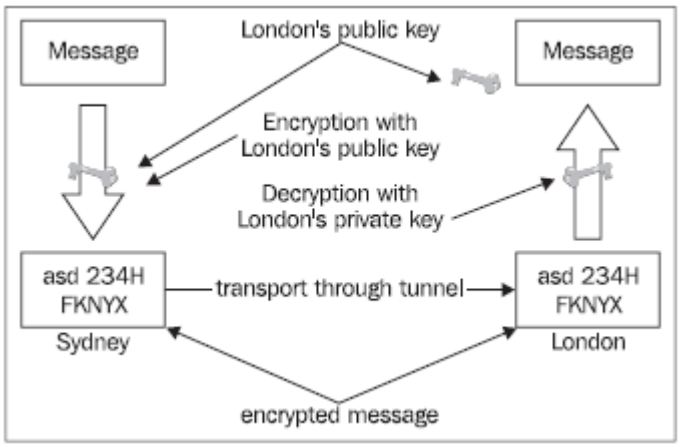
Gambar 2.13 : *Three steps of using symmetric encryption*

Hal inilah yang menjadikan VPN sedikit lebih kompleks dan sulit.

2. *Asymmetric Encryption*

Asymmetrical Key Encryption mengenkripsi informasi dengan suatu *key* dan mendekripsi dengan *key* yang lain. Sistem ini menggunakan kombinasi dari dua buah *key*, yaitu *private key* yang disimpan untuk diri sendiri, dan *public key* yang diberikan untuk *remote user*.

SSL/TLS menggunakan salah satu metode pengenkripsian *asymmetric encryption* ini untuk memastikan identifikasi dari masing-masing pengguna VPN.



Gambar 2.14 : *Asymmetric Encryption*

Pada contoh di atas, sebuah pesan di enkripsi di Sidney menggunakan *public key* dari London. Hasil dari enkripsi tersebut berupa kode dikirim ke London yang hanya dapat di buka menggunakan London *private key*.

Prosedur yang sama dapat juga dilakukan untuk melakukan otentikasi. London mengirim sejumlah angka *random* ke Sidney, dimana akan di *encode* di Sidney menggunakan *private key* dan dikirim kembali. Di London, menggunakan Sidney *public key* angka tersebut dapat di *decode*. Jika angka yang dikirimkan kembali benar, maka pasti yang mengirim kembali adalah pemegang *private key* Sidney. Sistem ini disebut ***digital signature*.**

2.6.3.2 *Authentication*

Selain *encryption*, salah satu aspek penting dalam VPN, yaitu memastikan identitas suatu *user* (*User Authentication*) dan data sampai tanpa adanya kerusakan atau modifikasi *(Data Authentication*).

2.6.3.2.1 *User Authentication*

Dengan *user authentication*, orang yang tidak berhak masuk ke *network* dapat dikenali. Ada beberapa metode *user authentication* antara lain :

1. *Pre-Shared Key*

   Pre-shared key adalah *password* yang diberikan kepada *user* yang tidak memiliki hubungan dengan infrastruktur VPN. *Password* ini memberikan cara mudah bagi *remote user* tertentu untuk masuk ke dalam VPN.

2. *Digital Signatures*

   Digital Signatures adalah bukti elektronik untuk membuktikan identitas *user*. Sertifikat / *Signature* ini disimpan di *remote computer* atau *token* yang dibawa *user*. Sekarang ini algoritma *public key* RSA dan *Digital Signature Standard* (DSS) telah didukung oleh *digital signature*.

3. *Hybrid Mode Authentication*

   Hybrid Mode Authentication memperbolehkan organisasi untuk menginterasikan sistem *authentication* seperti SecureID, TACACS+, dan RADIUS dengan VPN.

2.6.3.2.2 *Data Authentication*

Untuk memastikan apakah data tidak berubah dalam perjalanan, sistem VPN menggunakan *data authentication*. Salah satu teknik *data authentication* adalah *hash function*. Teknik ini membuat suatu angka, yang disebut *hash*, berdasarkan dari panjang bit tertentu. Pengirim menambahkan angka *hash* tersebut ke dalam paket data sebelum *encryption*. Ketika penerima mendapatkan data dan melakukan *decription*, penerima akan melakukan perhitungan *hash* kembali. Apabila kedua angka *hash* tersebut cocok, maka dipastikan data tidak mengalami perubahan dalam perjalanan.

2.7 OpenVPN

OpenVPN adalah aplikasi *open-source* untuk *Virtual Private Network* (VPN), dimana aplikasi tersebut dapat membuat koneksi *point-to-point tunnel* yang telah terenkripsi. OpenVPN menggunakan *private keys*, *certificate*, atau *username/password* untuk melakukan authentikasi dalam membangun koneksi, dimana untuk enkripsi menggunakan *OpenSSL*.

2.7.1 Sejarah OpenVPN

Table 2.4 : Sejarah OpenVPN versi 1

Tanggal	Versi	*Feature* Penting / Perubahan
13 Mei 2001	0.90	Peluncuran awal, dengan hanya sedikit fungsi seperti protokol internet melalui UDP, dan hanya memiliki satu mekanisme enkripsi.
26Desember2001	0.91	Penambahan mekanisme enkripsi.
23 Maret 2002	1.0	Pengesahan berbasis TLS dan mekanisme perubahan kunci di tambahkan. Buku manual pertama.
28 Maret 2002	1.0.2	Perbaikan pada *bug* dan pengembangan, terutama bagi sistem yang *rpm-based* seperti Redhat.
9 April 2002	1.1.0	Dukungan terhadap SSL/TLS diperluas.

		Traffic Sharping ditambahkan. OpenBSD *port* yang pertama kali. Perluasan pengamanan yang berulang membuat OpenVPN menjadi lebih aman. Adanya penambahan dan modifikasi pada buku manual.
22 April 2002	1.1.1	Pilihan untuk adanya konfigurasi otomatis pada jaringan OpenVPN. Fitur kontrol pada keadaan tidak aktif.
22 Mei 2002	1.2.0	File konfigurasi pendukung di tambahkan. SSL/TLS sebagai *background process* – kunci enkripsi yang lebih panjang dimungkinkan. Berbagai macam port di tambahkan (Solaris, OpenBSD, Mac OSX, x64). Pengembangan situs web, termasuk penambahan ”how to” pada web. Instalasi tanpa *automake* dimungkinkan.
12 Juni 2002	1.2.1	File Binary RPM untuk instalasi pada sistem berbasis Redhat disediakan.

		Perbaikan yang intensif dalam penanganan sinyal dan pengaturan kunci enkripsi pada saat *restart*. Dukungan untuk perubahan dinamik pada paket yang datang (seperti IP yang dinamik). Penambahan *support* untuk pengenalan *downgrade* setelah instalasi OpenVPN dapat dijalankan sebagai *user* tanpa hak khusus.
10 Juli 2002	1.3.0	”*Housekeeping Releases”* : Perbaikan *bug*, sedikit pengembangan, dan fitur-fitur baru, sekarang dapat berjalan dengan OpenSSL 0.9.7 beta 2.
10 Juli 2002	1.3.1	*Port* NetBSD. Mendukung instalasi *inetxd/xinetd* pada Linux.
23Oktober 2002	1.3.2	Pengembangan sederhana pada sertifikat SSL/TLS ditambahkan. Mendukung Ipv6 melalui TUN ditambahkan.
7 Mei 2003	1.4.0	Perbaikan pada proteksi berulang (keamanan). Sejumlah perbaikan pada *bug*, pengembangan, dan penambahan.

15 Mei 2003	1.4.1	Pengembangan dengan dukungan pada kernel 2.4.
15 Juli 2003	1.4.2	Pertama kali mendukung windows *port* (tetapi masih belum mendukung *driver kernel* pada windows). *Gentoo Init Script.*
4 Agustus 2003	1.4.3	Mengeluarkan beberapa perbaikan pada *bug.*
20November2003	1.5.0 (dan 14 buah beta sebelumnya)	Penarikan kembali daftar sertifikat. Dukungan terhadap TCP. Port Windows 2000 dan XP, termasuk installer Win32. Meningkatkan pemeriksaan yang baik dalam konfigurasi parameter. Penambahan *Proxy Support.* Perluasan fungsi *routing*. Mengembangkan dukungan pada TLS, perluasan kunci dan kode.
9 Mei 2004	1.6.0	Dukungan *proxy* SOCKS. Berbagai macam pengembangan pada sifat jaringan

		Windows – DHCP. Berbagai macam perbaikan *bug.*

Pengembangan OpenVPN versi 2

Beberapa fitur baru yang di tambahkan pada OpenVPN versi 2 :

1. Dukungan *Multi-Client* : OpenVPN menawarkan suatu modus koneksi yang khusus, dimana *client TLS-authenticated* (yang tidak di *blacklist* dalam CRL) disediakan.
2. Pilihan Pull/Push : Penyusunan jaringan *client* dapat diatur oleh *server*. Setelah penyusunan *tunnel* sukses, server dapat mengatakan kepada *client* (baik Windows maupun Linux) untuk menggunakan susunan jaringan yang berbeda dengan segera.
3. Pengadaan manajemen antarmuka Telnet.
4. *Driver* dan *Software Windows* yang telah meningkat secara luas.

2.7.2 Jaringan dengan OpenVPN

Struktur modular dari OpenVPN tidak hanya bisa ditemukan dalam model keamanannya, tetapi terdapat juga di dalam kerangka jaringan. ***James Yonan*** memilih *driver Universal* TUN/TAP untuk lapisan *networking* dari OpenVPN.

TUN/TAP *driver* adalah sebuah proyek *open-source* yang terdapat di dalam semua distribusi-distribusi Linux/UNIX yang modern

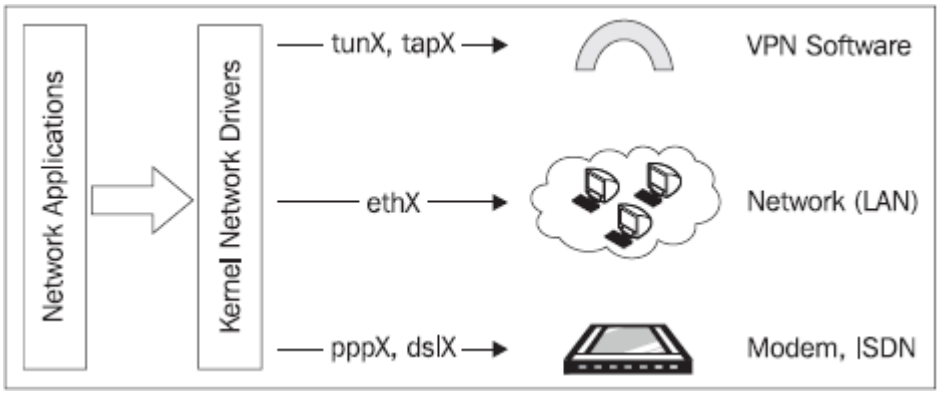
seperti juga Windows dan MacOS X. Seperti SSL/TLS, TUN/TAP juga dipakai dalam banyak proyek, oleh karena itu sehingga TUN/TAP dengan rutin ditingkatkan, dan banyak fitur baru ditambahkan.

Dengan menggunakan alat-alat TUN/TAP, banyak kompleksitas dari struktur OpenVPN dapat disingkirkan. Dengan strukturnya yang sederhana dapat lebih meningkatkan keamanan dibandingkan dengan solusi-solusi VPN yang lain. Kompleksitas merupakan musuh yang utama dari keamanan. Sebagai contoh, IPsec mempunyai suatu struktur kompleks dengan modifikasi-modifikasi yang kompleks di dalam *kernel* dan tumpukan protokol internet, hal ini menyebabkan adanya kemungkinan tercipta banyak lubang kecil di dinding keamanan.

Driver Universal TUN/TAP dikembangkan untuk menyediakan dukungan pada Linux *kernel* bagi proses *tunnelling*. *Driver* ini merupakan suatu antar muka jaringan maya, yang kelihatan asli kepada semua aplikasi dan para pemakai, hanya nama tunX atau tapX mencirikannya dari alat yang lain. Setiap aplikasi yang mampu menggunakan antar muka jaringan dapat menggunakan antar muka terowongan. Setiap teknologi yang sedang anda jalankan di dalam jaringan, dapat berjalan dalam antar muka TUN atau TAP juga.

Driver ini merupakan salah satu faktor utama yang membuat OpenVPN mudah untuk dimengerti, mudah untuk dikonfigurasi, dan aman juga pada waktu yang bersamaan.

Gambar berikut menggambarkan OpenVPN yang menggunakan antar muka standar :



Gambar 2.15 : *Standard Interface OpenVPN*

Sebuah TUN dapat digunakan sebagai suatu antar muka *point-to-point* yang maya, seperti *modem* atau DSL. Ini disebut *routed mode,* karena rute-rute disiapkan kepada mitra VPN.

Sebuah TAP, bagaimanapun, dapat digunakan sebagai suatu *Adapter Ethernet* yang maya. Hal ini memungkinkan *daemon* membaca antar muka untuk menangkap *Frame eternet*, yang tidak mungkin ditangkap dengan alat-alat TUN. Modus ini disebut modus penghubung karena jaringan tersebut dihubungkan seolah-olah di atas suatu *hardware.*

Aplikasi-aplikasi dapat dibaca atau ditulis pada antar muka ini, perangkat lunak (*tunnel driver*) akan mengambil semua data dan menggunakan *cryptographic libraries* dari SSL/TLS ke untuk mengenkripsikan mereka. Data tersebut dibungkus dan dikirim kepada ujung lain dari *tunnel*. Pengemasan ini terselesaikan atas standardisasi UDP atau paket TCP opsional. UDP merupakan pilihan pertama, tetapi

protokol TCP dapat sangat menolong dalam beberapa hal. Anda hampir dengan sepenuhnya bebas untuk memilih parameter-parameter konfigurasi seperti angka-angka protokol atau *port*, sepanjang kedua-duanya tujuan *tunnel* sepakat menggunakan hal yang sama.

OpenVPN mendengarkan alat TUN/TAP, mengatur *traffic*, melakukan enkripsi, dan mengirimkan kepada mitra VPN yang lain, di mana proses OpenVPN yang lain menerima data, melakukan deskripsi, dan menyampaikannya kepada alat jaringan maya, di mana aplikasi sedang menunggu data.

2.7.3 Konfigurasi OpenVPN

OpenVPN itu mempunyai suatu keamanan dan pendekatan keamanan sehingga mudah digunakan OpenVPN juga merupakan suatu model jaringan yang fleksibel. Sebagai konsekuensi, suatu sintaks konfigurasi yang sangat sederhana dan dokumentasi baik menandai antarmuka dari OpenVPN. Konfigurasi dilaksanakan dengan *editing* suatu *file* teks yang sederhana, sintaksnya sama pada setiap sistem operasi. Ini adalah salah satu contoh dari konfigurasi sederhana sepanjang 13 baris :

```
remote feilner-it.dynalias.net
float
dev tun
tun-mtu 1500
ifconfig 10.79.10.1 10.79.10.2
secret my_secret_key.txt
port 5050
route 10.94.0.0 255.255.0.0 10.79.10.2
```

```
comp-lzo
keepalive 120 600
resolv-retry 86400
route-up "/sbin/firewall restart"
log-append /var/log/openvpn/ultrino.log
```

Suatu baris perintah antar muka mengizinkan anda untuk memulai *tunnel* sesuai keinginan anda, yang sangat bermanfaat ketika anda sedang menguji susunan susunannya. Parameter-parameter yang sama seperti di file konfigurasi ditambahkan kepada baris perintah, dan *tunnelling* dimulai.

Di dalam *mode server*, OpenVPN dapat mendorong berbagai macam konfigurasi data kepada klien-klien melalui *tunnel*. *Tunnel* ganda dapat berjalan dalam *port*-nya tunggal, baik protokol UDP atau TCP. OpenVPN dapat di-*tunnel* melalui *firewall* dan *proxies*, jika mereka mengizinkan koneksi HTTPS, dan server dapat mengatakan kepada klien untuk menggunakan *tunnel* sebagai rute pada internet.