BAB 2

LANDASAN TEORI

2.1 Jaringan Komputer

Jaringan komputer adalah sebuah jaringan yang didalamnya terdapat media komunikasi, peralatan komuikasi, dan *software* yang dibutuhkan untuk menghubungkan dua atau lebih sistem komputer dan / atau peralatan komputer.

Jaringan komputer lebih banyak dipilih karena jaringan komputer ini memungkinkan untuk membagi *hardware, resources,* aplikasi komputer dan *database* yang ada. Jaringan komputer juga memungkinkan komputer tersebut untuk lebih fleksibel dan lebih cepat beradaptasi dengan kondisi yang dibutuhkan, memungkinkan para pekerja dan *workgroup* yang dipisahkan oleh faktor geografi untuk dapat membagi data, ide, pendapat, pemikiran baru, serta dapat berinteraksi secara lebih efektif dan efisien.

2.2 Tipe Jaringan Komputer

Tipe jaringan komputer dapat dikelompokkan berdasarkan luas wilayah cangkupannya, adapun beberapa tipe jaringan komputer bedasarkan luas wilayah cangkupannya :

a. *Local-Area Networks* (LAN)

LAN merupakan sebuah jaringan yang menghubungkan banyak komputer disebuah wilayah yang relatif kecil seperti rumah, kantor, atau kampus.

b. *Metropolitan-Area Networks* (MAN)

MAN adalah sebuah jaringan komputer besar yang mencangkup sebuah kota atau sebuah kampus besar. MAN biasanya merupakan gabungan dari LAN yang menggunakan teknologi *backbone* berkecepatan tinggi dan menyediakan layanan ke jaringan yang lebih besar seperti WAN dan *Internet*.

c. *Wide-Area Networks* (WAN)

WAN adalah sebuah jaringan yang disediakan oleh operator telekomunikasi yang mencangkup wilayah geografi yang besar.

d. Internet

Internet adalah sebuah sistem global yang menghubungkan jaringan komputer menggunakan protokol standar yang disebut *Internet Protocol Suite* (TCP/IP) untuk melayani milyaran pengguna di dunia.

2.3 Klasifikasi Jaringan Berdasarkan Hubungan Fungsional

Pada dasarnya setiap jaringan komputer ada yang berfungsi sebagai *client* dan ada juga yang berfungsi menjadi *server*. Tetapi ada jaringan tertentu yang memiliki komputer khusus yang bertugas sebagai *server* sedangkan komputer yang lain sebagai *client*. Maka berdasarkan fungsinya ada dua jenis jaringan komputer, yaitu :

a. Peer-to-Peer Networks

Dengan menggunakan teknologi LAN dan WAN, banyak komputer saling terhubung untuk menyediakan pelayanan kepada penggunanya. Dalam *peer-to-peer metwork*, jaringan komputer berperan sebagai partner, atau *peer*

antara yang satu dengan yang lain. Sebagai contoh, komputer A meminta sebuah file dari komputer B, yang merespon dengan mengirimkan file tersebut ke komputer A. Komputer A berfungsi sebagai *client* dan komputer B berfungsi sebagai *server*. Komputer A dan B mempunyai hubungan timbal-balik atau hubungan *peer* terhadap satu dan yang lainnya. Dalam *peer-to-peer network,* pengguna individual mengontrol sumber-sumber mereka sendiri. Ketika sebuah komputer berperan sebagai *server,* pengguna dari mesin itu akan mengalami performa yang menurun karena mesin itu melayani permintaan dari sistem lain. Seiring dengan perkembangan jaringan, hubungan *peer-to-peer* menjadi semakin sulit untuk dikoordinasi. Karena mereka tidak dapat beradaptasi dengan baik, keefisensian mereka menurun dengan cepat seiring dengan semakin banyaknya komputer yang terdapat dalam sebuah jaringan.

b. Client/Server Networks

Dalam *client/server*, layanan jaringan bertempat pada sebuah *dedicated* komputer yang disebut *server*, yang bereaksi kepada permintaan *client*. *Server* adalah pusat komputer yang terus-menerus tersedia untuk merespon kepada permintaan *client* untuk *file*, cetak, aplikasi, dan layanan lainnya. *Server* didesain untuk menangani permintaan dari banyak *client* secara bersamaan. Sebelum seorang *client* dapat mengakses sumber *server*, *client* tersebut harus mengidentifikasi dirinya sendiri dan mempunyai hak untuk menggunakan sumber. Ini dilakukan dengan cara mengeset setiap *client* dengan sebuah nama akun atau kata kunci yang diverifikasi dengan sebuah layanan autentikasi yang berperan sebagai penjaga akses ke jaringan.

Tabel 2.1 – Keuntungan *peer-to-peer* dan *client/server*

Keuntungan *Peer-to-Peer Networks*	**Keuntungan *Client/Server Networks***
Murah untuk diimplementasikan	Menyediakan keamanan yang lebih baik
Tidak memerlukan perangkat lunak NOS (*Network Operating System*)	Lebih mudah untuk dikelola ketika jaringan besar, karena administrasinya terpusat
Tidak memerlukan seorang administrator jaringan yang *dedicated*	Semua data dapat dibuat cadangannya dalam satu lokasi yang terpusat

Tabel 2.1 – Keuntungan *peer-to-peer* dan *client/server*

Kerugian *Peer-to-Peer Networks*	**Kerugian *Client/Server Networks***
Tidak beradaptasi dengan baik untuk jaringan yang besar; administrasi menjadi tidak terkendali	Memerlukan perangkat lunak NOS yang mahal seperti Windows NT, Windows 2000 server, atau Novell Netware
Setiap pengguna harus dilatih untuk melakukan pekerjaan administratif	Memerlukan perangkat keras yang mahal dan lebih kuat untuk mesin server
Kurang aman	Memerlukan administrator yang profesional
Semua mesin membagikan sumber yang buruk yang dapat berpengaruh pada kinerja	Mempunyai satu poin kegagalan jika hanya ada satu *server*; data pengguna tidak dapat tersedia jika server lagi tidak dapat bekerja

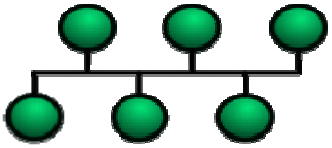
2.4 Topologi Jaringan

Topologi jaringan komputer dapat dikelompokkan lagi berdasarkan jenisnya, yaitu topologi fisik jaringan dan topologi logikal jaringan.

2.4.1 Topologi Fisik Jaringan

Topologi fisik jaringan merupakan gambaran sebenarnya sebuah jaringan secara fisik. Topologi fisik yang umumnya dikenal adalah *bus, star, ring, tree,* dan *mesh*.

a. Topologi Bus

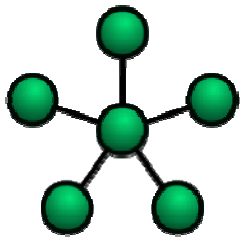


Gambar 2.1 – Topologi Bus

Biasanya disebut linear bus, semua alat dalam topologi bus dihubungkan dengan sebuah kabel *single*, yang mengalihkan dan menghubungkan dari sebuah komputer ke yang berikutnya seperti jalur bus yang bergerak ke sebuah kota. Kabel utama berakhir dengan sebuah terminator yang mengambil signal ketika mencapai akhir dari jalur atau kabel. Jika tidak ada terminator, signal elektrik mewakili data kembali ke akhir kabel, sehingga menyebabkan *error* dalam jaringan. Hanya boleh ada satu paket yang ditransmisikan pada saat bersamaan. Jika lebih dati satu paket ditransmisikan, mereka bertabrakan dan harus dikirim ulang. Sebuah topologi bus dengan banyak host dapat berjalan sangat lamabat dikarenakan oleh kolisi.

Topologi ini jarang digunakan dan cocok hanya untuk kantor rumahan atau bisnis kecil dengan hanya ada beberapa host.

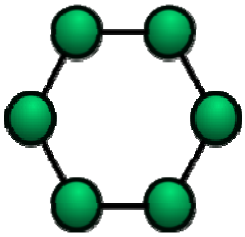
b. Topologi Star



Gambar 2.2 – Topologi Star

Bentuk topologi jaringan yang berupa konvergensi dari node tengah ke node pengguna disekitarnya.

c. Topologi Ring

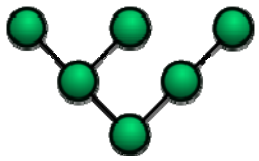


Gambar 2.3 – Topologi Ring

Topologi ring adalah topologi lainnya yang penting dalam konektifitas LAN. Host dihubungkan dalam bentuk cincin atau lingkaran. Tidak seperti topologi bus, topologi ini tidak mempunyai

awal atau akhir yang perlu diakhiri. Sebuah rangka yang disebut *token* menjelajahi cincin dan berhenti di setiap node. Jika sebuah node ingin mentransmisikan data, maka *token* akan menambahkan data dan mengalamatkan informasi ke rangka tersebut. Rangka itu terus bergerak sampai menemukan tujuan node, yang kemudian mengambil data keluar dari rangka.

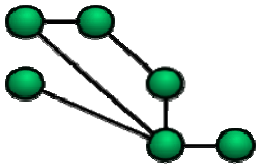
d. Topologi Tree



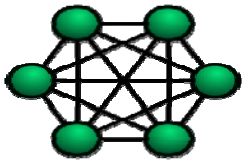
Gambar 2.4 – Topologi Tree

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

e. Topologi Mesh



Gambar 2.5 – Topologi Partial Mesh



Gambar 2.6 – Topologi Full Mesh

Topologi ini merupakan suatu bentuk hubungan antar node dimana setiap node terhubung secara langsung ke node lainnya yang ada di dalam jaringan. Akibatnya, dalam topologi mesh setiap perangkat dapat berkomunikasi langsung dengan perangkat yang dituju.

2.4.2 Topologi Logikal Jaringan

Merupakan gambaran secara abstrak bagaimana sebuah *host* dapat berkomunikasi melalui medium tertentu. Bentuk umum yang digunakan adalah *broadcast* dan *token parsing.*

a. Broadcast

Pada model ini semua komputer diharuskan menerima paket-paket data yang dikirimkan tiap-tiap komputer. Aturan yang digunakan adalah *First Come First Serve*.

b. Token Parsing

Pada model ini, jaringan komputer dikendalikan oleh sebuah *token*. Hanya komputer yang memiliki *token* yang dapat mengirimkan data ke jaringan. Kepemilikan *token* digilir secara bergantian.

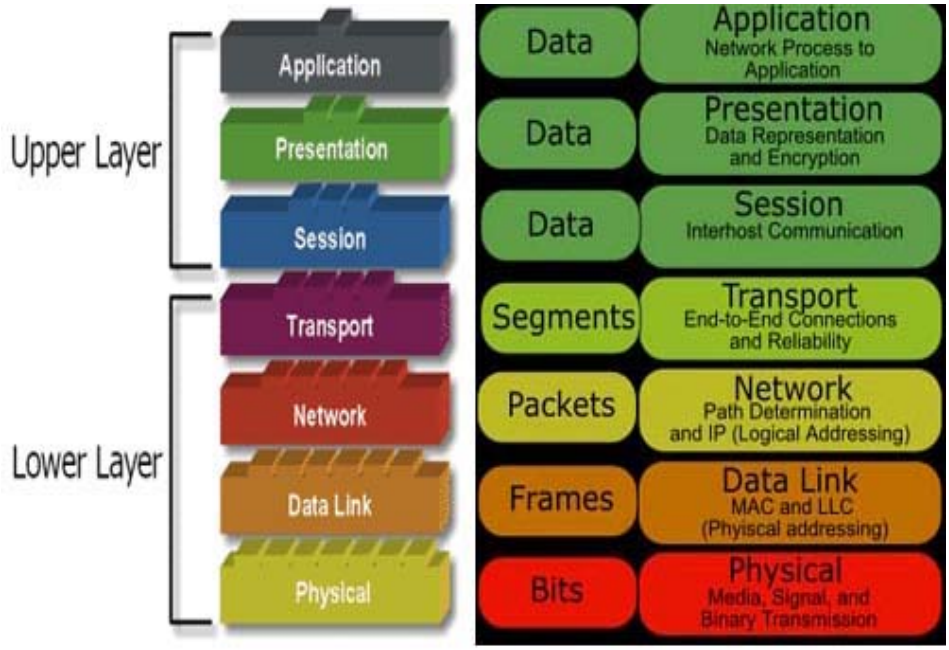
2.5 Protokol Jaringan

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

2.5.1 Open System Interconnection (OSI Layers)

Pada tahun 1984, organisasi internasional untuk standarisasi mengembangkan model *Open System Interconnection* (OSI) untuk mendukung interoperabilitas dalam jaringan. Model OSI ini memisahkan fungsi jaringan ke dalam tujuh lapis. Pada setiap lapis, fungsi yang disediakan bergantung pada pelayanan lapisan yang ada di bawahnya. Lapisan-lapisan itu dipisahkan dengan batasan yang jelas, atau

antarmuka. Setelah layer yang diberikan telah mengerjakan fungsinya, kemudian akan melewatkan informasi ke batasan di layer dibawahnya. Sebagai sebuah sarana, model OSI menawarkan kerangka, atau arsitektur, untuk mengembagkan protokol yang mengimplementasikan fungsi dari tiap lapisan.



Gambar 2.7 – OSI Layers dan fungsi per-Layernya

a. Layer 1 – *Physical Layer*

Physical Layer / lapisan fisik, berhubungan ke perangkat keras jaringan (kabel, media). Lapisan ini menentukan bagaimana data *binary* diterjemahkan ke dalam bentuk elektrik, optik, atau tipe lain dari signal fisik yang ditransmisikan di antara sistem.

b. Layer 2 – *Data Link Layer*

Data Link Layer / lapisan hubungan data, menerangkan bagaimana paket data diorganisasikan ke dalam kerangka pada tipe yang khusus dari jaringan dan peraturan untuk memasukkan kerangka ke dalam media jaringan. Lapisan hubungan data adalah dimana perangkat keras seperti switch dan bridge beroperasi. Switch dan bridge tidak melakukan fungsi router. Melainkan, mereka mengirim data jaringan bergantung pada informasi seperti alamat MAC.

c. Layer 3 – *Network Layer*

Network Layer / lapisan jaringan, menjelaskan bagaimana alamat ditetapkan dan bagaimana paket data diteruskan dari satu jaringan ke yang lainnya akhirnya sampai ke tujuan. Router beroperasi pada tingkat ini karena mereka mengarahkan informasi ke lokasi yang tepat sesuai dengan tipe dari protokol routing yang sedang digunakan.

d. Layer 4 – *Transport Layer*

Transport Layer / lapisan transport, menyediakan layanan pengiriman untuk setiap sesi. Layer ini melakukan segmentasi data ke dalam bagian yang lebih bisa diatur. Juga menyediakan sebuah layanan pengiriman yang terpecaya yang menjamin data sampai ke tujuannya, atau juga dapat menyediakan layanan pengiriman yang tidak terpecaya yang membawa data tanpa pengecekan error.

e. Layer 5 – *Session Layer*

Session Layer / lapisan sesi, menetapkan peraturan dari pembicaraan antara dua aplikasi. Lapisan sesi juga mengijinkan beberapa salinan dari sebuah aplikasi untuk berkomunikasi dengan aplikasi lain pada saat yang sama dengan mengidentifikasi setiap instansi dari aplikasi sebagai sebuah sesi yang terpisah.

f. Layer 6 – *Presentation Layer*

Presentation Layer / lapisan presentasi, menerangkan susunan, atau sintaks, dari data yang diinginkan aplikasi. Karena aplikasi pada sistem yang berbeda dapat diwakili atau format data mereka dalam cara yang berbeda, lapisan presentasi memasukkan translasi dari satu format ke yang lainnya. Lapisan presentasi juga memasukkan fungsi keamanan dan efisiensi (enkripsi dan kompresi). Hal ini menjelaskan aplikasi apa yang digunakan secara bebas dari sistem operasi yang digunakan untuk memperlihatkan data. .gif, .jpeg, .mpeg, dan .avi dari beberapa contoh dari apa yang lapisan presentasi kenali untuk membuat data tersedia untuk pengguna pada lapisan aplikasi.

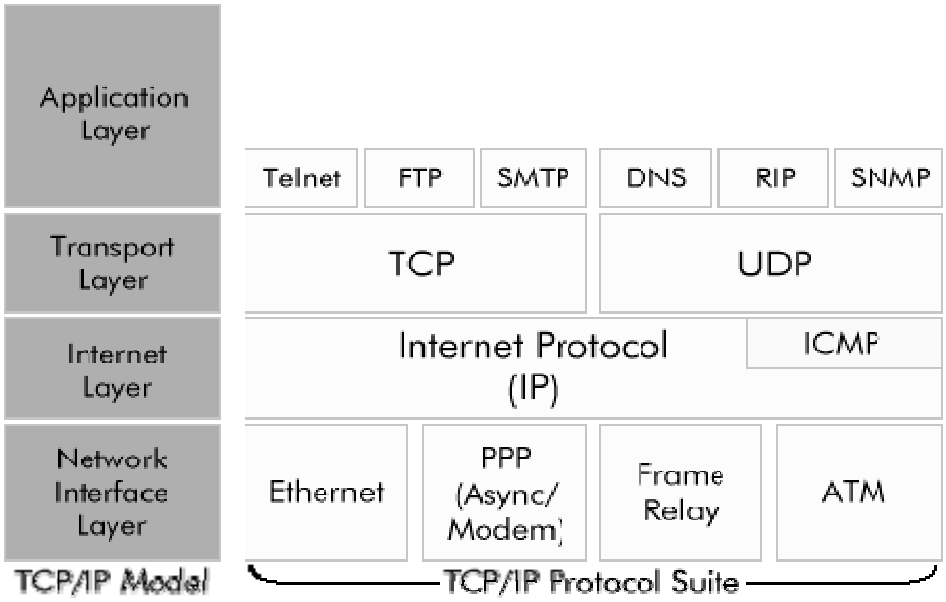
g. Layer 7 - *Application Layer*

Application Layer / lapisan aplikasi, menyediakan titik koneksi untuk aplikasi. Menerangkan detail dari bagaimana sebuah aplikasi membuat permintaan dan bagaimana aplikasi pada mesin lain merespon. Sebuah permintaan untuk layanan

jaringan berasal dari lapisan ini. FTP, Telnet, ping, dan e-mail protokol khusus pada lapisan aplikasi ini.

2.5.2 *Internet Protocol Suite* (TCP/IP)

TCP/IP adalah serangkaian atau koleksi dari protokol yang berbeda, setiap protokol melakukan sebuah tugas yang spesial. Seperti spesialis pada tim konstruksi, setiap protokol melakukan fungsi tertentu pada waktu yang tertentu juga.



Gambar 2.8 – TCP/IP Layers

a. 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 ke dalam suatu layer dan menjamin

data dipaketkan dengan benar sebelum masuk ke layer berikutnya. Beberapa program yang berjalan pada layer ini menyediakan layanan langsung kepada user. Program-program ini dan protokol yang berhubungan dengan layer ini meliputi HTTP (*The World Wide Web Protocol*), FTP, TFTP (*File Transport*), SMTP (*Email*), Telnet, SSH (*Secure remote login*), DNS (*Domain name management*).

b. Transport Layer

Transport layer menyediakan layanan transportasi dari *host* sumber ke *host* tujuan. *Transport layer* merupakan suatu koneksi logikal diantara *endpoints* dari suatu jaringan, yaitu *sending host* dan *receiving host. Transport protocol* membuat segmen dan mengumpulkan kembali layer aplikasi 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 mempunyai tugas untuk menjamin bahwa pesan diterima ke tujuan atau, jika mereka tidak dapat dikirim, maka mereka akan menginformasikan program aplikasi bahwa mereka gagal. Sesegera setelah koneksi TCP dibuat antara dua aplikasi, semua pesan mengalir dari asal ke tujuan melalui koneksi logikal. Aplikasi e-mail dan web browser menggunakan protokol ini.

- ***User Datagram protocol* (UDP)**

  UDP menyediakan layanan yang tidak terpecaya untuk aplikasi yang dapat mentoleransi kehilangan beberapa pesan tetapi masih dapat berfungsi. Aplikasi yang mengirim stream atau video atau data audio jatuh ke dalam kategori ini. Mereka dapat menemukan kehilangan beberapa data tetapi masih bisa berfungsi dengan cara yang masih bisa diterima oleh pengguna jaringan. Aplikasi lain yang menggunakan UDP termasuk DNS dan beberapa bentuk transfer file lainnya, termasuk TFTP. Setiap pesan UDP dikirim secara bebas dari yang lainnya tanpa memerlukan koneksi logikal telebih dahulu antara asal dan tujuan. Karakteristik dari protokol UDP adalah :

  1. Tidak terpecaya
  2. Cepat
  3. Diasumsikan aplikasi akan mengirim ulang jika error
  4. Sering digunakan *workstation* yang tidak memiliki disk

c. 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, DCHP, dan ICMP.

- ***Internet Protcol* (IP)**

  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 paket (*fragmentation*) dan penyatuan paket-paket, fungsi meneruskan paket (*routing*).

- ***Dynamic Host Configuration Protocol* (DHCP)**

  Tujuan dari DHCP adalah untuk mengijinkan komputer individual pada sebuah IP network untuk mengekstrak konfigurasi mereka dari sebuah server DHCP. Ketika sebuah komputer memperlihatkan sebuah alamat IP, dia mengirim sebuah permintaan ke server DHCP. Server DHCP dapat menyediakan komputer *host* dengan semua informasi konfigurasi yang diperlukannya, termasuk alamat IP, *subnet mask*, *gateway*, DNS dan WINS *server*, dan *domain*. DHCP juga mengijinkan untuk pemulihan dan kemampuan untuk secara otomatis memperbaharui alamat IP jaringan memalui sebuah *leasing mechanism*, dimana dialokasikan sebuah alamat IP untuk waktu tertentu dan kemudian melepasnya dan menentukan sebuah alamat IP baru. DHCP merupakan sebuah metode pengurangan kerja yang penting untuk mengelola jaringan IP yang besar.

- ***Internet Control Message Protocol* (ICMP)**

  ICMP menyediakan sebuah set dari error dan kontrol pesan untuk membantu pelacakan dan penyelesaian masalah jaringan. Sebuah

contoh, jalur fisik pada jaringan gagal, dan beberapa host menjadi tidak dapat dicapai. ICMP biasanya mengirim pesan "*Destination Unreachable*" ketika ada sebuah *error* di suatu tempat pada jaringan yang mencegah kerangka atau paket diteruskan ke sistem tujuan atau alat. ICMP memasukkan sebuah pesan yang disebut *echo request* yang dapat dikirim dari satu *host* ke yang lain untuk melihat apakah dia dapat dicapai dalam jaringan. Jika dapat dicapai maka host tujuan membalas dengan pesan ICMP *echo*. Program *ping* menggunakan ICMP untuk mengirim pesan *echo request* dan menerima balasan dari pesan *echo*.

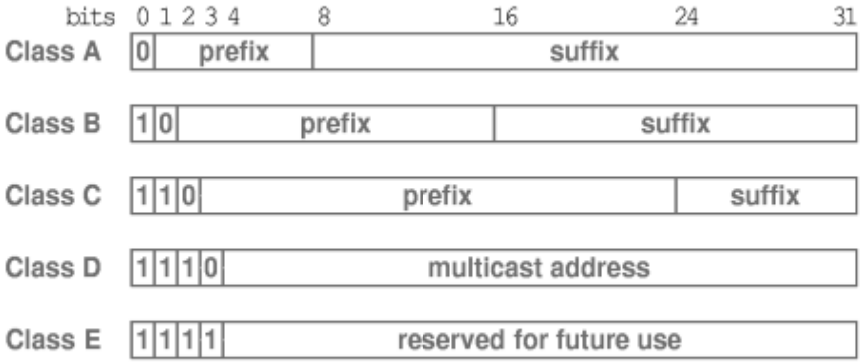
d. Network Interface Layer

Network Interface Layer atau yang biasa disebut *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 / Network Interface Layer* mengawasi pengalamatan secara *hardware* dan mendefinisikan protokol transmisi fisik data.

2.6 Pengalamatan IPv4

Sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IPv4. Panjang totalnya adalah 32-bit, dan secara teoritis dapat mengalamati hingga 4 milyar *host* komputer atau lebih tepatnya 4.294.967.296 host di seluruh dunia, jumlah *host* tersebut didapatkan dari 256 (didapatkan dari 8 bit) dipangkat 4 (karena terdapat 4 oktet) sehingga nilai maksimal dari alamt IP versi 4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga nilai nilai host yang dapat ditampung adalah 256x256x256x256=4.294.967.296 host.

Pada pengalamatan IPv4 terdapat 5 (lima) buah *class* yang masing-masing memiliki jumlah *host* yang berbeda. Adapun pembagian 5 (lima) *class* itu adalah sebagai berikut :



Gambar 2.9 – Kelas pengalamatan IP

a. Kelas A

Dalam kelas ini, angka pertama (oktet) dalam porsi jaringan, dan tiga angka terakhir dari porsi host. Formatnya adalah *jaringan.host.host.host*, atau N.H.H.H. untuk contoh, alamatnya 56.1.2.3, oktet pertama (56) mengidentifikasi jaringan, dan tiga oktet terakhir (1.2.3) mengidentifikasi host pada jaringan. Alamat 56.1.2.4 mengidentifikasi sebuah host yang berbeda (1.2.4) pada jaringan yang sama. Alamat 57.1.2.3 mengidentifikasi host 1.2.3 pada jaringan 57.

Karena dengan cara ini alamat disimpan dalam *binary* di komputer, oktet pertama selalu jatuh di antara 1 dan 127. Jika oktet pertama dalam sebuah alamat IP jatuh pada kisaran 1 sampai 127, itu adalah alamat kelas A. Hanya 1 sampai 126 yang valid untuk jaringan kelas A, karena 127.0.0.1 sudah dipesan. Alamat ini, dikenal sebagai alamat *local loopback*, yang digunakan untuk mengetes sistem lokal NIC.

b. Kelas B

Alamat kelas B membagi porsi jaringan dan porsi host antara oktet kedua dan ketiga. Formatnya adalah N.N.H.H. sebagai contoh, alamat 165.5.6.7 mewakili jaringan 165.5 dan host 6.7.

Karena cara ini alamat kelas B disimpan di *binary*, oktet pertama selalu jatuh pada kisaran 128 sampai 191. Jika oktet pertama alamat IP lebih besar dari 127 tetapi kurang dari 192, ini adalah IP kelas B.

c. Kelas C

Alamat kelas C memisahkan porsi jaringan dan porsi host antara oktet ketiga dan keempat. Formatnya adalah N.N.N.H. Sebagai contoh, alamat 192.8.9.10 mewakili jaringan 192.8.9 dan host 10.

Karena cara ini alamat kelas B disimpan di *binary*, oktet pertama selalu jatuh pada kisaran 192 sampai 223. Jika oktet pertama alamat IP lebih besar dari 191 tetapi lebih kecil dari 224, ini adalah alamat kelas C.

d. Kelas D dan E

Alamat kelas D dan E digunakan untuk tujuan tertentu. Kelas D dipesan untuk sebuah teknik yang dinamakan *multicast*, dan alamat kelas E digunakan untuk tujuan eksperimen. Organisasi komersial menggunakan alamat kelas A, B atau C untuk mengidentifikasi jaringan dan host.

2.7 *Virtual Private Network* (VPN)

VPN adalah sebuah emulasi dari *Wide Area Network* (WAN) pribadi yang menggunakan fasilitas berbagi IP atau IP publik, seperti *internet* atau *private* IP *backbones*. Dalam arti yang lebih sederhana, VPN adalah sebuah ekstensi dari *private intranet* melalui jaringan publik (*internet*) yang memastikan keamanan dan konektivitas dengan biaya yang efektif antara dua komunikasi akhir. *Private intranet* diperluas dengan bantuan dari *private logical* "*tunnels*". Tunnel ini memungkinkan dua pengguna akhir untuk menukar data dengan catatan komunikasi *point-to-point*.

Pada VPN terdapat 3 (tiga) mekanisme penting, yaitu enkripsi, autentikasi dan autorisasi.

Enkripsi adalah proses mengubah data ke dalam bentuk yang hanya bisa dibaca oleh penerima yang diinginkan. Untuk membaca pesan yang telah dienkripsi tersebut, penerima dara harus mempunyai kunci dekripsi yang benar. *Public-key encryption* menggunakan dua kunci. Satu kunci dikenal sebagai *public key*, yang oleh setiap orang boleh gunakan selama enkripsi dan dekripsi. Walaupun nama kuncinya adalah *public key*, kunci ini dipunyai oleh sebuah entiti. Jika entiti kedua perlu untuk berkomunikasi dengan pemilik kunci, entiti kedua menggunakan *public key* untuk melakukan komunikasi itu. *Public key* mempunyai *corresponding private key. Private key* adalah key yang bersifat pribadi kepada entiti. Sebagai hasilnya, dengan enkripsi *public key* setiap orang dapat menggunakan pemilik *public key* unyuk mengenkripsi dan mengirim pesan. Tetapi, hanya pemilik yang mempunyai *private key* untuk mendekripsi pesan. Dalam berkomunikasi, pengirim menggunakan *public key*-nya untuk mengenkripsi pesan. Penerima menerima pesan dan mendekripsi pesan yang telah didecode menggunakan *private key*. *Pretty Good Privacy* (PGP) dan *Data Encryption Standard* (DES) adalah dua dari *public key* enkripsi yang paling populer.

Autentikasi adalah proses untuk memastikan data dikirim kepada penerima yang diinginkan. Sebagai tambahan, autentikasi juga memastikan integritas penerima dari pesan dan sumbernya. Dalam bentuk yang paling sederhana, autentikasi memerlukan paling sedikit *username* dan *password* untuk menerima akses ke sumber spesifik. Dalam bentuk yang kompleks, autentikasi dapat didasari dari *secret-key encryption* atau *public-key encryption.*

Autorisasi adalah proses memberikan atau menolak akses ke sumber yang berlokasi dalam jaringan setelah pengguna telah berhasil diidentifikasi dan diautentiksi.

Pada VPN juga terdapat protokol yang disebut dengan *VPN Tunneling Protocols*, protokol-protokol ini berguna untuk memastikan aspek keamanan dari transaksi melalui VPN. Protokol yang biasa digunakan, yaitu *IP Security* (IPSec), *Point-to-Point Tunneling Protoco*l (PPTP), *Layer 2 Tunneling Protocol* (L2TP), dan protokol-protokol lainnya seperti SSL/TLS.

IP Security (IPSec). Dikembangkan oleh IETF, IPSec adalah standar terbuka yang memastikan keamanan trasmisi dan autentikasi pengguna melalui jaringan publik. Tidak seperti teknik enkripsi lainnya, IPSec beroperasi pada *Network Layer* dari model tujuh layer OSI. Oleh karena itu, dapat diimplementasikan secara bebas ke aplikasi yang berjalan melalui jaringan. Sebagai hasilnya jaringan dapat diamankan tanpa perlu mengimplementasikan dan mengkoordinasi keamanan untuk setiap aplikasi.

Point-to-Point Tunneling Protocol (PPTP). Dikembangkan oleh Microsoft, 3COM, dan Ascenf Communicarions, PPTP dimaksudkan sebagai alternatif untuk IPSsec. Tetapi, IPSec masih menjadi favorit tunneling protokol. PPTP beroperasi pada layer kedua (*Data Link Layer*) dari model OSI dan digunakan untuk mengamankan transmisi dari trafik Windows.

Layer 2 Tunneling Protocol (L2TP). Dikembangkan oleh Cisco System, L2TP juga dimaksudkan untuk mengganti IPSec sebagai tunneling protocol. Tetapi IPSec masih terus-menerus menjadi protokol yang dominan untuk komunikasi yang aman melalui *internet.* L2TP adalah kombinasi dari *layer 2*

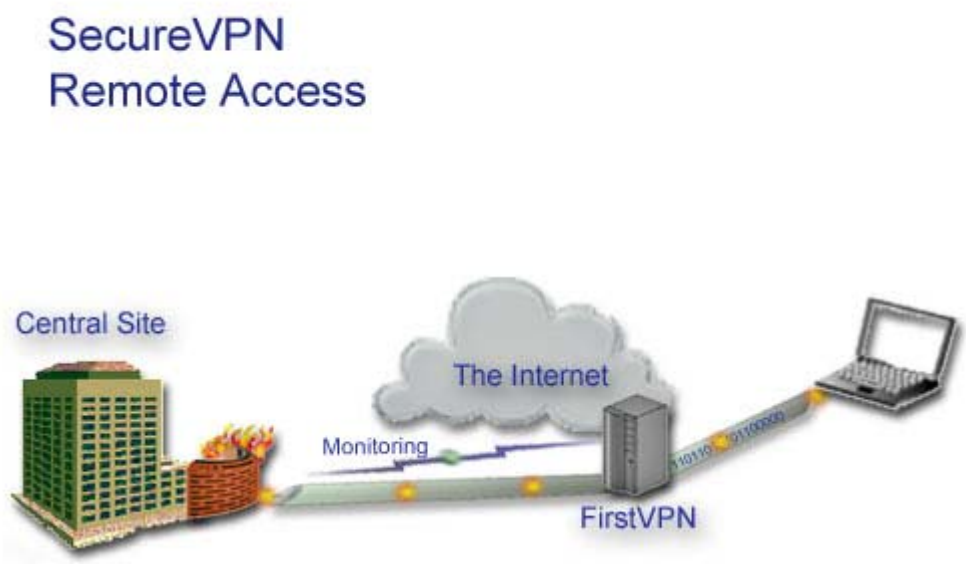
forwarding (L2F) dan PPTP dan digunakan untuk mengenkapsulasi *frame Point-to-Point Protocol* (PPP) yang dikirim melalui X.25, FR, dan jaringan ATM.

Faktor lain yang membedakan antara sistem dan protokol yang dijelaskan di atas adalah :

- Ketersediaan dari mekanisme autentikasi
- Mendukung untuk fitur *advanced networking* seperti *Network Address Translation* (NAT)
- Alokasi dinamis dari IP address untuk partner tunnel dalam mode dial-up
- Mendukung untuk *Public Key Infrastructures* (PKI)

VPN sendiri memiliki beberapa tipe, tipe-tipe VPN yang biasa dikenal adalah *Remote-Access VPN* dan *Site-to-Site VPN*. Kemudian *Site-to-Site VPN* ini dapat dibagi lagi menjadi *Intranet VPN* dan *Extranet VPN*.

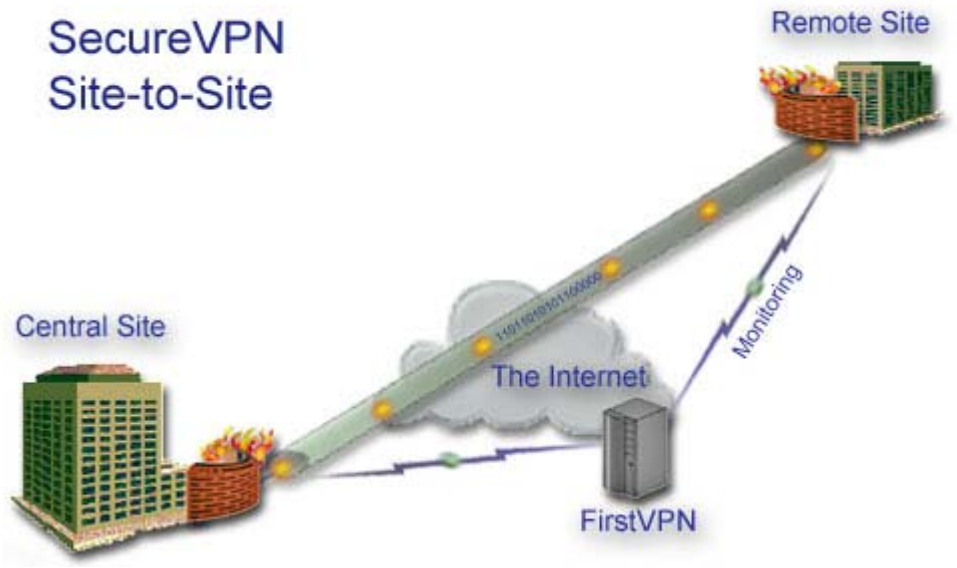
a. *Remote Access* VPN



Gambar 2.10 – Remote Access VPN

Seperti namanya, *Remote Access* VPN menyediakan akses dengan *remot*e, *mobile*, dan komunikasi karyawan dari sebuah organisasi ke jaringan sumber korporasi. Secara khusus, permintaan *remote* akses dibuat oleh pengguna yang selalu berkembang yang ingin mengakses jaringan LAN perusahaan. Dengan mengimplementasikan *Remote Access* VPN, pengguna remote dan cabang kantor hanya perlu melakukan setting koneksi lokal *dial-up* ke ISP atau ISP POP dan mengkoneksikan ke jaringan perusahaan melalui *internet*.

b. *Site-to-Site* VPN



Gambar 2.11 – Site-to-Site VPN

Site-to-Site VPN ini dibagi menjadi dua tipe, yaitu :

- **Intranet VPN**

  Intranet VPN digunakan untuk mengkoneksikan cabang kantor dari sebuah organisasi ke *intranet* perusahaan. Dalam setup *intranet*, tanpa menggunakan teknologi VPN setiap remote site harus dihubungkan ke *intranet* perusahaan (*backbone router*) menggunakan *campus routers*.

- **Ekstranet VPN**

  Tidak seperti *intranet* dan *remote access* VPN, extranet VPN tidak sepenuhnya dipisahkan dari "dunia luar". Kenyataanya, ekstranet VPN mengijinkan akses kontrol untuk sumber jaringan yang penting ke entiti bisnis eksternal, seperti partner, pelanggan, dan pemasok yang memainkan peran yang besar dalam bisnis organisasi.

2.7.1 Enkripsi Data VPN

Enkripsi data atau *cryptography* adalah salah satu komponen yang paling penting dari keamanan VPN dan memainkan peran yang besar dalam mengamankan data selama transit. Enkripsi ini adalah mekanisme dari mengkonversi dara ke dalam format yang tidak bisa dibaca, dikenal sebagai *ciphertext*, jadi akses ke data yang tidak sah dapat dicegah ketika data ditransmisikan melalui medium transmisi yang tidak aman.

Enkripsi data dapat mencegah :

a. Penangkapan data dan melihat data

b. Modifikasi data dan pencurian
c. Pemalsuan data
d. Penolakan data
e. Interupsi dari layanan jaringan

Ketika menerima pesan, penerima mendekripsi data kembali ke format semula. Bahkan jika *ciphertext* ditangkap selama transmisi, untuk membuatnya masuk akal bagian penangkapan itu harus mengetahui metode yang digunakan untuk mengkonversi data mentah ke dalam format yang acak. Jika tidak, data itu tidak berguna.

Pada pengirim dan penerima, bersamaan dengan proses enkripsi, membentuk *cryptosystem.* Enkripsi data atau *cryptography* ini terbagi menjadi dua jenis, yaitu *symmetric cryptosystems* dan *asymmetric cryptosystems*.

a. Symmetric Cryptosystems

Berdasarkan pada sebuah *single key*, yang merupakan sebuah bit string dari panjang yang sudah pasti. Untuk itu, mekanisme enkripsi ini juga dikenal sebagai *single-key encryption*. Key nya bersifat rahasia dan digunakan untuk enkripsi dan dekripsi. Sebelum dua bagian dapat menukar data maka key harus dibagi terlebih dahulu di antara mereka. Pengirim kemudian mengenkripsi pesan asli menggunakan *private key* ini dan mengirim pesan ke penerima. Pada

saat menerima pesan yang dienkripsi, penerima menggunakan key yang sama untuk mendekripsinya.

Berdasarkan panjang keynya, banyak algoritma symmetric encryption yang telah dikembangkan selama bertahun-tahun. Beberapa dari algoritma symmetric yang paling banyak digunakan dalam VPN adalah :

- ***Data Encryption Standard* (DES)**. Sebelumnya DES memiliki panjang key yang mencapai 128 bit. Tetapi, ukuran key dikurangi menjadi 56 bit oleh pemerintah Amerika Serikat untuk membuat algoritmanya menjadi cepat. Pengurangan ini, dapat mempercepat proses sistem komputer, tetapi mejadikan DES menjadi algoritma yang lemah dengan *Brute Force Attack*. Dalam penyerangan ini, key dihasilkan secara acak dan diterapkan ke text asli sampai key yang benar ditentukan. Semakin kecil ukuran key, semakin mudah untuk menghasilkan key yang benar.
- ***Triple Data Encryption Standard* (3DES)**. Seperti pendahulunya, DES-3DES juga menggunakan key sepanjang 56 bit. Tetapi, 3DES ini lebih aman karena tiga key yang berbeda digunakan untuk mengenkripsi data. Prosesnya sebagai berikut : key yang pertama mengenkripsi data. Kemudian, key yang kedua mendekripsi data baru yang dienkripsi. Terakhir, key ketiga mengenkripsi data untuk yang kedua kalinya. Seluruh proses ini membuat algoritma 3DES yang mempunyai keamanan tinggi.

Tetapi, karena kompleksitas algoritma ini, 3DES lebih lambat tiga kali dari DES.

- ***Ron's Code 4* (RC4)**. Dikembangkan oleh Ron Rivest, algoritma ini menggunakan key yang panjangnya mencapai 256 byte. Karena panjang key ini, RC4 dikategorikan sebagai mekanisme enkripsi yang kuat. Juga relatif cepat. RC4 membuat sebuah arus dari byte acak dan XOR mereka dengan text asli. Karena byte dihasilkan secara acak, RC4 memerlukan kunci baru untuk setiap pesan keluar.

Symmetric cryptosystem ini mempunyai dua masalah utama. Pertama, karena hanya satu key yang digunakan untuk mengenkripsi dan dekripsi, jika diketahui oleh pihak pengganggu, semua komunikasi yang menggunakan key ini terancam. Karena itu, key harus diubah secara berkala.

Masalah lainnya, jika jumlah komunikasinya besar, pengaturan key menjadi sulit. Sebagai tambahan, *overhead* yang berhubungan dengan *initial key-pair setup*, distribusi, dan penggantian key secara berkala mahal dan memakan waktu.

b. Asymmetric Cryptosystems

Menggunakan sepasang key matematik yang berhubungan. Salah satu key adalah rahasia dan hanya diketahui oleh pemilik pasangan key. Key yang kedua adalah publik dan didistribusikan

secara bebas. *Publik key* digunakan untuk tujuan enkripsi dimana *private key* digunakan untuk mendekripsi pesan.

Dalam solusi VPN, dua *asymmetric cryptosystems* sering digunakan. Termasuk algoritma *Diffie Hellman* (DH) dan *Rivest Shamir Adleman* (RSA).

- ***Diffie Hellman Algorithm***

  Setiap entiti komunikasi menerima sepasang kunci, salah satu kunci didistribusikan ke entiti komunikasi yang lain sementara yang lain disimpan secara rahasia. Proses algoritmanya sebagai berikut :

  - Pengirim menerima *publik key* si penerima, yang tersedia untuk semua pasangan komunikasi.
  - Pengirim kemudian mengerjakan sebuah kalkulasi yang memasukkan *private key* dan *publik key* penerima. Hasil kalkulasi di dalam kunci rahasia yang dibagikan.
  - Pesan dienkripsi menggunakan hasil key rahasia yang dibagikan
  - Pesan yang dienkripsi kemudian diteruskan ke penerima
  - Dalam penerimaan pesan yang dienkripsi, penerima menghasilkan key rahasia yang dibagikan dengan menggunakan sebuah kalkulasi yang serupa dan memasukkan *private key* miliknya dan *public key* si pengirim.

Ada satu masalah dalam algortima ini, sebagai contohnya, jika dua komunikasi pengguna akhir menukar *public key* melalui medium yang tidak aman, seperti internet, ini mungkin pengganggu sebelumnya dapat menangkap permintaan untuk *public key* dan mengirim *public key* miliknya ke dua komunikasi pengguna akhir. Dalam kasus ini, pengganggu dapat dengan mudah masuk ke dalam komunikasi karena kedua pengguna akhir itu sekarang menukar data menggunakan publik key pengganggunya. Kasus ini dikenal dengan penyerangan *Man-in-the-Middle.*

- ***The Rivest Shamir Adleman* (RSA) Algorithm**

  Karena mekasime enkripsinya yang kuat, RSA menjadi standard *asymmetric cryptosystems*. Tidak seperti *Diffie-Hellman*, pesan yang asli dienkripsi menggunakan *public key* penerima. Si penerima menerima pesan yang asli menggunakan *public key* pengirim.

  Algoritma RSA diimplementasikan untuk autentikasi menggunakan digital signature sebagai berikut :

  - *Public key* pengirim diminta oleh penerima yang bersangkutan dan kemudian diteruskan.
  - Pengirim menggunakan campuran fungsi untuk menurangi ukuran dari pesan asli. Hasil pesannya dikenal dengan *message digest* (MD).

- Pengirim mengenkripsi *message digest* dengan *private key* miliknya dengan generasi dari sebuah *digital signature* yang unik.
- Pesan dan *digital signature* dikombinasikan dan diteruskan ke penerima.
- Setelah penerimaan menerima pesan yang dienkripsi, penerima menghasilkan *message digest* menggunakan fungsi campuran yang sama dengan pengirim.
- Penerima kemudian mendekripsikan *digital signature* menggunakan *public key* pengirim.
- Penerima kemudian membandingkan *message digest* yang regenerasi (langkah kelima) dan *message digest* diterima dari digital signatute (langkah keenam). Jika keduanya cocok, data tidak akan dapat ditangkap, dipalsukan atau dimodifikasi selama transmisi. Jika sebaliknya, pesan ditolak.

2.7.2 *User Authentication*

Daya yang disimpan dalam beragam sumber jaringan sangat gampang terkena serangan. Penyerangan ini biasanya berupa :

a. Interupsi layanan jaringan. Kadang-kadang, karena penyerangan yang jahat baik dari luar maupun dari dalam jaringan, beragam sumber jaringan dan layanan tidak dapat digunakan lagi untuk waktu yang lama. Dalam kasus penyerangan ini, seluruh jaringan menjadi tidak dapat diakses oleh penggunanya

b. Penangkapan data. Selama dalam transit, data mungkin ditangkap oleh pihak yang tidak bertanggung jawab. Sebagai hasilnya, kerahasiaan data hilang. Untuk kasus ini, sebuah organisasi kehilangan banyak (dalam hal bisnis dan uang) jika data sensitif jatuh ke tangan yang salah
c. Modifikasi data. Penangkapan data juga dapat memicu pemodifikasian data. Dalam kasus ini, pihak tujuan menerima data yang rusak atau data pulsa. Ini dapat menyebabkan organisasi kehilangan uang, terutama jika data itu sangat penting
d. Pembuatan data. Dalam penyerangan ini, pihak yang tidak berwenang dapat menjadi pihak yang berwenang dan pengguna jaringan yang terpecaya. Setelah mempunyai akses ke jaringan, orang ini kemudian dapat menyebarkan kepalsuan dan pesan yang berbahaya ke pengguna jaringan lainnya. Ini dapat memicu penutupan sebagian atau seluruh jaringan, mengganggu layanan.

Mekanisme autentikasi pengguna diimplementasikan pada poin akses VPN dan digunakan untuk menagautentikasi pengguna ketika mengakses sebuah sumber dari jaringan. Sebagai hasilnya, hanya pihak yang berwenang yang dapat mengakses sumber jaringan, jadi dapat mengurangi kemungkinan pihak yang tidak berwenang untuk mengakses daya yang disimpan di jaringan.

Skema autentikasi yang diimplementasikan secara terpisah atau dikombinasikan dengan skema lain adalah :

a. *Login ID* dan *Password*. Skema ini menggunakan sistem operasi *login ID* dan *password* untuk verifikasi identitas pengguna dalam mengakses VPN

b. *S/Key Password.* Dalam skema ini, pengguna menginisialisasi S/Key dengan memilih sebuah *password* rahasia dan sebuah *integer*, n. *Integer* ini menunjukkan jumlah waktu sebuah fungsi campuran yang aman (MD4) yang akan diaplikasikan ke *password* rahasia. Hasilnya kemudian disimpan dalam *server* yang bersangkutan. Ketika pengguna ingin *log in*, *server* mengeluarkan tantangan. Perangkat lunak dalam mesin klien meminta password rahasia, diberlakukan iterasi n-1 dari fungsi campuran itu, dan mengirim respons ke *server*. *Server* memberlakukan fungsi campuran ke respons itu. Jika hasil yang diterimanya sama dengan nilai yang disimpan sebelumnya, pengguna berhasil diautentikasi. Pengguna diijinkan masuk, dan *server* mengganti nilai yang disimpan dengan respon yang diterima dari klien dan pengurangi *password counter*.

c. *Remote Access Dial-In user Service* (RADIUS). RADIUS adalah protokol keamanan internet yang didasarkan pada model *client/server*, dimana mesin yang mengakses jaringan adalah *client* dan RADIUS *server* pada jaringan mengautentikasi *client.* Pada umumnya, sebuah RADIUS *Server* mengautentikasi seorang pengguna menggunakan daftar internal *username/password* yang diaturnya. RADIUS juga dapat berperan sebagai client untuk mengautentikasi pengguna sistem operasi, seperti UNIX, NT dan

NetWare. Sebagai tambahan, RADIUS *Server* dapat berperan sebagai *client* untuk RADIUS *Server* lainnya, untuk lebih mengamankan data selama transit, transaksi antara *client* dan RADIUS *Server* dapat dienkripsi menggunakan mekanisme autentikasi, seperti *Password Authentication Protocol* (PAP) dan *Challenge Handshake Authentication Protocol* (CHAP).

d. *Two-Factor Token-Based Technique*. Seperti namanya, skema ini mengimplementasi dual autentikasi untuk verifikasi mandat pengguna. Ini mengkombinasikan penggunaan *token* dan *password*. Selama proses autentikasi, *server* elektronik perangkat keras sebagai token dan identifikasi unik, seperti *Personal Identification Number* (PIN) digunakan sebagai *password. Token* telah menjadi alat perangkat keras (seperti card), tetapi sekarang beberapa vendor menawarkan *token* yang berbasis perangkat lunak.

2.8 OpenVPN

OpenVPN adalah aplikasi *open-source* untuk membuat 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 autentikasi dalam membangun koneksi, dimana untuk enkripsi OpenVPN sendiri menggunakan SSL/TLS yang dimana pembuatan *certificate* SSL-nya dilakukan oleh *OpenSSL* yang telah disediakan oleh Linux.

Cara kerja OpenVPN adalah sebelumnya pada kedua sisi (*server – client*) harus memiliki jalur internet yang permanen. Apabila perusahaan memiliki router maka router tersebut harus dikonfigurasi *firewall*-nya agar dapat mencegah akses terhadap jaringan didalamnya dan juga harus dikonfigurasi agar OpenVPN dapat melewati router tersebut.

Aplikasi OpenVPN harus terinstall didalamnya, dan harus terkonfigurasi agar koneksi dapat terbuat. Apabila hal ini telah dilakukan maka dua sisi (*server – client*) akan dapat terhubung melalui jaringan virtual.

Setiap data yang dilewatkan pada OpenVPN dienkripsi terlebih dahulu dan didekripsi sesudah transmisi. Enkripsi menjamin keamanan data seperti sebuah terowongan kereta api di gunung yang menjaga agar kereta api aman melewati gunung tersebut. Terowongan inilah yang lebih dikenal dengan nama *tunnel*.

Sebuah koneksi OpenVPN biasanya dibuat diantara dua buah akses *internet* dengan *firewall* dan aplikasi OpenVPN. Aplikasi tersebut harus disetting agar koneksi antara partner VPN dapat dilakukan. *Firewall* juga harus disetting agar membolehkan akses dan pertukaran data antara partner VPN yang telah aman sebelumnya karena telah dilakukan enkripsi. Key enkripsi harus disediakan untuk semua partner VPN sehingga pertukaran data hanya bisa dilakukan oleh partner VPN yang telah terotorisasi.

OpenVPN bekerja secara sempurna bersamaan dengan *firewall*. *Firewall* adalah sebuah router yang member rute khusus kepada data yang telah diseleksi

sebelumnya. Peraturan pada *firewall* mendefinisikan bagaimana cara untuk menghandel data dan *traffic* yang spesifik. *Firewall* dapat berupa peralatan ataupun aplikasi pada PC, *server* ataupun pada perangkat lainnya.

Di Linux, kebanyakan *firewall* adalah berbasis kepada program **iptables**. **Iptables** ini merupakan sebuah *interface* pada kernel linux dan juga menawarkan semua yang dapat dilakukan oleh *firewall* modern. Ada kemungkinan cara terbaik untuk memproteksi LAN adalan dengan cara membuat *rules* pada **iptables** dengan menggunakan *shell script.*

OpenVPN ini memiliki banyak sekali keunggulan, yaitu :

1. OpenVPN bersifat *open-source* dan merupakan salah satu *software* yang dapat dipakai diberbagai macam jenis sistem operasi (*multi platform).*
2. Instalasi OpenVPN sangat mudah dilakukan di sistem operasi apapun (*easy to install).*
3. OpenVPN menyediakan *interface* yang mudah digunakan.
4. OpenVPN menawakan tingkat *mobility* yang tinggi kepada penggunanya.
5. OpenVPN menawarkan dua mode VPN, yaitu VPN pada Layer 2 ataupun VPN pada Layer 3.
6. Menyediakan *internal firewall* yang menjamin para karyawan dan staff yang tidak berada di perusahaan tetapi ingin mengakses jaringan yang ada di perusahaan.

   OpenVPN juga menawarkan *tunnel* VPN sebagai tempat lewatnya data sehingga keamanan data menjadi terjamin.

7. OpenVPN memungkinkan konfigurasi dirinya untuk berjalan dengan layanan TCP atau UDP dan sebagai *client* ataupun *server.*
8. OpenVPN memiliki tingkat fleksibilitas yang tinggi dengan berbagai kemungkinan *scriping* baik *script* untuk autentikasi sampai dengan *scripting* lainnya dengan tujuan tertentu.
9. *Support* dengan *dynamic* IP, apabila ada pergantian IP maka pengguna akan diberi pemberitahuan.
10. *Support* dengan NAT, *client* atau *server* dapat menggunakan IP *address private.*

Apabila dilakukan perbandingan antara OpenVPN dengan IPSec VPN, Meskipun IPSec mempunyai standar *de facto,* masih ada banyak pendapat yang menyebutkan lebih baik menggunakan OpenVPN. Di bawah ini terdapat penjelasan tentang apa kekurangan dan kelebihan OpenVPN dibandingkan dengan IPSec.

Tabel 2.3 – Kekurangan dan Kelebihan IPSec VPN

Kelebihan	**Kekurangan**
IPSec merupakan standar untuk teknologi VPN	Sulit untuk memodifikasi IP stack
Support untuk banyak platform hardware (aplikasi dan peralatan)	Modifikasi kritikal terhadap kernel diperlukan
Lebih dikenal	Harus berada pada mode administrator
Banyak GUI untuk administrator	Perbedaan implementasi IPSec dari perusahaan yang berbeda dapat menyebabkan terjadinya masalah dalam kompatibilitas

	Konfigurasi rumit, dan teknologi yang kompleks
	Butuh waktu untuk belajar dan membiasakan diri
	Beberapa port dan protocol pada *firewall* diperlukan
	Terdapat masalah pada IP *Address* yang dinamik (NAT)
	Masalah keamanan pada teknologi IPSec

Tabel 2.4 – Kekurangan dan Kelebihan OpenVPN

Kelebihan	**Kekurangan**
Teknologi yang simple	Masih kurang dikenal dan tidak kompatibel dengan IPSec
Interface jaringan dan paket-paketnya sudah terstandarisasi	Hanya berjalan pada computer, tetapi sudah bisa digunakan dihampir semua sistem operasi
Teknologi enkripsinya telah terstandarisasi	Teknologi baru yang sedang berkembang
Mudah, terstruktur dengan rapi, teknologi yang modular dan mudah untuk dikonfigurasi	Tidak ada GUI professional, tetapi mulai bermunculan proyek-proyek tentang GUI
Mudah dipelajari	
Berjalan sempurna pada DynDNS dan rekoneksi yang lebih cepat	
SSL/TLS sebagai layer kriptografi yang memenuhi standar industry	

Traffic shaping	
Kompatibel dengan *firewall* dan *proxy*	
Tidak ada masalah dengan NAT (dikedua sisinya)	

Asymmetric Encryption dengan SSL/TLS

SSL/TLS menggunakan satu yang terbaik dari teknologi enkripsi yang disebut dengan *asymmetric encryption* untuk memastikan identitas dari partner VPN. Kedua partner enkripsi memiliki dua key, yang satu adalah key public dan satu lagi adalah key pribadi. Key public menangani komunikasi antara partner yang mengenkripsi data dengan SSL/TLS. Karena pemilihan algoritma matematika yang digunakan untuk membuat pasangan key pribadi/publik, dan hanya key pribadi dari penerimalah yang bisa melakukan dekripsi terhadap data yang telah dienkripsi oleh key publiknya.

Keamanan SSL/TLS

Library SSL/TLS dapat digunakan untuk melakukan autentikasi dan enkripsi. Library ini adalah bagian dari OpenSSL yang terpasang pada hampir semua sistem operasi modern. SSL, yang juga terkenal sebagai TLS adalah sebuah protokol yang didisain oleh *Netscape Communications Corporation* untuk meyakinkan kemudahan dari integritas dan autentikasi data untuk mengimbangi perkembangan internet pada tahun 1990an. SSL/TLS adalah sebuah teknologi yang sangat baik yang digunakan hampir disemua website

milik bank, *e-commerce* ataupun aplikasi yang membutuhkan keamanan dan kerahasiaan.

Pada SSL/TLS terdapat sertifikat yang bernama *Trusted Certificates.* Sertifikat ini merupakan sertifikat yang sebelumnya telah dibuat oleh organisasi tertentu (Bank, *E-Commerce*, dll.) yang digunakan untuk menjamin keaslian identitas dari pemilik sertifikat tersebut.

Pada SSL/TLS juga terdapat sertifikat yang disebut dengan *Self-Signed Certificates* yang merupakan sertifikat yang tidak membutuhkan autentikasi seperti pada *Trusted Certificates*, tetapi dengan menggunakan sertifikat yang disebut dengan *Certificate Authority* (CA).

Pada OpenVPN, sertifikat SSL/TLS ini dibuat dan didefinisikan dan semua sertifikat yang valid yang dikeluarkan oleh otorisasi merupakan sertifikat yang akan diterima oleh VPN. Setiap *client* harus mempunyai sertifikat yang valid berdasarkan CA dan yang akan diijinkan untuk membuat koneksi ke VPN.

Certificate Revocation List (CRL) dapat digunakan untuk melakukan pencabutan sertifikat yang dipunyai *client* yang tidak diperbolehkan untuk melakukan koneksi dengan VPN. Koneksi akan ditolak apabila tidak ada sertifikat yang dimaksud, sertifikat yang berbeda dan memiliki CA yang salah, sertifikat yang telah dicabut haknya sebelumnya. Sertifikat-sertifikat ini dapat digunakan untuk berbagai macam tujuan. HTTPS dan OpenVPN adalah hanya dua aplikasi yang menggunakan ini dari berbagai macam aplikasi lainnya.

Jaringan dengan OpenVPN

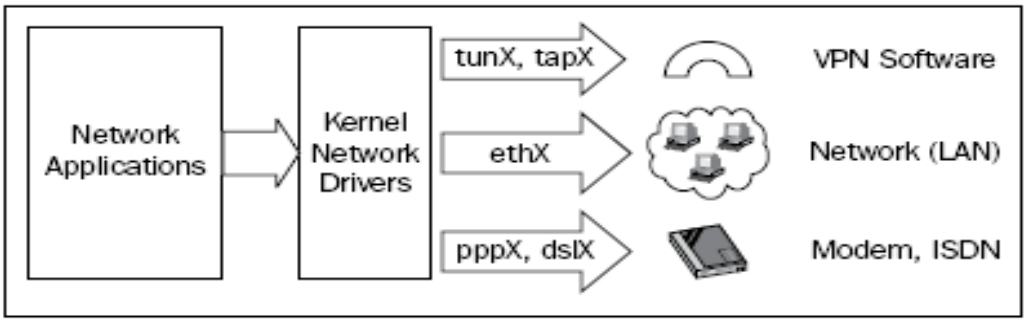
Struktur modular dari OpenVPN tidak hanya bisa ditemukan dalam model keamanannya sendiri, tetapi juga bisa ditemukan di dalam kerangka jaringan. **James Yonan** memilih *driver Universal* TUN/TAP untuk lapisan jaringan dari OpenVPN.

TUN/TAP *driver* adalah sebuah proyek *open-source* yang terdapat di dalam semua distribusi Linux/UNIX yang modern seperti juga Windows dan MacOS X. seperti SSL/TLS, TUN/TAP juga dipakai dalam banyak proyek, oleh karena itu TUN/TAP dengan rutin ditingkatkan dan ditambahkan banyak fitur. Penggunaan TUN/TAP mengebelakangakan banyak kompleksitas dari struktur OpenVPN itu sendiri sehingga dengan strukturnya yang sederhana tersebut dapat meningkatkan keamanan VPN dibandingankan dengan VPN lainnya. Contohnya, IPSec yang memiliki struktur kompleks dengan modifikasi kompleksnya pada *kernel* dan *IP Stack,* yang dapat menyebabkan terciptanya celah-celah kecil pada keamanannya sendiri.

Driver Universal TUN/TAP dikembangkan untuk dapat menyediakan dukungan pada Linux *kernel* untuk keperluan proses *tunneling*. *Driver* ini merupakan sebua *virtual network interface* yang muncul sebagai otentik untuk semua aplikasi dan pengguna; yang mencirikannya dari peralatan lainnya adalah dari penamaannya dengan tunX atau tapX. Setiap aplikasi yang memungkinkan penggunaan *network interface* dapat menggunakan *tunnel* ini.

Driver ini merupakan salah satu faktor utama yang membuat OpenVPN mudah untuk dimengerti, mudah untuk dikonfigurasi dan tidak lupa keamanannya.

Gambar berikut ini menunjukan *interface* sederhana yang digunakan oleh OpenVPN :



Gambar 2.12 – OpenVPN *Standard Interface*

Sebuah TUN dapat digunakan seperti sebuah *virtual interface* untuk melakukan koneksi *point-to-point*, seperti sebuah modem atau DSL *link*. Ini disebut dengan mode *routed*, karena rute antara pasangan VPN telah dikonfigurasi sebelumnya.

Sebuah TAP dapat digunakan seperti sebuah *virtual Ethernet adapter*. Hal ini memungkinkan *daemon* membaca *interface* untuk menangkap *Ethernet frames* yang tidak mungkin dilakukan oleh TUN. Mode ini disebut dengan *bridging mode* karena jaringan-jaringan yang terhubung seolah-olah berada dalam satu *hardware* yang sama.

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

TCP dapat sangat membantu dalam beberapa hal. Pemilihan protocol ini diserahkan kepada penggunanya.

OpenVPN mendengarkan TUN/TAP, mengatur *traffic*, melakukan enkripsi, dan mengirimkan data kepada pasangan VPN yang lain, dimana proses OpenVPN yang lain akan menerima data, melakukan dekripsi, dan menyampaikannya kepada alat jaringan, dimana aplikasi lainnya sedang menunggu data.