BAB II
DASAR TEORI

2.1 Virtual Private Network (VPN)

Virtual Private Network (VPN) merupakan suatu cara untuk membuat jaringan *private* (pribadi) secara aman dengan menggunakan jaringan publik. VPN dapat mengirimkan data antara dua jaringan komputer yang berbeda dengan melewati jaringan publik. Untuk mendapatkan hubungan yang bersifat pribadi, data yang dikirimkan harus dienkripsi agar terjaga kerahasiaannya pada saat melewati jaringan publik. Paket data yang dienkripsi atau dibungkus menjadi paket data lain disebut sebagai enkapsulasi. Proses enkapsulasi ini disebut juga dengan proses *tunneling*.[1]

VPN menggunakan jaringan *internet* untuk berkomunikasi antara pusat dengan cabang, dan pada saat bersamaan menjamin kemanan dan kerahasiaan dari data yang dikirimkan. Secara singkat, VPN dapat diartikan jaringan yang menyediakan koneksi yang aman antara pusat dan cabang dengan melalui internet dengan harga yang murah. Pada VPN kantor atau perusahaan pusat digunakan sebagai *server* dan kantor atau perusahaan cabang digunakan sebagai *client*. Berikut merupakan definisi yang mendasar mengenai dekripsi VPN. VPN merupakan :

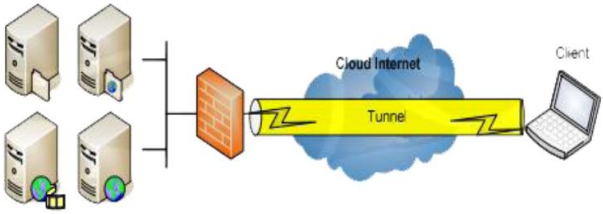
1. *Virtual*
   Dikatakan *virtual* karena tidak ada koneksi jaringan yang nyata antara dua atau lebih mitra komunikasi, hanya koneksi virtual yang disediakan oleh perangkat lunak VPN, tetapi tetap menggunakan koneksi *internet* publik.
2. *Private*
   Dikatakan *private* karena hanya anggota dari perusahaan atau kantor yang dihubungkan oleh perangkat lunak VPN dan diizinkan untuk membaca data yang telah ditransfer.

Dengan menggunakan VPN, pegawai yang berada di lokasi yang jauh dari kantor atau perusahaan pusat dapat saling berhubungan seolah – olah berada pada lokasi yang sama.

Semua data yang telah dikirim dari kantor cabang ke kantor pusat atau sebaliknya harus dienkripsi dan didekripsi sebelum dikirimkan. Enkripsi pengamanan data dalam koneksi dapat disamakan dengan dinding terowongan yang mengamankan kereta dari gunung yang ada disekitarnya. Hal ini menjelaskan bahwa VPN sering dikenal sebagai jaringan terowongan yang menggunakan teknologi *tunneling*.

Koneksi VPN biasanya dibangun antara dua *router* atau lebih

menggunakan akses *internet* yang dilengkapi dengan *firewall* dan perangkat lunak. Perangkat lunak ini harus diatur untuk menyambungkan ke mitra VPN, *firewall* harus dibentuk untuk memungkinkan pengaksesan, dan data yang dipertukarkan antara VPN harus dijamin dengan menggunakan enkripsi. Kunci enkripsi harus disediakan untuk semua mitra VPN sehingga pertukaran data hanya dapat dibaca oleh pihak yang berwenang.



Gambar 2. 1 *Virtual Private Network* (VPN) [2]

Dalam pemanfaatannya, VPN memiliki berbagai keunggulan, diantaranya adalah sebagai berikut :

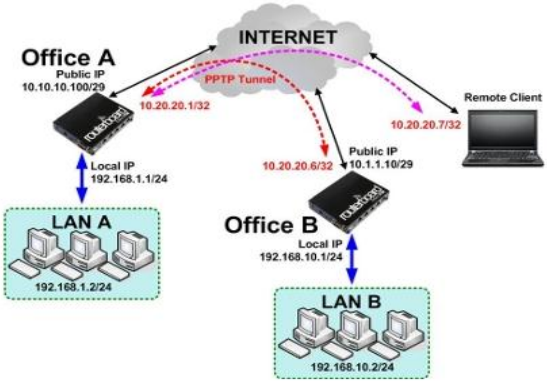
1. VPN dapat mengakses jaringan kantor maupun komputer dimana saja dan kapan saja, dengan menggunakan *remote access*.
2. VPN dapat menghemat biaya infrastruktur jaringan, karena VPN dapat digunakan sebagai teknologi alternatif yang menghubungkan suatu jaringan lokal melalui jaringan publik yang sudah ada, tanpa perlu membangun jaringan baru.
3. VPN lebih aman digunakan dalam hal *transfer* data karena adanya proses enkripsi data.[2]

2.2 *Point to Point Tunneling Protocol* (PPTP)

Point to Point Tunneling Protocol (PPTP) merupakan salah satu tipe VPN yang fleksibel dan konfigurasi yang sederhana. Mayoritas sistem operasi sudah mendukung dengan PPTP *client* baik pada perangkat *client* (komputer, *laptop*, dan lain sebagainya) maupun pada *smartphone* seperti android. Komunikasi pada PPTP menggunakan protokol TCP (*Transmission Control Protocol*) dan *tunnel* GRE (*Generic Routing Encapsulation*) untuk enkapsulasi paket datanya.[7] TCP merupakan lapisan *transport* yang memiliki fungsi untuk mengirimkan data ke tujuan yang bersifat *connection- oriented*, yang berarti bahwa perangkat yang digunakan untuk pertukaran data, harus melakukan hubungan terlebih dahulu sebelum pertukaran data berlangsung.

Dalam konfigurasinya, PPTP menentukan *network security protocol* yang digunakan untuk proses autentifikasi PPTP yang sama seperti PPP pada mikrotik seperti *Password Authentication Protocol* (PAP), *Challenge*

Handshake Protocol (CHAP), *Microsoft version of the Challenge Handshake Authentication Protocol* (MSCHAP) dan *Microsoft Challenge Handshake Authentication Protocol version* 2 (MSCHAP2). Data yang ditransmisikan melalui *tunneling* PPTP akan dienskripsi menggunakan *Microsoft Point to Point Encryption* (MPPE). [3]

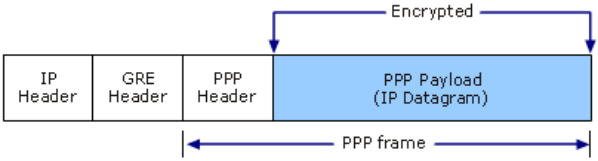


Gambar 2. 2 Topologi Jaringan VPN PPTP [4]

2.2.1 Arsitektur PPTP

Tunneling PPTP melalui tiga proses untuk membuat suatu komunikasi yang aman, dimana setiap proses saling memiliki keterhubungan. Proses tersebut adalah sebagai berikut:

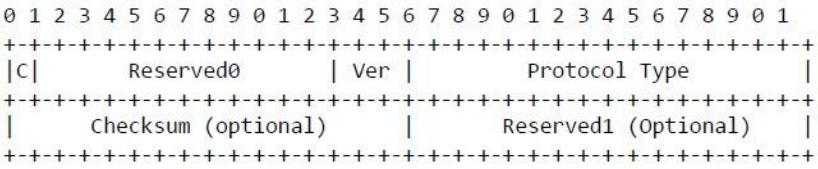
1. PPTP *Connection and Communication*
   Pada proses ini, *client* PPTP menggunakan PPP agar terhubung dengan ISP (*Internet Service Provider*). Penggunaan protokol PPP digunakan untuk menghubungkan koneksi dan enkripsi paket data.
2. PPTP *Control Connection*
   Setelah koneksi internet telah dibangun oleh *tunneling* PPTP, kemudian *tunneling* melanjutkan proses *control connection* dari *client* ke *server* PPTP. Koneksi PPTP menggunakan TCP untuk komunikasi dan disebut dengan PPTP *tunnel.*
3. PPTP *Data Tunneling*
   Proses terakhir pada *tunneling* PPTP adalah pembuatan IP *datagrams* yang didalamnya terdapat enkripsi paket PPP. Paket ini kemudin dikirimkan PPTP tunnel ke *server* PPTP. *Server* PPTP kemudian membongkar IP *datagrams* dan mendeskripsikan paket PPP dan selanjutnya paket yang telah dideskripsikan tersebut ke jaringan pribadi.[5]



Gambar 2. 3 Struktur PPTP [3]

2.2.2 *Generic Routing Encapsulation* (GRE)

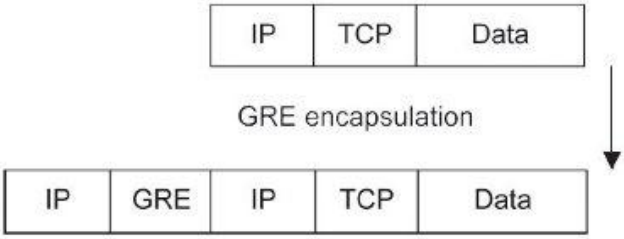
GRE adalah merupakan sebuah *tunneling* yang dikembangkan oleh cisco dan memiliki fungsi untuk melakukan enkapsulasi paket yang berada dalam lapisan *network protocol* dalam *tunneling*-nya. Proses enkapsulasi pada GRE adalah dengan cara membuat virtual komunikasi *point to point* dari router asal menuju router tujuan dengan menggunakan IP pada komunikasi *internetworking*.



Gambar 2. 4 GRE *Header* [6]

Keterangan :

1. *Checksum Present*
   Apabila *checksum* diatur pada bit ke satu, maka kolom *checksum* dan *reserved*1 akan dihadirkan, checksum berisi informasi *invalid*.
2. *Reserved* 0
   Penerima harus mengabaikan paket dimana setiap bit adalah non-zero,
3. *Version Number*
   Kolom ini harus bernilai 0
4. *Protocol Type*
   Kolom ini berisi jenis protokol *payload*.
5. *Checksum*
   Kolom ini berisi checksum IP dari GRE *header* dan *payload* packet.
6. *Reserved*1
   Kolom Reserved1 dihadirkan hanya ketika kolom checksum tersedia (checksum diatur ke bit 1). [6]

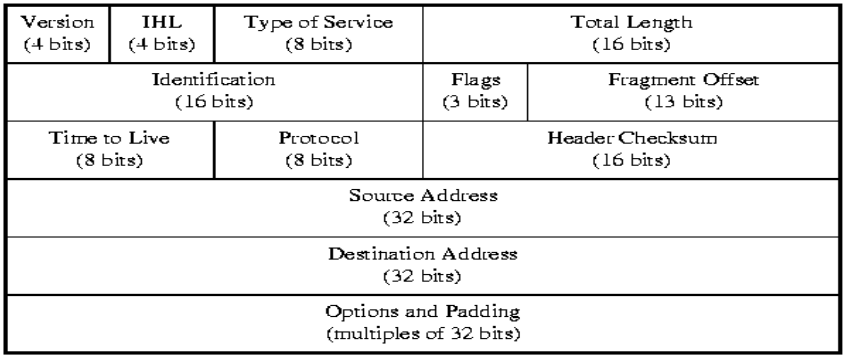


Gambar 2. 5 Proses enkapsulasi GRE

Gambar 2.5 menunjukan proses enkapsulasi GRE. GRE memiliki kemampuan untuk membawa beberapa protokol pengalamatan komunikasi. Bukan hanya paket yang memiliki alamat IP saja, namun juga protokol seperti CNLP, IPX, dan lain sebagainya. Enkapsulasi yang dilakukan oleh GRE yaitu dengan membungkus atau mengenkapsulasi protokol pengalamatan IP menjadi sebuah paket yang bersistem pengalamatan IP. Selanjutnya paket tersebut akan didistribusikan melalui system *tunnel* yang bekerja pada protokol komunikasi IP. Router yang berada pada ujung *tunnel* melakukan enkapsulasi paket – paket protokol lain di dalam header protokol IP dengan cara mendistribusikannya melalui sistem *tunneling*. Sehingga dengan adanya metode ini ini, protokol – prorokol yang dibawa oleh paket IP akan lebih bebas bergerak ke tujuan selama terjangkau oleh pengalamatan IP. [5]

2.2.3 *Internet Protocol Version 4* (IPv4)

IP adalah merupakan protokol yang digunakan untuk melakukan mekanisme pengalamatan. Salah satu jenis pengalamatan IP adalah IPv4. IPv4 merupakan salah satu jenis pengalamatan jaringan yang digunakan didalam protokol jaringan TCP/IP dan menggunakan protokol IP versi 4. Jumlah bit yang digunakan pada pengalamat versi 4 ini adalah sebanyak 32 bit.



Gambar 2. 6 IP *Header*[7]

Keterangan :

1. *Version*
   Bagian ini berisi keterangan versi IP *Protocol*, versi IP yang digunakan adalah IP *version* 4.
2. *Internet Header Length* (IHL)
   IHL digunakan untuk mengetahui ukuran *header* IP.
3. *Type of Services* (TOS)
   Type of Services digunakan untuk mengidentifikasi parameter kualitas layanan yang akan digunakan dalam transmisi paket IP.
4. *Total Length*
   Total *datagram* IP, yaitu berisi *header* IP dan muatannya.
5. *Identification*
   Identification digunakan untuk mengidentifikasi sebuah paket IP tertentu yang akan difragmentasi
6. *Flag*
   Bagian ini berisi dua buah flag yang digunakan untuk mengetahui bahwa sebuah *datagram* IP mengalami fragmentasi atau tidak.
   a. Bit 0 = *reserved*, diisi 0.
   b. Bit 1 = apabila bernilai 0 akan difragmentasi, namun apabila bernilai 1 tidak dapat difragmentasi.
   c. Bit 1 = apabila bernilai 0 maka fragmentasi akan berakhir, namun apabila bernilai 1 maka akan ada fragmentasi
7. *Fragment Offset*
   Fragment Offset digunakan untuk mengidentifikasi *offset*, dimulai saat fragment bersangkutan, dan dihitung dari permulaan muatan IP yang belum dipecah.
8. *Time to Live*
   Time to Live digunakan untuk mengetahui waktu maksimal paket harus tiba pada tujuan.
9. *Protocol*
   Protocol digunakan untuk mengetahui jenis protokol lapisan yang lebih tinggi yang dikandung oleh muatan IP.
10. *Header Checksums*
    Header Checksums digunakan untuk pengecekan integritas terhadap IP *header*.
11. *Source and Destination IP Address*
    Berisi mengenai sumber dan tujuan *datagram* IP.

12. *Options*

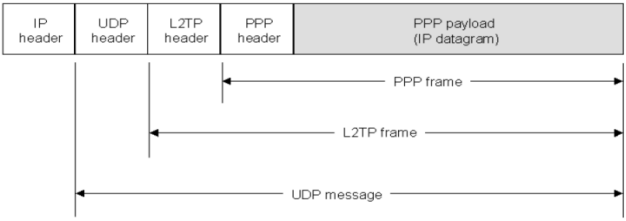
    Option digunakan untk mengkodekan permintaan pengirim seperti *record routing, time stamping, security label,* dan *source routing*.

13. *Padding*

    Digunakan untuk memastikan bahwa *header* paket berhenti pada bit ke -32. [7]

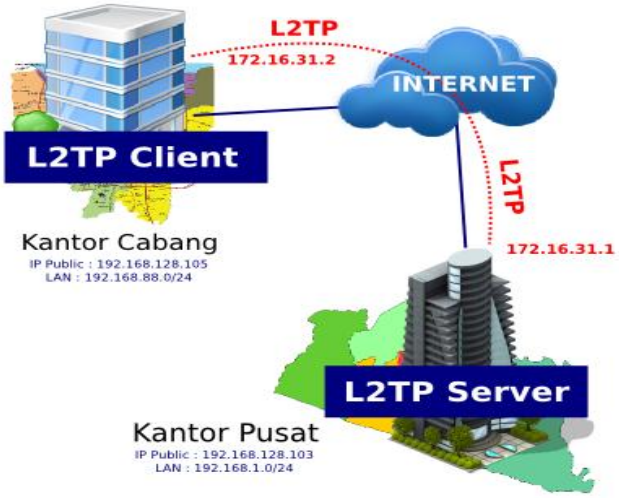
2.3 *Layer Two Tunneling Protocol* (L2TP)

Layer Two Tunneling Protocol merupakan pengembangan dari PPTP ditambah L2F (*Layer* 2 *Forwarding*) milik Cisco. Komunikasi pada L2TP menggunakan protokol UDP (*User Datagram Protocol*) untuk enkapsulasi paket datanya. UDP merupakan lapisan *transport* protokol yang mendukung komunikasi yang tidak handal (tanpa pengecekkan kesalahan) dan tanpa koneksi (*connectionless*). L2TP memiliki dua komponen utama yaitu L2TP *Network Server* (LNS) yang digunakan untuk mengakhiri dan authentifikasi aliran PPP dan L2TP *Access Concentrator* (LAC) yang secara fisik digunakan untuk mengakhiri suatu panggilan. Pada dasarnya L2TP menggunakan protocol UDP untuk mengirimkan PPP *frame* yang dienkapsulasi sebagai data yang akan dikirim melalui *tunnel.*



Gambar 2. 7 Struktur L2TP[3]

Network security protocol yang dan enkripsi yang digunakan oleh L2TP untuk authentifikasi sama dengan yang digunakan PPTP. Namun, pada L2TP untuk mendapatkan keamanan yang lebih baik, maka L2TP dikombinasikan dengan IPSec yang menjadi L2TP/IPSec. Dari segi enkripsi, L2TP/IPSec memiliki tingkat keamanan yang lebih tinggi daripada PPTP yang menggunakan *Microsoft Point to Point Encryption* (MPPE).[4]



Gambar 2. 8 Topologi Jaringan VPN L2TP [4]

2.4 IP *Security* (IPsec)

IPsec adalah sekumpulan ekstensi dari keluarga *tunneling* IP (*Internet Protocol*). IPSec menyediakan layanan kriptografi untuk keamanan transmisi data. Layanan ini termasuk *authenticity*, *integrity*, *access control*, *confidentiality*, dan *replay protection*. Layanan IPsec mirip dengan SSL (*Secure Socket Layer*) namun, IPsec melayani lapisan *network*, dan dilakukan secara transparan. Layanan tersebut dideskripsikan sebagai berikut :

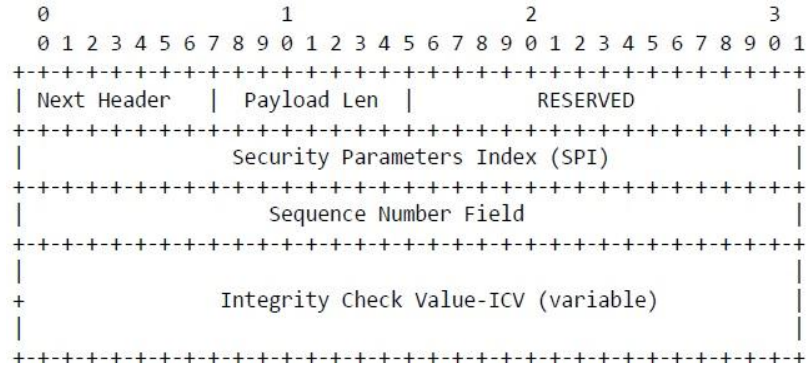
1. *Confidentiality* digunakan untuk menjaga keamanan data dengan cara enkripsi.
2. *Integrity* digunakan untuk menjamin bahwa data tidak berubah dalam perjalanan menuju tujuan.
3. *Authenticity* digunakan untuk menandai bahwa data yang dikirimkan memang berasal dari pengirim yang benar.
4. *Replay Protection*, untuk memastikan bahwa transaksi data hanya dilakukan sekali, kecuali sudah memiliki izin untuk perulangan. [8]

IPsec memiliki *tunneling* yang digunakan untuk melindungi data, diantaranya adalah :

a. *Authentication Header* (AH)

AH menyediakan layanan *authentication*, *integrity* dan *replay protection*, namun tidak dengan *confidentiality*. AH juga melakukan pengamanan terhadap *header* IP. Namun, AH tidak menawarkan fungsi enkripsi terhadap

data yang ditransmisikan. Informasi AH dapat digunakan sendiri atau dapat digunakan bersamaan dengan protocol ESP.



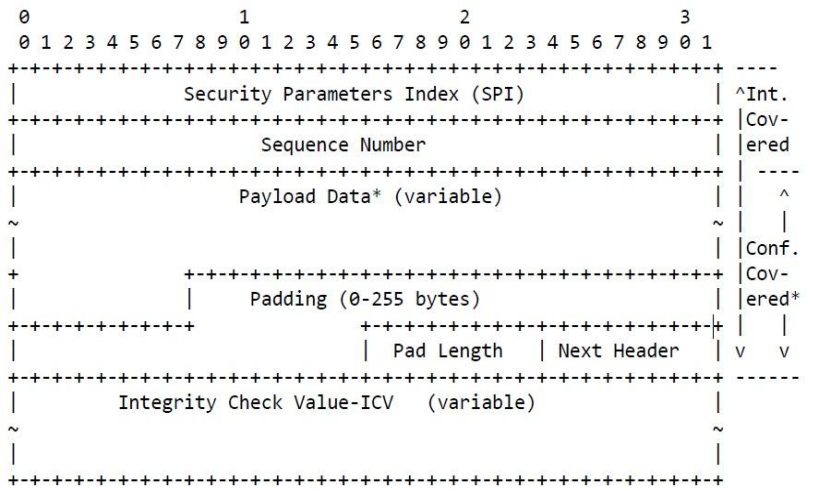
Gambar 2. 9 AH *Header*[9]

Keterangan :

1. *Next Header*
   Next Header adalah *field* 8-bit yang digunakan untuk identifikasi tipe dari *next payload* setelah AH.
2. *Payload Length*
   Payload Length adalah *field* 8-bit yang digunakan untuk menspesifikasikan panjang AH dalam 32 bit *word* (4 byte unit), kemudian panjang ini dikurangi dengan bilangan 2.
3. *Reserved*
   Reserved adalah *field* 16-bit yang dicadangkan untuk penggunaanya di masa datang.
4. *Security Parameter Index* (SPI)
   SPI adalah nilai 32-bit yang digunakan oleh penerima untuk mengidentifikasi dari SA mana paket akan dikirimkan.
5. *Sequence Number*
   Sequence Number adalah *field* 32-bit yang berisi nilai pencacah akan bertambah besar pada setiap paket yang dikirimkan.
6. *Integrity Check Value* (ICV)
   ICV adalah *field* yang berisi nilai cek untuk integritas. *Field* ini harus memilili panjang kelipatan sebesar 32-bit. [9]

b. *Encapsulapted Security Payload* (ESP)

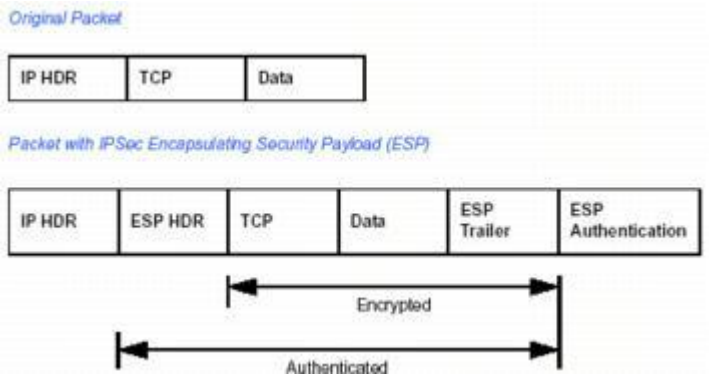
ESP menyediakan layanan *authentication*, *integrity*, *replay protection*, dan *confidentiality* terhadap data. ESP akan melakukan pengamanan data tehadap segala sesuatu paket data setelah *header* IP. Protokol ini dapat digunakan bersamaan dengan AH maupun digunakan sendirian.



Gambar 2. 10 ESP *Header* [11]

Keterangan :

1. *Sequence Parameter Index* (SPI)
   SPI adalah nilai 32-bit yang digunakan oleh penerima untuk mengidentifikasi dari SA mana paket akan dikirimkan.
2. *Sequence Number*
   Sequence Number adalah field 32-bit yang berisi nilai pencacah akan bertambah besar pada setiap paket yang dikirimkan.
3. *Payload Data*
   Payload adalah *field* yang berisi data (dari paket IP asli) yang dideskripsikan dengan *field next header.*
4. *Padding*
   Padding digunakan untuk enkripsi data dan untuk memastikan *chipertext* berakhir pada batas 4 byte.
5. *Padding Length*
   Padding Length digunakan untuk menentukan ukuran *padding field* dalam byte.
6. *Next Header*
   Next Header digunakan untuk mengidentifikasi format *payload*, seperti TCP atau UDP.
7. *Integrity Check Value* – ICV (*variable*)
   ICV adalah *field* yang memiliki otentikasi yang digunakan untuk memverifikasi identitas dan integritas pesan.



Gambar 2. 11 Proses Enkapsulasi ESP [10]

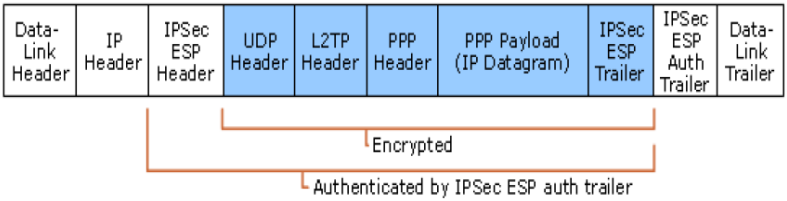
Gambar 2.11 adalah proses enkapsulasi dari ESP. ESP menyediakan otentikasi, integritas, kerahasiaan dan perlindungan terhadap data. Selain itu ESP juga menyediakan layanan enkripsi yang digunakan untuk IPSec. Proses enkripsinya adalah dengan mengubah pesan yang terbaca menjadi tidak terbaca atau disembunyikan. Proses kebalikan dari enkripsi yaitu dekripsi yang mengubah data tersembunyi menjadi data yang dapat dibaca. Pesan yang terbaca tersebut hanya dapat dibaca oleh pengirim dan penerima saja. ESP menyediakan otentikasi dan integritas data, namun proses otentikasi ini hanya digunakan untuk *payload*, bukan digunakan untuk IP header. [10]

2.5 *Layer Two Tunneling Tunneling/ IP Security* (L2TP/IPSec)

Tunneling L2TP memiliki kekurangan dalam hal kerahasian. Hal ini mendorong penggunaan L2TP bersama – sama dengan IPSec untuk menjamin kerahasianan data yang dilalui. Proses dalam pembentukan L2TP/IPSec adalah sebagai berikut :

1. Proses negosiasi IPSec *Security Association* (SA) dengan *Internet Key Exchange* (IKE).
2. Pembentukan komunikasi *Encapsulating Security Payload* (ESP) dengan *Transport Mode*.
3. Proses negosiasi dan pembentukan L2TP *tunnel*.

Keuntungan penggunaan L2TP/IPSec adalah memberikan perlindungan ganda melalui otentifikasi IPSec dan L2TP. Namun penggunaan L2TP/IPSec juga memiliki kerugian pada sisi performansi karena terjadinya *overhead* pada prosesnya.[11]



Gambar 2. 12 Struktur L2TP/IPSec[11]

2.6 Mikrotik

Router merupakan suatu perangkat jaringan yang digunakan untuk menghubungkan beberapa jaringan, baik jaringan yang sama maupun berbeda teknologi. *Router* memiliki berbagai jenis bentuk dan merek, salah satunya yaitu Mikrotik.

Mikrotik merupakan perangkat jaringan yang berbasis *kernel linux* yang memiliki fitur – fitur yang cukup lengkap. Fitur – fitur yang tersedia pada Mikrotik diantaranya *firewall*, NAT, *routing*, *hotspot*, *tunneling*, *IPSec*, *web Proxy*, DNS, DHCP, *Monitoring*, dan fitur lainnya yang tersedia pada mikrotik.

Selain fitur – fitur mikrotik yang beraneka ragam, mikrotik juga terdiri dari beberapa jenis, diantaranya adalah :

1. *RouterBOARD*

   RouterBOARD adalah sebuah *router* mikrotik yang berupa perangkat keras atau alat. *RouterBOARD* memiliki beragam seri dan *interface* sesuai dengan kebutuhan. Sebuah *routerBOARD* memiliki *RouterOS* yang digunakan sebagai sistem operasi. Contoh Mikrotik *routerBOARD* adalah Mikrotik RB750, Mikrotik RB2011UiAS-2HnD-IN, dan lain sebagainya
2. *RouterOS*

   RouterOS merupakan perangkat lunak mikrotik yang dapat diterapkan pada Mikrotik *RouterBOARD* maupun komputer. *RouterOS* yang diterapkan pada komputer akan menjadi sebuah *router* mikrotik yang handal dan memiliki fitur yang lengkap.

Walaupun mikrotik dibuat berdasarkan *kernel linux* yang *open source*, tetapi mikrotik memiliki lisensi yang dapat menggunakan fitur – fitur mikrotik. Setiap lisensi memiliki fitur – fitur yang berbeda. Jenis – jenis lisensi pada mikrotik adalah sebagai berikut :

1. *Level* 0 (Gratis)

   Level ini tidak memerlukan lisensi, namun penggunaan lisensinya hanya dibatasi selama 24 jam.

2. *Level* 1 (Demo)
   Level ini dapat menggunakan mikrotik secara penuh dan seluruh fitur yang tersedia dapat digunakan. Namun waktu penggunaan untuk *level* ini hanya 24 jam, setelah itu fitur – fitur yang aktif sebelumnya akan dikunci secara otomatis.
3. *Level* 3
   Lisensi *level* ini sudah mencakup lisensi *level* 1 dan dilengkapi dengan kemampuan untuk mengatur semua perangkat keras yang berbasis IP *Address.*
4. *Level* 4
   Lisensi *level* ini sudah mencakup lisensi *level* 1 dan 3, serta ditambah fitur untuk mengelola jaringan *wireless* tipe *access point*.
5. *Level* 5
   Lisensi *level* ini mencakup lisensi *level* 1, 3 dan 4, serta ditambah fitur untuk mengelola hotspot lebih banyak.
6. *Level* 6
   Lisensi *level* ini merupakan lisensi tertinggi di Mikrotik, sehingga diberikan fitur – fitur yang tersedia pada mikrotik tanpa ada limitasi apa pun.[1]

2.7 Model Referensi OSI (*Open System Interconnection*)

Open System Interconnection merupakan sebuah *tunneling* interkoneksi sistem terbuka yang dibuat oleh ISO (*International Organization for Standarzation*) untuk menyediakan model dasar sehingga dapat memodelkan semua protokol. Model OSI telah dipergunakan dalam praktik. Namun, model ini umumnya berfungsi sebagai prototipe teoritis berupa grafik dan blok – blok diagram untuk membuat sistem protokol jaringan yang baik. Protokol – protokol jaringan yang ada saat ini mengambil sebagian atau keseluruhan fungsi dasar pada model OSI.

2.7.1 Karakeristik Lapisan OSI

Model OSI terbagi menjadi tujuh lapisan dan setiap lapisan memiliki fungsi masing – masing. Lapisan – lapisan ini merepresentasikan tipe – tipe fungsi yang seharusnya didukung oleh protokol. Lapisan – lapisan disusun berdasarkan blok fungsi model logik, dari atas ke bawah. Bagian atas lapisan merupakan lapisan yag dekat dengan pengguna atau aplikasi, sedangkan lapisan bagian bawah memiliki fungsi yang lebih dekat dengan fisik atau antarmuka jaringan.

2.7.2 Lapisan – lapisan Model OSI

Model OSI memiliki tujuh lapisan, diantaranya :

1. Lapisan Fisik (*Physical*)
   Lapisan fisik merupakan lapisan pertama, atau lapisan terbawah pada model OSI. Lapisan ini berfungsi untuk mengirim dan menerima data dari atau ke media fisik seperti konektor, kabel, perangkat keras dan media radio maupun satelit.
2. Lapisan Hubungan Data (*Data Link*)
   Lapisan hubungan data merupakan lapisan kedua. Awalnya lapisan ini dibuat sebagai lapisan fungsional tunggal (*single function layer*). Namun karena kebutuhan yang semakin banyak, sehingga lapisan ini terbagi menjadi dua sub lapisan yaitu *Logical Link Control* (LLC) dan *Media Access Control* (MAC). Kedua lapisan ini memiliki fungsi untuk memindahkan paket menuju dan keluar dari jaringan, Pada lapisan ini bit dan *byte* bergabung menjadi *frame*, atau sebaliknya. Sub lapisan LLC memaketkan *byte* yang diterima dari sublapisan MAC yang ada dibawah sehingga menjadi format yang mudah dibaca oleh lapisan jaringan diatasnya. Contoh : *Point to Point Protocol* (PPP)
3. Lapisan Jaringan (*Network*)
   Lapisan jaringan adalah lapisan ketiga pada model OSI yang memiliki fungsi untuk melakukan rutinitas paket melalui multiple jaringan. Lapisan ini bekerja tanpa memperhatikan protokol pokok yang digunakan. Hal ini menyebabkan alat seperti *router* dapat beroperasi pada *level* ini dapat digunakan untuk menghubungkan jaringan – jaringan pada lapisan *datalink* dan lapisan fisik. Contoh : *Routing Information Protocol* (RIP).
4. Lapisan *Transport*
   Lapisan *transport* merupakan lapisan keempat pada model OSI. Lapisan ini berfungsi untuk mentransmisikan pesan dari *host* pengirim ke penerima. Lapisan *transport* bertugas membuat sirkuit *virtual* diantara dua titik didalam jaringan dan memastikan integritas data (jika *level* atau *tunneling* dibawahnya tidak menyediakan servis ini). Contoh : *Transmission Control Protocol* (TCP), *Name Binding Protocol* (NBP) dan *User Datagram Protocol* (UDP).
5. Lapisan Sesi (*Session*)
   Lapisan kelima ini memiliki fungsi untuk mengadakan, mempertahankan dan memutuskan komunikasi diantara aplikasi – aplikasi atau proses – proses yang berjalan dijaringan. Contoh : SQL, *Apple Talk Session Protocol*

(ASP), dan *Digital Network Architecture Session Control Program* (DNASCP).

6. Lapisan Presentasi (*Presentation*)
   Lapisan presentasi merupakan lapisan keenam pada model OSI. Lapisan ini sangat erat kaitannya dengan lapisan aplikasi. Tugas utamanya adalah untuk memastikan bahwa data yang sedang dilewatkan menuju lapisan aplikasi sudah dikonversi menjadi format yang diketahui oleh lapisan aplikasi. Contoh : ASCII, JPEG, dan MIDI
7. Lapisan Aplikasi (*Application*)
   Lapisan ini merupakan lapisan teratas pada model OSI. Lapisan aplikasi digunakan untuk menyediakan akses jaringan untuk program aplikasi. Program aplikasi pengguna dan servis sistem biasanya memperoleh akses jaringan melalui interaksi dengan proses yang sedang berjalan pada lapisan OSI ini. Contoh : *E-mail* (pop3 dan SMTP), *File Transfer Protocol* (FTP), dan *Hypertext Transfer Protocol* (HTTP).

2.8 Model Referensi TCP/IP (*Transmission Control Protocol* / *Internet Protocol*)

TCP/IP bukan merupakan protokol tunggal, dari namanya dapat diketahui kalau TCP/IP merupakan sepasang protokol. TCP/IP juga berupa sederet protokol. Dengan kata lain, TCP/IP merupakan kumpulan protokol yang saling berkerja sama. TCP/IP digunakan untuk membuat sebuah protocol yang dapat melintasi lingkungan jaringan yang beraneka ragam dan mempunyai kemampuan menjalankan rute ganda untuk sampai ke tujuan akhir. Pada mulanya, TCP/IP disebut NCP (*Network Control Protocol*) dan merupakan proyek penelitian dari departemen pertahanan Amerika untuk ARPAnet (*Advanced Research Project Agency Network*). Protokol percobaan tersebut melewati *packet switch*. Tujuannya adalah memastikan bahwa jika suatu bagian jaringan mengalami kerusakan, transmisi data akan berlangsung melewati alternatif lain.

2.8.1 Karakeristik Lapisan TCP/IP

TCP/IP digunakan untuk mengatur komunikasi data komputer di *internet* dan memastikan pengiriman data ke alamat yang dituju. Lapisan – lapisan protokol TCP/IP melayani permintaan pengguna untuk mengirim dan menerima data, mengatur komunikasi antar *host*, melakukan pengecekan kesalahan, menyampaikan paket ke alamat yang benar, dan mengirim atau menerima data dari media fisik.

2.8.2 Lapisan – lapisan Model TCP/IP

Protokol TCP/IP memiliki empat lapisan yang merupakan penyederhanaan dari lapisan – lapisan yang ada di OSI, diantaranya adalah sebagai berikut :

1. Lapisan Antarmuka Jaringan atau Fisik
   Lapisan antaramuka jaringan atau fisik merupakan lapisan paling bawah. Lapisan ini bertugas mengirimkan paket data yang berisi IP. Lapisan ini bekerja dekat dengan ARP (*Address Resolution Protocol*) untuk menentukan *header* informasi yang tepat yang perlu ditambahkan pada masing – masing *frame.* ARP adalah lapisan *internet* pada *tunneling* yang bertugas untuk menentukan alamat komputer. ARP disebut juga sebagai alamat MAC (*MAC Address*). Lapisan ini membuat *frame* yang cocok dengan tipe jaringan yang digunakan seperti *Ethernet* atau Token *Ring*, kemudian meletakan paket IP ke daerah muatan pada sebuah *frame* dan mengirimkan data.
2. Lapisan *Internet*
   Lapisan *internet* bertugas untuk menjalankan data didalam dan diantara jaringan yang berbeda. *Router* sangat berfungsi pada lapisan protokol ini dan bertugas meneruskan paket data dari suatu jaringan atau segmen. Contoh : IP (*Internet Protocol*) dan ARP (*Address Resolution Protocol*)
3. Lapisan *Transport* (Transportasi)
   Lapisan Transportasi berfungsi mengatur komunikasi antar *host* dan melakukan pengecekkan kesalahan. Lapisan ini melakukan dan mempertahankan komunikasi *point to point* diantara dua *host.* Fungsi utamanya adalah memberi balasan terhadap informasi yang diterima, mengontrol aliran, mengurutkan dan mentransmisikan paket-paket data. Contoh : TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*).
4. Lapisan Aplikasi
   Lapisan Aplikasi pada model protokol TCP/IP adalah lapisan yang melayani permintaan pengguna untuk mengirim dan menerima data. Lapisan ini merupakan tempat dimana aplikasi dan servis – servis lain memperoleh akses ke jaringan. Contoh : NetBIOS.[12]

2.9 *Quality of Services* (QoS)

Quality of Service (QoS) merupakan kualitas atau jaminan terhadap layanan (*service*) yang diberikan kepada pengguna jaringan. Kinerja jaringan VPN dievaluasi berdasarkan parameter – parameter kualitas layanan seperti

delay, *jitter*, *packet loss*, dan *throughput*. Berikut ini adalah definisi singkat dari keempat parameter layanan VPN tersebut.

1. *Throughput* adalah sebuah ukuran tentang berapa banyak data yang bisa dialirkan dalam sebuh media yang sebenarnya per satuan waktu. Besaranya nilai *throughput* dapat dipengaruhi oleh beberapa hal, diantaranya jumlah pengguna dan *delay*. [13]
2. *Delay* adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media transmisi, dan waktu.[14]

Tabel 2.1 Standarisasi *Delay* pada TIPHON TR 101 329 v2.1.1 (1999-06) [15]

Kategory *Delay*	**Besar *Delay***	**Indeks**
Sangat Bagus	<150 ms	4
Bagus	150 – 300 ms	3
Sedang	300 – 450 ms	2
Jelek	>450 ms	1

3. *Jitter* adalah variasi – variasi dalam panjang antrian, dalam waktu pengolahan data dan juga dalam waktu penghimpunan ulang paket – paket perjalanan akhir. *Jitter* juga merupakan masalah dalam sebuah jaringan jika terdapat paket data yang berbeda dengan *delay* yang berbeda. Sehingga aplikasi yang menggunakan data yang rentan terhadap waktu menjadi terganggu, seperti suara dan video. [2]

Tabel 2. 2 Standarisasi *Jitter* pada TIPHON TR 101 329 v2.1.1 (1999-06) [15]

Kategory *Jitter*	***Peak Jitter***	**Indeks**
Sangat Bagus	0 ms	4
Bagus	0 - 75 ms	3
Sedang	75 - 125 ms	2
Jelek	125 - 225 ms	1

4. *Packet loss* adalah suatu parameter yang menggambarkan suatu kondisi yang menunjukan jumlah total paket yang hilang dalam proses pengiriman data dari sumber ke tujuan.[14]

Tabel 2. 3 Standarisasi *Packet loss* pada TIPHON TR 101 329 v2.1.1 (1999-06) [15]

Kategory ***Packet loss***	***Packet loss***	**Indeks**
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

Dengan adanya parameter QoS seperti *delay, jitter, packet loss,* dan *throughput*, maka kualitas jaringan dapat diukur tingkat kualitasnya. Adanya QoS juga dapat digunakan untuk menentukan teknologi yang tepat digunakan pada suatu perusahaan ataupun perkantoran.

2.10 Layanan Jaringan

Layanan suatu jaringan sangat bermacam- macam jenisnya. Layanan ini didapat dengan cara mengakses suatu *server* yang ada didalam suatu *server* yang ada pada suatu jaringan. Bentuk dari layanan jaringan tersebut bermacam – macam tergantung dari *server* yang diakses.

2.10.1 *Hypertext Transfer Protocol* (HTTP)

Hypertext Transfer Tunneling (HTTP) adalah sebuah protokol jaringan pada lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif dan menggunakan *hypermedia*. HTTP diprakarsai oleh sistem informasi global World-Wide Web sejak tahun 1990. Versi pertama HTTP (HTTP/0.9) merupakan protokol yang sederhana untuk data raw yang melakukan transfer melintasi *internet*. Realisasi penggunaan aplikasi yang memanfaatkan protokol HTTP dilakukan dengan pendekatan *client - server*. Pada kondisi nyatanya, posisi *server* ditempatkan dengan aplikasi *server* web seperti Apache dan IIS, sedangkan aplikasi *client* dapat diilustrasikan itu adalah peramban yang dapat menggunakan protokol HTTP. [16]

2.10.2 *File Transfer Protocol* (FTP)

File Transfer Protocol (FTP) merupakan protokol yang bertanggung jawab pada pertukaran *file* di *internet*. FTP berkerja seperti layaknya protokol HTTP dalam menyajikan halaman web. Umumnya, FTP *server* merupakan suatu komputer yang khusus menyimpan *file – file* yang dapat diunduh dari *internet*. Model protokol FTP adalah sebagai berikut :

1. *Control Connection*

   Control Connection digunakan untuk hubungan antara *server* dan *client. Server* membuka diri secara pasif di sebuah *port* khusus yaitu *port* 21.
2. *Data Connection*

   Data Connection yang dibangun setiap kali sebuah *file* ditranfer dari *server* ke *client*. *Port* yang digunakan yaitu *port* 20.[12]