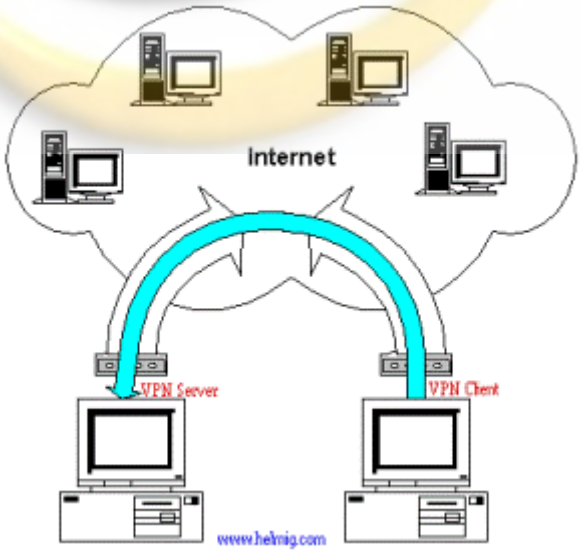
BAB II

LANDASAN TEORI

2.1 *Virtual Private Network* (*VPN*)

VPN adalah sebuah koneksi *private* melalui jaringan publik atau *internet*, *virtual network* berarti jaringan yang terjadi hanya bersifat *private* dimana tidak semua orang bisa mengaksesnya. Data yang dikirimkan terenkripsi sehingga tetap rahasia mesk*IP*un melalui jaringan Publik. Jika menggunakan *VPN* kita seolah-olah membuat jaringan didalam jaringan atau biasa disebut *tunnel*. *VPN* menggunakan salah satu dari tiga teknologi tunneling yang ada yaitu: *PPTP*, *L2TP* dan standar terbaru, *Internet Protocol Security* (biasa disingkat menjadi *IP*Sec). *VPN* merupakan perpaduan antara teknologi *tunneling* dan enkripsi [1]. Proses perjalanan informasi pada *VPN* dimulai dengan pembuatan identitas khusus yang dilakukan oleh *server* dan kemudian identitas itu selanjutnya diketahui oleh *client* untuk membentuk koneksi secara *private*. Apabila koneksi antar kedua *tunnel end-point* telah terjadi, maka *virtual tunnel* telah terbentuk dan kedua jaringan terhubung.

.



Gambar 2.1 Teknologi VPN Server dan VPN Client [helmig.com]

Pada teknologi *VPN* akan dikenal istilah *tunnel* dan enkripsi-dekr*IP*si. *Tunnel* dapat diartikan sebagai suatu pembentukan jalur khusus didalam jaringan publik internet yang bertujuan untuk mengamankan komunikasi antar *server* dan *client* dari serangan jaringan publik *internet*. Enkripsi adalah suatu proses untuk mengamankan paket data atau informasi dengan cara menambahkan beberapa *header* pada data yang melewati *VPN*. Selanjutnya data yang telah terenkripsi dengan *header VPN* dikirimkan melalui jaringan tunnel dan akan didekr*IP*si kembali sebelum sampai lokasi tujuan. Implementasi *VPN* pada *router* sangat bervariasi, akan tetapi untuk dasarnya hampir semua sama. Untuk *VPN PPTP* dan *L2TP* sudah didukung oleh beberapa *brand router* didunia [2].

2.2.1 *Tunneling*

Tunneling merupakan inti dari teknologi *VPN*. Tunneling merupakan suatu teknik untuk melakukan enkapsulasi pada seluruh data pada suatu paket yang menggunakan suatu format protokol tertentu. Dengan kata lain, *header* dari suatu protokol *tunneling* ditambahkan pada *header* paket yang asli. Kemudian barulah paket tersebut dikirimkan ke dalam jaringan paket data. Ketika paket yang telah "ditunnel" dirutekan ke terminal tujuan. Paket-paket tersebut akan melewati suatu jalur logika yang dikenal dengan nama kanal. Ketika penerima menerima paket tersebut, maka akan dibuka dan dikembalikan lagi ke dalam format aslinya.

2.2.2 Jenis-Jenis Jaringan *VPN*

Ada 3 jenis jaringan *VPN* yang diimplementasikan antara lain :

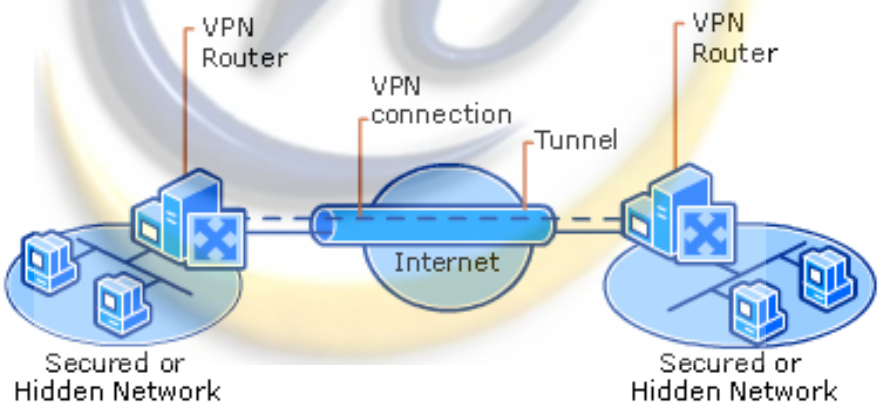
a. *Access VPN*, menyediakan *remote access* ke jaringan *intranet* atau *extranet* perusahaan yang memiliki kebijakan yang sama sebagai jaringan privat. *Access VPN* memungkinkan user untuk dapat mengakses data perusahaan kapanpun dimanapun dan bagaimanapun mereka mau.
b. *Inrarnet VPN* (*site to site VPN*), menghubungkan antara kantor pusat suatu perusahaan dengan kantor cabang, kantor pembantu melalui *shared network* menggunakan koneksi yang permanent (*dedicated*).
c. *Extranet VPN*, menghubungkan konsumen, *suppliers*, mitra bisnis atau berberapa komunitas dengan kepentingan yang sama ke jaringan intranet

perusahaan melalui infrastruktur yang terbagi menggunakan koneksi permanent (dedicated).

2.2.3 Topologi *VPN*

Koneksi *VPN* menghubungkan suatu jaringan dengan dengan jaringan lain melalui jaringan publik dan menc*IP*takan komunikasi yang aman. Jalur *VPN* akan diarahkan melalui *Internet* yang *dedicated*. Ketika jaringan *VPN* terhubung melalui *Internet*, *router* yang bertindak sebagai *server* akan meneruskan paket ke *router client* lain di koneksi *VPN* dan begitu juga sebaliknya. *VPN* juga disini juga beroprasi menjadi *data link* pada *layer 2* [1].

Topologi VPN sebagai layout dasar koneksi jaringan pada masing-masing site yang berbeda. *VPN server* menyediakan fasilitas *remote system* untuk *client router* yang ingin terhubung dengan *VPN server*. *VPN Client* yang terkoneksi akan melakukan proses otorisasi dan otentifikasi pada *router VPN Server* [13].



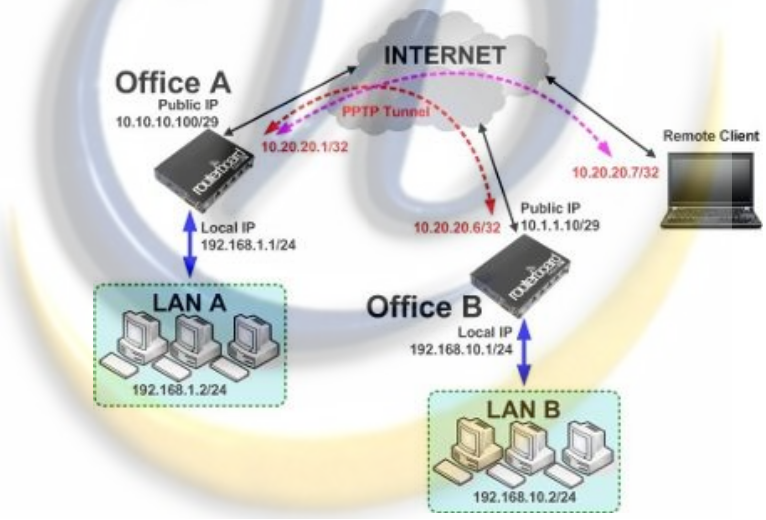
Gambar 2.2 Topologi VPN [13]

2.2.4 *VPN Point to Point Tunneling Protocol (PPTP)*

PPTP merupakan Salah satu *service* yang biasa digunakan untuk membangun sebuah jaringan *VPN* adalah *Point to Point Tunnel Protocol (PPTP).* Sebuah koneksi *PPTP* terdiri dari *Server* dan *Client* [1]. *MikroTik RouterOS* bisa difungsikan baik sebagai *server* maupun *client* atau bahkan diaktifkan keduanya bersama dalam satu

mesin yang sama. *Feature* ini sudah termasuk dalam *package PPP* sehingga anda perlu cek di menu *system package* apakah paket tersebut sudah ada di *router* atau belum. Fungsi *PPTP Client* juga sudah ada di hampir semua *OS*, sehingga kita bisa menggunakan *Laptop/PC* sebagai *PPTP Client* [1].

Biasanya *PPTP* ini digunakan untuk jaringan yang sudah melewati *multihop router (Routed Network).* Jika anda ingin menggunakan *PPTP* pastikan di *router* anda tidak ada rule yang melakukan *blocking* pada protocol TCP 1723 dan *IP Protocol 47/GRE* karena *service PPTP* menggunakan *protocol* tersebut [3]. Pada *PPTP*, *protokol point-to-point (PPP)* yang dibungkus dalam protokol *TCP / IP*, yang menyediakan koneksi *internet*. Oleh karena itu, mesk*IP*un sambungan dibuat melalui *Internet*, koneksi *PPTP* meniru hubungan langsung antara dua lokasi, yang memungkinkan untuk koneksi yang aman [1].



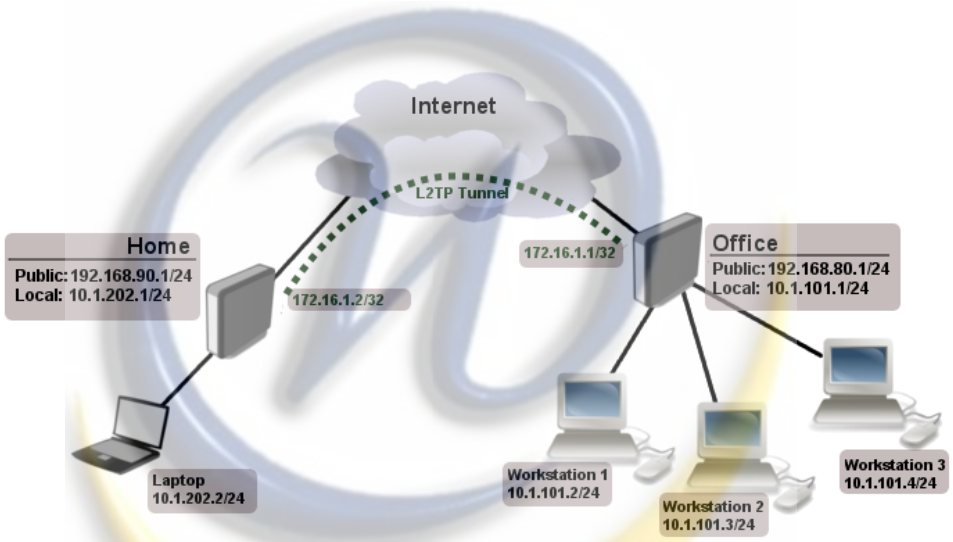
Gambar 2.3 Topologi VPN PPTP [mikrotik.co.id]

2.2.5 *VPN Layer 2 Tunnel Protocol* (*L2TP*)

L2TP merupakan pengembangan dari *PPTP* ditambah *L2F*. *Network security Protocol* dan enkripsi yang digunakan untuk autentikasi sama dengan *PPTP*. Akan tetapi untuk melakukan komunikasi, *L2TP* menggunakan *UDP port 1701*. *L2TP* merupakan *protocol* yang tinnggi dengan standar dalam *IETF RFC 3193*. *L2TP* biasanya digunakan dalam membuat *Virtual Private Dial Network (VPDN)* yang dapat bekerja membawa semua jenis protokol komunikasi di dalamnya. *L2TP*

memungkinkan penggunanya untuk tetap dapat terkoneksi dengan jaringan lokal milik mereka dengan *policy* keamanan yang sama dan dari manapun mereka berada, melalui koneksi *VPN* atau *VPDN*. Koneksi ini sering kali dianggap sebagai sarana memperpanjang jaringan lokal milik penggunanya, namun melalui media *public* [1].

Protokol ini menawarkan kapabilitas yang sangat tinggi antar vendor komputer dan jaringan komputer yang bahkan tidak dimiliki oleh protokol *tunneling* lainnya. *Protocol L2TP* juga sering juga disebut sebagai protokol *dial-up-virtual*, karena *L2TP* memperluas suatu *session Point to Point Protocol (PPP) dial-up* melalui jaringan publik *internet* [2].



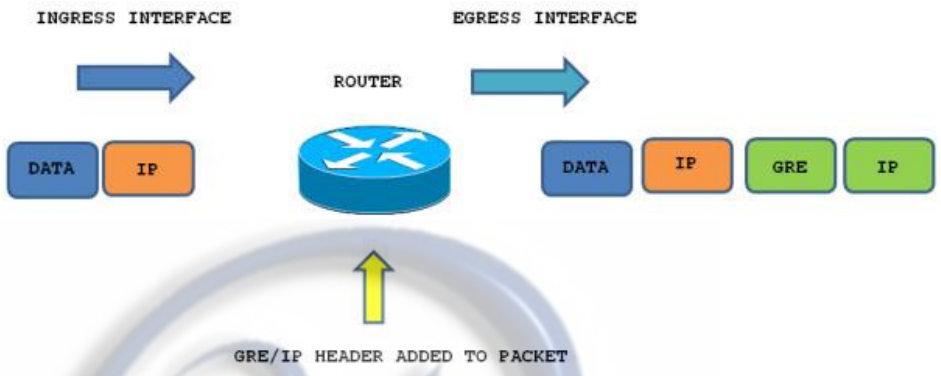
Gambar 2.4 Topologi VPN L2TP [mikrotik.co.id]

2.2.6 Generic Routing Encapsulation (GRE)

Protokol *tunneling* mampu membawa lebih dari satu jenis protokol pada layer 3 melalui jaringan *IP*. Pengalamatan komunikasi *GRE* dapat membuat koneksi *point to point* serperti halnya *VPN*. *GRE* bekerja dengan enkapsulasi payload pada *header packet*. Salah satu *GRE tunnel endpoint* mengirimkan *payload* melalui *tunnel* dengan paket *routing* yang dienkapsulasi melalui jaringan *IP*. Setelah mencapai *endpoint* yang lain *enkapsulasi GRE* akan di hapus pada *payload* yang akan diteruskan ke tujuan [2].

Router yang mendukung protocol GRE pada masing-masing *tunnel akan* melakukan enkapsulasi paket-paket protokol lain di dalam *header* dari protokol *IP*. Hal ini akan membuat paket-paket tadi dapat dibawa kemanapun. Dengan adanya

kemampuan ini, maka protokol-protokol yang dibawa oleh paket *IP* tersebut dapat lebih bebas bergerak kemanapun lokasi yang dituju, asalkan terjangkau secara pengalamatan *IP*. *GRE* banyak digunakan untuk memperpanjang dan mengekspansi jaringan lokal yang dimiliki suatu perusahaan [3].



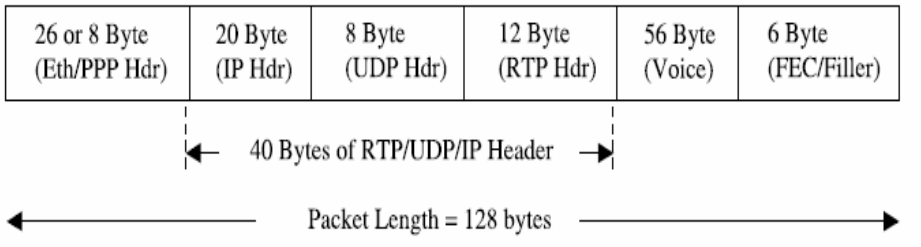
Gambar 2.5 Proses penambahan Header protocol GRE

2.3 *Voice Over Internet Protocol* (*VoIP*)

VoIP (*Voice over Internet Protocol*) dedefinisikan sebagai suatu sistem yang menggunakan jaringan internet untuk mengirimkan paket data suara dari suatu tempat ke tempat lainnya menggunakan perantara protokol *IP*. *VoIP* mentransmisikan sinyal suara dengan mengubahnya ke dalam bentuk digital, dan dikelompokkan menjadi paket–paket data yang dikirim dengan menggunakan platform *IP* (*Internet Protocol*). *VoIP* adalah teknologi yang mampu melewatkan "panggilan suara", *video*, dan data melalui jaringan *IP*. Bentuk panggilan analog dikonversikan menjadi bentuk digital dan dijalankan sebagai data *oleh internet protocol (IP).*

Ada berapa komponen yang dibutuhkan dalam membuat jaringan *VoIP*, [4] yaitu:

a. *Codecs.*
b. *TCP/IP* dan *VoIP protocols*.
c. *IP telephony servers* dan *PBXs*.
d. *VoIP gateways* dan *layer 3 routers.*
e. *IP phones* atau *softphones.*



Gambar 2.6 Contoh protokol stack VoIP [4]

2.3.1 Kualitas *VoIP*

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

a. Codec

Sebuah *codec* (yang merupakan kepanjangan dari *compressor/decompresser* atau *coder/decoder*) adalah suatu *hardware* atau *software* yang melakukan *sampling* pada sinyal suara analog, kemudian mengkonversi ke dalam bit – bit digital dan mengeluarkannya. Beberapa jenis *codec* melakukan kompresi agar dapat menghemat *bandwidth*. Dibawah ini terdapat beberapa jenis *codec* yang digunakan secara umum, bersama dengan data rate dan *Delay* paket[4].

Tabel 2.1 Jenis Codec

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

B. *Frame Loss*

Dalam jaringan *internet*, setiap paket yang dikirim ke penerima belum tentu sampai, hal ini disebabkan karena sifat jaringan *internet* yaitu *best effort*, dimana jaringan hanya berusaha menjaga agar paket sampai di penerima. Jika paket lama

berputar di jaringan *(looping)*, paket itu akan dibuang. Begitu pula dengan *QoS*. Jaringan internet tidak menyediakan *QoS* bagi penggunanya. Hal itu yang menyebabkan *VoIP* menggunakan protokol *RTP*. Salah satu kelemahan protokol ini adalah tidak adanya transmisi ulang jika paket yang dikirim hilang[4].

C. *Delay*

Delay merupakan faktor yang penting dalam menentukan kualitas *VoIP*. Semakin besar *Delay* yang terjadi maka akan semakin rendah kualitas *VoIP* yang dihasilkan[4]. Adapun beberapa sumber *Delay* antara lain:

a. *Delay Algoritmic*, Merupakan *Delay* yang dihasilkan oleh *Codec* dalam melakukan pengkodean.
b. *Delay Paketisasi*, *Delay* paketisasi merupakan *Delay* yang terjadi antara satu paket *RTP* dengan paket lainnya. *RFC* 1890 menspesifikasikan bahwa *Delay* sebaiknya 20 ms.
c. *Delay* Serialisasi, Merupakan waktu yang dibutuhkan untuk mengirimkan paket dari pengirim. *Delay* serialisasi juga terjadi pada *router* atau switch.
d. *Delay* Propagasi, Merupakan waktu yang diperlukanuntuk sinyal listrik atau sinyal optik untuk melewati media transmisi dari pengirim ke penerima. *Delay* ini tergantung dari media transmisi dan jarak yang harus ditempuh sinyal.
e. *Delay* komponen, Merupakan *Delay* yang terjadi yang disebabkan proses yang terjadi pada komponen contohnya *router*. Dalam *router* terdapat *Delay* yang terjadi jika paket masuk dan d*IP*roses kemudian dikeluarkan ke *port output*.

Agar tidak terjadi suara *overlap* dalam komunikasi *VoIP*, G.114 :

a. Menyediakan pedoman untuk batasan *Delay* satu arah :
b. 0 s/d 150 ms *Delay* ini masih dapat diterima.
c. 150 s/d 400 ms masih dapat diterima jika *administrator* waspada.
d. Pada waktu yang dapat mempengaruhi.
e. Kualitas transmisi.
f. Diatas 400 ms tidak dapat diterima untuk layanan *VoIP*.

2.3.2 Session Initation Protokol (S*IP*)

Session Initiation Protocol atau SIP merupakan standar *IETF* untuk suara atau layanan multimedia melalui jaringan *internet*. *SIP (RFC 2543)* di ajukan pada tahun 1999. Penc*IP*ta standar ini adalah Henning Schulzrinne. *SIP* merupakan protokol *layer* aplikasi yang digunakan untuk manajemen pengaturan panggilan dan pemutusan panggilan. *SIP* digunakan bersamaan dengan protokol *IETF* lain seperti *SAP, SDP, MGCP (MEGACO)* untuk menyediakan layanan *VoIP* yang lebih luas. Arsitektur *SIP* mir*IP* dengan arsitektur *HTTP* (*protocol client - server*). Arsitekturnya terdiri dari request yang dikirim dari user *SIP* ke *server SIP*. *Server* itu memproses request yang masuk dan memberikan respon kepada *client*. Permintaan request itu, bersama dengan komponen respon pesan yang lain membuat suatu komunikasi *SIP*. Arsitektur *SIP* terdiri dari dua buah komponen seperti tertera dibawah ini [5].

A. User Agent

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

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

B. *SIP Server*

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

a. *Registration Server*: *server* ini menerima request registrasi dari *user SIP* dan melakukan *update* pada lokasi *user* dengan *server* ini.
b. *Proxy Server*: *server* ini menerima request *SIP* dan meneruskan ke *server* yang dituju yang memiliki informasi tentang user yang d*IP*anggil.
c. *Redirect Server*: *server* ini setelah menerima request S*IP*, menentukan *server* yang dituju selanjutnya dan mengembalikan alamat *server* yang dituju

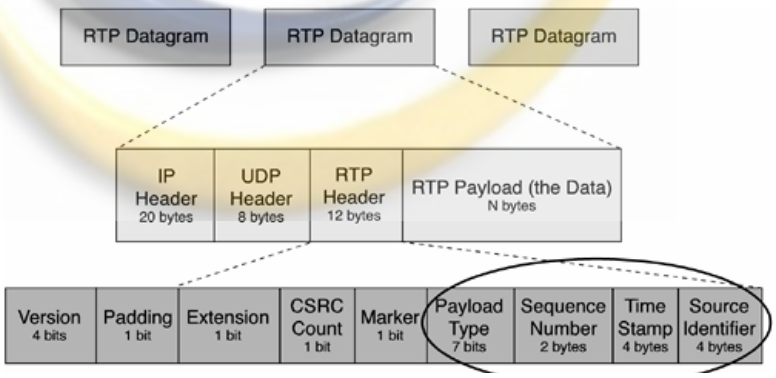
selanjutnya kepada *client* daripada meneruskan request ke *server* yang dituju tersebut[5].

2.3.3 Protokol *VoIP*

Voice over IP telah diimplementasikan dalam berbagai macam bentuk yang menggunakan hak milik dan standar serta protokol terbuka. Contoh protokol jaringan yang digunakan untuk mengimplementasikan *VoIP* meliputi:

a. *Real – time Transport Protocol* (*RTP*)

RTP digunakan untuk mengirim paket *real-time*, seperti *audio* dan *video*, melalui jaringan paket switch. Protokol ini digunakan oleh kedua protocol, *SIP* dan *H.323*. Protokol menyediakan informasi waktu kepada penerima sehingga dapat memberbaiki *Delay* karena *jitter*. *RTP* juga memungkinkan penerima untuk mendeteksi paket yan g hilang. *Header* dari *RTP* berisikan informasi yang membantu penerima agar dapat merekonstruksi ulang paket dan juga informasi mengenai bagaimana bit – bit dibagi menjadi paket oleh *codec*. *RTP* menyediakan informasi yang cukup kepada penerima sehingga sehingga dapat d*IP*ulihkan kembali bahkan jika terjadi kehilangan paket atau *jitter*[5].



Gambar 2.7 Header dengan informasi Payload dan Sequence[5]

b. Real Time Control Protocol (RTCP)

RTCP adalah suatu protocol control yang bekerja berhubungan dengan *RTP*. Dalam sebuah sesi *RTP*, terminal akhir (*endpoint*) secara periodik mengirimkan paket *RTCP* untuk menyebarkan informasi yang bermanfaat mengenai *QoS*. Terminal akhir

ini dapat mengetahui takaran yang tepat unuk mengirimkan paket dengan lebih efisien melalui sesi *RTP*. Secara umum fungsi dari *RTCP* adalah untuk menyediakan *feedback QoS*, kontrol sesi, identifikasi user, dan sinkronisasi inter media, yaitu mensinkronkan antara *audio* dan *video* seperti mensinkronkan antara gerakan bibir pada *video* dengan suara yang keluar pada *audio*.

c. *Real – time Streaming Protocol (RTSP)*

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

2.4 *Remote Access*

Remote Access adalah kemampuan untuk mendapatkan akses ke komputer atau jaringan dari jarak jauh. Di perusahaan, orang-orang di kantor cabang, *telecommuters*, dan orang-orang yang bepergian mungkin perlu akses ke jaringan korporasi. Pengguna rumah mendapatkan akses ke *internet* melalui akses *remote* ke penyedia layanan *Internet (ISP)*. Koneksi *dial-up* melalui *desktop*, *notebook*, atau modem komputer genggam yang melalui saluran telepon biasa, merupakan metode umum dari akses *remote*.

Remote Access juga mungkin menggunakan *dedicated line* antara komputer atau jaringan area lokal jarak jauh dan jaringan area lokal perusahaan "pusat" atau utama. Sebuah *dedicated line* adalah lebih mahal dan kurang fleksibel tetapi menawarkan kecepatan data yang lebih cepat [8]. Beberapa fungsi *remote access* diantaranya:

a. Mengendalikan komputer lain untuk mengakses *software* di komputer yang ada dibagian tertentu.
b. Mematikan komputer dari jarak jauh.
c. Menghidupkan ulang/*restart* komputer dari jarak jauh.
d. Memodifikasi *setting registry* komputer lain dari jarak jauh.
e. Mengawasi penggunaan komputer lain dari jarak jauh.

f. Mengawasi penggunaan program berjalan / *internet* dari jarak jauh.
g. Pemeliharaan (*maintenance*) komputer dari jarak jauh.
h. *Sharing resource* dari jarak jauh.

2.5 Quality of Services (QoS)

Quality of Service adalah kemampuan suatu jaringan untuk menyediakan layanan yang lebih baik pada trafik data tertentu pada berbagai jenis *platform* teknologi. *QoS* tidak d*IP*eroleh langsung dari infrastruktur yang ada, melainkan d*IP*eroleh dengan mengimplementasikannya pada jaringan yang bersangkutan. Tujuan akhir dari *QoS* adalah memberikan *network service* yang lebih baik dan terencana dengan *dedicated bandwidth*, *jitter* dan *Delay* yang terkontrol. Beberapa parameter yang menyatakan *QoS* antara lain *jitter*, *Delay*, *Packet Loss*, dan *Throughput* [2].

a. *Throughput*, adalah ukuran dari berapa banyak jumlah *bit* informasi sistem yang dapat memproses dalam jumlah waktu tertentu.

$$Throughput = \frac{Data\ received}{Beetween\ first\ and\ last\ packet} * \left(\frac{8}{1000}\right) \quad [2]$$

b. *Delay*, adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. Menurut, rumus dan toleransi nilai *Delay* adalah sebagai berikut.

$$Delay = \frac{Waktu\ jeda\ antara\ paket\ pertama\ dan\ terakhir}{Total\ paket\ yang\ diterima} \quad [2]$$

Tabel 2.2 Standar Delay international telecomunication union ITU-T G.114[3]

Waktu Tunda (ms)	Kualitas
0-150ms	Baik
150-400ms	Cukup
>400ms	Buruk

c. *Jitter*, adalah variasi waktu dari sinyal periodik dalam elektronik dan telekomunikasi, sering kali dalam kaitannya dengan sumber jam. Berikut rumus penentun *Delay* [2].

$$jitter = \frac{Total\ Variasi\ Delay}{Total\ paket\ yang\ diterima - 1} \qquad [2]$$

$$Total\ Variasi\ delay = (delay\ 2 - delay\ 1) + (delay\ 3 - delay\ 2)$$
$$\ldots\ldots + (delay\ n - (delay\ n - 1) \qquad [2]$$

Tabel 2.3 Standar Jitter Telecommunications and Internet Protocol Harmonization Over Network (TIPHON) [8]

Variasi Waktu Tunda (ms)	Kualitas
0-20ms	Baik
20-50ms	Cukup
>50ms	Buruk

d. *Packet Loss*, Merupakan keadaan dimana suatu paket yang ditransmisikan gagal mencapai tujuan.

$$Packet\ Loss = \frac{Paket\ yang\ dikirim - paket\ yang\ diterima}{paket\ yang\ dikirim} * 100\%$$

Tabel 2.4 Standar Packet Loss Telecommunications and Internet Protocol Harmonization Over Network (TIPHON) [8]

Paket Hilang (%)	Kualitas
0 – 0,5%	Sangat baik
0,5%-1,5%	Baik
>1,5%	Buruk

2.6 *Network Analysis*

Analisis jaringan (*Network Analysis*) juga dikenal sebagai '*protocol analysis*' merupakan cara mendengarkan (*listening*) dalam komunikasi data dan jaringan, biasanya dilakukan untuk memastikan bagaimana peralatan-peralatan berkomunikasi dan menentukan kualitas dari jaringan tersebut [2].

Beberapa tugas yang dilakukan selama sesi analisis jaringan adalah sebagai berikut :

a. Menangkap trafik yang diinginkan

b. Melihat trafik yang telah ditangkap

c. Menyaring trafik yang diinginkand. Hasil dokumentasi temuan

2.7. *Wireshark*

Wireshark merupakan salah satu tools atau aplikasi "*Network Analyzer*" atau Penganalisa Jaringan. Penganalisaan Kinerja Jaringan itu dapat melingkupi berbagai hal, mulai dari proses menangkap paket-paket data atau informasi yang berlalu-lalang dalam jaringan, sampai pada digunakan pula untuk *sniffing* (memperoleh informasi penting seperti *password, username*, dll). *Wireshark* sendiri merupakan *open source tools* untuk untuk *Network Analyzer* yang ada saat ini. Kelebihan aplikasi *Network Analyzer Wireshark* antara lain :

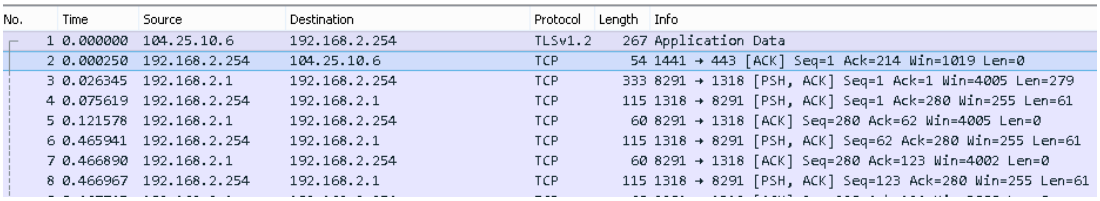
a. Menangkap dan *Capture* paket data secara langsung dari sebuah network interface.

b. Menampilkan informasi yang sangat detail mengenai hasil capture tersebut.

c. Memiliki kemampuan dan fitur yang baik serta lengkap.

Beberapa fungsi menu pada wireshark yang sering digunakam untung melakukan capture traffic jaringan yang dianalisis.

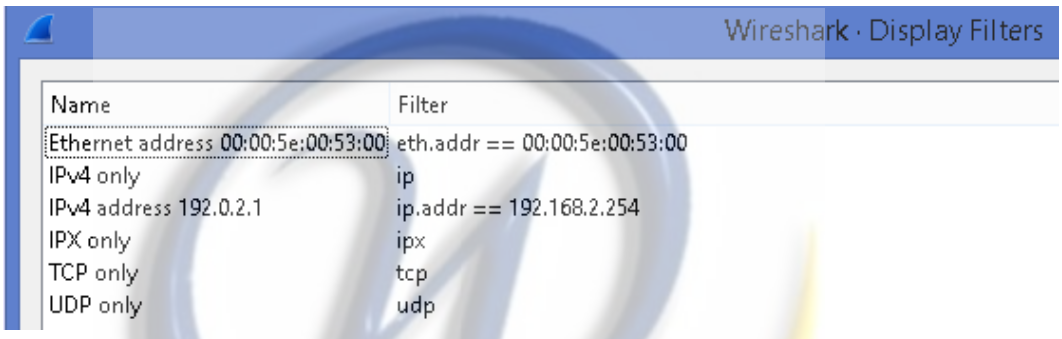
a. Tabel informasi dari paket yg di analisis , yang diperhitungkan adalah keterangan *time* yang merupakan variasi *Delay* waktu .



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	104.25.10.6	192.168.2.254	TLSv1.2	267	Application Data
2	0.000250	192.168.2.254	104.25.10.6	TCP	54	1441 → 443 [ACK] Seq=1 Ack=214 Win=1019 Len=0
3	0.026345	192.168.2.1	192.168.2.254	TCP	333	8291 → 1318 [PSH, ACK] Seq=1 Ack=1 Win=4005 Len=279
4	0.075619	192.168.2.254	192.168.2.1	TCP	115	1318 → 8291 [PSH, ACK] Seq=1 Ack=280 Win=255 Len=61
5	0.121578	192.168.2.1	192.168.2.254	TCP	60	8291 → 1318 [ACK] Seq=280 Ack=62 Win=4005 Len=0
6	0.465941	192.168.2.254	192.168.2.1	TCP	115	1318 → 8291 [PSH, ACK] Seq=62 Ack=280 Win=255 Len=61
7	0.466890	192.168.2.1	192.168.2.254	TCP	60	8291 → 1318 [ACK] Seq=280 Ack=123 Win=4002 Len=0
8	0.466967	192.168.2.254	192.168.2.1	TCP	115	1318 → 8291 [PSH, ACK] Seq=123 Ack=280 Win=255 Len=61

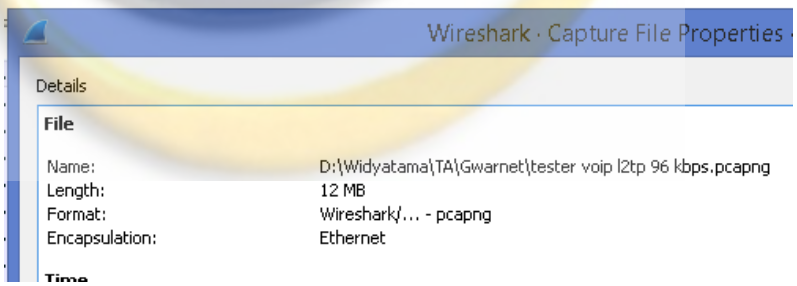
Gambar 2.8 Tabel Informasi Wirehark

b. Display filter, untuk memasukkan parameter filter seperti, jenis pritokol, alamat, tujuan informasi yang akan di analisis.



Gambar 2.9 DISPlay Filter Wireshark

d. Capture file propertis atau hasil summary , adalah kumpulan hasil analisis traffic pada wireshark



Gambar 2.10 hasil Summary Wireshark