# BAB 2
# DASAR TEORI

## 2.1 KAJIAN PUSTAKA

Pada tahun 2021, Iryani dan Andika [7] melakukan studi dengan permasalahan dibutuhkannya sistem jaringan *Wide Area Network* (WAN) berupa VPN yang *scalable* apabila mempunyai kebutuhan perangkat yang banyak antar lokasi yang berbeda namun apakah dengan jaringan VPN yang *scalable* mempunyai QoS yang mumpuni atau tidak. Penelitian ini dilakukan dengan menganalisis perfomansi proses pengiriman paket berkoneksi *Transmission Control Protocol* (TCP) antar *client* pada jaringan DMVPN dengan FR*Routing* IBGP dan EBGP yang diimplementasikan pada arsitektur *hub* dan *spoke*. Berdasarkan hasil pengujian menunjukkan hampir semua parameter tergolong dalam kategori yang sangat bagus terkecuali *delay*.

Pada tahun 2019, Ismah [3] melakukan studi dengan permasalahan manakah sistem jaringan *Wide Area Network* (WAN) dengan kinerja yang bagus terhadap layanan *file transfer* dan video *streaming* serta mampu untuk mengamankan proses pengiriman paket antara lokasi yang jauh berbeda. Penelitian ini mengujikan sistem jaringan *hub* dan *spoke* yaitu DMVPN *phase* 3 dengan sistem jaringan yang lain yaitu MPLS-L3VPN, dan VPLS. Skema untuk pengujian layanan *file transfer* berformat RAR dengan *size* 10 MB dan 20 MB. Skema untuk pengujian layanan video *streaming* berformat MP4 dan MKV dengan *size* 6 MB beresolusi 360p. Parameter *quality of service* meliputi nilai dari *throughput, delay,* dan *packet loss*. Hasil pengujian untuk skenario *file transfer* menghasilkan kinerja yang bagus pada sistem jaringan DMVPN *phase* 3 dan MPLS-L3VPN. Sementara hasil pengujian skenario video *streaming* menghasilkan kinerja yang bagus pada sistem jaringan MPLS-L3VPN dan VPLS.

Pada tahun 2018, Arvi [8] melakukan studi dengan permasalahan membandingkan sistem jaringan *Wide Area Network* (WAN) yang mempunyai kinerja yang mumpuni serta mampu untuk mengamankan proses transmisi paket

yang lokasinya berjauhan. Penelitian ini mengujikan sistem jaringan *hub* dan *spoke* yaitu DMVPN *phase* 3 dengan sistem jaringan yang lain yaitu MPLS-VPN. Kedua sistem jaringan tersebut diimplementasikan *routing protocol* OSPF-BGP. Skema untuk pengujian layanan *file transfer* berformat RAR dengan *size* 10 MB dan 20 MB. Skema untuk pengujian layanan video *streaming* berformat MP4 dan MKV dengan *size* 3 MB beresolusi 240p. Parameter *quality of service* meliputi nilai dari *throughput*, *delay*, dan *packet loss*. Hasil pengujian untuk skenario *file transfer* menghasilkan kinerja yang bagus pada sistem jaringan DMVPN phase 3. Sementara hasil pengujian skenario video *streaming* menghasilkan kinerja yang bagus pada sistem jaringan MPLS-VPN.

Pada tahun 2018, Rizal, dkk [9] melakukan studi dengan permasalahan dibutuhkannya jaringan dengan cakupan *Wide Area Network* (WAN) melalui internet namun diharapkan mempunyai perfomansi yang bagus. Penelitian ini menganalisis perfomansi berupa *quality of service* jaringan DMVPN *phase* 1, *phase* 2, *phase* 3 dengan *routing* protokol RIPv2, OSPF, EIGRP dengan BGP yang diimplementasikan pada arsitektur *hub* dan *spoke*. Penelitian ini menggunakan paket TCP dengan *window size* 2, 4, 8, 16, dan 32 KBytes dan pengujian paket UDP menggunkan *bandwidth* 2 Mbps dengan variasi waktu 20, 30, 40, 50, dan 60 detik. Berdasarkan pengujian parameter *throughput*, *jitter*, *packet loss*, dan *convergence time* menghasilkan nilai *Quality of Service* terbagus diperoleh DMVPN *phase* 1 dengan *routing* protokol EIGRP-BGP, DMVPN *phase* 2 dengan *routing* protokol EIGRP-BGP, dan DMVPN *phase* 3 dengan *routing* protokol RIPv2-BGP. Secara keseluruhan rata-rata *routing* protokol EIGRP-BGP merupakan kombinasi terbaik bagi DMVPN.

Pada tahun 2019, Mardianto [10] melakukan studi dengan permasalahan berkembangnya internet, ISP diharapkan mampu berkembang juga dengan mempunyai *quality of service* yang bagus. Penelitian ini dilakukan dengan membandingkan *quality of service* antara jaringan VPN-L2TP dengan MPLS-L3VPN. Besar *file* untuk pengujian yaitu 102 KB, 220 KB, 412 KB 1.16 MB, 2,08 MB, 4,18 MB, 8,07 MB dan 10.0 MB dengan *bandwidth* 128 Kbps, 256 Kbps, dan 512 Kbps. Hasil *delay, throughput, dan packet loss* berdasarkan pengujian kedua jaringan tersebut menghasilkan nilai *delay* yang sangat bagus.

Pada Tabel 2.1 merupakan rangkuman singkat mengenai penelitian sebelumnya yang berkaitan.

**Tabel 2.1 Penelitan yang berkaitan**

| No | Penulis dan Judul | Metode dan Proses | Kelebihan | Kekurangan |
|---|---|---|---|---|
| 1 | Iryani, Andika, (2021), Performansi *Dynamic Multipoint Virtual Private Networ*k pada *Routing Protocol* BGP dengan FR*Routing* | Metode Simulasi, Membandingkan kinerja FRRouting IBGP dan EBGP pada DMVPN untuk mendapatkan parameter *throughput*, *delay*, *jitter*, dan *packet loss* | Menggunakan arsitektur *hub* dan *spoke* | Skenario pengujian hanya menggunakan paket TCP tanpa dijelaskan aplikasi yang digunakan beserta objek yang di uji |
| 2 | Ismah, (2019), Evaluasi Kinerja Jaringan DMVPN *Phase* 3, MPLS-L3VPN, dan VPLS terhadap Layanan Video *Streaming* dan *File Transfer* | Metode Simulasi, Membandingkan kinerja jaringan DMVPN *Phase* 3, MPLS-L3VPN, dan VPLS unutk mendapatkan parameter *throughput*, *delay*, dan *packet loss* | Membandingkan kinerja sistem *hub* dan *spoke* dengan dua sistem yang lainnya, Skenario pengujian TCP dan UDP jelas beserta aplikasi yang digunakan | Besar *file* pengujian video *streaming* tergolong kecil, parameter QoS yang digunakan tidak melibatkan *jitter* |
| 3 | Arvi, (2018), Evaluasi Kinerja Jaringan DMVPN *Phase* 3 dan MPLS-VPN terhadap Layanan Video *Streaming* dan *File Transfer* | Metode Simulasi, Menggunakan *routing protocol* OSPF-BGP dalam membandingkan kinerja jaringan DMVPN *Phase* 3 dengan MPLS-VPN untuk mendapatkan nilai parameter *throughput*, *delay*, dan *packet loss* | Membandingkan kinerja sistem *hub* dan *spoke* dengan satu sistem yang lainnya, Skenario pengujian TCP dan UDP jelas beserta aplikasi yang digunakan | Besar *file* pengujian video *streaming* tergolong kecil, parameter QoS yang digunakan tidak melibatkan *jitter* |

| No | Penulis dan Judul | Metode dan Proses | Kelebihan | Kekurangan |
|---|---|---|---|---|
| 4 | Rizal, Arini, Masruroh, (2018), Evaluasi Kinerja Jaringan DMVPN Menggunakan *Routing Protocol* RIPv2, OSPF, EIGRP dengan BGP | Metode Simulasi, Membandingkan semua jenis DMVPN menggunakan *routing protocol* RIPv2, OSPF, EIGRP dengan BGP untuk mendapatkan parameter *throughput*, *jitter*, *packet loss*, dan *convergence time* | Menggunakan semua jenis mekanisme *hub* dan *spoke* secara lengkap untuk mengimplementasikan DMVPN | Hasil nilai parameter QoS yang didapatkan tidak jelas asal usulnya misalnya aplikasi yang digunakan untuk pengujian TCP maupun UDP, format yang digunakan, ukuran yang di uji coba |
| 5 | Mardianto, (2019), Analisis *Quality of Service* (QoS) pada Jaringan VPN dan MPLS VPN Menggunakan GNS3 | Metode Simulasi, Membandingkan kinerja MPLS VPN-L2TP dengan MPLS-L3VPN menggunakan topologi *full mesh* unutk mendapatkan parameter *throughput*, *delay*, dan *packet loss* | *Router* yang digunakan banyak, Topologi yang digunakan sama untuk membandingkan kedua sistem jaringan | Topologi jaringan tidak menggunakan sistem *hub* dan *spoke*, Skenario pengujian hanya *file transfer* (TCP) dengan *bandwidth* yang sangat kecil dan ukuran file yang kecil |
| 6 | Ardhandy, (2023), Analisis Kinerja Jaringan FlexVPN *over* MPLS Menggunakan Parameter *Quality of Service* | Metode Simulasi, Melakukan pengujian kinerja jaringan FlexVPN *over* MPLS terhadap layanan *file transfer* menggunakan transport protocol TCP dan UDP untuk mendapatkan nilai parameter *delay*, *jitter*, *packet loss*, dan *throughput* | Skenario sistem sangat jelas menggunakan arsitektur *hub* dan *spoke*, pengujian mencantumkan *transport protocol* dan aplikasi yang digunakan, dan besaran *file* yang diujikan relatif besar | Hanya menggunakan satu skenario sistem yang di ujikan |

## 2.2 DASAR TEORI

### 2.2.1 Jaringan Komputer

Jaringan komputer merupakan kumpulan dari beberapa komputer yang dapat saling terhubung antara satu dengan yang lainnya sehingga dapat saling berkomunikasi. Media kabel telepon, kabel *fiber* optik, sinyal *infrared*, serta *wireless* yang memanfaatkan gelombang radio merupakan contoh media yang dapat digunakan untuk menghubungkan beberapa komputer untuk saling terhubung. Pada saat komputer saling berkomunikasi, mereka dapat saling berkolaborasi dengan bermacam cara. Misalnya dengan cara membagi *resources* antara satu sama lain dan membagi *workload* dari suatu pekerjaan maupun bertukar informasi [11]. Terdapat 3 jenis jaringan komputer, antara lain: *Local Area Network* (LAN), *Metropolitan Area network* (MAN), *Wide Area Network* (WAN), dan Internet [12].

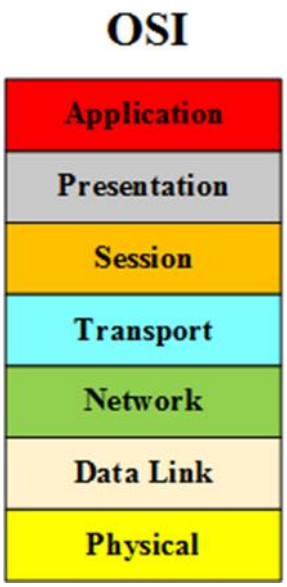
### 2.2.2 *Wide Area Network* (WAN)

*Wide Area Network* (WAN) merupakan jenis jaringan yang dapat menjangkau area geografis yang luas dengan menggunakan jaringan penyedia infrastruktur penyedia layanan untuk menghubungkan dua sites di lokasi yang jauh berbeda [1]. Perangkat router dalam jaringan *Wide Area Network* (WAN) digunakan untuk menentukan jalur paket yang dilalui antara komputer yang saling berkomunikasi [13]. Jaringan *Wide Area Network* (WAN) digunakan untuk menghubungkan dua atau lebih jaringan *Local Area Network* (LAN) sehingga dapat saling terhubung [1]. Memanfaatkan jaringan *Wide Area Network* (WAN) memungkinkan untuk melakukan proses pengiriman paket secara jarak jauh dimana saja selama terkoneksi dengan layanan yang ada pada jaringan *Wide Area Network* (WAN) [14].

### 2.2.3 Pemodelan Lapisan OSI

Lapisan OSI turut membantu dalam mengirimkan data antara *host* yang berbeda [15]. Model lapisan OSI mempunyai total 7 lapisan dengan dibagi menjadi 2 bagian yaitu lapisan bawah dan lapisan atas. Bagian lapisan bawah meliputi lapisan *physical, data link, network,* dan *transport* yang berfungsi bagaimana data dikirmkan dari ujung ke ujung. Sementara lapisan atas meliputi lapisan *session*,

*presentation*, dan *application* yang berfungsi bagaimana aplikasi yang berada di dalam dapat berkomunikasi antara satu dengan yang lainnya. Gambar 2.1 menunjukkan bagian dari lapisan OSI [16].




**Gambar 2.1 Lapisan OSI**

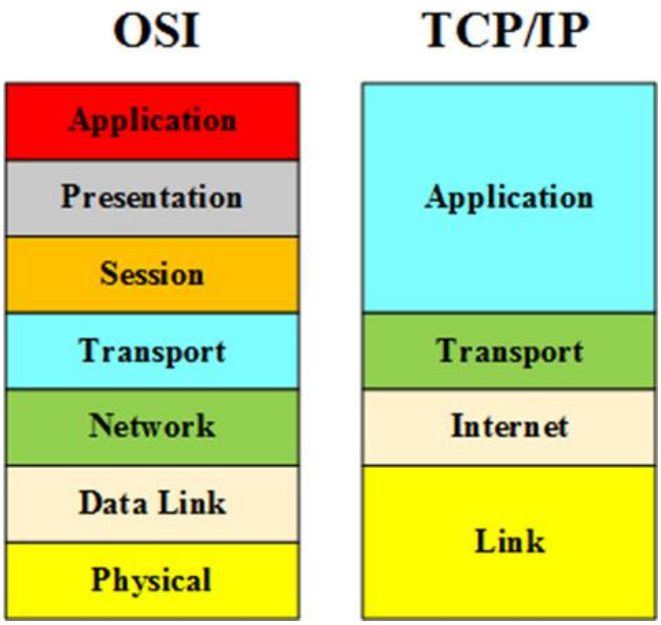
Berdasarkan Gambar 2.1 berikut adalah penjelesannya [11]. Pertama untuk bagian bawah dimulai dari lapisan *physical* yang berperan untuk media pengiriman dalam jaringan, topologi jaringan, apapun yang terkait dengan pensinyalan, serta sinkronisasi bit. Unit data pada lapisan *physical* disebut bit. Lapisan *data link* berperan untuk menentukan operasi *hardware* jaringan, mengoreksi kesalahan, *flow control*, serta pengalamatan pada *hardware*. Unit data pada lapisan *data link* disebut *frame*. Lapisan *network* berperan untuk mengartikan *address* komputer dalam jaringan, melakukan proses *routing*, serta membuat *header packet*. Unit data pada lapisan *network* disebut datagram. Lapisan *transport* berperan untuk memecah paket data ke dalam beberapa bagian dan dilakukan proses penomoran. Penomoran dapat mempermudah proses penyatuan kembali paket data yang terjadi pada pengirim ke penerima. Unit data pada lapisan *transport* disebut segmen.

Bagian atas dimulai dari lapisan *session* yang berperan untuk melakukan proses dalam hal pembuatan suatu koneksi, *maintenance* koneksi, serta penghancuran dan berakhirnya dari suatu koneksi. Lapisan presentation berperan untuk mengartikan data yang dikirimkan oleh aplikasi ke dalam format yang dapat

dikirimkan oleh jaringan komputer. Lapisan *application* berperan untuk mendeskripsikan terkait spesifikasi aplikasi supaya dapat berkomunikasi dalam jaringan komputer, pengaksesan jaringan, serta *interface* aplikasi dengan jaringan. Protokol pada lapisan application meliputi HTTP, FTP, dan lain sebagainya. Unit data pada lapisan *session*, *presentation*, dan *application* disebut data.

### 2.2.4 Lapisan *Transmission Control Protocol*/Internet *Protocol* (TCP/IP)

TCP/IP mempunyai peran dalam melindungi integritas data dengan mengibaratkan menjaga komunikasi dalam situasi kekacauan dalam perang [15]. Lapisan TCP/IP mempunyai 4 lapisan sedangkan lapisan OSI mempunyai 7 lapisan. Lapisan TCP/IP merupakan versi ringkasnya dibandingkan dengan lapisan OSI. Gambar 2.2 menunjukkan perbandingan antara lapisan TCP/IP dengan lapisan OSI [16].




**Gambar 2.2 Perbandingan lapisan OSI dan lapisan TCP/IP**

Berdasarkan Gambar 2.2 berikut adalah penjelesannya [11]. Lapisan *Link* pada TCP/IP mencakup lapisan *physical* dan *data link* pada lapisan OSI. Lapisan *link* berfungsi untuk mendeskripsikan apapun yang digunakan pada jaringan komputer misalnya protokol, topologi, *interface*, dan lain sebagainya untuk menjalankan komunikasi. Unit data pada lapisan link disebut *frame* yang meliputi

*frame header*, *frame data*, dan *frame footer*.

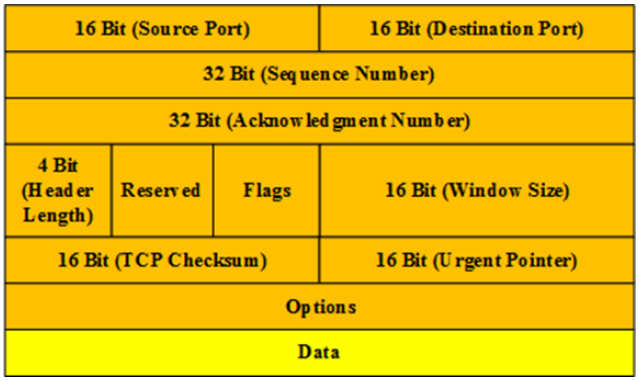
Lapisan internet pada TCP/IP mencakup lapisan *network* pada lapisan OSI. Lapisan internet berfungsi untuk peralihan datagram pada jaringan. Menyiapkan *interface* jaringan, pengalamatan, dan *routing* merupakan contoh fungsi dari lapisan internet. IP *header* dan IP data merupakan bagian dari lapisan internet.

Lapisan *transport* pada TCP/IP mencakup lapisan *transport* pada lapisan OSI. Lapisan *transport* berfungsi untuk mempersiapkan proses konektivitas, *channel* pergantian data untuk aplikasi, serta pengiriman *end to end message* dengan protokol TCP dan UDP.

Lapisan *application* pada TCP/IP mencakup lapisan *session*, *presentation*, dan *application* pada lapisan OSI. Lapisan *application* berfungsi untuk menjalin komunikasi data yang mencakupi antar aplikasi dan komputer. FTP, HTTP, SMTP merupakan contoh protokol jaringan yang ada pada lapisan *application*.

### 2.2.5 *Transmission Control Protocol* (TCP)

Protokol TCP merupakan suatu protokol untuk mentransimisakan data dengan menyediakan jalur data pada jaringan komputer misalnya internet yang dapat menyesuaikan dengan *tools* jaringan dengan berbagai permasalahannya. TCP mempunyai kelebihan dapat menjamin data karena bersifat *connection oriented* dalam mentransimisikan data dengan memanfaatkan *Automatic Repeat Request* (ARQ). Kelebihan tersebut akan secara otomatis mengirimkan ulang data apabila terjadi permasalahan pada saat mengirimkan data tersebut berdasarkan informasi gagalnya berupa *Acknowledgement* (ACK) dari pihak penerima [17]. Gambar 2.3 menunjukkan struktur TCP *header* [11].




**Gambar 2.3 Struktur TCP *header***

Berdasarkan Gambar 2.3 berikut adalah penjelasannya [9] [11].

1) *Source Port* (16 Bit)

   Memberitahukan modul yang melibatkan aplikasi untuk mengeluarkan serta mengirim TCP segmen *payload*.

2) *Destination Port* (16 Bit)

   Memberitahukan modul yang melibatkan aplikasi untuk menerima segmen *payload* dari TCP.

3) *Sequence Number* (32 Bit)

   Memberitahukan nomor *sequence* pada segmen *payload* TCP dengan penerima boleh menentukan apakah segmen *payload* bakal diterima berkali-kali atau hilang.

4) *Acknowledgment Number* (32 Bit)

   Nilai dari *acknowledgment number* merupakan oktet dari TCP.

5) *Header Length* (4 Bit)

   Memberitahukan terkait panjang dari TCP *segment header*. *Header* dapat berisikan opsi dengan banyak variasi asalkan panjang *header* mempunyai kelipatan 4.

6) *Flag* (6 Bit)

   Dari 6 bit tersebut setiap bitnya mempunyai nama masing-masing. Misal contoh *flag* yaitu FIN, URG, RST, PSH, SYN, dan ACK.

7) *Window Size* (16 Bit)

   Ukuran maksimum dari suatu data yang dapat ditransmisikan tanpa *acknowledge* dengan semakin kecil nilai *window size* maka dapat memperlambat *transfer* yang dihasilkan. Hal tersebut dikarenakan berlimpahnya total pecahan paket data yang perlu di *acknowledge*.

8) TCP *Checksum* (16 Bit)

   Melakukan pengecekan apabila terjadi kesalahan dalam *header* maupun data.

9) *Urgent Pointer* (16 Bit)

   Apabila *urgent pointer* bit kodenya telah diatur pada suatu wilayah/lokasi yang valid. Hal tersebut menandakan *urgent pointer* dalam kondisi *offset* berdasarkan *sequence number* yang ada dalam oktet, dimana segmen dari

*non-urgent* data tersebut dapat dimulai.

10) *Option*

    Secara umum nilai *option* adalah 0 yang mengisyaratkan tidak ada yang dicantumkan. Namun, bisa juga mempunyai nilai 32 bit dengan *padding* 0s harus dicantumkan yang fungsinya untuk memastikan data dimulai pada batas 32 bit yang berupa kata-kata.

11) Data

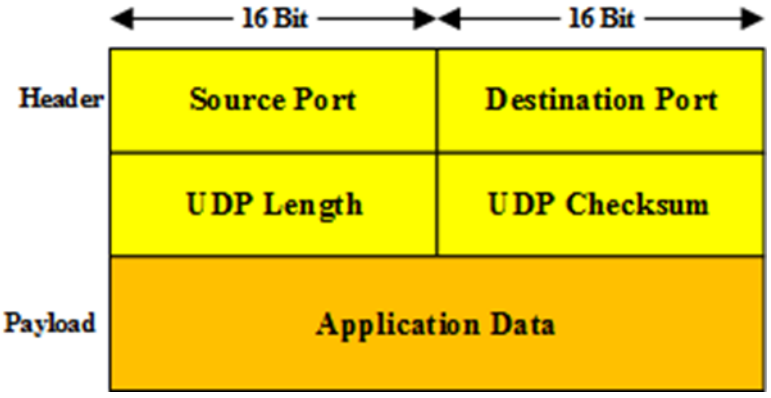
    Data yang diperoleh dari lapisan yang ada diatasnya yaitu lapisan *transport*.

### 2.2.6 *File Transfer Protocol* (FTP)

*File Transfer Protocol* (FTP) merupakan standarisasi yang digunakan untuk mengirimkan file antara dua komputer yang saling berkomunikasi dalam jaringan dengan menggunakan *transport protocol* TCP. Sesuai dengan konsep TCP maka FTP juga menerapkan konsep *connection oriented* dengan mekanisme *client-server* sehingga memberikan kehandalan yang lebih. Aplikasi untuk menerapkan FTP dilengkapi dengan sistem masuk yang jelas misalnya *username*, kata sandi, *port*, dan alamat IP *host* dari *server* [8].

### 2.2.7 *User Datagram Protocol* (UDP)

Protokol UDP berbeda dengan TCP, hal tersebut karena UDP merupakan protokol yang bersifat *connectionless* dalam mengirimkan data dan tidak akan ada pengecekan saat terjadi *error* dalam pengiriman data. UDP adalah komunikasi dalam jaringan yang memberikan cara cepat dan sederhana untuk mengirimkan data antara perangkat. Mengandung IP dengan menambahkan *header* singkat merupakan dasar dari protokol UDP. Protokol UDP tidak menyediakan mekanisme otomatis untuk memastikan pengiriman data atau urutan pengiriman. UDP tidak menyediakan mekanisme otomatis untuk memastikan pengiriman data atau urutan pengiriman. Protokol UDP tidak mengenal *data flow*, serta permasalahan atau kesalahan dalam hal pengiriman ulang karena UDP hanya menyediakan *interface* ke protokol IP sehingga sangat berguna dalam penggunaan *client-server* [17]. Gambar 2.4 menunjukkan struktur UDP *datagram* [16].




**Gambar 2.4 Struktur UDP datagram**

Berdasarkan Gambar 2.4 bahwa struktur UDP datagram khususnya *source port* dan *destination port* yang terdapat pada bagian *header* sama dengan apa yang ada pada *header* TCP. Namun header pada UDP tidak mempunyai *sequence number* seperti pada TCP. Hal tersebut karena UDP bersifat tanpa koneksi atau *connectionless* yang artinya UDP tidak dilakukan adanya klasifikasi ke dalam paket data ataupun paket kontrol sehingga tidak memerlukan proses *acknowledgment* dan *retransmission*. Lapisan aplikasi digunakan pada modul UDP pengirim untuk mengirimkan UDP datagram ke modul IP. Penerima melakukan proses ekstrak pada *payload* dari UDP datagram dan mengirimkan *payload* tersebut ke modul aplikasi sesuai *destination port* yang bersangkutan. UDP tidak handal terkait permasalahan *delay*, *packet loss*, dan permasalahan lainnya karena semua proses diatur oleh lapisan aplikasi. Lapisan aplikasi tidak menyediakan *acknowledgment* dan *retransmission* [11].

### 2.2.8 Jaringan MPLS

Jaringan *Multi-Protocol Label Switching* (MPLS) adalah sebuah sistem yang menerapkan konsep penyampaian paket data dengan mengimplementasikan beberapa karakteristik dari jaringan *circuit switched* dengan jaringan *packet-switched*. Konsep dari MPLS pada mekanisme *switching* yaitu menerapkan label pada paket. Mekanisme tersebut digunakan untuk meneruskan paket. Label pada mekanisme MPLS berperan pada ujung jaringan, dan mekanisme penerusan pada MPLS hanya tergantung dari label tersebut [16].

Mekanisme *forwarding* protokol selain TCP/IP memang diperuntukkan untuk jaringan MPLS. Misalnya dalam jaringan yang lebih luas, hasil dari pelabelan

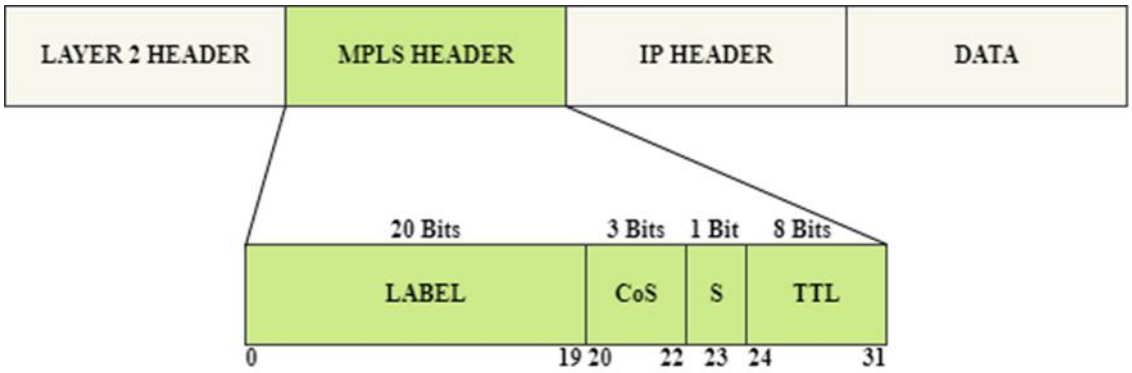
pada MPLS menunjukkan hanya *router edge* yang melakukan pencarian proses *routing*. Semua router inti akan meneruskan paket yang sesuai dengan labelnya dan membuat penerusan paket berdasarkan ISP yang membuat jauh lebih cepat. Jaringan MPLS dibuat untuk memperuntukkan *forwading* protokol selain TCP/IP [11].

### 2.2.9 Aristektur MPLS

MPLS (*Multiprotocol Label Switching*) adalah sebuah arsitektur dalam jaringan komunikasi yang digunakan untuk mengarahkan dan mengelola lalu lintas data melalui jaringan dengan lebih efisien dan dapat diatur. Mekanisme dari arsitektur MPLS yaitu dengan melakukan label *switching*, dimana menggabungkan kecepatan dari paket *forwarding* berdasarkan *layer* 2 *switching* dengan manfaat dari *layer* 3 *routing*.

Label pada MPLS digunakan paket untuk transportasi di seluruh jaringan berbasis paket. Manfaat tersebut memeberikan pengurangan beban kerja pada IP *routers* di layer 3 [18]. Bertukar label merupakan mekanisme *forwarding* pada seluruh jaringan. Unit data akan membawa paket dan *fixed-length* label yang menginformasikan *switching node* sepanjang jalur paket terkait memproses dan meneruskan data [16].

MPLS berada diantara *layer* 2 dan *layer* 3. MPLS merupakan suatu mekanisme yang menerapkan konsep *forwarding* berupa meneruskan paket data yang melewati suatu jaringan dengan menggunakan informasi dalam bentuk label yang dilekatkan pada paket IP [11]. MPLS ditujukan untuk penggunaan *private network* dengan aman dan cepat untuk bertukar informasi antara lokasi yang jauh berbeda [19]. Setiap paket IP diterapkan header MPLS berupa label yang menginformasikan paket serta rute berdasarkan prioritas. Setiap paket IP dalam *router* pertama yang dilewati paket IP menerapkan *header* MPLS guna mengambil keputusan terkait pengiriman paket IP bagi *router* lain [16]. MPLS telah menjadi salah satu teknologi kunci dalam pengembangan jaringan *modern* dan telah digunakan secara luas dalam lingkungan ISP, perusahaan besar, dan penyedia layanan jaringan lainnya. MPLS mempunyai *Header* yang dapat ditunjukkan pada Gambar 2.1 [11]:




**Gambar 2.5 Header MPLS**

Berdasarkan Gambar 2.5 berikut adalah penjelasan terkait 32 bits yang membentuk *header* MPLS [20]:

1) Suatu bidang label yang berisikan nilai nyata dari MPLS Label terletak pada 20 bits pertama.
2) *Field Class of Service* (CoS) digunakan untuk mempengaruhi antrian paket data dan algoritma paket data yang tidak diperlukan terletak pada 3 bits berikutnya.
3) *Bottom of stack flag* terletak pada *1 bits berikutnya*. *Header* MPLS sedang melakukan kontak dengan *header* protokol berikutnya yaitu IP header apabila di atur ke 1 bits header MPLS tidak memiliki area apapun, apabila diatur ke 0 bits *header* terakhir sebelum *payload* MPLS merupakan fungsi dari *Bit bottom of stack flag*.
4) *Time to Live* MPLS terletak pada 8 bits berikutnya. TTL MPLS berfungsi dalam penerapan mekanisme untuk membuang paket dalam suatu proses *forwarding loop*.

### 2.2.11 Prinsip Kerja MPLS

Teknik pelabelan pada setiap paket yang dikirim melalui jaringan merupakan konsep jaringan MPLS. Bekerja dengan memberi label kepada paket-paket data merupakan cara kerja dari MPLS. Menentukan rute atau prioritas digunakan MPLS untuk proses pengiriman paket. Informasi penting terkait informasi *routing* dari suatu paket berasal dari pemberian label. Tujuan paket dan

prioritas paket mana yang akan dikirimkan terlebih dahulu merupakan contoh dari informasi yang diperoleh [3].

Menyisipkan label diantara *header layer* 2 dan *layer* 3 pada paket merupakan cara kerja dari MPLS [21]. Menggunakan *layer* 2 dengan memanfaatkan kecepatan *switching* yang digabungkan dengan *layer* 3 yaitu kemampuan routing dan skalabilitas merupakan prinsip kerja jaringan MPLS [21]. Prinsip tersebut diharapkan mampu menjadi solusi jaringan yang dapat menangani masalah terkait *scalability*, kecepatan, rekayasa trafik, serta *Quaility of Service* (QoS) [3].

### 2.2.12 Virtual *Private Network* (VPN)

Suatu jaringan yang bersifat private dengan menghubungkan satu *node* jaringan ke *node* lainnya (*tunneled*) merupakan definisi dari *Virtual Private Network* (VPN) [3]. VPN memanfaatkan jaringan *public* atau *open connection* atau *virtual circuits* yaitu internet [11]. Salah satu fungsi VPN yaitu untuk melindungi data perusahaan karena dapat menciptakan koneksi dan menjaga keamanan data pada saat proses transmisi data [22].

Teknologi VPN mampu terhubung secara *point-to-point* dalam pengiriman data antara dua komputer yang melewati jaringan publik [11]. Pengimplementasian VPN biasanya diterapkan oleh perusahaan besar karena teknologi VPN dapat membangun koneksi yang aman antara kantor yang berbeda dengan bergabung dengan jaringan lokal [11] [16].

### 2.2.13 FlexVPN

FlexVPN merupakan solusi *Cisco* untuk menyederhanakan penerapan VPN dan mencakup semua jenis VPN. FlexVPN meliputi konfigurasi *framework* berupa kumpulan perintah CLI/API yang bertujuan untuk menyederhanakan pengaturan *Site-to-site, Hub and spoke* (termasuk *spoke-to-spoke traffic*), dan *Remote access*. FlexVPN adalah teknologi VPN yang memungkinkan perusahaan untuk membangun koneksi secara aman antara kantor yang berbeda lokasi. FlexVPN mampu memberikan layanan yang jauh lebih hemat dibandingkan dengan berbagai jenis solusi VPN lainnya yang terpisah. Misalnya; GRE (*Generic Routing Encapsulation*), *Crypto Map*, dan *Virtual Tunnel Interface* (VTI). FlexVPN

mengandalkan *open-standards-based* (ASR 1000) yang menjalankan IOS-XE, termasuk *Cloud Services Router* 1000v (CSR 1000v) dan router seri ISR4000. Sebagian besar perintah konfigurasi FlexVPN dimulai dengan *crypto* ikev2, dan *dynamic tunnel* [23].

### 2.2.14 Manfaat FlexVPN

Berikut adalah beberapa manfaat apabila menggunakan FlexVPN [23]:

1) FlexVPN dapat digunakan di kedua jaringan publik berupa internet dan juga di jaringan private seperti jaringan MPLS-VPN.
2) FlexVPN ditujukan untuk VPN akses *site-to-site* dan *remote access*. Satu FlexVPN yang digunakan dapat mengenali dua jenis permintaan koneksi secara bersamaan.
3) FlexVPN didasarkan pada protokol *Internet Key Exchange version* 2 (IKEv2). IKEv2 dapat menghasilkan peningkatan kinerja yang lebih.

### 2.2.15 Komponen FlexVPN

Berikut adalah komponen yang meliputi dari FlexVPN :

1) IKEv2

   FlexVPN didasarkan pada IKEv2 di-karenakan menghadirkan semua fitur protokol IKEv2. Fitur tersebut meliputi; pertukaran konfigurasi, pengalihan IKEv2 untuk server load balancing, tantangan *cookie* untuk Mitigasi DoS, NAT *traversal*, IKEv2 *fragmentation*, dan fitur Cisco IOS IKEv2 seperti IKEv2 *call admission control*, penghapusan sesi pada kedaluwarsa sertifikat, atau *revocation* ke semua topologi VPN yang didukung. FlexVPN menggunakan konfigurasi IKEv2 untuk bertukar parameter kebijakan antara *peer* seperti antara klien dan *server* FlexVPN. Sistem untuk bertukar informasi menggunakan perutean antara *peer*. Sistem tersebut dapat menambahkan rute ke alamat *overlay* jarak jauh dan subnet yang dilindungi berfungsi sebagai *overlay* ringan dengan mekanisme *routing* [23].

2) Cisco IOS *Point-to-Point* (P2P) *Tunnel Interfaces*

   FlexVPN menggunakan per-*peer point-to-point* (P2P) *tunnel interfaces* untuk semua topologi VPN yang didukung dengan GRE atau *native* IPsec

*encapsulation*. Enkapsulasi GRE menawarkan manfaat *native support* untuk IP *dual stack* karena dapat membawa *layer* IPv4 dan IPv6 *overlay traffic* melalui IPv4 atau IPv6 *transport*. Manfaatnya dapat beroperasi dengan implementasi yang tidak mendukung GRE. P2P *tunnel interface* dikonfigurasi secara statis. Pada inisiator FlexVPN dibuat secara dinamis dari *virtual template interface* pada FlexVPN *responders* [23].

### 2.2.16 Arsitektur FlexVPN *over* MPLS

Berikut adalah beberapa istilah yang digunakan dalam FlexVPN *over* MPLS:

1) Arsitektur *Hub* dan *Spoke*

   Arsitektur *hub dan spoke* adalah desain jaringan yang diperuntukkan untuk jaringan area luas atau WAN. Arsitektur *hub* dan *spoke* memiliki perangkat pusat (*hub*) yang terhubung ke beberapa perangkat lain (*spoke*) melalui perantara internet. Perangkat hub mempunyai rute individual untuk setiap *spoke* [24]. Konsep tersebut cocok untuk perusahaan besar yang memungkinkan untuk mengirim atau menerima data secara aman. Arsitektur *hub* dan *spoke* mempunyai dua mekanisme yang berbeda untuk menyampaikan paket kepada antar *spoke*nya. Mekanisme pertama yaitu melalui perangkat *hub* sebagai titik transit paket untuk sampai ketujuannya. Mekanisme kedua tidak melibatkan perangkat *hub* sebagai titik transit paket karena alasan keamanannya [3].

2) IPsec

   *Internet Protocol Security* (IPsec) adalah sebuah protokol yang digunakan untuk memberikan keamanan pada komunikasi *Internet Protocol* (IP) dengan melakukan otentikasi dan enkripsi pada setiap paket IP yang dikirim dalam sebuah sesi komunikasi. Menggunakan IPsec dapat memberikan komunikasi antara dua atau lebih titik dapat dienkripsi sehingga hanya pihak yang memiliki kunci yang tepat yang dapat membaca isi pesan yang dikirimkan. IPsec dapat membantu melindungi privasi dan keamanan data yang dikirimkan melalui jaringan internet. Pada awal sesi, IPsec mencakup protokol untuk mengaktifkan otentikasi bersama dan negosiasi kunci

kriptografi yang akan digunakan selama sesi [22]. IPsec memfasilistasi enkripsi data melalui *tunneling* pada lapisan transport ketika melakukan transmisi data sehingga dapat memberikan keamanan, integritas, serta keaslian data yang ditransfer melalui jaringan IP berskala luas [25]. IPsec digunakan untuk mengamankan data *traffic* antara perangkat *spoke* dengan perangkat hub setelah perangkat spoke jarak jauh ditemukan secara dinamis [4].

3) IKEv2

FlexVPN menggunakan konfigurasi IKEv2 untuk bertukar parameter kebijakan antara *peer* seperti antara klien (*spoke*) dan server (*hub*) FlexVPN. Sistem untuk bertukar informasi menggunakan perutean antara *peer* berupa pertukaran kunci dan otentikasi misalnya *pre-shared key* antara dua ujung jaringan VPN yaitu *hub* dan *spoke*. *Spoke* mempunyai ujung jaringan VPN jaringan lokalnya, sementara *hub* mempunyai ujung jaringan VPN antar *spoke* lain yang berarti mengarahkan [23]. IKEv2 memanfaatkan protokol *tunneling* IPsec untuk mengamankan *traffic* data dan VTI digunakan untuk menyediakan saluran untuk kebutuhan IPsec [23].

4) *Next Hop Resolution Protocol* (NHRP)

NHRP adalah solusi yang mirip dengan ARP yang memungkinkan sebuah sistem untuk dinamis mempelajari alamat NBMA (*Non-Broadcast Multi-Access*) dari perangkat yang terhubung ke dalam jaringan, sehingga memungkinkan sistem tersebut untuk berkomunikasi secara langsung tanpa perlu melewati hop perantara. NHRP membantu mengidentifikasi lokasi perangkat di jaringan dan mengarahkan lalu lintas ke alamat tujuan secara efisien dan langsung [7]. Hasilnya router *hub* akan membuat database NHRP untuk pengalihan antar router *spoke* sehingga router *spoke* dapat terhubung satu sama lainnya tanpa melibatkan *router hub* sebagai titik transit paket untuk sampai ke tujuan [3]. NHRP dapat digunakan untuk menyelesaikan alamat *overlay* jarak jauh dan secara dinamis menemukan *end-point transport* yang diperlukan untuk membuat *tunnel* yang aman [4]. NHRP memerlukan *router* untuk meminta *next-hop* IP dari asosiasi keamanan IKEv2. VTI *dynamic* dalam NHRP digunakan untuk

mengefisiensikan pembentukan koneksi VPN melalui *tunnel* berdasarkan informasi dari *hub* karena koneksi terbentuk apabila *interface virtual* diperlukan.

5) VPN *routing* dan *forwarding* (VRF)

VPN *Routing* and *Forwarding* (VRF) merupakan elemen kunci dalam teknologi MPLS Cisco VPN yang digunakan untuk memisahkan *traffic* sehingga memungkinkan penggunaan beberapa tabel *routing* yang terpisah di dalam satu router, seperti tabel *routing* global dan tabel *routing* per pelanggan. Pemisahan tersebut menyebabkan *traffic* dari satu VRF tidak dapat melihat atau berinteraksi dengan lalu lintas dari VRF lainnya tanpa adanya informasi dari hub. Fitur ini memungkinkan isolasi lalu lintas antara berbagai pelanggan yang menggunakan jaringan yang sama. Hasilnya setiap pelanggan memiliki lingkungan jaringan yang terisolasi dan dapat beroperasi secara independen satu sama lain [11]. MPLS dalam VRF memiliki kemampuan untuk menetapkan label per VRF yang mengidentifikasi VRF yang benar ke mana data perlu dirutekan [12].

6) MPLS *Forwarding Infrastructure* (MFI)

MFI berfungsi untuk mengelola struktur data MPLS yang digunakan untuk penerusan. Cisco mengganti *Label Forwarding Information Base* (LFIB) dengan MFI pada IOS dan IOS XE terbaru. NHRP adalah aplikasi yang memanggil MFI untuk manajemen label [4].

7) *Multi-Protocol* BGP (MP-BGP)

BGP digunakan untuk mendistiribusikan atau bertukar informasi berdasarkan routing internalnya. ISP bertukar informasi berdasarkan *routing table* menggunakan BGP. BGP digunakan untuk bertukar informasi jaringan satu dengan beberapa jaringan lainnya, sehingga *router* dapat mengetahui rute jaringan tersebut [11]. Pada jaringan FlexVPN *over* MPLS, MP-BGP merupakan protokol routing yang memiliki peran untuk mendistribusikan atau mengiklankan label *overlay* untuk jaringan yang berbeda seperti pada masing-masing VRF sehingga masing-masing jaringan yang berbeda dapat mengetahui jalur jaringan yang sudah di iklankan atau di distribusikan tersebut [4]. Informasi label MPLS digunakan untuk

memetakan paket data ke jalur yang tepat pada jaringan sehingga antar VRF dapat saling menjangkau melalui *tunnel overlay*.

### 2.2.17 *Emulated Network Environment Next Generation* (EVE-NG)

EVE-NG (Emulated Virtual Environment Next Generation) merupakan suatu aplikasi untuk mengimplementasikan jaringan dalam bentuk virtualisasi berbagai perangkat jaringan dan sistem operasi berjalan seolah-olah nyata. Sama halnya dengan aplikasi cisco *packet tracer*, dan GNS3, karakteristik yang dihasilkan oleh EVE-NG jauh lebih lengkap dari segi fitur-fiturnya. EVE-NG dijalankan pada *virtual machine*, dan *interface* penggunaan grafisnya dijalankan di *web browser* dengan mengakses ip lokal pada laptop ataupun PC [26].

### 2.2.18 *Quality of Service* (QoS)

*Quality of Service* merupakan sekumpulan parameter yang digunakan untuk standarisasi kinerja dari suatu jaringan yang berkaitan dengan jaminan dan tingkat kualitas pelayanan yang diperoleh [11]. Apabila suatu sistem jaringan tidak mampu memberikan jaminan kualitas layanan kepada pengguna, sudah dipastikan pengguna di dalam jaringan tidak akan nyaman menggunakan sistem jaringan tersebut. Ketika mengimplementasikan suatu sistem pada jaringan, teknologi yang diterapkan sangat berpengaruh pada QoS yang dihasilkan karena setiap sistem pasti mempunyai kekurangan dan kelebihan masing-masing [3].

### 2.2.19 Parameter Penelitian

Berikut adalah parameter-parameter yang digunakan dalam penelitian ini.

1) *Delay*

   *Delay* merupakan lamanya waktu yang dibutuhkan supaya paket atau data sampai ke sisi penerima. Salah satu faktor yang mempengaruhi delay yaitu beban jaringan dalam situasi tertentu, jarak, perangkat yang digunakan, dan lain sebagainya. Secara umum, tabel nilai kategori *delay* berdasarkan *standard* TIPHON ditunjukkan pada tabel 2.2 berikut [7] [27].

**Tabel 2.2 Kategori *delay***

| **Kategori *Delay*** | ***Delay* (ms)** |
|---|---|
| Sangat Bagus | < 150 |
| Bagus | < 250 |
| Sedang | < 350 |
| Buruk | < 450 |

Berdasarkan standard TIPHON pada Tabel 2.2 mencari nilai *delay* dapat digunakan persamaan 2.1 sebagai berikut [7]:

$$Delay = \frac{\text{Panjang Paket (bit)}}{\text{Link Bandwidth (bps)}}\ Second \quad \textbf{(2.1)}$$

2) *Jitter*

*Jitter* merupakan nilai variasi dari waktu yang dibutuhkan oleh *Internet Protocol Network* untuk menyampaikan paket dari sumber ke tujuannya [11]. *Jitter* disebabkan adanya variasi-variasi panjang antrian, waktu pengolahan data, serta waktu penghimpunan ulang paket-paket di akhir perjalanan [16]. Secara umum, tabel nilai kategori *jitter* berdasarkan *standard* TIPHON ditunjukkan pada tabel 2.3 berikut [7] [27].

**Tabel 2.3 Kategori *jitter***

| **Kategori *Jitter*** | ***Peak Jitter* (ms)** |
|---|---|
| Sangat Bagus | 0 |
| Bagus | 75 |
| Sedang | 125 |
| Buruk | 225 |

Berdasarkan *standard* TIPHON pada Tabel 2.3 mencari nilai *jitter* dapat digunakan persamaan 2.2 sebagai berikut [7]:

$$Jitter = \frac{\text{Total Variasi } Delay}{\text{(Total Paket yang Diterima)}}\ Second \quad \textbf{(2.2)}$$

3) *Packet Loss*

*Packet Loss* merupakan jumlah paket dalam bentuk persentase yang tidak sampai ke tujuan [16]. Hal tersebut dapat terjadi dikarenakan *collision* dan *congestion* yang terjadi pada suatu jaringan [11]. Secara umum, tabel nilai kategori dari *packet loss* berdasarkan *standard* TIPHON ditunjukkan pada tabel 2.4 berikut [7] [27]:

**Tabel 2.4 Kategori *packet loss***

| **Kategori *Packet Loss*** | ***Packet Loss* (%)** |
|---|---|
| Sangat Bagus | 0 |
| Bagus | 3 |
| Sedang | 15 |
| Buruk | 25 |

Berdasarkan standard TIPHON pada Tabel 2.4 mencari nilai *packet loss* dapat digunakan persamaan 2.3 sebagai berikut [7]:

$$Packet\ Loss = \frac{\text{Paket Data Dikirim - Paket Data Diterima}}{\text{Paket Data yang Dikirim}} \times 100\% \quad \textbf{(2.3)}$$

4) *Throughput*

*Throughput* merupakan jumlah data aktual yang dapat diterima dan dikirimkan pada setiap waktunya dalam suatu sesi koneksi [11]. Secara umum, tabel nilai kategori *throughput* berdasarkan *standard* TIPHON ditunjukkan pada Tabel 2.5 berikut [28].

**Tabel 2.5 Kategori *throughput***

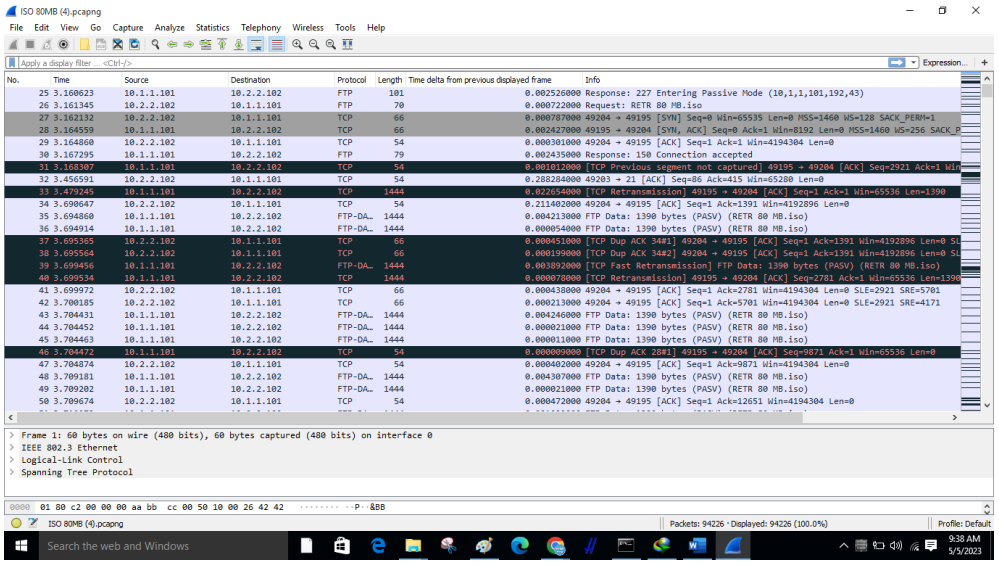
| **Kategori *Throughput*** | ***Throughput* (kbps)** |
|---|---|
| Sangat Bagus | > 2100 kbps |
| Bagus | 1200 - 2100 kbps |
| Sedang | 700 - 1200 kbps |
| Buruk | 338 - 700 kbps |
| Sangat Buruk | 0 - 338 kbps |

Berdasarkan *standard* TIPHON pada Tabel 2.5 mencari nilai *throughput* dapat digunakan persamaan 2.4 sebagai berikut [7].

$$Throughput = \frac{\text{Jumlah packet Dikirim (bit)}}{\text{Lamanya Waktu Pengiriman}}\ bps \qquad \textbf{(2.4)}$$

### 2.2.20 *Wireshark*

*Wireshark* merupakan perangkat lunak yang digunakan untuk menangkap paket dalam lalu lintas data yang berada di ruang lingkup jaringan komputer. *Wireshark* digunakan untuk mendapatkan informasi pada suatu sistem jaringan misalnya saja dalam pentransmisian data yaitu TCP dan UDP. Mengetahui layak atau tidaknya sistem jaringan dalam melakukan pentransmisian data dapat diketahui dengan menganalisis hasil informasi pada paket-paket saat proses komunikasi berlangsung sehingga dapat diketahui kesimpulan dari hasil informasi terkait paket-paket tersebut. Gambar 2.6 menunjukkan tampilan *wireshark* saat menangkap paket-paket.




**Gambar 2.6 Tampilan *wireshark***