BAB III METODE PENELITIAN

3.1 PEMODELAN SISTEM

Penelitian ini menggunakan wireshark dalam menganalisis jaringan yang tersedia. Analisis dilakukan dengan menggunakan tools Wireshark, dan parameter yang dianalisis adalah delay, jitter, throughput dan packet loss. Pengolahan data dilakukan dengan cara membandingkan hasil dari pengukuran dengan menggunakan standar TIPHON (*Telecommunications* and Internet Harmonization Over Network). TIPHON merupakan standar penilaian parameter QoS yang dikeluarkan oleh badan standar ETSI (*European Telecommunications Standars Institute*). Kemudian dianalisis bagaimana kriteria jaringan tersebut dan diambil kesimpulan dari hasil parameter-parameter tersebut.



3.2 ALUR PENELITIAN

Gambar 3.1 Diagram Alur Penelitian

Gambar 3.1 menunjukkan tahapan dalam perencanaan dan simulasi penelitian yang akan dilakukan. Alur kerja dimulai dengan melakukan studi literatur yang mengangkat topik tentang perancangan jaringan ipv4 dan ipv6 serta penggunaan jaringan MPLS. Pada tahap ini juga dilakukan pengumpulan artikel penelitian selanjutnya yang telah dilakukan terkait dengan protokol routing OSPF. Apabila bahan keseluruhan telah terpenuhi selanjutnya melakukan perancangan topologi jaringan yang akan digunakan. Pada perancangan topologi jaringan meliputi kebutuhan hardware dan software. Setelah itu dilanjutkan dengan memulai tahap simulasi atau pengujian dengan OSPF MPLS pada IPv4 dan IPv6. Setelah tahap simulasi selesai selanjutnya proses pengambilan data dilakukan dengan menggunakan software wireshark dengan sesi waktu 15, 30 dan 60 detik, masing-masing sesi diulang selama lima kali. Apabila berhasil maka penelitian akan diteruskan ke tahapan selanjutnya, apabila gagal maka akan dilakukan simulasi ulang. Pengambilan data dilakukan setelah diuji berdasarkan skenario, maka didapat nilai QoS (Quality of Service) berupa nilai delay, jitter, packet loss dan throughput. Setelah itu akan dilakukan pengecekan ulang jika terjadi RTO (*Request Time Out*) atau destination host unreachable pada saat pengujian. Dan akan dilakukan analisa untuk menyimpulkan hasilnya.

3.3 PERANCANGAN SISTEM



Gambar 3.2 Desain Topologi Jaringan

3.3.1 Perancangan Desain Topologi Jaringan

Langkah awal dalam penelitian yang dilakukan adalah merancang desain topologi jaringan, adapun desain topologi jaringan dapat dilihat pada gambar 3.2. Topologi jaringan yang akan diuji coba dibuat menggunakan perangkat lunak simulasi GNS3. Pada desain topologi jaringan yang digunakan terdapat lima router yang saling terhubung dengan memiliki 1 *router provider* (P), 2 *router provider edge* (PE), 2 *router customer edge* (CE) dan dua buah PC yang berfungsi sebagai *client* dan *server*.

Router	Interface	IP Address	Subnet Mask
PC1	eO	192.168.10.2	255.255.255.0
PC2	eO	192.168.20.2	255.255.255.0
CE1	Fa0/0	192.168.10.1	255.255.255.0
	Fa1/0	172.16.1.1	255.255.255.0
CE2	Fa0/0	172.16.1.2	255.255.255.0
	Fa1/0	192.168.20.1	255.255.255.0
PE1	Fa0/0	172.16.1.2	255.255.255.0
	Fa1/0	17.10.11.1	255.255.255.240
	Fa1/1	17.20.25.1	255.255.255.240
	Lo/0	12.12.1.1	255.255.255.0
PE2	Fa0/0	17.30.30.2	255.255.255.240
	Fa1/0	172.16.2.1	255.255.255.0
	Fa1/1	17.20.25.2	255.255.255.240
	Lo/0	12.12.3.1	255.255.255.0
Р	Fa0/0	17.10.11.2	255.255.255.240
	Fa1/0	17.30.30.1	255.255.255.240
	Lo/0	12.12.2.1	255.255.255.0

Tabel 3.1 Adressing Topologi Jaringan

Langkah selanjutnya adalah mengkonfigurasi alamat IP pada setiap *router*, alamat IP yang digunakan di setiap *router* dapat dilihat pada Tabel 3.1.

Setelahnya, dilakukan pengujian konektivitas dengan menggunakan *ping point-to-point* antar *router*. Berikut hasil konfigurasi IP *address*:

PE1(config)#int f0/0 PE1(config-if)#ip add 172.16.1.2 255.255.255.0 PE1(config-if)#no sh

3.3.2 Perancangan Routing Open Shortest Path First

Open Shortest Path First (OSPF) adalah protokol routing yang digunakan untuk IP dan termasuk dalam kategori protokol link-state. OSPF menggunakan algoritma Djikstra untuk memilih jalur terpendek menuju setiap tujuan. Dalam konteks komunikasi perangkat pada jaringan MPLS, IP *loopback* digunakan pada masing-masing perangkat. Sebelum membuat routing OSPF, langkah awalnya adalah membuat *interface* pada *bridge* dengan nama *loopback*. Interface *loopback* adalah suatu *interface* yang sebenarnya tidak ada secara fisik dan diciptakan untuk mengatasi masalah pada jaringan. Keunggulan dari IP *loopback* adalah kestabilan koneksi, karena IP ini tidak terhubung ke suatu jaringan tertentu atau *interface* fisik, sehingga tidak akan mengalami *down* seperti halnya *interface* yang terhubung ke jaringan tertentu.

Berikut hasil konfigurasi IP loopback:

```
PE1(config-if)#int lo0
PE1(config-if)#ip add 12.12.1.1 255.255.255.0
PE1(config-if)#no sh
```

Berikut hasil konfigurasi protokol routing OSPF:

Berikut hasil konfigurasi protokol routing OSPF: PE1(config)#router ospf 1 PE1(config-router)#net 12.12.1.1 0.0.0.255 area 0 PE1(config-router)#net 172.16.1.0 0.0.0.255 area 0 PE1(config-router)#net 17.10.11.0 0.0.0.15 area 0 PE1(config-router)#net 17.20.25.0 0.0.0.15 area 0 PE1(config-router)#do wr Building configuration... [OK]

3.3.3 Perancangan Jaringan Multi Protocol Label Switching

Multi Protocol Label Switching adalah teknologi pengiriman paket pada jaringan backbone (utama) dengan kecepatan tinggi. Prinsip dasar *label switching* adalah adanya penambahan *label* khusus dan unik pada setiap paket data dan *label* tersebut digunakan untuk mengarahkan paket melalui jaringan. Setelah melakukan konfigurasi *routing* OSPF, langkah berikutnya adalah membangun jaringan MPLS. Penting untuk memastikan bahwa *routing* ke IP *Loopback* pada setiap router MPLS telah terkonfigurasi dengan benar. Setelah semua IP *Loopback* terdaftar dalam routing, konfigurasi jaringan MPLS dimulai dengan menambahkan LDP *interface*. LDP *interface* digunakan untuk menentukan antarmuka dari router yang terhubung ke router lainnya dalam jaringan MPLS.

Berikut hasil konfigurasi MPLS:

PE1(config)#ip cef
PE1(config)#mpls ip
PE1(config)#mpls label protocol ldp
PE1(config)#mpls ldp router-id f0/0
PE1(config)#mpls ldp router-id f1/0
PE1(config)#mpls ldp router-id f1/1
PE1(config)#int f0/0
PE1(config-if)#mpls ip
PE1(config-if)#mpls label protocol ldp
PE1(config-if)#int f1/0
PE1(config-if)#mpls ip
PE1(config-if)#mpls label protocol ldp
PE1(config-if)#int f1/1
PE1(config-if)#mpls ip

PE1(config-if)#mpls label protocol ldp PE1(config-if)#do wr Building configuration... [OK]

3.4 SKENARIO PENGUJIAN

Pengujian yang dilakukan pada penelitian ini menggunakan dua skenario, yakni: pengujian pada jaringan MPLS OSPF untuk IPV6. Proses pengambilan data dilakukan dengan menggunakan *software Wireshark* dan dengan sesi waktu 15 detik, 30 detik dan 60 detik, masing-masing sesi diulang sebanyak 5 kali. Pengujian dilakukan untuk memahami kinerja jaringan dari IPv4 dan IPv6 serta mengamati stabilitas kinerja melalui hasil rata-rata dari beberapa pengujian. Hasil pengujian kemudian dianalisis menggunakan beberapa parameter QoS untuk mengevaluasi kualitas jaringan. Beberapa standar dapat diajukan acuan untuk mengetahui bagus tidaknya nilai dalam parameter QoS. Standar kategori performansi jaringan yang digunakan sesuai dengan versi TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*).

Tahapan yang dilakukan pada skenario pengujian untuk pengambilan data:

- 1. Pada konfigurasi jaringan yang sudah dibuat menggunakan *ipterm server* dan *client*, serta *iperf* yang digunakan untuk menaikkan trafik pada jaringan.
- 2. Gunakan GNS3 VM2. GNS3 Virtual Machine memungkinkan pengguna untuk membuat dan menguji topologi jaringan yang kompleks tanpa perlu perangkat keras fisik. GNS3 VM2 mengintegrasikan berbagai perangkat jaringan virtual dan fisik, seperti router, switch dan firewall dalam satu topologi, memungkinkan untuk mengembangkan dan menguji konfigurasi jaringan tanpa perlu mengganggu infrastruktur fisik yang ada. GNS3 VM2 diinstal sebagai bagian dari instalasi GNS3 pada komputer

tujuan, yang memberikan fleksibilitas dan kinerja yang lebih baik dan memastikan simulasi jaringan yang lebih lancar dan responsif.



Gambar 3.3 Informasi pada GNS3 VM2

Pada gambar 3.3 menunjukkan informasi seperti IP, *port* dan *password* yang akan digunakan dalam konfigurasi selanjutnya.

3. Selanjutnya, buka FileZilla. FileZilla dapat dijalankan pada berbagai platform yang memungkinkan untuk dengan mudah mengelola *transfer file* seperti mengunggah dan mengunduh *file*.

Site Manager	×
Select entry:	General Advanced Transfer Settings Charset
→ My Sites ↓ 및 New site	Protocol:
	Logon Type: Normal
	Password: 9155
	Background color: None v
New site New folder New Bookmark Rename	A
Delete Duplicate	•
	Connect OK Cancel

Gambar 3.4 Tampilan pada FileZilla

Gambar 3.4 adalah tampilan pada FileZilla untuk memasukkan informasi berupa *host, port, username* serta *password* yang digunakan.

4. Tahap selanjutnya klik *connect*, dan status akan berubah menjadi *connected to* 192.168.56.102.

Host:	192.168.56.102	Username:	admin	Password:	Port:	Quickcon	nect 💌
Status:	Connecting to	192.168.56.102					
Status:	Using usernam	e "gns3".					
Status:	Connected to 1	92.168.56.102					
Status:	Retrieving direct	tory listing					
Status:	Listing director	y /home/gns3					
Status:	Directory listing	g of "/home/g	ns3" successful				
Local s	ite: C:\Users\HP\				~	Remote site:	/home/gns3
	Defa	ult User				<mark>?</mark> /	
	🕀 – 🔁 HP						me
	🕀 📒 Publ	ic					gns3
	🗄 📩 Window	5			1		

Gambar 3.5 Tampilan Status

Gambar 3.5 menunjukkan tampilan pada saat *host* sudah terhubung dan status berubah menjadi "*connected*".

5. Selanjutnya, *upload file* yang akan dikirimkan dan status akan berubah menjadi "*file transfer successful*".

Host:	192.168.56.102	Username:	admin	Password:	Pc
Status:	Starting upload	of C:\Users\H	P\Pictures\Screensh	ots\Screenshot 2	2023-05-31 073801.png
Status:	File transfer suc	cessful, transf	erred 27.735 bytes in	1 second	
Status:	Retrieving direc	tory listing of	"/opt/gns3/images/	QEMU"	
Status:	Listing director	y/opt/gns3/in	nages/QEMU		
Status:	Directory listing	of "/opt/gns	3/images/QEMU" sug	cessful	
Status:	Disconnected fi	rom server	-		

Gambar 3.6 Tampilan status setelah upload file selesai

Gambar 3.6 berisi tampilan setelah *upload file* berhasil dilakukan dan status berubah menjadi "*file transfer successful*". Data/*file* yang dikirimkan dengan ukuran 27.735 *bytes* (27,4 KB) dengan format png.

6. Setelah *transfer file* selesai dilakukan, pengambilan data akan dilakukan dengan menggunakan *software wireshark* sebagai *network analyzer*. Wireshark akan dipasangkan pada *interface* f1/0 router CE2 yang mengarah pada VPCS sebagai *client*.

Ime Source Destration Photoal Leight Infi 10 9.232011 17.20.25.1 172.16.2.1 LDP 72 Keep Alive Message 11 9.379302 172.16.2.1 17.20.25.1 TCP 60 2586 + 646 [ACK] Seq=1 Ack=19 Min=3678 Len=0 19 18.05932 17.30.30.1 17.20.25.1 LDP 72 Keep Alive Message 20 18.215731 17.20.25.1 17.30.30.1 TCP 60 646 + 56323 [ACK] Seq=1 Ack=19 Min=3678 Len=0 38 36.193025 172.16.2.1 17.20.25.1 DP 72 Keep Alive Message 39 36.193025 172.16.2.1 17.20.25.1 DP 72 Keep Alive Message 39 36.193025 172.16.2.1 17.20.25.1 DP 72 Keep Alive Message 39 36.346097 17.20.25.1 170 60 646 + 22586 [ACK] Seq=19 Ack=19 Min=3714 Len=0 52 49.153122 17.30.30.1 DP 72 Keep Alive Message 127.30.30.1 53 49.540803 17.30.30.1 DP 72 Keep Aliv	
10 9.232011 17.20.25.1 172.16.2.1 LDP 72 Keep Alive Message 11 9.379302 172.16.2.1 17.20.25.1 TCP 60 22586 + 646 [ACK] Seq=1 Ack=19 Win=3678 Len=0 19 18.035922 17.30.30.1 17.20.25.1 LDP 72 Keep Alive Message 20 18.115731 17.20.25.1 17.30.30.1 TCP 60 646 + 56323 [ACK] Seq=1 Ack=19 Win=3678 Len=0 30 36.193825 172.16.2.1 17.30.25.1 LDP 72 Keep Alive Message 30 36.193825 172.16.2.1 17.30.25.1 LDP 72 Keep Alive Message 30 36.44697 17.20.25.1 172.40.25.1 LDP 72 Keep Alive Message 24 9.31521 17.20.25.1 172.40.25.1 LDP 72 Keep Alive Message 314 9.446981 17.30.30.1 LDP 72 Keep Alive Message Min=3660 Len=0 52 54.9.546981 17.30.30.1 17.20.25.1 TCP 54 Seq=19 Ack=19 Win=3660 Len=0 52 56.306499 17.2	
11 9.379302 172.16.2.1 17.20.25.1 TCP 60 25586 + 646 [ACK] Seq=1 Ack=19 Win=3678 Len=0 19 18.035522 17.30.30.1 17.20.25.1 10P 72 Keep Alive Message 20 18.215731 17.20.25.1 17.30.30.1 TCP 60 464 + 56323 [ACK] Seq=1 Ack=19 Win=3678 Len=0 38 36.193825 172.16.2.1 17.20.25.1 10P 72 Keep Alive Message 99 36.344697 17.20.25.1 172.16.2.1 TCP 60 664 + 25806 [ACK] Seq=1 Ack=19 Win=3678 Len=0 52 49.153122 17.20.25.1 172.16.2.1 TCP 60 664 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Len=0 53 49.546808 17.30.30.1 10P 72 Keep Alive Message 10P 72 Keep Alive Message 64 59.66081 17.30.30.1 10P 72 Keep Alive Message 10P 12 Wessage 62 58.630849 17.20.25.1 172.06.2.1 TCP 64 64 [ACK] Seq=19 Ack=19 Win=3660 Len=0 62 58.630849 17.20.25.1 172.16.2.1 10P 72 Keep Alive Message 64 59.660951 172.16.2.1 17.20.25.1 172.16.2.1 170 80 2586 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0 65 59.660951 17.20.25.1 172.16.2.1 172 P	
19 18.835922 17.30.30.1 17.20.25.1 LDP 72 Keep Alive Message 20 18.215731 17.20.25.1 17.30.30.1 TCP 60 646 + 56323 [ACK] Seq=1 Ack=19 Win=3678 Len=0 30 36.193025 172.16.2.1 17.20.25.1 LDP 72 Keep Alive Message 30 36.193025 172.16.2.1 17.20.25.1 LDP 72 Keep Alive Message 30 36.193025 172.0.25.1 172.16.2.1 TCP 60 646 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Len=0 52 49.153122 17.20.25.1 17.30.30.1 LDP 72 Keep Alive Message 54 95.46808 17.30.30.1 LDP 72 Keep Alive Message 18.6569 Len=0 52 58.630849 17.20.25.1 172.15.2.1 LDP 72 Keep Alive Message 64 59.660851 172.16.2.1 17.09 25.1 CP 52 Ke45 [ACK] Seq=19 Ack=3 Win=3660 Len=0 65 59.660851 172.2.5.1 172.08.25.1 TCP 60 2586 + 646 [ACK] Seq=19 Ack=3 Win=3660 Len=0 65 59.660851 172.2.2.1 17.20.25.1 172.16.2.1 TCP 60 2586 + 2586 [ACK] Seq=19 Ack=3 Win=3660 Len=0 65 59.660851 172.2.2.1 172.16.2.1 172 P2 [TCP Spurious Retransmission] 646 + 22586 [ACK] Seq=19 Ack=19 Win=3714	
20 18.215731 17.20.25.1 17.30.30.1 TCP 60 646 + 56323 [ACK] Seq=1 Ack=19 kin=3678 Lem=0 30 36.139325 172.16.2.1 17.20.25.1 LDP 72 Keep Alive Message 30 36.344607 17.20.25.1 172.16.2.1 TCP 60 646 + 25586 [ACK] Seq=19 Ack=19 kin=3678 Lem=0 52 49.153122 17.20.25.1 172.16.2.1 TCP 60 646 + 22586 [ACK] Seq=19 Ack=19 kin=3714 Lem=0 53 49.54080 17.30.30.1 LDP 72 Keep Alive Message 17.30.30.1 LDP 54 9.54080 17.30.30.1 17.20.25.1 TCP. 54 5632 + 46 [ACK] Seq=19 Ack=19 kin=3660 Lem=0 62 56.630849 17.20.25.1 172.16.2.1 LDP 72 Keep Alive Message 64 59.060951 172.16.2.1 17.20.25.1 TCP 60 2586 + 646 [ACK] Seq=19 Ack=37 kin=3660 Lem=0 62 58.630849 17.20.25.1 TCP 60 2586 + 646 [ACK] Seq=19 Ack=37 kin=3660 Lem=0 65 59.060951 172.16.2.1 TCP 60 2586 + 646 [ACK] Seq=19 Ack=37 kin=3660 Lem=0 65 59.060951 17.20.25.1 TCP 60 2586 + 646 [ACK] Seq=19 Ack=37 kin=3660 Lem=0 65 59.060951 17.20.25.1 TCP 72 [TCP Spurious Retransmission] 646 + 22586 [ACK] Seq=19 Ack=19 kin=3714 Lem=18	
38 36.193825 172.16.2.1 17.28.25.1 LDP 72 Keep Alive Message 39 36.344697 17.28.25.1 172.16.2.1 TCP 60 645 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Len=0 52 49.153122 17.28.25.1 17.30.30.1 LDP 72 Keep Alive Message 53 49.546988 17.30.30.1 LDP 72 Keep Alive Message 53 49.546988 17.30.30.1 LDP 72 Keep Alive Message 64 58.60949 17.20.25.1 TCP 54 56323 + 646 [ACK] Seq=19 Ack=19 Win=3660 Len=0 64 59.660951 172.16.2.1 17.20.25.1 TCP 60 2586 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0 65 59.660951 172.26.25.1 172.08.25.1 TCP 60 2586 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0 65 59.660951 172.26.25.1 172.9.25.1 TCP 60 2586 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0 65 59.660951 172.26.25.1 172.16.2.1 TCP 72 [TCP Spurious Retransmission] 646 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Len=18	
39 36.344697 17.20.25.1 172.16.2.1 TCP 60 646 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Lem=0 52 49.153122 17.20.25.1 17.30.30.1 LDP 72 Keep Alive Message 53 49.546988 17.30.30.1 17.20.25.1 TCP 54 55323 + 646 [ACK] Seq=19 Ack=19 Win=3660 Lem=0 62 58.630649 17.20.25.1 172.16.2.1 LDP 72 Keep Alive Message 64 59.060951 172.16.2.1 172.06.25.1 TCP 60 2256 + 646 [ACK] Seq=19 Ack=37 Win=3660 Lem=0 65 59.060951 172.16.2.1 17.20.25.1 TCP 60 2256 + 646 [ACK] Seq=19 Ack=37 Win=3660 Lem=0 65 59.060951 17.20.25.1 172.16.2.1 TCP 72 [TCP Spurious Retransmission] 646 + 22506 [ACK] Seq=19 Ack=19 Win=3714 Lem=18	
52 49.153122 17.20.25.1 17.30.30.1 LDP 72 Keep Alive Message 53 49.546988 17.30.30.1 17.20.25.1 TCP 54 55323 + 646 [ACK] Seq=19 Ack=19 Win=3660 Len=0 62 58.630649 17.20.25.1 172.16.2.1 LDP 72 Keep Alive Message 64 59.060951 172.16.2.1 17.20.25.1 TCP 60 22566 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0 65 59.060951 17.20.25.1 17.20.25.1 TCP 60 22566 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0 65 59.060951 17.20.25.1 172.16.2.1 TCP 72 [TCP Spurious Retransmission] 646 + 22505 [ACK] Seq=19 Ack=19 Win=3714 Len=18	
53 49.546988 17.30.30.1 17.20.25.1 TCP 54 56323 + 646 [ACK] Seq=19 Ack=19 Win=3660 Len=0 62 58.630649 17.20.25.1 172.16.2.1 LDP 72 Keep Alive Message 64 59.060951 172.16.2.1 17.20.25.1 TCP 60 22586 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0 65 59.060951 17.20.25.1 172.16.2.1 TCP 72 [TCP Spurious Retransmission] 646 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Len=18	
62 58.638949 17.20.25.1 172.16.2.1 LDP 72 Keep Alive Message 64 59.068951 172.16.2.1 17.20.25.1 TCP 60 22586 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0 65 59.068951 17.20.25.1 172.16.2.1 TCP 72 [TCP Spurious Retransmission] 646 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Len=18	
64 59.060951 172.16.2.1 17.20.15.1 TCP 60 22586 + 646 [ACK] Seq=19 Ack=37 Win=3660 Lem=0 65 59.060951 17.20.15.1 172.16.2.1 TCP 72 [TCP Spurious Retransmission] 646 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Lem=18	
65 59.060951 17.20.25.1 172.16.2.1 TCP 72 [TCP Spurious Retransmission] 646 + 22586 [ACK] Seq=19 Ack=19 Win=3714 Len=18	
66 59.149133 172.16.2.1 17.20.25.1 TCP 60 [TCP Dup ACK 64#1] 22586 + 646 [ACK] Seq=19 Ack=37 Win=3660 Len=0	
75 69.319932 17.30.30.1 17.20.25.1 LDP 72 Keep Alive Message	
76 69.526661 17.20.25.1 17.30.30.1 TCP 60 646 + 56323 [ACK] Seq=19 Ack=37 Win=3660 Len=0	
96 89.952054 172.16.2.1 17.20.25.1 LDP 72 Keep Alive Message	
98 90.253312 17.20.25.1 172.16.2.1 TCP 60 646 + 22586 [ACK] Seq=37 Ack=37 Win=3696 Len=0	

Gambar 3.7 Tampilan wireshark saat capture data

Gambar 3.7 adalah tampilan pada *software wireshark* pada saat *capture* data dilakukan. Menampilkan paket-paket yang berhasil ditangkap, berurutan mulai dari paket pertama dan seterusnya. Menunjukkan waktu yang dibutuhkan untuk *capture*, sumber dan tujuan paket, protokol serta informasi dari paket tersebut.

7. Selanjutnya akan dilakukan pengukuran dan pengambilan data QoS, meliputi *throughput, packet loss, delay* dan *jitter*. Pengukuran dilakukan selama lima kali dengan sesi waktu 15 detik, 30 detik dan 60 detik. Data yang sudah didapat dari wireshark, akan di *save as* dalam format .csv untuk melakukan perhitungan *delay* dan *jiter* pada *Microsoft Excel* dan selanjutnya hasil akan dianalisis.

3.5 VARIABEL PENELITIAN

Quality of Service (QoS) merupakan metode yang digunakan untuk mengukur dan mengetahui karakteristik dan sifat dari satu *service*. QoS juga digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan di asosiasikan dengan suatu servis. 1. Mengamati besarnya *delay*

Ketika nilai delay yang dihasilkan semakin kecil dalam sebuah jaringan tersebut akan semakin baik, begitu juga sebaliknya, apabila nilai *delay* yang dihasilkan semakin besar maka kualitas jaringan tersebut akan semakin buruk. Semakin besar nilai *delay* yang dihasilkan akan menyebabkan semakin lama paket data akan di terima atau dengan kata lain kinerja jaringan tersebut akan menjadi lebih lambat.

2. Mengamati jumlah packet loss

Semakin kecil nilai *packet loss* yang dihasilkan dari jaringan internet maka kualitas jaringan tersebut akan semakin baik. Begitu juga sebaliknya, ketika nilai dari *packet loss* yang dihasilkan semakin besar, maka kualitas dari jaringan tersebut bisa dikatakan semakin buruk. Karena dengan semakin besar dari nilai *packet loss* maka paket data yang di terima akan semakin berkurang atau hilang.

3. Mengamati besarnya nilai jitter

Ketika nilai *jitter* yang dihasilkan dari monitoring menggunakan aplikasi *wireshark* dalam sebuah jaringan adalah kecil, maka kualitas jaringan tersebut akan semakin baik. Sebaliknya, ketika nilai *jitter* yang dihasilkan adalah besar maka kualitas jaringan tersebut akan semakin buruk.

4. Mengamati besarnya throughput

Semakin tinggi nilai *throughput* maka kualitas jaringan tersebut semakin baik karena dapat memaksimalkan penggunaan *bandwidth* pada jaringan sehingga lebih efisien. Sebaliknya semakin kecil nilai *throughput* yang dihasilkan maka kualitas jaringan tersebut akan semakin buruk.