

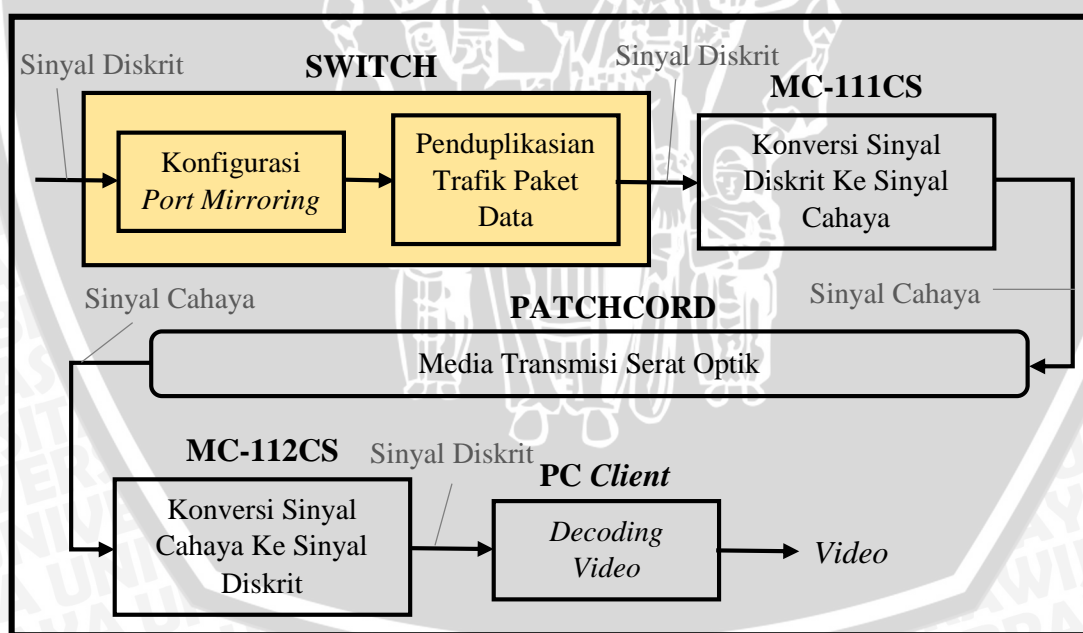
BAB IV HASIL DAN PEMBAHASAN

Bab IV ini menjelaskan hasil dan pembahasan dari penelitian mengenai pengaruh *monitoring* menggunakan *port mirroring* terhadap performansi jaringan untuk layanan *Video on Demand* (VoD) melalui media serat optik berdasarkan parameter *throughput*, *packet loss*, dan *delay*. Terdapat beberapa tahapan dalam penelitian ini, yaitu:

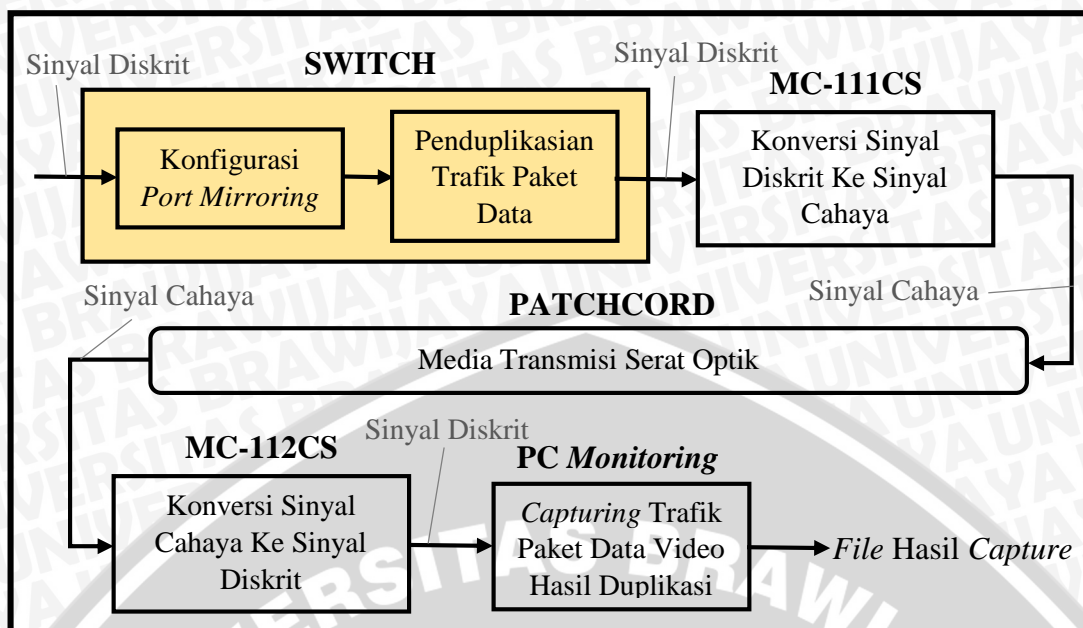
- Konfigurasi dan instalasi perangkat, serta pengujian sistem.
- Pengambilan data sesuai dengan parameter yang dibutuhkan yaitu *throughput*, *packet loss*, dan *delay*.
- Analisis dari data yang telah didapatkan dari penelitian.

4.1 Konfigurasi dan Instalasi Perangkat

Sebelum melakukan konfigurasi dan instalasi perangkat, telah dilakukan perancangan blok diagram penelitian *PC client* seperti yang ditunjukkan pada Gambar 4.1(a) dan Gambar 4.1(b) merupakan blok diagram penelitian *PC Monitoring*.

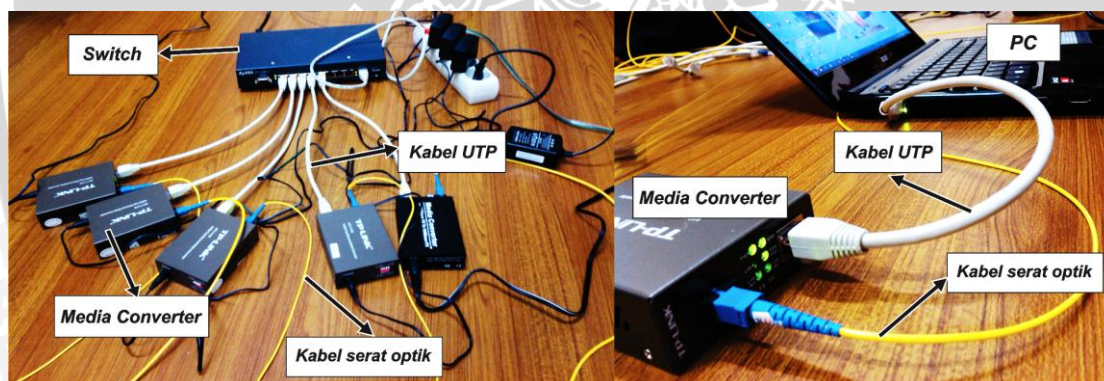


Gambar 4.1(a) Blok Diagram Penelitian *Client*
(Sumber: Perancangan)



Gambar 4.1(b) Blok Diagram Penelitian *Monitoring*
(Sumber: Perancangan)

Berdasarkan perancangan blok diagram penelitian, maka penyusunan perangkat untuk pengambilan data ditunjukkan pada Gambar 4.2.



Gambar 4.2 Penyusunan Perangkat
(Sumber: Penelitian)

Spesifikasi dan penjelasan dari masing-masing perangkat yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Server*

Server yang dipakai menggunakan *Google Cloud Computing*. Perangkat lunak yang digunakan sebagai *server Video on Demand (VoD)* adalah *Wowza Streaming Engine Manager*.

2. *ISP (Internet Service Provider)*

Jaringan internet yang digunakan di lingkup Gedung C Jurusan Teknik Elektro Universitas Brawijaya.

3. Switch

Switch yang digunakan sebagai *switch* Gedung C Jurusan Teknik Elektro Universitas Brawijaya adalah Huawei s5700-28C-PWR-El. Spesifikasi perangkat telah terlampir pada Lampiran 1.



Gambar 4.3 *Switch* Huawei s5700
(Sumber: www.huawei.com)

Switch laboratorium yang digunakan *Switch Manageable* ZyXEL ES-2108 G. Bentuk dan spesifikasi perangkat ditunjukkan pada Gambar 4.4 dan Tabel 4.1.



Gambar 4.4 *Switch* ZyXEL ES-2108 G
(Sumber: Penelitian)

Tabel 4.1 Spesifikasi *Switch* ZyXEL ES-2108 G

Spesifikasi	Keterangan
Jumlah Port Fast Ethernet	8 Port 10/100 Base-TX
Jumlah Port Gigabit	1 Port Gigabit Ethernet 1 Port mini-GBIC
Kapasitas Switching	3,6 Gbps
Switching Forwarding Rate	2,7 Mpps
Maksimum Ukuran Frame	1522 bytes (termasuk tag/CRC)
Packet Buffer	256 Kbytes
MAC Address	8 K

(Sumber: www.ZyXEL.com)

4. Komputer Client

Spesifikasi minimum komputer *client* yang digunakan dalam penelitian ini ditunjukkan pada Tabel 4.2.

Tabel 4.2 Spesifikasi Minimum Komputer *Client*

Spesifikasi	Keterangan
Processor	Intel® Core™ i3
Standard Memory	2 GB
Hard Drive Type	500 GB HDD
Network Card	Integrated 10/100 BASE-T Ethernet LAN
Network Interface	RJ-45

5. Komputer *Monitoring*

Pada penelitian ini komputer *monitoring* mengamati trafik terhadap *client* saat melakukan *streaming Video on Demand* (VoD) melalui *port mirroring*. Komputer *monitoring* tersebut memiliki spesifikasi perangkat yang ditunjukkan Tabel 4.3.

Tabel 4.3 Spesifikasi Komputer *Monitoring*

Spesifikasi	Keterangan
Processor	AMD Quad Core Processor A6-5200 (2 GHz)
Standard Memory	2 GB
Hard Drive Type	500 GB HDD
Network Card	Gigabit Ethernet
Network Interface	RJ-45

6. *Media Converter*

Media Converter merupakan perangkat yang berfungsi untuk mengubah sinyal elektrik menjadi optik (cahaya) dan sebaliknya. Jenis *media converter* yang digunakan dalam penelitian ini adalah TP-LINK MC-111CS dan MC-112CS. Masing-masing memiliki 1 *port* 100 Mbps SC untuk kabel serat optik (*patch cord*) *single mode* dan 1 *port* 100 Mbps RJ45 untuk kabel UTP *category 5* atau 5e.

MC-111CS bekerja pada panjang gelombang 1550nm untuk mentransmisikan data dan menerima data pada panjang gelombang 1310nm. MC-112CS mentransmisikan data pada panjang gelombang 1310nm dan menerima data pada panjang gelombang 1550nm. Meskipun terdapat dua panjang gelombang yang berbeda, dengan MC-111CS dan MC-112CS menerapkan teknologi WDM (*Wavelength Division Multiplexing*), sehingga hanya membutuhkan sebuah kabel serat optik saja untuk mengirim dan menerima data. Bentuk perangkat ditunjukkan Gambar 4.5.



Gambar 4.5 Media Converter TP-LINK MC-111CS dan MC-112CS
(Sumber: Penelitian)

7. Kabel UTP (*Unshielded Twisted Pair*)

Kabel UTP yang digunakan dalam penelitian ini adalah kabel UTP *category 5e* dengan konektor RJ45. Kabel UTP *category 5e* memiliki kecepatan transmisi data digital hingga 250 Mbps. Panjang kabel UTP yang digunakan total sepanjang 6,6 meter. Salah satu bentuk kabel UTP penelitian dengan panjang 0,3 meter ditunjukkan Gambar 4.6.



Gambar 4.6 Kabel UTP *Category 5e*
(Sumber: Penelitian)

8. Kabel Serat Optik (*Patch Cord*)

Patch Cord merupakan kabel serat optik yang dikhususkan untuk pemakaian *indoor* yang memiliki konektor pada kedua ujungnya yang berfungsi untuk menghubungkan dengan perangkat lain. Jenis *patch cord* yang digunakan adalah *single mode* dengan diameter *core* dan *cladding* sebesar 9 μ m dan 125 μ m, serta mampu melewati sinyal dengan panjang gelombang 1310nm, 1490nm, dan 1550nm. Konektor *patch cord* penelitian adalah SC-SC, yang berarti di kedua ujung *patch cord* memiliki konektor SC. *Patch cord* penelitian ditunjukkan Gambar 4.7.



Gambar 4.7 Patch Cord
(Sumber: Penelitian)

4.2 Prosedur Pengambilan Data

Berikut ini merupakan prosedur pengambilan data saat kondisi 0 *port*, dimana tidak ada *port* yang diamati melalui *port mirroring* atau *port mirroring* tidak aktif.

1. Menghubungkan seluruh *client* dan PC Monitoring dengan internet melalui *switch* ZyXEL ES-2108G.
2. Seluruh *client* melakukan *streaming video* melalui perangkat lunak VLC Media Player secara bersamaan. Prosedur *streaming video* dijelaskan pada sub bab 4.2.2.
3. Setiapp *client* melakukan *capture* trafik selama 60 detik menggunakan perangkat lunak Wireshark pada waktu yang bersamaan dengan *client* saat melakukan *streaming video*. Prosedur *capturing* trafik menggunakan Wireshark dijelaskan pada sub bab 4.2.3.

Berikut ini merupakan prosedur pengambilan data saat penerapan *port mirroring* pada *switch* untuk kondisi 2, 3, dan 4.

1. Mengaktifkan *port mirroring* pada *switch* ZyXEL ES-2108G, dan melakukan konfigurasi *port mirroring*. Prosedur konfigurasi *port mirroring* dijelaskan pada sub bab 4.2.1.
 - Kondisi 2: Jumlah *port* yang diamati melalui *port mirroring* adalah 1 *port*, seperti yang ditunjukkan Gambar 4.11(a).
Monitor port : Port 5 (Terhubung dengan PC Monitoring)
Mirrored port : Port 1 (Terhubung dengan PC Client 1)
 - Kondisi 3: Jumlah *port* yang diamati melalui *port mirroring* adalah 2 *port*, seperti yang ditunjukkan Gambar 4.11(b).
Monitor port : Port 5 (Terhubung dengan PC Monitoring)
Mirrored port : Port 1 (Terhubung dengan PC Client 1)
 Port 2 (Terhubung dengan PC Client 2)

- Kondisi 4: Jumlah *port* yang diamati melalui *port mirroring* adalah 3 *port*, seperti yang ditunjukkan Gambar 4.11(c).

Monitor port : Port 5 (Terhubung dengan PC Monitoring)

Mirrored port : Port 1 (Terhubung dengan PC Client 1)

Port 2 (Terhubung dengan PC Client 2)

Port 3 (Terhubung dengan PC Client 3)

- Kondisi 5: Jumlah *port* yang diamati melalui *port mirroring* adalah 3 *port*, seperti yang ditunjukkan Gambar 4.11(d).

Monitor port : Port 5 (Terhubung dengan PC Monitoring)

Mirrored port : Port 1 (Terhubung dengan PC Client 1)

Port 2 (Terhubung dengan PC Client 2)

Port 3 (Terhubung dengan PC Client 3)

Port 4 (Terhubung dengan PC Client 4)

2. Menghubungkan seluruh *client* dan PC Monitoring dengan internet melalui *switch* ZyXEL ES-2108G.
3. Seluruh *client* melakukan *streaming video* melalui perangkat lunak VLC Media Player secara bersamaan. Prosedur *streaming video* dijelaskan pada sub bab 4.2.2.
4. *Administrator* pada PC Monitoring melakukan *capture* trafik selama 60 detik menggunakan perangkat lunak Wireshark pada waktu yang bersamaan dengan *client* saat melakukan *streaming video*. Prosedur *capturing* trafik menggunakan Wireshark dijelaskan pada sub bab 4.2.3.

4.2.1 Prosedur Konfigurasi Switch ZyXEL ES-2108 G

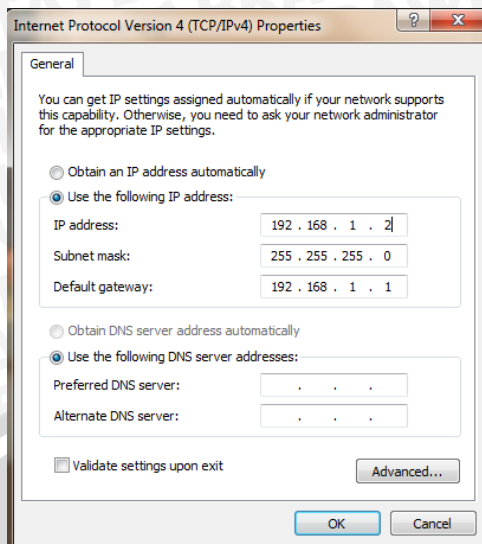
Konfigurasi *switch* ZyXEL ES-2108 G dilakukan melalui *web configurator* ZyXEL. Berikut merupakan beberapa langkah untuk *log in* ke *web configurator* dari *switch* ZyXEL ES-2108 G:

1. Menghubungkan PC Monitoring ke *switch* melalui *port Gigabit Ethernet*.
2. Mengatur TCP/IPv4 seperti pada Gambar 4.8.

IP Address : 192.168.1.2

Subnet mask : 255.255.255.0

IP Gateway : 192.168.1.1



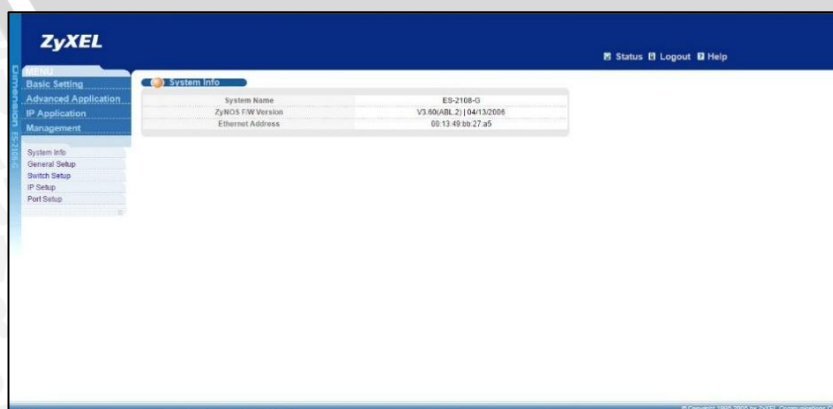
Gambar 4.8 Pengaturan TCP/IPv4
(Sumber: Penelitian)

3. Membuka *web browser*, kemudian ketik IP Address Switch yaitu **192.168.1.1** pada *Address field*, tekan 'Enter'.
4. Muncul layar *log in* untuk *web configurator* seperti pada Gambar 4.9. Kemudian mengisi *user name* dan *password*.



Gambar 4.9 Layar *log in* untuk *web configurator*
(Sumber: Penelitian)

5. Setelah berhasil *log in*, akan muncul halaman *web configurator* seperti pada Gambar 4.10.



Gambar 4.10 Halaman *Web Configuration* ZyXEL
(Sumber: Penelitian)

6. Klik *Advanced Application* kemudian klik *Mirroring* untuk mengaktifkan dan mengkonfigurasi *port mirroring*. Kemudian mengkonfigurasi *port mirroring* sesuai dengan Tabel 3.1 seperti yang ditunjukkan pada Gambar 4.11.



Gambar 4.11 Konfigurasi *Port Mirroring*

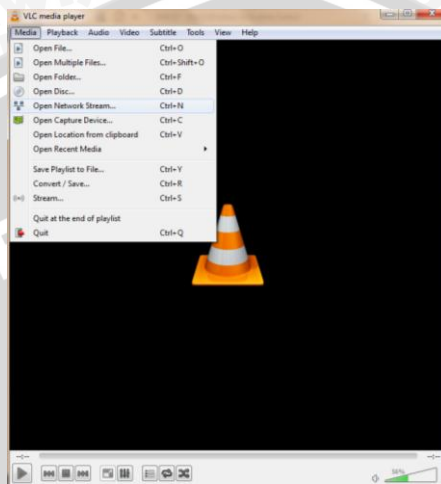
(a) Kondisi ke-1 (b) Kondisi ke-2 (c) Kondisi ke-3 (d) Kondisi ke-4
(Sumber: Penelitian)

7. Ketika sudah selesai melakukan konfigurasi, klik 'Apply'. Kemudian pilih klik 'Logout' untuk keluar dari *Web Configurator*.

4.2.2 Prosedur *Streaming VoD* oleh *PC Client*

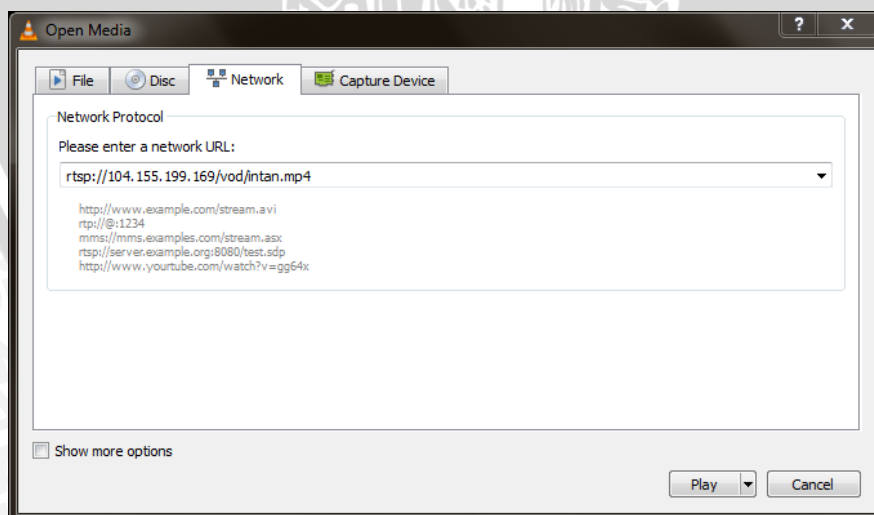
Pada *PC client* dilakukan instalasi *VLC Media Player*. Instalasi dilakukan agar *client* dapat *streaming Video on Demand (VoD)* melalui internet dan *VLC media player* tersebut.

1. Membuka perangkat lunak *VLC Media Player*. Klik '*Media*' kemudian pilih '*Open Network Stream*' seperti yang ditunjukkan oleh Gambar 4.12.



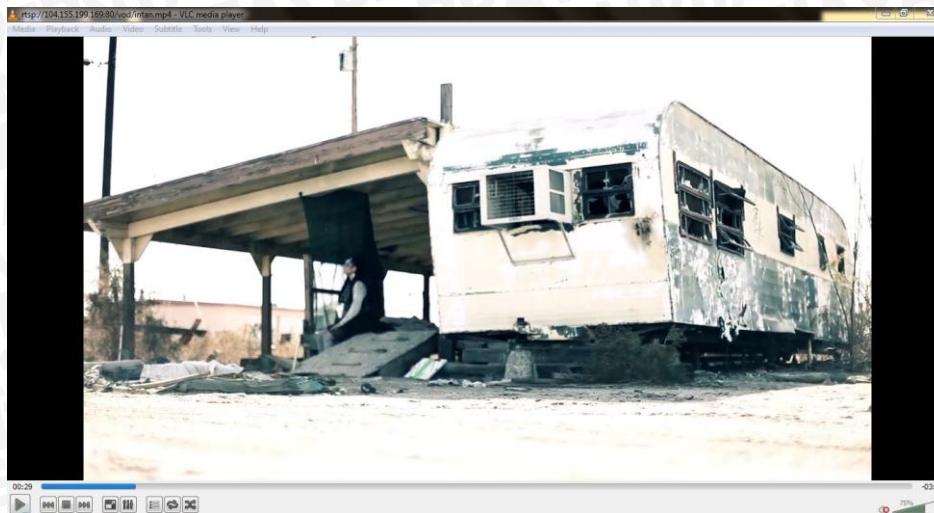
Gambar 4.12 *Home screen VLC Media Player*
(Sumber: Penelitian)

2. Menulis URL (*Uniform Resource Locator*) *file video* yang terdapat di *server* yaitu **rtsp://104.155.199.169/vod/intan.mp4** seperti pada Gambar 4.13. Setelah itu klik '*Play*'.



Gambar 4.13 *URL File Video*
(Sumber: Penelitian)

3. *VLC Media Player* kemudian memutar *file video* tersebut, seperti yang ditunjukkan pada Gambar 4.14.

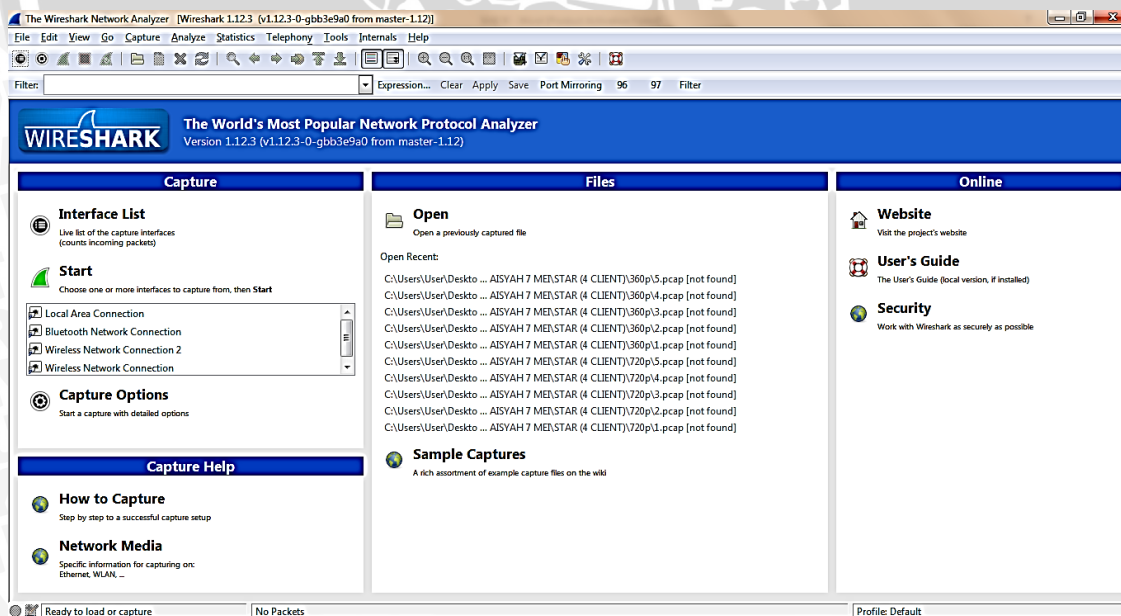


Gambar 4.14 Tampilan *streaming video* pada *VLC Media Player*
(Sumber: Penelitian)

4.2.3 Posedur *Capturing data trafik* pada *PC Monitoring*

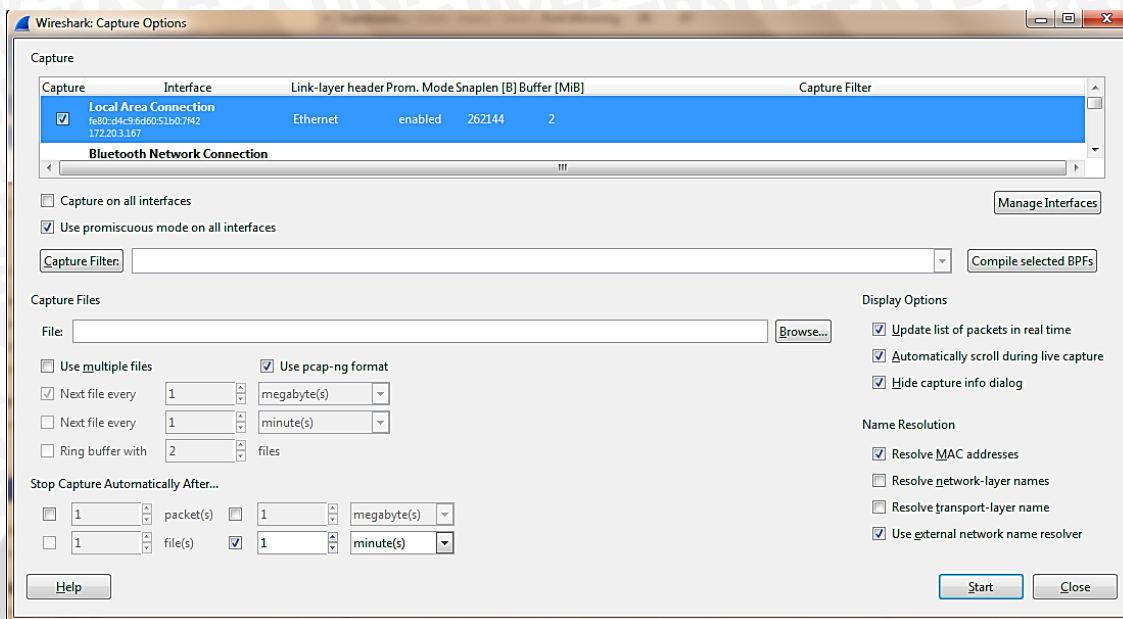
Pada *PC monitoring* dilakukan instalasi *Network Protocol Analyzer* yaitu Wireshark. Instalasi dilakukan agar *PC Monitoring* dapat *capturing data trafik* dan menyimpan hasil *file capture*. Berikut merupakan langkah-langkah untuk *capturing data trafik* dan penyimpanan hasil *capture* menggunakan Wireshark.

1. Membuka perangkat lunak Wireshark. Gambar 4.15 menunjukkan tampilan utama Wireshark.



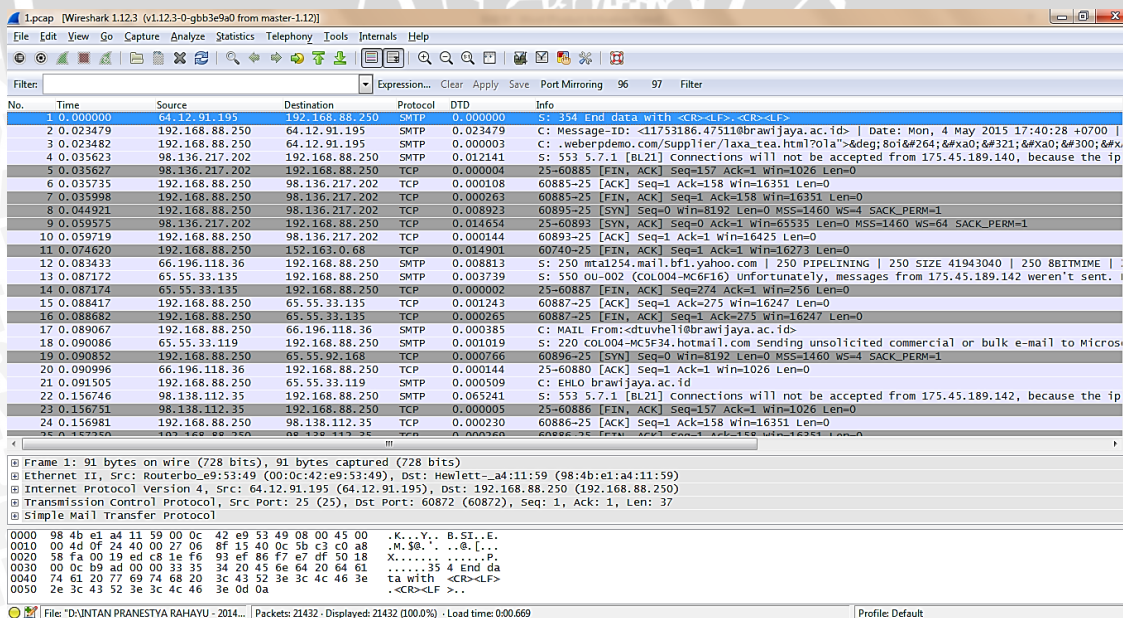
Gambar 4.15 Tampilan Utama Wireshark
(Sumber: Penelitian)

2. Kemudian klik 'Interface List'. Muncul kotak dialog Wireshark: *Capture Options* dan *checklist* seperti pada Gambar 4.16. Setelah itu klik 'Start' maka Wireshark akan mulai *capture* data trafik selama 60 detik.



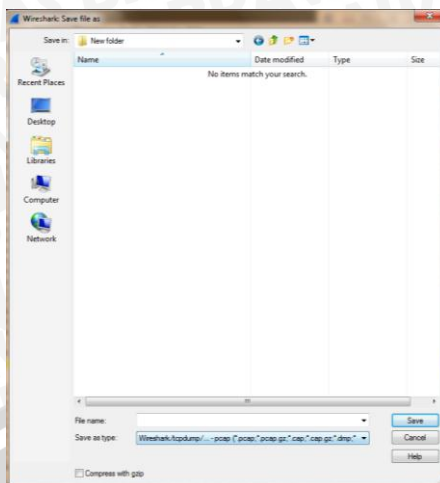
Gambar 4.16 Kotak Dialog Wireshark: *Capture Options* (Sumber: Penelitian)

3. Hasil *Capturing* data trafik seperti pada Gambar 4.17.



Gambar 4.17 Hasil *Capturing* Data Trafik pada Wireshark (Sumber: Penelitian)

4. Menyimpan hasil *capture* data trafik pada Wireshark. Klik 'File' kemudian 'Save As'. Pilih *save as type* menjadi *pcap seperti yang ditunjukkan Gambar 4.18.



Gambar 4.18 Penyimpanan Hasil *Capture* Wireshark
(Sumber: Penelitian)

4.3 Hasil dan Pembahasan Performansi Jaringan

Hasil penelitian pengaruh jumlah *port* yang diamati melalui *port mirroring* terhadap performansi jaringan dilihat dari parameter *throughput*, *packet loss*, dan *delay* beserta pembahasannya akan dijelaskan pada sub bab ini. Hasil penelitian didapatkan dari pengambilan data primer yang dilakukan dengan *capturing* trafik menggunakan *Network Protocol Analyzer* Wireshark.

Pengambilan data primer dilakukan pada jam tidak sibuk yaitu pada pukul 02.00 WIB, dan pada jam sibuk yaitu pada pukul 09.00 WIB. Terdapat 5 (lima) kondisi berbeda saat pengukuran, kondisi tersebut adalah variasi jumlah *port* yang diamati melalui *port mirroring* seperti ditunjukkan pada tabel 4.4.

Tabel 4.4. Kondisi Pengambilan Data

Kondisi		Jumlah Port yang Diamati	Keterangan
1	Semua PC Client melakukan streaming VoD	0 Port	-
2		1 Port	PC Client 1
3		2 Port	PC Client 1, PC Client 2
4		3 Port	PC Client 1, PC Client 2, PC Client 3
5		4 Port	PC Client 1, PC Client 2, PC Client 3, PC Client 4

4.3.1 Throughput

Throughput didefinisikan sebagai suatu pengukuran yang menyajikan informasi aktual terkait dengan kemampuan suatu perangkat mentransmisikan paket data untuk kurun waktu satu detik.

File video yang digunakan dalam penelitian ini yaitu *video* dengan resolusi 1920x1080, *frame rate* sebesar 23 *frame* per sekon, dan *codec* H.264. Berdasarkan persamaan 2.1b dapat diperoleh nilai *throughput video* dengan *codec* H.264 tanpa *losses* secara teoritis sebagai berikut:

$$\lambda_{FLVideo} = Width \times Height \times Frame Rate \times 0,07 \times Motion Rank$$

$$\lambda_{FLVideo} = 1.920 \times 1.080 \times 23 \times 0,07 \times 1$$

$$\lambda_{FLVideo} = 3.338,5 \text{ kbps}$$

Standar *audio* pada *video* penelitian ini menggunakan standar AAC (*Advanced Audio Coding*) yang diketahui *throughput audio* tanpa *losses* sebesar 128 kbps. Maka dengan menggunakan persamaan 2.1a *throughput* total tanpa *losses* secara teoritis sebesar:

$$\lambda_{FL} = \lambda_{FLVideo} + \lambda_{FLAudio}$$

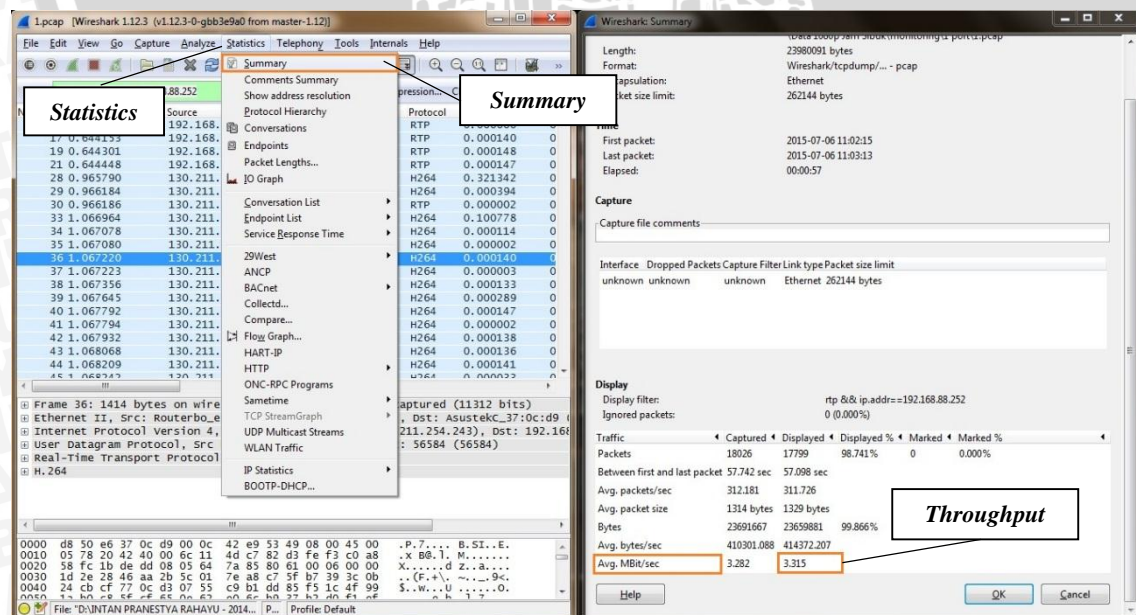
$$\lambda_{FL} = 3.338,5 \text{ kbps} + 128 \text{ kbps}$$

$$\lambda_{FL} = 3.466,5 \text{ kbps}$$

Hasil perhitungan *throughput* total tanpa *losses* secara teoritis didapatkan sebesar 3,4665 Mbps.

4.3.1.1 Throughput Rata-Rata Tiap Client

Tabel 4.5 merupakan *throughput* rata-rata tiap *client* yang didapatkan dari hasil *capturing* data menggunakan Wireshark seperti ditunjukkan pada Gambar 4.19 saat seluruh *client streaming Video on Demand (VoD)* secara bersamaan.



Gambar 4.19 Langkah Memperoleh Nilai *Throughput* Pada Wireshark (Sumber: Penelitian)

Tabel 4.5 Nilai *Throughput* Rata-Rata Tiap *Client*

Jumlah Port yang Diamati	Percobaan Ke-	<i>Throughput</i> (Mbps)	
		02.00 (Jam Tidak Sibuk)	10.00 (Jam Sibuk)
0 Port	1	3,384	3,354
	2	3,416	3,36
	3	3,414	3,35
	4	3,391	3,359
	5	3,4	3,313
	Rata-Rata	3,401	3,347
1 Port	1	3,38	3,315
	2	3,38	3,321
	3	3,373	3,334
	4	3,4	3,305
	5	3,404	3,305
	Rata-Rata	3,387	3,316
2 Port	1	3,374	3,280
	2	3,37	3,288
	3	3,377	3,290
	4	3,42	3,257
	5	3,397	3,226
	Rata-Rata	3,387	3,268
3 Port	1	3,376	3,339
	2	3,347	3,347
	3	3,371	3,203
	4	3,382	3,159
	5	3,383	3,2
	Rata-Rata	3,372	3,25
4 Port	1	3,360	3,246
	2	3,363	3,198
	3	3,325	3,280
	4	3,385	3,250
	5	3,373	3,240
	Rata-Rata	3,361	3,243

(Sumber: Perhitungan)

4.3.1.2 *Throughput* Pada Port Mirroring

Nilai *throughput* pada *port mirroring* pada jam tidak sibuk dapat dihitung dengan menggunakan persamaan 2.1c, dimana diketahui panjang tiap paket yang diterima sebesar 1330 byte dan *header* paket 59 byte. Perhitungan *throughput* pada *port mirroring* adalah sebagai berikut:

Untuk 1 *port* yang diamati, jumlah paket yang diterima benar adalah 18.679 paket dalam waktu transmisi 58,544 sekon, maka:

$$\text{Throughput} = \frac{(L + L') \times N_T}{t_v}$$

$$\text{Throughput} = \frac{1389 \times 8 \times 18.679}{58,544}$$

$$\text{Throughput} = \mathbf{3,545 \text{ Mbps}}$$

Untuk 2 *port* yang diamati, jumlah paket yang diterima benar adalah 37.804,2 paket dalam waktu transmisi 59,393 sekon, maka:

$$\text{Throughput} = \frac{(L + L') \times N_T}{t_v}$$

$$\text{Throughput} = \frac{1389 \times 8 \times 37.804,2}{59,393}$$

$$\text{Throughput} = \mathbf{7,073 \text{ Mbps}}$$

Untuk 3 *port* yang diamati, jumlah paket yang diterima benar adalah 56.232,8 paket dalam waktu transmisi 59,178 sekon, maka:

$$\text{Throughput} = \frac{(L + L') \times N_T}{t_v}$$

$$\text{Throughput} = \frac{1389 \times 8 \times 56.232,8}{59,178}$$

$$\text{Throughput} = \mathbf{10,559 \text{ Mbps}}$$

Untuk 4 *port* yang diamati, jumlah paket yang diterima benar adalah 75.151,4 paket dalam waktu transmisi 59,658 sekon, maka:

$$\text{Throughput} = \frac{(L + L') \times N_T}{t_v}$$

$$\text{Throughput} = \frac{1389 \times 8 \times 75.151,4}{59,658}$$

$$\text{Throughput} = \mathbf{13,998 \text{ Mbps}}$$

Nilai *throughput* pada *port mirroring* pada jam sibuk diperoleh dengan menggunakan cara perhitungan yang sama. Kedua hasil perhitungan *throughput* pada *port mirroring* pada jam tidak sibuk dan jam sibuk disajikan dalam bentuk Tabel 4.6.

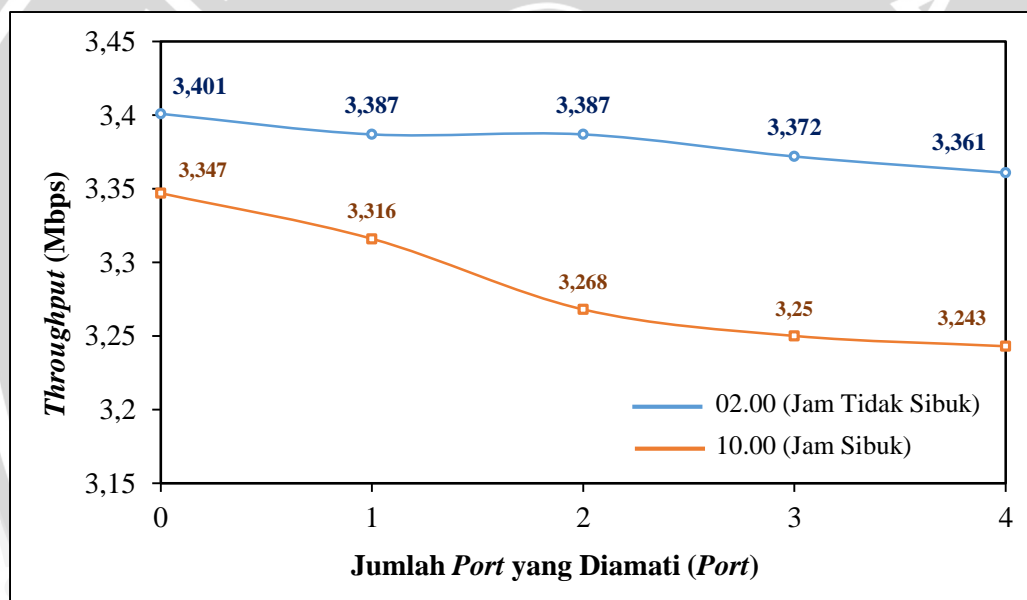
Tabel 4.6 *Throughput* pada *Port Mirroring*

Jumlah Port yang Diamati	02.00 (Jam Tidak Sibuk)		10.00 (Jam Sibuk)	
	N _T (Paket)	<i>Throughput</i> (Mbps)	N _T (Paket)	<i>Throughput</i> (Mbps)
0 Port	0	0	0	0
1 Port	18.679	3,545	18.412	3,514
2 Port	37.804,2	7,073	36.327	6,815
3 Port	56.232,8	10,559	54.139,8	10,125
4 Port	75.151,4	13,998	72.022	13,424

(Sumber: Perhitungan)

4.3.2 Analisis Hubungan antara *Throughput* Dengan Jumlah Port yang Diamati Melalui *Port Mirroring*

Hubungan parameter performansi *throughput* rata-rata tiap *client* terhadap jumlah port yang diamati melalui *port mirroring* ditunjukkan pada Gambar 4.20.

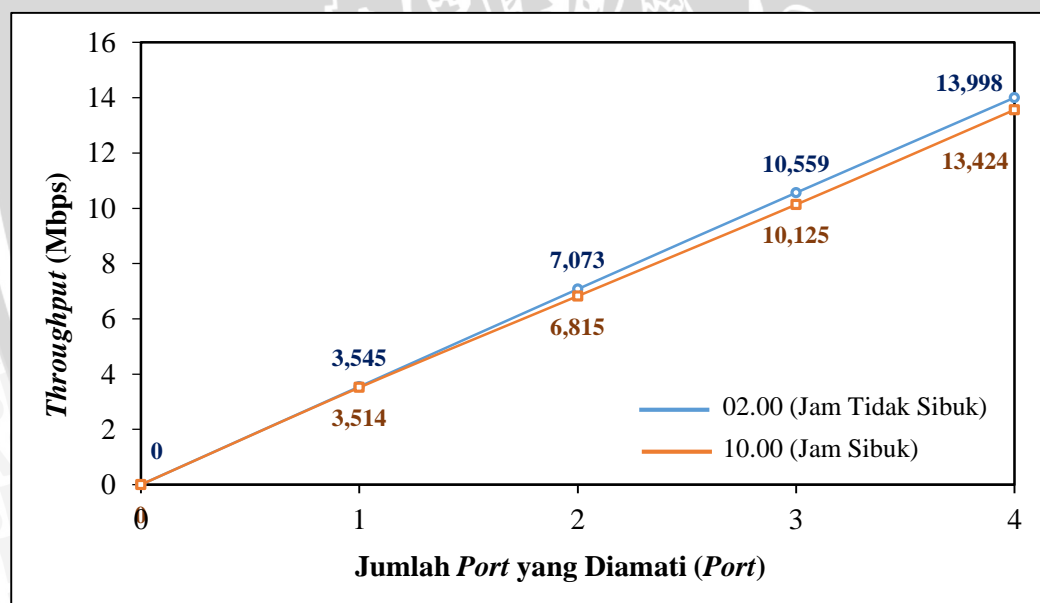


Gambar 4.20 Grafik Hubungan Antara Jumlah Port yang Diamati dengan *Throughput* Rata-Rata Tiap *Client* (Sumber: Penelitian)

Untuk saat jam tidak sibuk, pada kondisi 0 port yaitu saat tidak ada port yang diamati melalui *port mirroring*, dihasilkan *throughput* rata-rata pada tiap *client* terbesar yaitu 3,401 Mbps. Pada kondisi 4 port yang diamati melalui *port mirroring*, dihasilkan nilai *throughput* rata-rata pada tiap *client* terkecil yaitu 3,361 Mbps. *Throughput* rata-rata tiap *client* mengalami penurunan dari kondisi 0 port ke 4 port yang diamati melalui *port mirroring* yaitu sebesar 0,04 Mbps (1,176%) meskipun tiap kondisi *client streaming video* yang sama dan *capture* dilakukan pada durasi yang sama pula.

Nilai *throughput* pada saat jam sibuk lebih rendah dibandingkan saat jam tidak sibuk, pada kondisi 0 *port* yaitu saat tidak ada *port* yang diamati melalui *port mirroring*, dihasilkan *throughput* rata-rata pada tiap *client* terbesar yaitu 3,347 Mbps. Pada kondisi 4 *port* yang diamati melalui *port mirroring*, dihasilkan nilai *throughput* rata-rata pada tiap *client* terkecil yaitu 3,243 Mbps. *Throughput* rata-rata tiap *client* mengalami penurunan dari kondisi 0 *port* ke 4 *port* yang diamati melalui *port mirroring* yaitu sebesar 0,104 Mbps (3,107%) meskipun tiap kondisi *client streaming video* yang sama dan *capture* dilakukan pada durasi yang sama pula.

Hasil penelitian menunjukkan bahwa *throughput* rata-rata pada tiap *client* semakin menurun seiring dengan semakin bertambahnya jumlah *port* yang diamati melalui *port mirroring*. Penurunan *throughput* rata-rata tiap *client* mengindikasikan bahwa penerapan *port mirroring* mengakibatkan jumlah paket atau *bit* yang diterima benar oleh *client* saat *streaming video* semakin berkurang dan berdampak pada kualitas *video* yang dihasilkan. Hubungan parameter performansi *throughput* pada *port mirroring* terhadap jumlah *port* yang diamati melalui *port mirroring* ditunjukkan pada Gambar 4.21.



Gambar 4.21 Grafik Hubungan Antara Jumlah *Port* yang Diamati dengan *Throughput* Pada *Port Mirroring* (Sumber: Perhitungan)

Untuk saat jam tidak sibuk, pada kondisi 0 *port* yang diamati melalui *port mirroring*, *throughput* pada *port mirroring* sebesar 0 Mbps dikarenakan pada kondisi tersebut tidak ada proses duplikasi paket dan tidak ada paket hasil duplikasi yang dikirimkan ke *port mirroring*. Sedangkan untuk jumlah *port* yang lain 1 *port*, 2 *port* 3

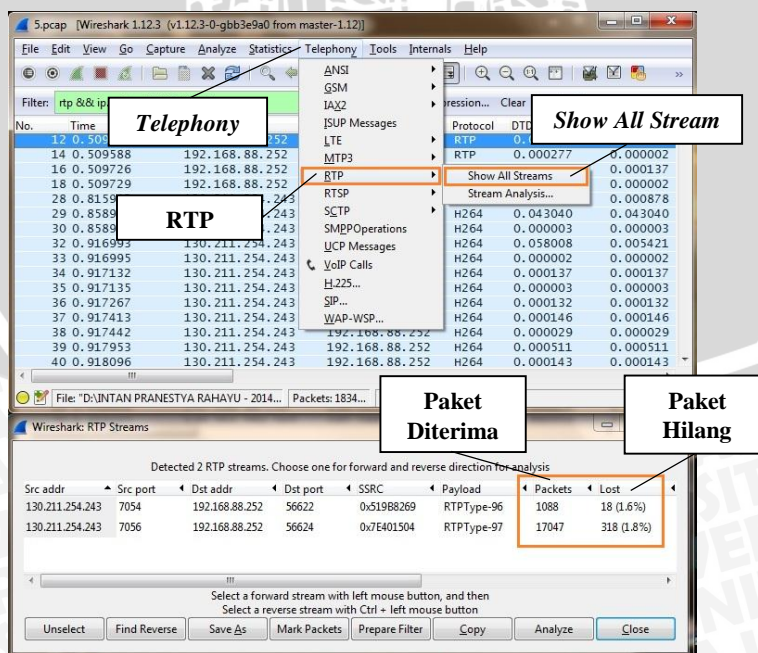
port, dan 4 port dihasilkan *throughput* sebesar 3,545 Mbps, 7,073 Mbps, 10,559 Mbps, dan 13,998 Mbps.

Nilai *throughput* pada saat jam sibuk lebih rendah dibandingkan saat jam tidak sibuk,, pada kondisi 0 port yang diamati melalui *port mirroring*, *throughput* pada *port mirroring* sebesar 0 Mbps dikarenakan pada kondisi tersebut tidak ada proses duplikasi data dan tidak ada paket hasil duplikasi yang dikirimkan ke *port mirroring*. Sedangkan untuk jumlah port yang lain 1 port, 2 port 3 port, dan 4 port dihasilkan *throughput* sebesar 3,516 Mbps, 6,826 Mbps, 10,137 Mbps, dan 13,567 Mbps.

Hasil penelitian menunjukkan bahwa *throughput* pada *port mirroring* semakin besar seiring dengan semakin bertambahnya jumlah port yang diamati melalui *port mirroring*. *Throughput* pada *port mirroring* yang semakin besar mengindikasikan beban trafik yang diamati melalui *port mirroring* yang semakin besar pula. Maka, *port mirroring* membutuhkan *bandwidth* yang semakin besar seiring dengan penambahan jumlah port yang diamati melalui *port mirroring*.

4.3.3 Packet Loss

Packet Loss dapat diartikan sebagai hilangnya sejumlah paket data pada jaringan komputer selama proses transmisi paket data. Perhitungan persentase *packet loss* menggunakan persamaan 2.2, dimana diketahui jumlah paket yang diterima dan jumlah paket yang hilang pada Tabel 4.7, kedua data primer tersebut didapatkan dari hasil *capturing* trafik menggunakan Wireshark seperti yang ditunjukkan Gambar 4.22.



Gambar 4.22 Langkah Memperoleh Jumlah Paket yang Diterima dan Hilang (Sumber: Penelitian)

Tabel 4.7 Data Primer Jumlah Paket yang Diterima dan Hilang

Jumlah Port yang Diamati	Percobaan Ke-	02.00 (Jam Tidak Sibuk)		10.00 (Jam Sibuk)	
		Jumlah Paket Rata-Rata yang Diterima	Jumlah Paket Rata-Rata yang Hilang	Jumlah Paket Rata-Rata yang Diterima	Jumlah Paket Rata-Rata yang Hilang
0 Port	1	18.806	0	18436	108
	2	19.071	0	17664	135
	3	19.077	0	18421	172
	4	18.737	0	18465	73
	5	19.126	35	18164	135
	Rata-Rata	18.963,4	7	18230	124,6
1 Port	1	18.673	43	17795	154
	2	18.472	1	18217	256
	3	18.431	14	18223	237
	4	18.806	0	18329	378
	5	19.013	16	18135	336
	Rata-Rata	18.679	14,8	18139,8	272,2
2 Port	1	18.695	104,5	18173	606,5
	2	18.678,5	46,5	18373	430
	3	18.802	68,5	18135	459,5
	4	19.272	2	18169,5	570,5
	5	19.063	2,5	17967	762,5
	Rata-Rata	18902,1	44,8	18163,5	565,8
3 Port	1	18.708	34	18674	374
	2	18.470,33	45	18373,00	1178
	3	18.644	53,33	17934	394,33
	4	18.949,33	84,33	17645,33	1101,87
	5	18.949,67	81,67	17606,33	403,33
	Rata-Rata	18.744,27	59,67	18046,60	690,37
4 Port	1	18.609,75	73,5	18135,75	689,75
	2	18.970,75	108	17612	818,75
	3	18.443,5	181,75	18123,25	607,75
	4	19.025,5	47	18097,5	834,25
	5	18.889,75	109,5	18059	938,75
	Rata-Rata	18.787,85	103,95	18005,5	777,85

(Sumber: Penelitian)

Perhitungan *packet loss* untuk jumlah *port* yang diamati 0 *port* adalah sebagai berikut:

$$Packet\ loss\ (\%) = \frac{N_{packet\ loss}}{N_{packet} + N_{packet\ loss}} \times 100\%$$

$$Packet\ loss\ (\%) = \frac{7}{18.963,4 + 7} \times 100\%$$

$$Packet\ loss\ (\%) = 0,037\%$$

Perhitungan *packet loss* untuk jumlah *port* yang diamati sebanyak 1 *port* adalah sebagai berikut:

$$\text{Packet loss (\%)} = \frac{N_{\text{packet loss}}}{N_{\text{packet}} + N_{\text{packet loss}}} \times 100\%$$

$$\text{Packet loss (\%)} = \frac{14,8}{18.679 + 14,8} \times 100\%$$

$$\text{Packet loss (\%)} = \mathbf{0,079\%}$$

Perhitungan *packet loss* untuk jumlah *port* yang diamati sebanyak 2 *port* adalah sebagai berikut:

$$\text{Packet loss (\%)} = \frac{N_{\text{packet loss}}}{N_{\text{packet}} + N_{\text{packet loss}}} \times 100\%$$

$$\text{Packet loss (\%)} = \frac{44,8}{18.902,1 + 44,8} \times 100\%$$

$$\text{Packet loss (\%)} = \mathbf{0,238\%}$$

Perhitungan *packet loss* untuk jumlah *port* yang diamati sebanyak 3 *port* adalah sebagai berikut:

$$\text{Packet loss (\%)} = \frac{N_{\text{packet loss}}}{N_{\text{packet}} + N_{\text{packet loss}}} \times 100\%$$

$$\text{Packet loss (\%)} = \frac{59,67}{18.744,27 + 59,67} \times 100\%$$

$$\text{Packet loss (\%)} = \mathbf{0,316\%}$$

Perhitungan *packet loss* untuk jumlah *port* yang diamati sebanyak 4 *port* adalah sebagai berikut:

$$\text{Packet loss (\%)} = \frac{N_{\text{packet loss}}}{N_{\text{packet}} + N_{\text{packet loss}}} \times 100\%$$

$$\text{Packet loss (\%)} = \frac{103,95}{18.787,85 + 103,95} \times 100\%$$

$$\text{Packet loss (\%)} = \mathbf{0,552\%}$$

Nilai *packet loss* pada jam sibuk diperoleh dengan menggunakan cara perhitungan yang sama. Kedua Hasil perhitungan masing-masing persentase *packet loss* pada jam tidak sibuk dan jam sibuk untuk setiap kondisi ditunjukkan pada Tabel 4.8.

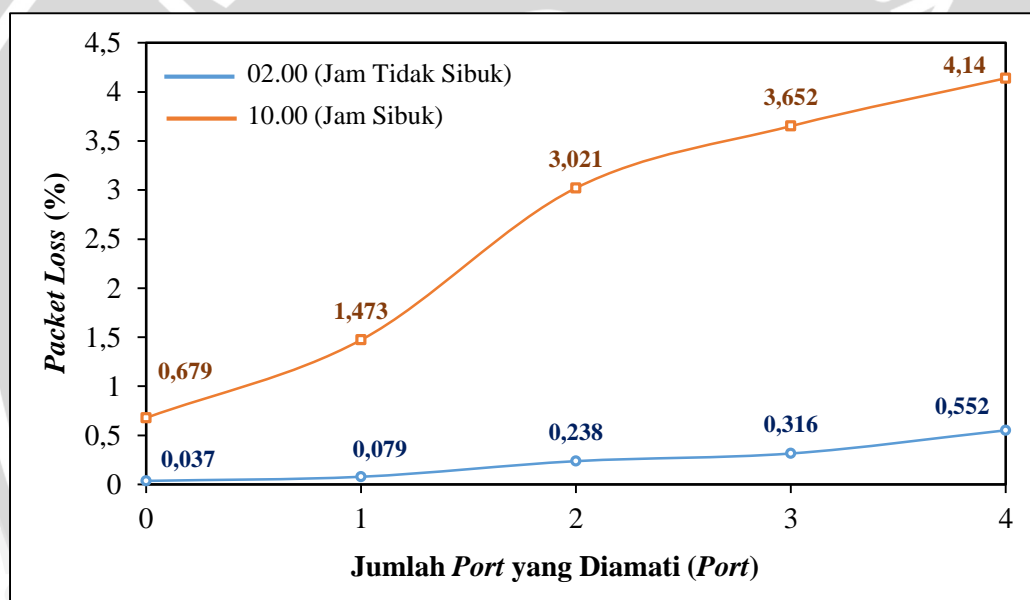
Tabel 4.8 Nilai *Packet Loss*

Jumlah Port yang Diamati	<i>Packet Loss (%)</i>	
	02.00 (Jam Tidak Sibuk)	10.00 (Jam Sibuk)
0 Port	0,037	0,679
1 Port	0,079	1,473
2 Port	0,238	3,021
3 Port	0,316	3,652
4 Port	0,552	4,140

(Sumber: Perhitungan)

4.3.4 Analisis Hubungan antara *Packet Loss* dengan Jumlah Port yang Diamati Melalui *Port Mirroring*

Hubungan parameter performansi *packet loss* terhadap jumlah port yang diamati melalui *port mirroring* ditunjukkan pada Gambar 4.23.



Gambar 4.23 Grafik Hubungan Antara Jumlah Port yang Diamati dengan *Packet Loss* (Sumber: Perhitungan)

Hasil penelitian menunjukkan bahwa *packet loss* semakin besar seiring dengan semakin bertambahnya jumlah port yang diamati melalui *port mirroring*. Untuk saat jam tidak sibuk, pada kondisi 0 port dimana tidak ada port yang diamati melalui *port mirroring* dihasilkan *packet loss* terkecil yaitu 0,037%, kemudian meningkat menjadi 0,079%, 0,238%, 0,316%, dan 0,552% pada kondisi 1 port, 2 port, 3 port, dan 4 port yang diamati melalui *port mirroring*. Sedangkan untuk saat jam sibuk, pada kondisi 0 port dimana tidak ada port yang diamati melalui *port mirroring* dihasilkan *packet loss* terkecil

yaitu 0,679%, kemudian meningkat menjadi 1,473%, 3,021%, 3,652%, dan 4,14% pada kondisi 1 *port*, 2 *port*, 3 *port*, dan 4 *port* yang diamati melalui *port mirroring*.

Hasil tersebut menunjukkan bahwa penerapan *monitoring* menggunakan *port mirroring* mengakibatkan *packet loss* semakin besar. Sebelumnya telah dijelaskan pada bab II bahwa saat penerapan *port mirroring* akan terjadi proses duplikasi trafik di *switch*. Konfigurasi *port mirroring* pada *switch* berbasis pada nomor *port switch* yang akan diamati, maka trafik paket pada *port switch* yang diamati tersebut akan diduplikasi. Trafik paket hasil duplikasi selanjutnya dikirim melalui *switch fabric* menuju *output queue port mirroring* terlebih dahulu sebelum diteruskan ke *PC Monitoring*. Sehingga penambahan jumlah *port* yang diamati melalui *port mirroring* pada penelitian menghasilkan penambahan jumlah paket yang harus diduplikasi dan dikirim ke *port mirroring*.

Konsekuensi hal tersebut adalah dapat terjadinya penumpukan paket pada *output queue port mirroring* dan saat jumlah paket yang mampu diterima *output queue* terbatas maka probabilitas *blocking* paket pada *output queue port mirroring* akan semakin besar pula. *Blocking* paket terjadi saat *output queue* penuh dan tidak mampu menerima paket lagi sehingga paket akan dibuang dan menghasilkan *packet loss*.

Pada layanan *streaming Video on Demand (VoD)*, semakin besarnya *packet loss* mengindikasikan bahwa tampilan *video* hasil *streaming* semakin buruk akibat semakin banyaknya potongan gambar *video* yang hilang. Berdasarkan rekomendasi standar *packet loss* ITU-T G.1010 Tabel 2.1 jaringan dapat dinyatakan memiliki kualitas yang baik karena jika hasil *packet loss* kurang dari 1%, maka dengan *packet loss* maksimum untuk saat jam tidak sibuk adalah 0,552% dapat dinyatakan memenuhi standar tersebut, sedangkan dengan *packet loss* maksimum untuk saat jam sibuk adalah 4,14% tidak memenuhi standar tersebut.

4.3.5 Delay

Delay didefinisikan sebagai lamanya waktu yang diperlukan oleh paket data untuk sampai ke tujuan. Terdapat dua macam *delay* yang diperoleh dari penelitian, yaitu *delay* rata-rata dan *delay end-to-end* maksimum.

4.3.5.1 Delay Rata-Rata

Perhitungan *delay* rata-rata menggunakan persamaan 4.1, dimana persamaan tersebut membutuhkan data primer seperti durasi waktu *streaming Video on Demand (VoD)* dan total paket data hasil *capture*.

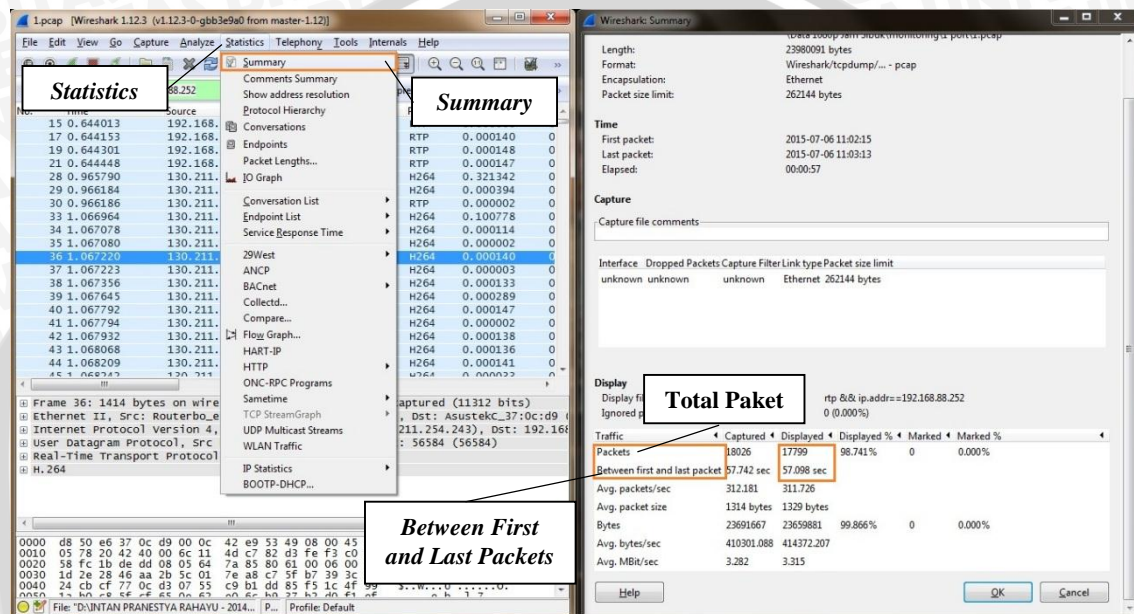
$$Delay \text{ rata - rata (sekon/paket) } = \frac{\text{Between First and Last Packet}}{\text{Packet}} \quad (4.1)$$

Keterangan:

Between First and Last Packet : Durasi *streaming* VoD (sekon)

Packet : Total paket data (paket)

Kedua data primer yang dibutuhkan tersebut didapatkan dari hasil *capturing* trafik menggunakan Wireshark seperti yang ditunjukkan Gambar 4.24.



Gambar 4.24 Langkah Memperoleh *Delay* Rata-Rata Pada Wireshark (Sumber: Penelitian)

Tabel 4.9 menunjukkan data primer yang dibutuhkan untuk menghitung *delay* rata-rata yang didapatkan dari hasil *capturing* Wireshark.

Tabel 4.9 Data Primer Untuk Perhitungan *Delay* Rata-Rata

Jumlah Port yang Diamati	Percobaan Ke-	02.00 (Jam Tidak Sibuk)		10.00 (Jam Sibuk)	
		Between First and Last Packet (s)	Total Paket Rata-Rata	Between First and Last Packet (s)	Total Paket Rata-Rata
0 Port	1	59,133	18806	57,954	18544
	2	59,433	19071	57,744	17799
	3	59,502	19077	58,046	18593
	4	58,799	18737	57,99	18538
	5	59,87	19161	57,85	18299
	Rata-Rata		59,347	18970,4	57,917

Tabel 4.9 (Lanjutan)

Jumlah Port yang Diamati	Percobaan Ke-	02.00 (Jam Tidak Sibuk)		10.00 (Jam Sibuk)	
		Between First and Last Packet (s)	Total Paket Rata-Rata	Between First and Last Packet (s)	Total Paket Rata-Rata
1 Port	1	58,776	18716	57,098	17949
	2	58,179	18473	58,4	18473
	3	58,157	18445	58,2	18460
	4	58,157	18806	59	18707
	5	59,45	19029	58,4	18471
	Rata-Rata	58,544	18693,8	58,220	18412,0
2 Port	1	58,965	18799,5	58,95	18779,5
	2	58,94	18725	59,5	18803
	3	59,253	18870,5	59	18594,5
	4	60,031	19274	59,4	18740
	5	59,775	19065,5	59,3	18729,5
	Rata-Rata	59,393	18946,9	59,230	18729,3
3 Port	1	58,952	18742	59,5	19049
	2	58,722	18515	58,9	19551
	3	58,879	18697	59,8	18328
	4	59,718	19034	59,6	18747
	5	59,621	19031	59,3	18010
	Rata-Rata	58,178	18803,9	59,420	18737,0
4 Port	1	58,943	18683,25	59,789	18825,5
	2	60,025	19078,75	59,147	18430,75
	3	59,872	18625,25	59,66775	18731
	4	59,84	19072,5	59,895	18931,75
	5	59,61	18999,25	59,5945	18997,75
	Rata-Rata	59,658	18891,8	59,619	18783,4

(Sumber: Penelitian)

Perhitungan *delay* rata-rata pada jam tidak sibuk diperoleh sebagai berikut:

Perhitungan *delay* rata-rata untuk jumlah *port* yang diamati melalui *port mirroring* sebanyak 0 *port*.

$$Delay (ms) = \frac{59,347}{18.970,4} \times 1000$$

$$Delay (ms) = 3,129 ms$$

Perhitungan *delay* rata-rata untuk jumlah *port* yang diamati melalui *port mirroring* sebanyak 1 *port*.

$$Delay (ms) = \frac{58,544}{18.693,8} \times 1000$$

$$Delay (ms) = 3,132 ms$$

Perhitungan *delay* untuk jumlah *port* yang diamati melalui *port mirroring* sebanyak 2 *port*.

$$\text{Delay (ms)} = \frac{59,393}{18.946,9} \times 1000$$

$$\text{Delay (ms)} = 3,135 \text{ ms}$$

Perhitungan *delay* untuk jumlah *port* yang diamati melalui *port mirroring* sebanyak 3 *port*.

$$\text{Delay (ms)} = \frac{59,178}{18.803,9} \times 1000$$

$$\text{Delay (ms)} = 3,147 \text{ ms}$$

Perhitungan *delay* untuk jumlah *port* yang diamati melalui *port mirroring* sebanyak 4 *port*.

$$\text{Delay (ms)} = \frac{59,658}{18.891,8} \times 1000$$

$$\text{Delay (ms)} = 3,158 \text{ ms}$$

Delay rata-rata pada jam sibuk diperoleh dengan menggunakan cara perhitungan yang sama. Kedua hasil perhitungan *delay* rata-rata yang dihasilkan pada penelitian ini pada jam tidak sibuk dan jam sibuk ditunjukkan pada Tabel 4.10.

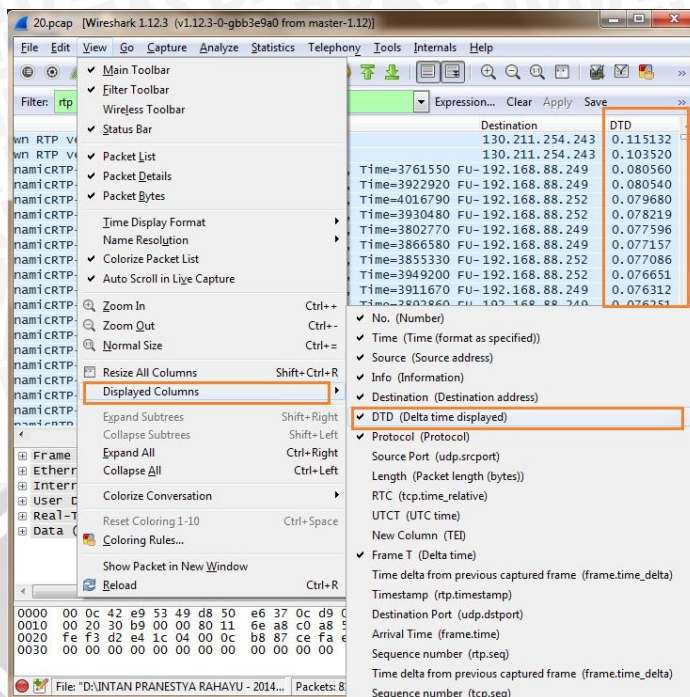
Tabel 4.10 *Delay* Rata-Rata

Jumlah <i>Port</i> yang Diamati	<i>Delay</i> Rata-Rata (ms)	
	Jam Tidak Sibuk	Jam Sibuk
0 <i>Port</i>	3,129	3,156
1 <i>Port</i>	3,132	3,162
2 <i>Port</i>	3,135	3,162
3 <i>Port</i>	3,147	3,174
4 <i>Port</i>	3,158	3,174

(Sumber: Perhitungan)

4.3.5.2 *Delay End-to-End* Maksimum

Nilai *delay end-to-end* maksimum didapatkan dari hasil *capturing* trafik menggunakan Wireshark (*View* → *Displayed Column* → *Delta Time Display* → sortir urutan nilai pada kolom *Delta Time Display* dari nilai maksimum ke minimum → nilai *delay end-to-end* maksimum merupakan nilai maksimum dari *Delta Time Display*) seperti yang ditunjukkan Gambar 4.25.



Gambar 4.25 Langkah Memperoleh *Delay End-to-End* Maksimum Pada Wireshark (Sumber: Penelitian)

Delay end-to-end maksimum yang dihasilkan pada penelitian ini ditunjukkan pada Tabel 4.11.

Tabel 4.11 Nilai *Delay End-to-End* Maksimum

Jumlah Port yang Diamati	Percobaan Ke-	<i>Delay End-to-End</i> (ms)	
		02.00 (Jam Tidak Sibuk)	10.00 (Jam Sibuk)
0 Port	1	104,227	103,58
	2	105,283	104,81
	3	105,747	121,17
	4	106,367	102,94
	5	106,203	106,21
	Rata-Rata	105,565	107,74
1 Port	1	105,473	107,43
	2	105,045	106,17
	3	107,254	116,24
	4	106,425	107,18
	5	106,594	107,28
	Rata-Rata	106,158	108,86
2 Port	1	109,402	109,90
	2	106,872	108,37
	3	105,758	109,02
	4	106,288	108,82
	5	108,535	111,66
	Rata-Rata	107,371	109,56

Tabel 4.11 (Lanjutan)

Jumlah Port yang Diamati	Percobaan Ke-	Delay End-to-End (ms)	
		02.00 (Jam Tidak Sibuk)	10.00 (Jam Sibuk)
3 Port	1	106,0	111,2
	2	107,333	110,9
	3	112,333	111,5
	4	107,0	109,7
	5	109,333	109,7
	Rata-Rata	108,4	110,6
4 Port	1	108,0	108,43
	2	107,375	106,88
	3	109,0	106,42
	4	109,525	110,67
	5	108,25	132,76
	Rata-Rata	108,43	113,03

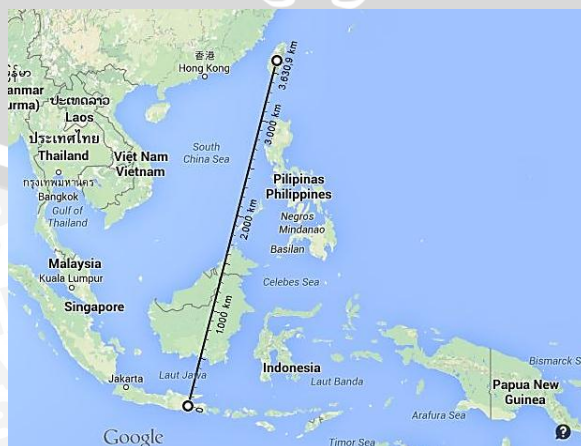
(Sumber: Penelitian)

4.3.5.3 Perhitungan Delay End-to-End Secara Teoritis

Perhitungan *delay end-to-end* secara teoritis terdiri dari *delay* propagasi, *delay* transmisi, *delay* proses (enkapsulasi dan dekapsulasi), dan *delay* antrian. Hasil penjumlahan kelima *delay* tersebut adalah nilai *delay end-to-end* seperti persamaan 2.3. Berikut ini merupakan perhitungan *delay end-to-end* secara teoritis pada jam tidak sibuk.

a. Delay Propagasi

Delay propagasi adalah waktu yang dibutuhkan oleh paket data untuk merambat dari *server* menuju *client* melalui media transmisi. Diketahui bahwa *server* penelitian berada di Taiwan sedangkan *client* penelitian berada di Malang. Jarak antara Taiwan-Malang adalah 3.630,9 km seperti yang ditunjukkan Gambar 4.26.



Gambar 4.26 Peta Jarak Taiwan-Malang
(Sumber: maps-google.com)

Berdasarkan persamaan 2.4 dan Tabel 2.3, maka perhitungan *delay* propagasi adalah sebagai berikut:

Diketahui,

Panjang kabel serat optik Taiwan-Malang	= 3.630,9 km
	= 3.630.900 meter
Panjang kabel serat optik (<i>Patch cord</i>)	= 3 meter
Panjang kabel UTP	= 6,6 meter

Untuk *delay* propagasi kabel serat optik,

$$t_p \text{ Serat Optik} = \frac{3.630.903}{(0,66 \times 3 \times 10^8)}$$

$$t_p \text{ Serat Optik} = 0,01833 \text{ s}$$

$$t_p \text{ Serat Optik} = \mathbf{18,33 \text{ ms}}$$

Untuk *delay* propagasi kabel UTP,

$$t_p \text{ UTP} = \frac{6,6}{(0,64 \times 3 \times 10^8)}$$

$$t_p \text{ UTP} = 3,4375 \times 10^{-8} \text{ sekon}$$

$$t_p \text{ UTP} = \mathbf{0,000034375 \text{ ms}}$$

Jadi, perhitungan total *delay* propagasi adalah sebagai berikut:

$$t_p = t_p \text{ Serat Optik} + t_p \text{ UTP}$$

$$t_p = 18,33 \text{ ms} + 0,000034375 \text{ ms}$$

$$t_p = \mathbf{18,33003438 \text{ ms}}$$

b. *Delay* Transmisi

Delay transmisi adalah waktu yang dibutuhkan untuk transmisi paket data. Perhitungan *delay* transmisi menggunakan persamaan 2.5, dengan diketahui data sekunder sebagai berikut:

- Panjang *header* IPv4 ($L_{\text{Header IPv4}}$) = 20 byte
- Panjang *header* NALU ($L_{\text{Header NALU}}$) = 1 byte
- Panjang *header* UDP ($L_{\text{Header UDP}}$) = 8 byte
- Panjang *header* RTP ($L_{\text{Header RTP}}$) = 12 byte
- Panjang *header* Ethernet ($L_{\text{Header Ethernet}}$) = 14 byte
- Panjang *header* CRC ($L_{\text{Header CRC}}$) = 4 byte

Panjang *header* (L'), didapatkan

$$L' = L_{Header\ Ipv4} + L_{Header\ NALU} + L_{Header\ UDP} + L_{Header\ RTP} + L_{Header\ Ethernet} + L_{Header\ CRC}$$

$$L' = (20 + 1 + 8 + 12 + 14 + 4) \times 8\ bit$$

$$L' = 472\ bit$$

Sedangkan diketahui dari data primer sebagai berikut:

- Panjang paket data (L) = 1.330 byte
= 10.640 bit
- *Bandwidth* Kanal (B) = 100 Mbps
= 104.857.600 bps

Sehingga, perhitungan *delay* transmisi adalah sebagai berikut:

$$t_t = \frac{(L + L')}{B}$$

$$t_t = \frac{(10.640 + 472)}{104.857.600}$$

$$t_t = 1,0597 \times 10^{-4}\ sekon$$

$$t_t = \mathbf{0,10597\ ms}$$

c. *Delay* Proses

Delay proses terdiri dari *delay* enkapsulasi dan *delay* dekapsulasi. *Delay* enkapsulasi adalah waktu yang dibutuhkan untuk proses pemaketan data dengan menambahkan *header* pada paket data sehingga paket data tersebut dapat dikirimkan ke tujuan. Proses enkapsulasi terjadi di *server*. *Delay* dekapsulasi adalah waktu yang dibutuhkan untuk proses pembacaan keseluruhan *header* dari sebuah paket yang diterima suatu *node*. Proses dekapsulasi terjadi di *client*.

Diketahui data sekunder diantaranya adalah sebagai berikut:

- Panjang *header* IPv4 ($L_{Header\ Ipv4}$) = 20 byte
- Panjang *header* NALU ($L_{Header\ NALU}$) = 1 byte
- Panjang *header* UDP ($L_{Header\ UDP}$) = 8 byte
- Panjang *header* RTP ($L_{Header\ RTP}$) = 12 byte
- Panjang *header* Ethernet ($L_{Header\ Ethernet}$) = 14 byte
- Panjang *header* CRC ($L_{Header\ CRC}$) = 4 byte

Perhitungan *delay* enkapsulasi dengan menggunakan persamaan 2.6 adalah sebagai berikut,

- Jumlah rata-rata paket yang dikirim (N) = 18.861,4 paket

- Panjang paket yang dikirim (L) = 1.330 byte
- Panjang *header* paket (L') = 59 byte
- Waktu pengiriman total data (T) = 60 s

Perhitungan kecepatan pemrosesan pada terminal pengirim adalah sebagai berikut,

$$C_{proses} = \frac{(L + L') \times N}{T}$$

$$C_{proses} = \frac{(1.330 + 59) \times 8 \times 18.861,4}{60}$$

$$C_{proses} = \frac{209.587.876,8}{60}$$

$$C_{proses} = 3.493.131,28 \text{ bps}$$

Jadi, perhitungan *delay* enkapsulasi adalah sebagai berikut:

$$t_{enc} = \frac{L_{Header IPv4} + L_{Header NALU} + L_{Header UDP} + L_{Header RTP} + L_{Header Ethernet} + L_{Header CRC}}{C_{proses}}$$

$$t_{enc} = \frac{(20 + 1 + 8 + 12 + 14 + 4) \times 8}{3.493.131,28}$$

$$t_{enc} = 1,3512 \times 10^{-4} \text{ sekon}$$

$$t_{enc} = \mathbf{0,13512 \text{ ms}}$$

Perhitungan *delay* dekapsulasi dengan menggunakan persamaan 2.7 adalah sebagai berikut,

- Jumlah rata-rata paket yang diterima (N) = 18.861,4 paket
- Panjang paket yang diterima (L) = 1.330 byte
- Panjang *header* paket (L') = 59 byte
- Waktu pengiriman total data (T) = 60 s

Perhitungan kecepatan pemrosesan pada terminal penerima adalah sebagai berikut,

$$C_{proses} = \frac{(L + L') \times N}{T}$$

$$C_{proses} = \frac{(1.330 + 59) \times 8 \times 18.861,4}{60}$$

$$C_{proses} = \frac{209.587.876,8}{60}$$

$$C_{proses} = 3.493.131,28 \text{ bps}$$

Jadi, perhitungan *delay* dekapsulasi adalah sebagai berikut:

$$t_{dec} = \frac{L_{Header\ Ipv4} + L_{Header\ NALU} + L_{Header\ UDP} + L_{Header\ RTP} + L_{Header\ Ethernet} + L_{Header\ CRC}}{C_{proses}}$$

$$t_{dec} = \frac{(20 + 1 + 8 + 12 + 14 + 4) \times 8\ bit}{3.493.131,28}$$

$$t_{dec} = 1,3512 \times 10^{-4}\ sekon$$

$$t_{dec} = \mathbf{0,13512\ ms}$$

Maka, *delay* proses diperoleh dengan menggunakan persamaan 2.8 sebagai berikut,

$$t_{proses} = t_{enc} + t_{dec}$$

$$t_{proses} = 0,13512\ ms + 0,13512\ ms$$

$$t_{proses} = \mathbf{0,27024\ ms}$$

d. Delay Antrian

Delay antrian adalah lamanya waktu yang diperlukan paket data untuk diproses pada suatu perangkat, terkait dengan pengiriman dan penerimaan paket data. Model antrian yang digunakan adalah M/M/1.

Perhitungan *delay* antrian *server* adalah sebagai berikut,

Diketahui dari data primer untuk perhitungan *delay* antrian *server* untuk tiap *client* diantaranya adalah sebagai berikut:

- Jumlah rata-rata paket yang dikirimkan (N) = 18.861,4 paket
- Waktu rata-rata pengiriman paket (T) = 60 sekon
- Panjang paket data (L) = 1.330 byte
- Panjang *header* paket (L') = 59 byte
- Kapasitas Kanal (C) = 100 Mbps
= 104.857.600 bps

Perhitungan kecepatan pelayanan *server* (μ) menggunakan persamaan 2.10 adalah sebagai berikut:

$$\mu = \frac{C}{(L + L')}$$

$$\mu = \frac{104.857.600}{(1.330 + 59) \times 8}$$

$$\mu = \mathbf{9.436,429\ paket/sekon}$$

Perhitungan kecepatan pengiriman paket *server* (λ) menggunakan persamaan 2.12 adalah sebagai berikut:

$$\lambda = \frac{N}{T}$$

$$\lambda = \frac{18.861,4}{60}$$

$$\lambda = 314,356 \text{ paket/sekon}$$

Perhitungan nilai faktor utilitas (ρ) menggunakan persamaan 2.11 adalah sebagai berikut:

$$\rho = \frac{\lambda}{\mu}$$

$$\rho = \frac{314,356}{9.436,429}$$

$$\rho = 0,033$$

Faktor utilitas ρ sebesar 0,033 menunjukkan bahwa kapasitas *server* yang digunakan kira-kira hanya 3,3% dari kapasitas maksimumnya. Jadi, perhitungan *delay* antrian (t_{queue}) *server* untuk tiap *client* adalah sebagai berikut:

$$t_{queue} = \frac{1}{\mu(1-\rho)}$$

$$t_{queue} = \frac{1}{9.436,429(1-0,033)}$$

$$t_{queue} = 1,09588 \times 10^{-4} \text{ sekon}$$

$$t_{queue} = 0,109588 \text{ ms}$$

Perhitungan *delay end-to-end* saat jam sibuk menggunakan cara yang sama. Hasil perhitungan *delay* propagasi, *delay* transmisi, *delay* proses, dan *delay* antrian ditunjukkan pada Tabel 4.12.

Tabel 4.12 Hasil Perhitungan *Delay*

Waktu Pengambilan Data	<i>Delay</i> Propagasi (ms)	<i>Delay</i> Transmisi (ms)	<i>Delay</i> Proses (ms)	<i>Delay</i> Antrian (ms)
02.00 (Jam Tidak Sibuk)	18,33003438	0,10597	0,27024	0,10959
10.00 (Jam Sibuk)	18,33003438	0,10597	0,273994	0,10957

Sehingga didapatkan nilai *delay end-to-end* pada jam tidak sibuk,

$$t_{end-to-end} = 18,33003438 \text{ ms} + 0,10597 \text{ ms} + 0,27024 \text{ ms} + 0,10959 \text{ ms}$$

$$t_{end-to-end} = 18,81583438 \text{ ms}$$

Sedangkan nilai *delay end-to-end* pada jam sibuk,

$$t_{end-to-end} = 18,33003438 \text{ ms} + 0,10597 \text{ ms} + 0,273994 \text{ ms} + 0,10957 \text{ ms}$$

$$t_{end-to-end} = \mathbf{18,81956838 \text{ ms}}$$

4.3.5.4 Perhitungan *Delay Total Sistem Video on Demand (VoD)*

Delay total (t_{total}) diperoleh dengan menjumlahkan *delay jaringan* ($t_{end-to-end}$) dengan *delay codec* (t_{codec}). Perhitungan *delay codec* sesuai dengan persamaan 2.14 dan Tabel 2.4, yaitu:

$$t_{codec} = t_{video} + t_{audio}$$

$$t_{codec} = 50 \text{ ms} + 60 \text{ ms}$$

$$t_{codec} = \mathbf{110 \text{ ms}}$$

Sehingga diperoleh *delay total sistem Video on Demand (VoD)* pada jam tidak sibuk,

$$t_{total} = t_{codec} + t_{end-to-end}$$

$$t_{total} = 110 \text{ ms} + 18,81583438 \text{ ms}$$

$$t_{total} = \mathbf{128,81583438 \text{ ms}}$$

Sedangkan, *delay total sistem Video on Demand (VoD)* pada jam sibuk,

$$t_{total} = t_{codec} + t_{end-to-end}$$

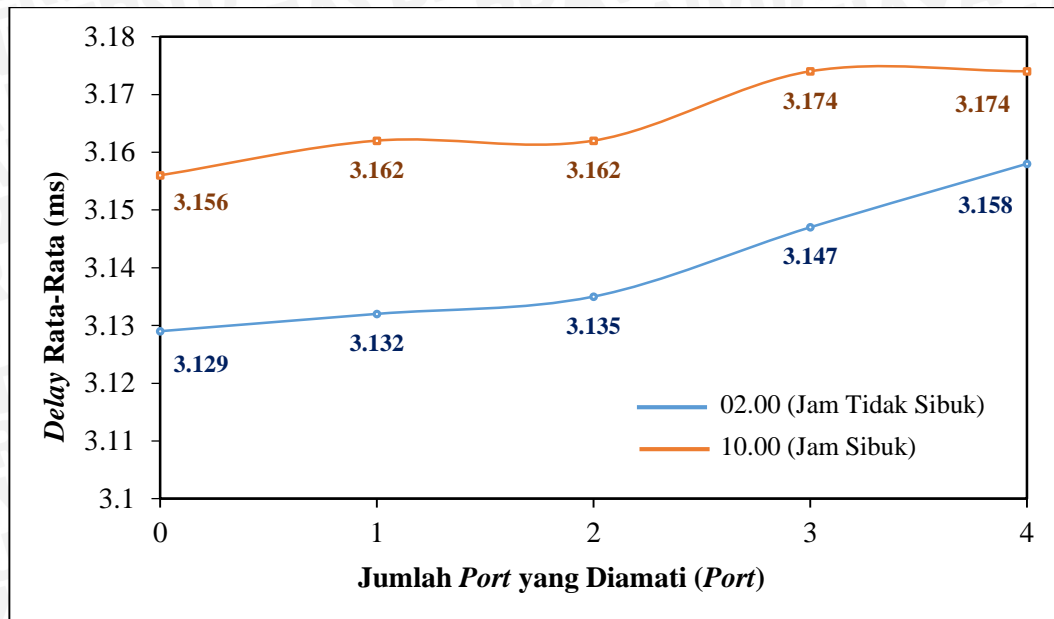
$$t_{total} = 110 \text{ ms} + 18,81956838 \text{ ms}$$

$$t_{total} = \mathbf{128,81956838 \text{ ms}}$$

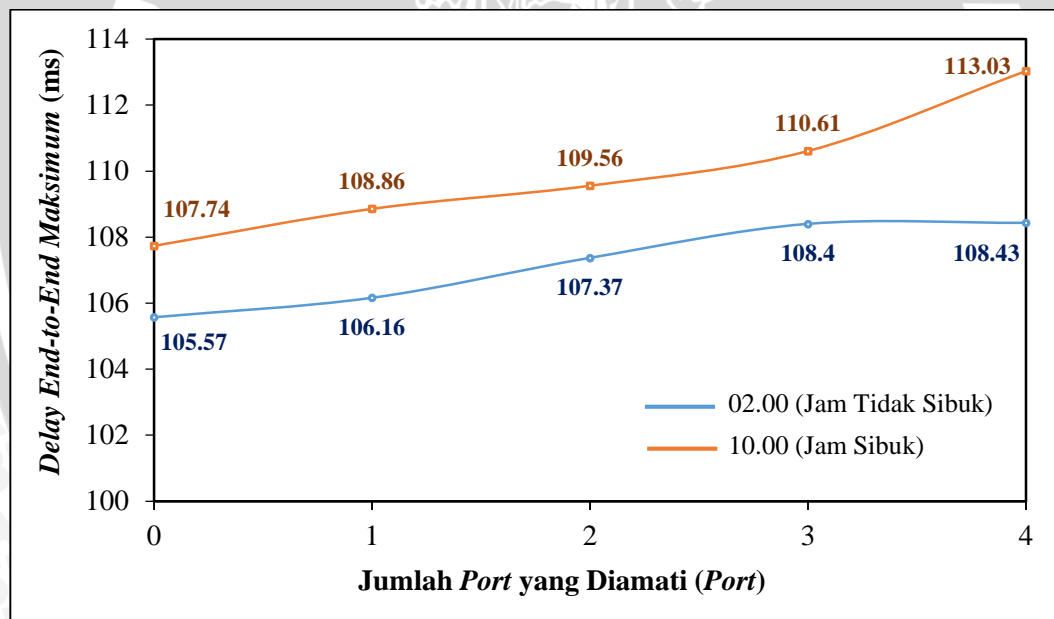
Berdasarkan perhitungan *delay end-to-end* secara teoritis, didapatkan *delay end-to-end* saat jam tidak sibuk sebesar 128,81583438 ms dan *delay end-to-end* saat jam sibuk sebesar 128,81956838 ms. *Delay end-to-end* saat jam tidak sibuk lebih kecil dibandingkan saat jam sibuk.

4.3.6 Analisis Hubungan antara *Delay* dengan Jumlah *Port* yang Diamati Melalui *Port Mirroring*

Hubungan parameter performansi *delay* rata-rata dan *delay end-to-end* maksimum terhadap jumlah *port* yang diamati melalui *port mirroring* saat jam tidak sibuk dan jam sibuk ditunjukkan pada Gambar 4.27 dan Gambar 4.28.



Gambar 4.27 Grafik Hubungan Antara Jumlah *Port* yang Diamati dengan *Delay Rata-Rata*
(Sumber: Perhitungan)



Gambar 4.28 Grafik Hubungan Antara Jumlah *Port* yang Diamati dengan *Delay End-to-End Maksimum*
(Sumber: Penelitian)

Untuk saat jam tidak sibuk, pada kondisi 0 *port* dimana tidak ada *port* yang diamati melalui *port mirroring* diperoleh *delay rata-rata* dan *delay end-to-end* maksimum terkecil yaitu 3,129 ms dan 105,57 ms. Kemudian *delay* semakin besar untuk kondisi 1 *port*, 2 *port*, 3 *port*, hingga 4 *port*. Pada kondisi 4 *port* yang diamati melalui *port mirroring* diperoleh *delay rata-rata* dan *delay end-to-end* maksimum terbesar yaitu 3,158 ms dan 108,43 ms.

Delay yang dihasilkan untuk saat jam sibuk, pada kondisi 0 *port* dimana tidak ada *port* yang diamati melalui *port mirroring* diperoleh *delay* rata-rata dan *delay end-to-end* maksimum terkecil yaitu 3,156 ms dan 107,74 ms. Kemudian *delay* semakin besar untuk kondisi 1 *port*, 2 *port*, 3 *port*, hingga 4 *port*. Pada kondisi 4 *port* yang diamati melalui *port mirroring* diperoleh *delay* rata-rata dan *delay end-to-end* maksimum terbesar yaitu 3,174 ms dan 113,03 ms.

Kedua hasil penelitian parameter performansi *delay* yaitu *delay rata-rata* dan *delay end-to-end* maksimum menunjukkan bahwa *delay* yang diperoleh semakin besar seiring dengan semakin bertambahnya jumlah *port* yang diamati melalui *port mirroring*. Hasil tersebut menunjukkan bahwa penerapan *port mirroring* pada *switch* menyebabkan penambahan nilai *delay*. Seperti yang sudah dijelaskan dalam analisis sebelumnya, bahwa terjadi proses duplikasi paket saat penerapan *port mirroring*. *Switch* membutuhkan waktu untuk melakukan proses duplikasi trafik paket yang terjadi di dalam *switch*. Berkaitan dengan penambahan jumlah *port* yang diamati mengakibatkan trafik paket pada *port mirroring* akan semakin padat atau bahkan terjadi penumpukan paket hasil duplikasi di *output queue port mirroring* sehingga waktu rata-rata paket menunggu di *ouput queue* menjadi semakin lama.

Hasil perhitungan dan penelitian untuk nilai *delay* masih kurang dari 150 ms. Berdasarkan standar *delay* Tabel 2.2 rekomendasi ITU-T G.114, jika nilai *delay* sebesar 0-150 ms dapat dinyatakan bahwa jaringan memiliki kualitas yang baik.