

# **KOMUNIKASI DATA**



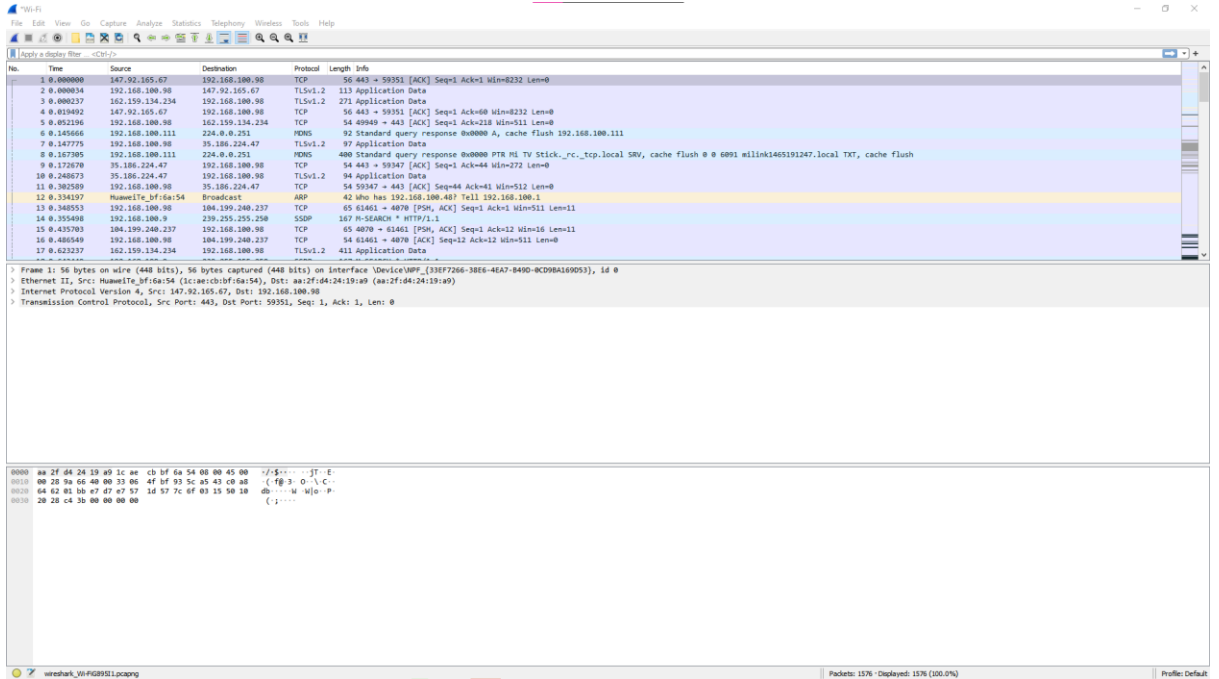
**Disusun**

**Oleh:**

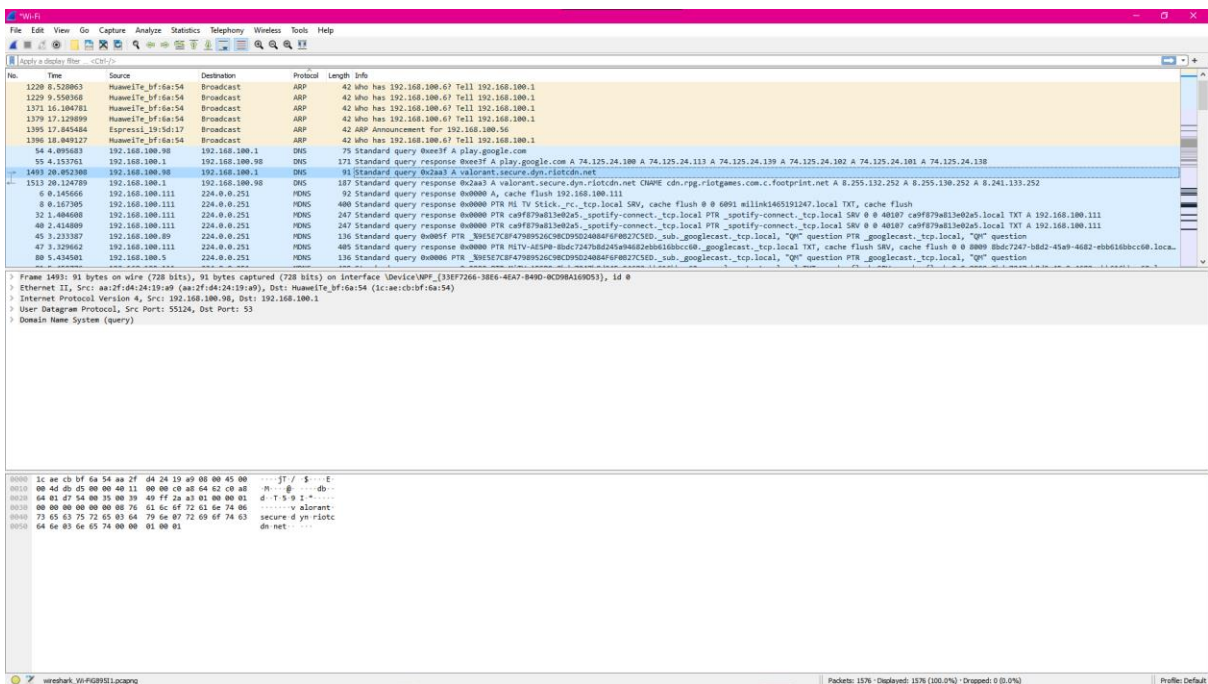
**Nama : Hanif Azfa Sadifatiasmi**  
**NIM : 09011382025135**  
**Dosen : Adi Hermansyah, S.KOM., M.T.**

**JURUSAN SISTEM KOMPUTER**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS SRIWIJAYA**  
**2022**

## 1. Menjalankan aplikasi sambal merecord jaringan pada wireshark



2. Pada proses ini saya menjalankan wireshark selama 22 detik sambil memainkan aplikasi game online yang tersambung ke server online game itu sendiri. Dapat dilihat pada gambar dibawah ini bahwa wireshark merecord jaringan server yang tersambung pada game yang saya mainkan



Saya memainkan game dengan developer RiotGames sehingga akan terdapat informasi mengenai server RiotGames pada hasil record wireshark seperti pada gambar diatas, valorant.secure.dyn.riotcdn.net merupakan service dari server RiotGames untuk game yang saya mainkan. Selain itu saya juga melakukan perhitungan throughput, packet loss, jitter, dan delay yang terjadi pada saat proses wireshark tersebut sedang berlangsung. Untuk hasil dan gambar untuk pencarian throughput, packet loss dan jitter dapat dilihat pada halaman selanjutnya.

Mencari Throughput, Packet Loss, Jitter dan Delay Pertama buka hasil statistic dari record pada wireshark tadi seperti pada gambar dibawah ini

File

Name:

C:\Users\hanif\AppData\Local\Temp\wireshark\_Wi-FiG895I1.pcapng

Length:

995 kB

Hash (SHA256):

97551081b302d64f991f49fbaf50fe690666f897d257b670aeb5a448df5c5276

Hash (RIPEMD160):

2db27acf8a7e59ecf7339f4aa9219ab5b665663c

Hash (SHA1):

86ace64f0313d64ca840d8bf49e21e2ca2d162d4

Format:

Wireshark/... - pcapng

Encapsulation:

Ethernet

Time

First packet:

2022-03-21 19:27:16

Last packet:

2022-03-21 19:27:38

Elapsed:

00:00:22

Capture

Hardware:

Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (with SSE4.2)

OS:

64-bit Windows 10 (20H2), build 19042

Application:

Dumpcap (Wireshark) 3.6.1 (v3.6.1-0-ga0a473c7c1ba)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

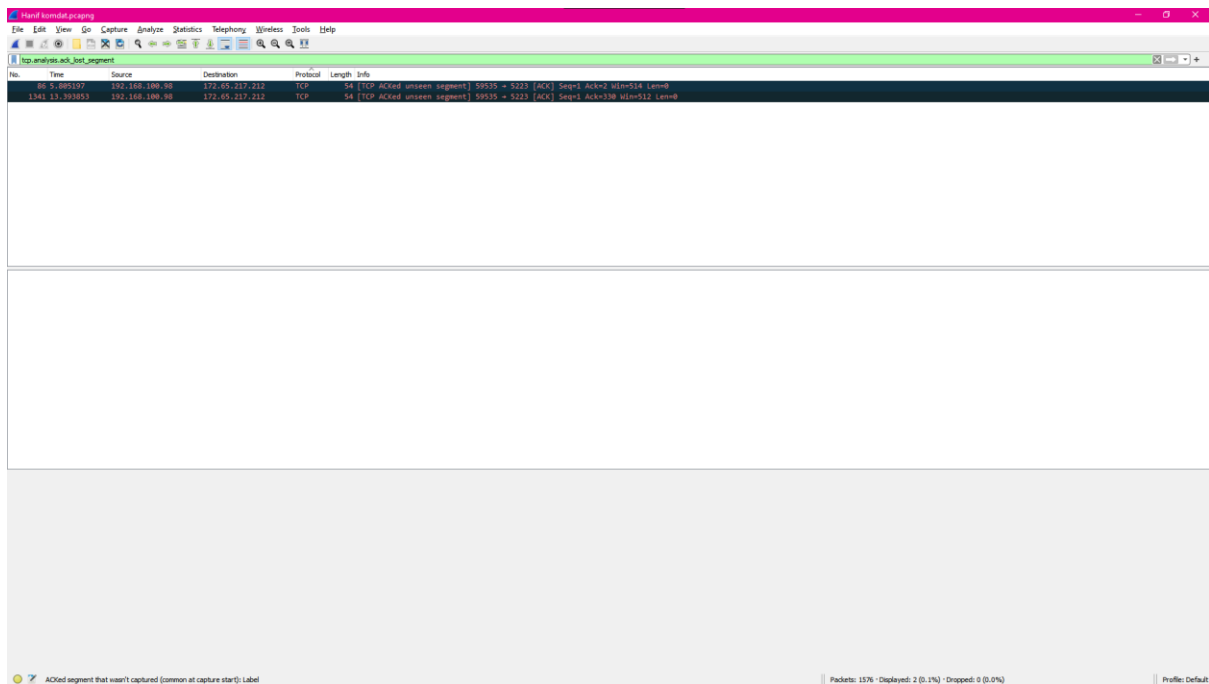
Statistics

Measurement	Captured	Displayed	Marked
Packets	1576	1576 (100.0%)	—
Time span, s	22.076	22.076	—
Average pps	71.4	71.4	—
Average packet size, B	598	598	—
Bytes	942362	942362 (100.0%)	0
Average bytes/s	42 k	42 k	—
Average bits/s	341 k	341 k	—

Ini adalah hasil statistic dari record wireshark yang saya lakukan, terdapat 1576 paket yang diproses dengan rentang waktu 22,076 second. Untuk mencari throughput bisa dilakukan dengan jumlah bytes data dibagi time span kemudian hasilnya yang bernilai bytes dikalikan dengan 8 bit(1 bytes = 8 bit). Sehingga didapat throughputnya sekitar 341k.

```
1.througput =jumlah bytes/waktu pengiriman=
              942362/22,076=42,68 bit/s
              lalu dikali 8 karena jumlah 1kb ialah 8bit
              42,68 x 8 = 341,49 k bits/s
```

Selanjutnya adalah mencari packet loss, untuk packet loss sendiri kita harus mencari tcp.analysis.segment untuk packet loss seperti pada gambar dibawah ini



Pada gambar didapat ada 2 packet loss dari 1576 packet. Maka kita hitung dengan cara paket dikirim dikurang paket diterima hasil dikurang 2 lalu dibagi paket dikirim kemudian dikali 100

Pada percobaan saya mendapatkan nilai packet loss sebesar 0,126

$$\begin{aligned}
 \text{2. packet loss} &= (\text{paket dikirim} - \text{paket diterima}) / \text{paket dikirim} \times 100\% \\
 &= ((1576 - 1574) / 1576) \times 100\% \\
 &= (2 / 1576) \times 100 \\
 &= 0,126
 \end{aligned}$$

Untuk jitter dan delay dapat dicari dengan mengexport hasil record wireshark menjadi CSV atau file dengan format excel seperti pada gambar dibawah

	C	I	J	K	L	M	N	O	P	Q
1448				9,486494	9,488712	0,002217669		-1,2E-14	0,002217669	0,002218
1449				9,488712	9,490929	0,002217669		1,78E-15	0,002217669	0,002218
1450				9,490929	9,493147	0,002217669		1,07E-14	0,002217669	0,002218
1451				9,493147	9,495365	0,002217669		-1,1E-14	0,002217669	0,002218
1452				9,495365	9,497582	0,002217669		-1,8E-15	0,002217669	0,002218
1453				9,497582	9,499800	0,002217669		1,78E-15	0,002217669	0,002218
1454				9,499800	9,502018	0,002217669		1,07E-14	0,002217669	0,002218
1455				9,502018	9,504235	0,002217669		-1,1E-14	0,002217669	0,002218
1456				9,504235	9,506453	0,002217669		-1,8E-15	0,002217669	0,002218
1457				9,506453	9,508671	0,002217669		1,78E-15	0,002217669	0,002218
1458				9,508671	9,510888	0,002217669		1,07E-14	0,002217669	0,002218
1459				9,510888	9,513106	0,002217669		-1,1E-14	0,002217669	0,002218
1460				9,513106	9,515324	0,002217669		-1,8E-15	0,002217669	0,002218
1461				9,515324	9,517541	0,002217669		1,78E-15	0,002217669	0,002218
1462				9,517541	9,519759	0,002217669		1,07E-14	0,002217669	0,002218
1463				9,519759	9,521977	0,002217669		-1,1E-14	0,002217669	0,002218
1464				9,521977	9,524194	0,002217669		-1,8E-15	0,002217669	0,002218
1465				9,524194	9,526412	0,002217669		1,24E-14	0,002217669	0,002218
1466				9,526412	9,528630	0,002217669		-1,1E-14	0,002217669	0,002218
1467				9,528630	9,530847	0,002217669		-1,8E-15	0,002217669	0,002218
1468				9,530847	9,533065	0,002217669				
1469										
1470					Total Delay	9,53306492			Total Jitter	9,535215
1471					Rata-rata Delay	0,00605274			Rata-rata Jitter	-1565,46
1472										
1473										
1474										
1475										
1476										
1477										
1478										
1479										
1480										
1481										
1482										
1483										
1484										
1485										

Mencari Delay sendiri adalah dengan pengurangan, yaitu waktu pertama dibagi waktu kedua kemudian ditotalkan, untuk rata ratanya dibagi 1575. Untuk jitter sendiri juga sama namun yang dikurangi adalah delay kedua dikurang delay pertama kemudian ditotalkan, untuk rata rata jitter dibagi dengan 1575 karena pada jitter total data berkurang 1.

```

3.delay = time2-time1
total delay = 9,53306492 s
rata-rata delay = total delay/(jumlah paket-1)=
9,53306492/1575 = 0,00605274

4.jitter = delay2-delay1
total jitter = 9,535214588 s
rata-rata jitter = total jitter/(jumlah paket-1)=
9,535214588/1575 = -1565,464785

```

Dan pada gambar dibawah ini adalah proses komunikasi datanya

```
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 59351, Seq: 1, Ack: 1, Len: 0
  Source Port: 443
  Destination Port: 59351
  [Stream index: 0]
  [Conversation completeness: Incomplete (12)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 3881246039
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2087650069
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 8232
  [Calculated window size: 8232]
```

Yang dimana port menjadi pembeda dalam proses transportasi data.

Link Repository Github : [https://github.com/hanifazfas/Komunikasi\\_Data](https://github.com/hanifazfas/Komunikasi_Data)