

**ANALISA PERFORMANSI *REAL TIME STREAMING*  
*PROTOKOL (RTSP)* UNTUK APLIKASI VIDEO STREAMING  
PADA JARINGAN MOBILE IPv4 DENGAN MENGGUNAKAN  
*VIDEO LAN CLIENT (VLC)***

**SKRIPSI**



**Disusun Oleh:  
BALQIS FITRI ANNA  
09.18.086**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1  
FAKULTAS TEKNIK INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2014**

REPUBLICAN PARTY OF THE STATE OF CALIFORNIA  
COUNTY OF LOS ANGELES  
COUNTY CLERK  
1000 WEST 10TH STREET  
LOS ANGELES, CALIFORNIA 90057

1000

1000  
1000  
1000

THE COUNTY OF LOS ANGELES  
COUNTY CLERK  
1000 WEST 10TH STREET  
LOS ANGELES, CALIFORNIA 90057  
P. 008

# LEMBAR PERSETUJUAN

**ANALISA PERFORMANSI *REAL TIME STREAMING PROTOKOL* (RTSP)  
UNTUK APLIKASI VIDEO STREAMING PADA JARINGAN MOBILE  
IPv4 DENGAN MENGGUNAKAN *VIDEO LAN CLIENT* (VLC)**

## SKRIPSI


*Disusun dan diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Teknik Informatika Strata Satu (S-1)*


Disusun Oleh :  
**BALQIS FITRI ANNA**  
09.18.086

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

  
**Ir. Yudi Limpraptono, MT.**  
NIP.Y. 1039500274

  
**Michael Ardita, ST, MT.**  
NIP.P. 1031000434

Ketua Jurusan Teknik Informatika S-1

  
**Joseph Dedy Irawan, ST, MT**  
NIP. 197404162005011002

**PROGRAM STUDI TEKNIK INFORMATIKA S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

2014

**MILIK  
PERPUSTAKAAN  
ITN MALANG**

**ANALISA PERFORMANSI *REAL TIME STREAMING*  
PROTOKOL (RTSP) UNTUK APLIKASI VIDEO STREAMING  
PADA JARINGAN MOBILE IPv4 DENGAN MENGGUNAKAN  
VIDEO LAN CLIENT (VLC)**

**Balqis Fitri Anna**

**09.18.086**

**Program Studi Teknik Informatika S-1**

**Fakultas Teknologi Industri**

**Institut Teknologi Nasional Malang**

**Jl. Raya Karanglo Km 2 Malang**

**Email : [fitrinarendra@ymail.com](mailto:fitrinarendra@ymail.com)**

**Dosen Pembimbing : 1. Ir. Yudi Limpraptono, MT.**

**2. Michael Ardita, ST, MT.**

***Abstrak***

*Streaming memungkinkan menampilkan media tanpa harus menunggu keseluruhan media diterima lengkap terlebih dahulu oleh client. Pada skripsi ini telah dibuat sebuah jaringan kecil yang berbasis Mobile IPv4.*

*Pengukuran Quality of Service berupa delay, jitter, packet loss dan throughput dilakukan dengan cara streaming video dari server ke client dengan menggunakan aplikasi VLC, kemudian menangkap paket-paket tersebut dengan menggunakan aplikasi Wireshark.*

*Dari hasil percobaan diketahui bahwa QoS pada parameter delay saat proses streaming dari server ke client dalam jaringan local sangat rendah yakni 8,3% dibandingkan saat proses streaming dari server ke client pada jaringan virtual server. Hal ini disebabkan ketidak stabilan jaringan provider. Untuk jitter masih memenuhi standard QoS, karena dari hasil pengukuran nilai jitter masih dibawah 30 ms yang merupakan standard jitter yang baik.*

*Kata Kunci : IPv4, Mobile IPv4, Video streaming, Delay, Jitter, Packet Loss, dan Throughput.*



## **SURAT PERNYATAAN ORISINALITAS**

Yang bertanda tangan di bawah ini:

Nama : Balqis Fitri Anna

NIM : 0918086

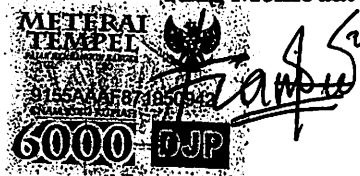
Program Studi : Teknik Informatika S-1 ITN Malang

Dengan ini menyatakan bahwa Skripsi dengan judul “ *ANALISA PERFORMANSI REAL TIME STREAMING PROTOKOL (RTSP) UNTUK APLIKASI VIDEO STREAMING PADA JARINGAN MOBILE IPv4 DENGAN MENGGUNAKAN VIDEO LAN CLIENT (VLC)*” yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, Februari 2014

Yang Membuat Pernyataan,



Balqis Fitri Anna  
NIM : 0918086

## KATA PENGANTAR

Alhamdulillah Puji Syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena telah memberikan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul *“Analisa Performansi Real Time Streaming Protokol (RTSP) Untuk Aplikasi Video Streaming Pada Jaringan Mobile IPv4 Dengan Menggunakan Video LAN Client (VLC)”* sesuai dengan waktu yang ditentukan.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang. Ucapan terima kasih yang sebesar-besarnya kami sampaikan kepada:

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor ITN Malang.
2. Bapak Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Bapak Joseph Dedy Irawan, ST, MT selaku Ketua Jurusan Teknik Informatika ITN Malang.
4. Bapak Ir. Yudi Limpraptono, MT selaku Dosen Pembimbing I.
5. Bapak Michael Ardita, ST, MT selaku Dosen Pembimbing II.
6. Semua pihak yang telah membantu terselesaikannya skripsi ini.
7. Bapak dan Ibu serta saudara-saudara kami yang selalu memberikan do'a dan dorongan serta semangat.
8. Teman-teman dan semua yang telah membantu dalam penyelesaian penulisan skripsi ini.

Penulis menyadari bahwa skripsi masih jauh dari sempurna, oleh karena itu penulis mengharapkan kritik dan saran dari pembaca. Semoga skripsi ini bisa bermanfaat bagi para pembaca sekalian.

Malang, Februari 2014

Penyusun

## KATA PENGANTAR

Keberhasilan bagi seorang penulis tergantung kepada Tuhan Yang Maha Esa. Kami telah menuliskan beberapa hal sebagai bentuk penghargaan kepada Tuhan Yang Maha Esa. Kami juga mengucapkan terima kasih kepada keluarga kami yang telah mendukung kami dalam menyelesaikan buku ini.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Sarjana (S1) Teknik Informatika. Melalui Teknologi Informatika di Institut Teknologi Nasional Malang, tempat kami kuliah yang telah banyak memberi pengalaman kepada kami.

1. Bapak Ir. Soeparno (Ipa) MT selaku Dekan ITN Malang.
2. Bapak Ir. Anwar Sobandi MT selaku Dekan Fakultas Teknologi Informatika ITN Malang.
3. Bapak Joseph Dedy (Ipa) ST MT selaku Ketua Jurusan Teknik Informatika ITN Malang.
4. Bapak Ir. Yudi (Ipa) MT selaku Dosen Pembimbing I.
5. Bapak Mochamad (Ipa) MT selaku Dosen Pembimbing II.
6. Semua pihak yang telah membantu terselesaikannya skripsi ini.
7. Bapak dan Ibu serta saudara-saudara kami yang selalu memberikan doa dan dukungan serta semangat.
8. Teman-teman dan semua yang telah membantu dalam penyelesaian penelitian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Semoga skripsi ini bisa bermanfaat bagi para pembaca sekalian.

Malang, 1 Januari 2014

Pengantar

## DAFTAR ISI

<b>LEMBAR PERSETUJUAN SKRIPSI</b> .....	<b>i</b>
<b>ABSTRAK</b> .....	<b>ii</b>
<b>SURAT PERNYATAAN ORISINALITAS</b> .....	<b>iii</b>
<b>KATA PENGANTAR</b> .....	<b>iv</b>
<b>DAFTAR ISI</b> .....	<b>v</b>
<b>DAFTAR GAMBAR</b> .....	<b>vii</b>
<b>DAFTAR TABEL</b> .....	<b>x</b>
<b>DAFTAR GRAFIK</b> .....	<b>xi</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian.....	2
1.4 Batasan Masalah .....	2
1.5 Metode Penelitian.....	2
1.6 Sistematika Penulisan.....	3
<b>BAB II LANDASAN TEORI</b> .....	<b>5</b>
2.1 Internet Protokol versi 4 ( <i>IPv4</i> ).....	5
2.2 <i>Mobile IPv4</i> .....	5
2.3 OSI Layer .....	6
2.4 <i>RTSP (Real Time Streaming Protokol)</i> .....	8
2.5 <i>VLC (Videl LAN Client)</i> .....	8
2.6 Wireshark .....	8
2.7 <i>Parameter Video Streaming</i> .....	8
2.8 Topologi Jaringan Komputer .....	9
2.9 Sistem Operasi Linux.....	13
<b>BAB III PERANCANGAN DAN DESAIN APLIKASI</b> .....	<b>15</b>
3.1 Analisa system.....	15
3.1.1 Deskripsi Umum .....	15
3.1.2 Fitur Aplikasi <i>Video LAN Client</i> .....	15
3.1.3 Analisa Kebutuhan Sistem .....	16
3.2 Perancangan Sistem.....	16

DAFTAR ISI

i	.....	DAFTAR ISI
ii	.....	ABSTRAK
iii	.....	SINGKAPAN
iv	.....	KATA PENGANTAR
v	.....	DAFTAR ISI
vi	.....	DAFTAR GAMBAR
vii	.....	DAFTAR TABEL
viii	.....	DAFTAR GRAFIK
1	.....	BAB I PENDAHULUAN
2	1.1	1.1 Latar Belakang
3	1.2	1.2 Rumusan Masalah
4	1.3	1.3 Tujuan Penelitian
5	1.4	1.4 Batasan Masalah
6	1.5	1.5 Metode Penelitian
7	1.6	1.6 Sistematika Penelitian
8	.....	BAB II LANDASAN TEORI
9	2.1	2.1 Konsep Tokoh (WTA)
10	2.2	2.2 Model WTA
11	2.3	2.3 Uji WTA
12	2.4	2.4 Uji WTA (WTA)
13	2.5	2.5 Uji WTA (WTA)
14	2.6	2.6 WTA (WTA)
15	2.7	2.7 WTA (WTA)
16	2.8	2.8 Topologi Jaringan Komputer
17	2.9	2.9 Sistem Operasi Jaringan
18	.....	BAB III PERANCANGAN DAN IMPLEMENTASI
19	3.1	3.1 Analisis Sistem
20	3.1.1	3.1.1 Deskripsi Sistem
21	3.1.2	3.1.2 Diagram Alir Data (DAD)
22	3.1.3	3.1.3 Analisis Kelembutan Sistem
23	3.2	3.2 Perancangan Sistem

3.2.1	Arsitektur Aplikasi <i>Video LAN Client</i> .....	16
3.2.2	Diagram Blok .....	18
3.3	Konfigurasi.....	18
3.3.1	Konfigurasi <i>Server Side</i> .....	19
3.3.2	Konfigurasi <i>Client Side</i> .....	25
3.3.3	Konfigurasi <i>Wireshark</i> .....	27
<b>BAB IV</b>	.....	<b>29</b>
<b>IMPLEMENTASI DAN ANALISIS</b>	.....	<b>29</b>
4.1	Implementasi Sistem .....	29
4.1.1	Implementasi perangkat lunak .....	29
4.1.2	Implementasi perangkat keras .....	29
4.2	Pengujian Sistem.....	30
4.2.1	Pengujian hasil pengukuran <i>delay</i> transmisi .....	30
4.2.2	Pengujian hasil pengukuran <i>Throughput</i> .....	53
4.2.3	Pengujian hasil pengukuran <i>PacketLoss</i> dan <i>Jitter</i> .....	69
<b>BAB V</b>	.....	<b>89</b>
<b>PENUTUP</b>	.....	<b>89</b>
5.1	Kesimpulan.....	89
5.2	Saran .....	90
<b>DAFTAR PUSTAKA</b>	.....	<b>91</b>
<b>LAMPIRAN</b>		



## DAFTAR GAMBAR

### BAB III PERANCANGAN DAN DESAIN APLIKASI

Gambar 3.1. Arsitektur <i>VLC</i> .....	17
Gambar 3.2. Diagram Blok Sistem .....	18
Gambar 3.3. Konfigurasi <i>server</i> tahap 1 .....	19
Gambar 3.4 Konfigurasi <i>server</i> tahap 2 .....	20
Gambar 3.5 Konfigurasi <i>server</i> tahap 3 .....	21
Gambar 3.6 Konfigurasi <i>server</i> tahap 4.....	21
Gambar 3.7 Konfigurasi <i>server</i> tahap 5 .....	22
Gambar 3.8 Konfigurasi <i>server</i> tahap 6.....	23
Gambar 3.9 Konfigurasi <i>server</i> tahap 7 .....	24
Gambar 3.10 Hasil <i>streaming server side</i> .....	24
Gambar 3.11 Konfigurasi <i>client</i> tahap 1 .....	25
Gambar 3.12 Konfigurasi <i>Client</i> tahap 2 .....	26
Gambar 3.13 Hasil video <i>streaming pada client</i> .....	26
Gambar 3.14 Konfigurasi <i>Wireshark</i> tahap 1 .....	27
Gambar 3.15 Konfigurasi <i>Wireshark</i> tahap 2 .....	28

### BAB IV IMPLEMENTASI DAN ANALISIS

Gambar 4.1 <i>Wireshark Capture ICMP Request</i> 1.....	31
Gambar 4.2 <i>Wireshark Capture ICMP Replay</i> 1 .....	31
Gambar 4.3 <i>Wireshark Capture ICMP Request</i> 2.....	32
Gambar 4.4 <i>Wireshark Capture ICMP Replay</i> 2 .....	32
Gambar 4.5 <i>Wireshark Capture ICMP Request</i> 3.....	33
Gambar 4.6 <i>Wireshark Capture ICMP Replay</i> 3 .....	33
Gambar 4.7 <i>Wireshark Capture ICMP Request</i> 4.....	34
Gambar 4.8 <i>Wireshark Capture ICMP Replay</i> 4 .....	34
Gambar 4.9 <i>Wireshark Capture ICMP Request</i> 1.....	36
Gambar 4.10 <i>Wireshark Capture ICMP Replay</i> 1 .....	37

Gambar 4.11 <i>Wireshark Capture ICMP Request 2</i> .....	37
Gambar 4.12 <i>Wireshark Capture ICMP Replay 2</i> .....	38
Gambar 4.13 <i>Wireshark Capture ICMP Request 3</i> .....	38
Gambar 4.14 <i>Wireshark Capture ICMP Replay 3</i> .....	39
Gambar 4.15 <i>Wireshark Capture ICMP Request 4</i> .....	39
Gambar 4.16 <i>Wireshark Capture ICMP Replay 4</i> .....	40
Gambar 4.17 <i>Wireshark Capture ICMP Request 1</i> .....	42
Gambar 4.18 <i>Wireshark Capture ICMP Replay 1</i> .....	42
Gambar 4.19 <i>Wireshark Capture ICMP Request 2</i> .....	43
Gambar 4.20 <i>Wireshark Capture ICMP Replay 2</i> .....	43
Gambar 4.21 <i>Wireshark Capture ICMP Request 3</i> .....	44
Gambar 4.22 <i>Wireshark Capture ICMP Replay 3</i> .....	44
Gambar 4.23 <i>Wireshark Capture ICMP Request 4</i> .....	45
Gambar 4.24 <i>Wireshark Capture ICMP Replay 4</i> .....	45
Gambar 4.25 <i>Wireshark Capture ICMP Request 1</i> .....	47
Gambar 4.26 <i>Wireshark Capture ICMP Replay 1</i> .....	48
Gambar 4.27 <i>Wireshark Capture ICMP Request 2</i> .....	48
Gambar 4.28 <i>Wireshark Capture ICMP Replay 2</i> .....	49
Gambar 4.29 <i>Wireshark Capture ICMP Request 3</i> .....	49
Gambar 4.30 <i>Wireshark Capture ICMP Replay 3</i> .....	50
Gambar 4.31 <i>Wireshark Capture ICMP Request 4</i> .....	50
Gambar 4.32 <i>Wireshark Capture ICMP Replay 4</i> .....	51
Gambar 4.33 <i>Video 1 Properties</i> .....	53
Gambar 4.34 <i>Video 2 Properties</i> .....	54
Gambar 4.35 <i>Video 3 Properties</i> .....	55
Gambar 4.36 <i>Throughput video 1</i> .....	56
Gambar 4.37 <i>Throughput video 2</i> .....	57
Gambar 4.38 <i>Throughput video 3</i> .....	58
Gambar 4.39 <i>Throughput video 1</i> .....	59
Gambar 4.40 <i>Throughput video 2</i> .....	60

Gambar 4.41 <i>Throughput video 3</i> .....	61
Gambar 4.42 <i>Throughput video 1</i> .....	62
Gambar 4.43 <i>Throughput video 2</i> .....	64
Gambar 4.44 <i>Throughput video 3</i> .....	65
Gambar 4.45 <i>Throughput video 1</i> .....	66
Gambar 4.46 <i>Throughput video 2</i> .....	67
Gambar 4.47 <i>Throughput video 3</i> .....	68
Gambar 4.48 <i>Video 1 Properties</i> .....	70
Gambar 4.49 <i>Video 2 Properties</i> .....	71
Gambar 4.50 <i>Video 3 Properties</i> .....	72
Gambar 4.51 <i>Analisa packetloss dan jitter video 1</i> .....	73
Gambar 4.52 <i>Analisa packetloss dan jitter video 2</i> .....	74
Gambar 4.53 <i>Analisa packetloss dan jitter video 3</i> .....	75
Gambar 4.54 <i>Analisa packetloss dan jitter video 1</i> .....	77
Gambar 4.55 <i>Analisa packetloss dan jitter video 2</i> .....	78
Gambar 4.56 <i>Analisa packetloss dan jitter video 3</i> .....	79
Gambar 4.57 <i>Analisa packetloss dan jitter video 1</i> .....	81
Gambar 4.58 <i>Analisa packetloss dan jitter video 2</i> .....	82
Gambar 4.59 <i>Analisa packetloss dan jitter video 3</i> .....	83
Gambar 4.60 <i>Analisa packetloss dan jitter video 1</i> .....	85
Gambar 4.61 <i>Analisa packetloss dan jitter video 2</i> .....	86
Gambar 4.62 <i>Analisa packetloss dan jitter video 3</i> .....	87

## DAFTAR TABEL

### BAB IV IMPLEMENTASI DAN ANALISIS

Tabel 4.1	Standarisasi ITU-T <i>Delay</i> .....	30
Tabel 4.2	Hasil pengujian <i>delay</i> transmisi dari perangkat <i>mobile</i> sebagai <i>client</i> ke komputer <i>server</i> . .....	35
Tabel 4.3	Hasil pengujian <i>delay</i> transmisi dari perangkat <i>mobile client</i> ke komputer <i>server</i> .....	41
Tabel 4.4	Hasil pengujian <i>delay</i> transmisi dari computer <i>client</i> ke computer <i>server</i> menggunakan kabel cat 6. ....	46
Tabel 4.5	Hasil pengujian <i>delay</i> transmisi dari perangkat <i>mobile client</i> ke komputer <i>server</i> menggunakan <i>wifi</i> . ....	51
Tabel 4.6	Perbandingan <i>bitrate</i> dengan <i>throughput bitrate</i> pada Kabel UTP CAT 5.....	59
Tabel 4.7	Perbandingan <i>bitrate</i> dengan <i>throughput bitrate</i> pada Kabel UTP CAT 5E.. ....	62
Tabel 4.8	Perbandingan <i>bitrate</i> dengan <i>throughput bitrate</i> pada Kabel UTP CAT 6.....	66
Tabel 4.9	Perbandingan <i>bitrate</i> dengan <i>throughput bitrate</i> menggunakan <i>wifi</i> .....	69
Tabel 4.10	Hasil pengujian <i>Packetloss</i> dan <i>Jitter</i> menggunakan kabel CAT 5.....	76
Tabel 4.11	Hasil pengujian <i>Packetloss</i> dan <i>Jitter</i> menggunakan kabel CAT 5e.....	80
Tabel 4.12	Hasil pengujian <i>Packetloss</i> dan <i>Jitter</i> menggunakan kabel CAT 6.....	84
Tabel 4.13	Hasil pengujian <i>Packetloss</i> dan <i>Jitter</i> menggunakan <i>wifi</i> .. .....	88

**DAFTAR TABEL**

**SARAF IMPLEMENTASI WAKELAS**

36	.....	Tabel 4.1 Standar UUT Desy	Tabel 4.1
	.....	Hasil pengujian deskriptor minimal dan pengujian mobile sebagai	Tabel 4.2
37	.....	View ke komputer server	Tabel 4.3
	.....	Hasil pengujian deskriptor minimal dan pengujian mobile view ke	Tabel 4.4
41	.....	komputer server	Tabel 4.5
	.....	Hasil pengujian deskriptor minimal dan komputer view ke komputer	Tabel 4.6
44	.....	server menggunakan label cat	Tabel 4.7
	.....	Hasil pengujian deskriptor minimal dan pengujian mobile view ke	Tabel 4.8
51	.....	komputer server menggunakan uji	Tabel 4.9
	.....	Perbandingan view dengan pengujian view ke	Tabel 4.10
59	.....	UTP CAT 3	Tabel 4.11
	.....	Perbandingan view dengan pengujian view ke	Tabel 4.12
62	.....	UTP CAT 3E	Tabel 4.13
	.....	Perbandingan view dengan pengujian view ke	Tabel 4.14
66	.....	UTP CAT 6	Tabel 4.15
	.....	Perbandingan view dengan pengujian view ke menggunakan	Tabel 4.16
69	.....	wifi	Tabel 4.17
	.....	Hasil pengujian view ke dan view menggunakan label	Tabel 4.18
73	.....	CAT 5	Tabel 4.19
	.....	Hasil pengujian view ke dan view menggunakan label	Tabel 4.20
80	.....	CAT 5e	Tabel 4.21
	.....	Hasil pengujian view ke dan view menggunakan label	Tabel 4.22
83	.....	CAT 6	Tabel 4.23
88	.....	Hasil pengujian view ke dan view menggunakan wifi	Tabel 4.24

## DAFTAR GRAFIK

### BAB IV IMPLEMENTASI DAN ANALISIS

Grafik 4.1. Hasil pengujian <i>delay</i> transmisi dari perangkat <i>mobile</i> sebagai <i>client</i> ke komputer <i>server</i> .....	36
Grafik 4.2. Hasil pengujian <i>delay</i> transmisi dari perangkat <i>mobile</i> sebagai <i>client</i> ke komputer <i>server</i> .....	41
Grafik 4.3. <i>Delay</i> Transmisi dari perangkat computer <i>client</i> ke komputer <i>server</i> menggunakan kabel cat 6.....	47
Grafik 4.4. <i>Delay</i> Transmisi dari perangkat <i>mobile client</i> ke komputer <i>server</i> menggunakan wifi secara <i>ad hoc peer to peer</i> .....	52

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan perkembangan zaman, hampir semua instansi di dunia telah memanfaatkan teknologi jaringan sebagai pendukung dari perkembangan teknologi informasi yang mereka gunakan. Hal ini dialami instansi baik berskala kecil maupun skala besar. *IP (Internet Protocol)* adalah protokol yang mengatur komunikasi data komputer di internet. Komputer-komputer yang terhubung ke internet berkomunikasi dengan protokol ini. *IP Address* yang lebih dikenal secara umum dan digunakan saat ini adalah *IP versi 4* atau IPv4.

*Internet Protokol versi 4 (IPv4)*, merupakan salah satu jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan *Transmission Control Protokol/ Internet Protokol (TCP/IP)* menggunakan protokol IPv4. Pada dasarnya alamat IPv4 terdiri dari 32-bit yang di bagi menjadi empat *octet* dan setiap *octet* terdiri dari 8-bit. IPv4 sendiri terbagi menjadi beberapa kelas, yaitu kelas A, B, C, D dan E. Secara teori IPv4 ini mampu mencakup hingga 4 miliar *host* komputer yang di alamatkannya. Sehingga bila suatu saat batas kuota tersebut melebihi *host* yang ada diseluruh dunia maka akan terjadi kekurangan pengalamatan untuk *host-host* baru yang bermunculan, sehingga dikembangkanlah pengalamatan jenis baru yang sekarang dikenal dengan *IP versi 6* atau IPv6.

Salah satu kelebihan yang dimiliki oleh *Internet Protokol* yaitu *Mobile IP*. *Mobile IP* dikembangkan di IPv4 dan IPv6. *Mobile IP* merupakan teknologi yang memungkinkan *node* untuk pindah ke subnet lain tanpa mengubah *IP address*. Skripsi ini dimaksudkan untuk menunjukkan dampak *user mobility* terhadap performansi *video streaming* di jaringan *mobile IPv4*. Parameter-parameter yang akan dianalisa untuk menguji performansi jaringan *Mobile IPv4* yaitu *Delay*, *Jitter*, *Packet Loss* dan *Throughput*.

BAB I  
PENDAHULUAN

1.1 Latar Belakang

Peningkatan perkembangan teknologi informasi dan komunikasi telah membawa perubahan signifikan dalam kehidupan masyarakat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat.

Perkembangan teknologi informasi yang semakin pesat telah membawa perubahan signifikan dalam kehidupan masyarakat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat.

Perkembangan teknologi informasi yang semakin pesat telah membawa perubahan signifikan dalam kehidupan masyarakat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat. Hal ini ditandai dengan berkembangnya teknologi informasi yang semakin pesat.

## 1.2 Rumusan Masalah

Dari permasalahan yang timbul adalah Bagaimana menganalisa suatu kinerja dari aplikasi video streaming pada jaringan *Mobile IPv4*. Yang meliputi *Delay, jitter, Packet Loss* dan *Troughput*.

## 1.3 Tujuan Penelitian

Adapun tujuan yang diharapkan dalam penelitian ini adalah merancang dan membangun suatu aplikasi *video streaming* serta menganalisa *delay, jitter, packet loss* dan *troughput* pada jaringan *Mobile IPv4* menggunakan aplikasi *Wireshark*.

## 1.4 Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan yang diharapkan, maka pembahasan dibatasi oleh hal – hal sebagai berikut:

1. Membahas konsep *IPv4* dan *Mobile IPv4*.
2. Rancangan jaringan local yang terdiri dari 2 buah laptop yang berfungsi sebagai 1 *client (Wireshark)* dan 1 server, 1 buah *Smartphone* yang berfungsi sebagai *client (user)*, 1 buah *Modem ADSL (WiFi Router)* yang berfungsi sebagai *virtual server* dan penghubung *server* dengan *client*.
3. Instalasi aplikasi yang akan digunakan untuk implementasi yaitu *Video LAN Client (VLC)* pada sisi *client* dan *server*.
4. Analisa parameter-parameter untuk menguji jaringan *Mobile IPv4* yaitu *Delay, Jitter, Packet Loss* dan *Throughput*.

## 1.5 Metode Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut:

### 1. Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

### 2. Perancangan Sistem

Rancangan jaringan local yang terdiri dari 2 buah laptop yang berfungsi sebagai *1 client (Wireshark)* dan 1 server, 1 buah *Smartphone* yang berfungsi sebagai *client (user)*, 1 buah *Modem ADSL (WiFi Router)* yang berfungsi sebagai *virtual server* dan penghubung *server* dengan *client*.

### 3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat dan diimplementasikan kedalam sistem.

### 4. Eksperimen, analisa hasil dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

## 1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

#### Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

#### Bab II : Landasan Teori

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

#### Bab III : Perancangan dan Desain Aplikasi

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat

2.1.1.1.1.1.1.1

... dan ...

2.1.1.1.1.1.1.2

... dan ...

2.1.1.1.1.1.1.3

... dan ...

2.1.1.1.1.1.1.4

... dan ...

2.1.1.1.1.1.2

... dan ...

Bab I

... dan ...

Bab II

... dan ...

Bab III

... dan ...

kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

**Bab IV : Implementasi dan Analisis**

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

**Bab V : Penutup**

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

## BAB II

### LANDASAN TEORI

#### 2.1 *Internet Protokol versi 4 (IPv4)*

Sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IP versi 4. Panjang totalnya adalah 32-bit, dan secara teoritis dapat mengalami hingga 4 miliar *host* komputer atau lebih tepatnya 4.294.967.296 *host* di seluruh dunia, jumlah *host* tersebut didapatkan dari 256 (didapatkan dari 8 bit) dipangkat 4 (karena terdapat 4 oktet) sehingga nilai maksimal dari alamat IPv4 tersebut adalah 255.255.255.255 dimana nilai dihitung dari nol sehingga nilai-nilai *host* yang dapat ditampung adalah  $256 \times 256 \times 256 \times 256 = 4.294.967.296$  *host*. Sehingga bila *host* yang ada diseluruh dunia melebihi kuota tersebut maka dibuatlah *IP versi 6* atau IPv6. Contoh alamat *IP versi 4* adalah 192.168.0.3.<sup>[1]</sup>

#### 2.2 *Mobile IPv4*

Saat ini *Internet Engineering Task Force (IETF)* telah melakukan standarisasi dalam bidang protokol internet yang nantinya akan mempunyai fitur dukungan terhadap pergerakan dari *mobile device*. Protokol tersebut adalah *Mobile Internet Protocol* atau biasa yang disebut dengan Mobile IP. Ada dua varian dari Mobile IP yang dikembangkan yaitu Mobile IPv4 dan Mobile IPv6. Fitur yang terdapat pada Mobile IPv4 mengijinkan *host* yang telah mempunyai dukungan terhadap IPv4 untuk meninggalkan *home subnetnya*, sementara *host* tersebut sedang memperbaharui koneksinya ke internet. Di *Mobile IPv4*, *mobile node* mempunyai dua *address* yaitu *home address* dan *care-of-address*. *Home address* merupakan alamat unik yang digunakan untuk mengalamatkan sebuah perangkat ketika berada di *home network* atau jaringan tempat berasal atau dengan kata lain alamat tetap dari perangkat *mobile* tersebut. Sedangkan *care-of-address* merupakan alamat yang diperoleh sebuah perangkat *mobile* ketika berada di luar *home network*. *Care-of-address* digunakan sebagai alamat pengganti ketika perangkat *mobile* berada di luar *home network (foreign network)*. Dua alamat tersebut penggunaannya diatur oleh dua *agent* yang terdapat dalam skema

BAB II

LANDASAN TEORI

2.1. Konsep dan Fungsi (Hidayat, 2014)

Salah satu konsep yang digunakan di dalam protokol jaringan yang menggunakan jaringan yang digunakan adalah jaringan TCP/IP yang menggunakan protokol IP versi 4. Panjang totalnya adalah 32-bit dan secara umum dapat menggunakan hingga 4 miliar host yang berbeda-beda. Untuk keperluan ini, protokol ini menggunakan dua kelas alamat IP (alamat 8 bit) dan alamat 24 bit (alamat 4 kelas) untuk membedakan antara jaringan yang satu dengan yang lainnya. Untuk keperluan ini, protokol ini menggunakan dua kelas alamat IP yang berbeda-beda, yaitu kelas A dan kelas B. Kelas A menggunakan alamat 8 bit untuk membedakan antara jaringan yang satu dengan yang lainnya, sedangkan kelas B menggunakan alamat 16 bit untuk membedakan antara jaringan yang satu dengan yang lainnya.

2.2. Mobile IP (Hidayat, 2014)

Salah satu konsep yang digunakan di dalam protokol jaringan yang menggunakan adalah Mobile IP yang menggunakan protokol IP versi 4. Panjang totalnya adalah 32-bit dan secara umum dapat menggunakan hingga 4 miliar host yang berbeda-beda. Untuk keperluan ini, protokol ini menggunakan dua kelas alamat IP (alamat 8 bit) dan alamat 24 bit (alamat 4 kelas) untuk membedakan antara jaringan yang satu dengan yang lainnya. Untuk keperluan ini, protokol ini menggunakan dua kelas alamat IP yang berbeda-beda, yaitu kelas A dan kelas B. Kelas A menggunakan alamat 8 bit untuk membedakan antara jaringan yang satu dengan yang lainnya, sedangkan kelas B menggunakan alamat 16 bit untuk membedakan antara jaringan yang satu dengan yang lainnya.

jaringan Mobile IPv4. Kedua *agent* tersebut adalah *Home Agent* dan *Foreign Agent*.<sup>[2]</sup>

*Home Agent* merupakan perangkat yang bertugas mendeteksi keberadaan perangkat *mobile*. Agent ini akan selalu memonitor keberadaan perangkat *mobile* walaupun perangkat tersebut berada pada *foreign agent*. *Foreign Agent* merupakan agent yang terdapat pada *foreign network* dan bertugas memberikan *care-of-address* kepada perangkat *mobile* dari jaringan lain di luar dirinya yang sedang berada pada jaringan tersebut.<sup>[2]</sup>

### 2.3 OSI Layer

*Open System Interconnection* atau yang biasa disebut dengan OSI Layer adalah standar komunikasi yang diterapkan di dalam jaringan komputer. Model OSI mempunyai tujuh lapisan serta tugas dan tanggung jawab masing-masing, yaitu :

1. *Physical Layer* : Berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan, topologi jaringan dan pengkabelan. Adapun perangkat-perangkat yang dapat dihubungkan dengan *Physical Layer* adalah *Network Interface Card* (NIC) berikut dengan kabel – kabelnya.
2. *Data Link Layer* : Befungsi untuk menentukan bagaimana bit-bit data dikelompokkan menjadi format yang disebut sebagai *frame*. Pada layer ini terjadi koreksi kesalahan, flow control, pengalamatan perangkat keras seperti halnya *MAC Address*, dan menentukan bagaimana perangkat-perangkat jaringan seperti *Hub*, *Bridge*, *Repeater*, dan *Switch layer 2 (Switch un-manage)* beroperasi. Spesifikasi *Institute Of Electrical And Electronics Engineers* (IEEE) 802, membagi layer ini menjadi dua layer anak, yaitu lapisan *Logical Link Control* (LLC) dan lapisan *Media Access Control* (MAC).
3. *Network Layer* : Berfungsi untuk mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket, dan kemudian melakukan

*routing* melalui *internetworking* dengan menggunakan *Router* dan *Switch layer-3 (Switch Manage)*.

4. *Transport Layer* : Berfungsi untuk memecah data ke dalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Selain itu, pada layer ini juga membuat sebuah tanda bahwa paket diterima dengan sukses (*acknowledgement*), dan mentransmisikan ulang terhadap paket-paket yang hilang di tengah jalan.
5. *Session Layer* : Berfungsi untuk mendefinisikan bagaimana koneksi dapat dibuat, dipelihara, atau dihancurkan. Selain itu, di layer ini juga dilakukan resolusi nama.
6. *Presentation Layer* : Berfungsi untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan. Protokol yang berada dalam layer ini adalah perangkat lunak redirektor (*redirector software*), seperti layanan *Workstation* (dalam Windows NT) dan juga *Network shell* (semacam *Virtual Network Computing (VNC)* atau *Remote Desktop Protocol (RDP)*).
7. *Application Layer* : Berfungsi sebagai antarmuka dengan aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan kesalahan. Protokol yang berada dalam layer ini adalah HTTP, FTP, SMTP, dan NFS.

Dari ke Tujuh layer tersebut mempunyai 2 (dua) tingkatan layer, yaitu:

- a. Lower Layer yang meliputi : Physical Layer, DataLink Layer, dan Network Layer.
- b. Upper Layer yang meliputi : Transport Layer, Session Layer, Presentation.
- c. Layer, dan Application Layer. <sup>[3]</sup>

... dengan menggunakan ...

1. ...

2. ...

3. ...

4. ...

... dan ...

a. ...

b. ...

... dan ...

## 2.4 RTSP (*Real Time Streaming Protocol*)

RTSP adalah sebuah protokol di level aplikasi untuk membangun dan mengontrol pengiriman data dengan real-time. RTSP menyediakan *framework* yang *extensible* untuk bisa dikontrol. Dalam permintaan pengiriman data yang *real time*, seperti audio dan video RTSP menggunakan pesan teks, meminta dan menanggapi, yang bisa jadi mengandung pesan biner. Protokol ini digunakan untuk menetapkan dan mengendalikan sesi media antara dua titik ujungnya.<sup>[4]</sup>

## 2.5 VLC (*Video LAN Client*)

VLC adalah sebuah program media player gratis yang dapat memainkan banyak jenis file format video dan audio. VLC media player juga dapat digunakan sebagai server untuk *streaming*.<sup>[5]</sup>

## 2.6 Wireshark

*Wireshark* merupakan perangkat lunak yang berfungsi untuk menganalisa jaringan. *Wireshark* dapat menganalisa beberapa parameter *QoS* seperti *delay*, *jitter*, *paket loss* dan *throughput* serta dapat mengcapture *protocol* yang sedang berjalan di jaringan tersebut.<sup>[4]</sup>

## 2.7 Parameter Video Streaming

Parameter- parameter yang dihitung untuk penilaian kualitas video streaming meliputi :

### a. *Delay*

*Delay* merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama.



**b. Jitter**

*Jitter*, atau variasi kedatangan paket, hal ini diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket di akhir perjalanan *jitter*.

**c. Packet Loss**

*Packet loss* merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang. Biasa disebabkan oleh *network switch* yang mempunyai kekurangan kapasitas *buffer* yang terbatas.

**d. Throughput**

*Throughput*, yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam *bps*. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut. <sup>[5]</sup>

## 2.8 Topologi Jaringan Komputer

Topologi menggambarkan struktur dari suatu jaringan atau bagaimana sebuah jaringan didesain. Pola ini sangat erat kaitannya dengan metode *access* dan media pengiriman yang digunakan. Topologi yang ada sangatlah tergantung dengan letak geografis dari masing-masing terminal, kualitas kontrol yang dibutuhkan dalam komunikasi ataupun penyampaian pesan, serta kecepatan dari pengiriman data. Dalam definisi topologi terbagi menjadi dua, yaitu topologi fisik (*physical topology*) yang menunjukkan posisi pemasangan kabel secara fisik dan topologi logik (*logical topology*) yang menunjukkan bagaimana suatu media diakses oleh *host*.

Adapun topologi fisik yang umum digunakan dalam membangun sebuah jaringan adalah :

### 1. *Point to Point* (Titik ke-Titik)

*Point to Point* merupakan jaringan kerja yang paling sederhana tetapi dapat digunakan secara luas. Dalam hal ini, kedua simpul mempunyai kedudukan yang setingkat, sehingga simpul manapun dapat memulai dan mengendalikan hubungan dalam jaringan tersebut. Data dikirim dari satu simpul langsung kesimpul lainnya sebagai penerima, misalnya antara terminal dengan CPU.<sup>[6]</sup>

### 2. *Star Network* (Jaringan Bintang)

Dalam konfigurasi bintang, beberapa peralatan yang ada akan dihubungkan kedalam satu pusat kontrol. Kontrol yang ada akan dipusatkan pada satu titik, misalnya mengatur beban kerja serta pengaturan sumber daya yang ada. Semua *link* harus berhubungan dengan pusat apabila ingin menyalurkan data kesimpul lainnya yang dituju. Dalam hal ini, bila pusat mengalami gangguan, maka semua terminal juga akan terganggu. Model jaringan bintang ini relatif sangat sederhana, sehingga banyak digunakan oleh pihak perbankan yang biasanya mempunyai banyak kantor cabang yang tersebar diberbagai lokasi. Dengan adanya konfigurasi bintang ini, maka segala macam kegiatan yang ada di-kantor cabang dapatlah dikontrol dan dikoordinasikan dengan baik.

Kelebihan :

- a. Kerusakan pada satu saluran hanya akan mempengaruhi jaringan pada saluran tersebut dan stasiun yang terpaut.
- b. Tingkat keamanan termasuk tinggi.
- c. Penambahan dan pengurangan station dapat dilakukan dengan mudah.

Kekurangan :

- a. Jika *node* tengah mengalami kerusakan, maka seluruh jaringan akan terhenti.<sup>[6]</sup>

### 3. *Ring Networks* (Jaringan Cincin)

Pada jaringan komputer ini terdapat beberapa peralatan saling dihubungkan satu dengan lainnya dan pada akhirnya akan membentuk

1. Point-to-Point (Tipe 1a-1b)

Pada tipe ini, bus berhenti di satu tempat yang kemudian melanjutkan perjalanan ke tempat berikutnya. Hal ini dilakukan untuk memudahkan penumpang yang ingin turun atau naik di tempat tersebut. Tipe ini biasanya digunakan untuk rute yang pendek dan padat penumpang.

2. Star System (Tipe 1c-1d)

Dalam sistem ini, bus berangkat dari satu titik pusat ke beberapa titik tujuan. Setelah selesai melayani penumpang di titik-titik tersebut, bus akan kembali ke titik pusat. Hal ini memudahkan penumpang yang ingin turun atau naik di titik-titik tersebut. Tipe ini biasanya digunakan untuk rute yang lebih panjang dan memiliki banyak titik tujuan.

Kelompok

dan kelompok bus lainnya akan melanjutkan perjalanan pada sistem tersebut dan stasiun yang tertera.

3. Sistem busway (Tipe 1e-1f)

Busway adalah jalur bus yang memiliki hak jalan khusus. Hal ini memudahkan bus untuk bergerak dengan cepat dan efisien.

Kelompok

dan bus lainnya akan melanjutkan perjalanan pada sistem busway tersebut.

4. Ring System (Tipe 1g-1h)

Dalam sistem ini, bus berangkat dari satu titik pusat ke beberapa titik tujuan dan kemudian melanjutkan perjalanan ke titik-titik tersebut. Hal ini memudahkan penumpang yang ingin turun atau naik di titik-titik tersebut.

bagan seperti halnya sebuah cincin. Jaringan cincin tidak memiliki suatu titik yang bertindak sebagai pusat ataupun pengatur lalu lintas data, semua simpul mempunyai tingkatan yang sama. Data yang dikirim akan berjalan melewati beberapa simpul sehingga sampai pada simpul yang dituju. Dalam menyampaikan data, jaringan bisa bergerak dalam satu ataupun dua arah. Walaupun demikian, data yang ada tetap bergerak satu arah dalam satu saat. Pertama, pesan yang ada akan disampaikan dari titik ke titik lainnya dalam satu arah. Apabila ditemui kegagalan, misalnya terdapat kerusakan pada peralatan yang ada, maka data yang ada akan dikirim dengan cara kedua, yaitu pesan kemudian ditransmisikan dalam arah yang berlawanan, dan pada akhirnya bisa berakhir pada tempat yang dituju.

Konfigurasi semacam ini relative lebih mahal apabila dibanding dengan konfigurasi jaringan bintang. Hal ini disebabkan, setiap simpul yang ada akan bertindak sebagai komputer yang akan mengatasi setiap aplikasi yang dihadapinya, serta harus mampu membagi sumber daya yang dimilikinya pada jaringan yang ada. Disamping itu, sistem ini lebih sesuai digunakan untuk sistem yang tidak terpusat (*decentralized-system*), dimana tidak diperlukan adanya suatu prioritas tertentu.

Kelebihan :

- a. Tidak ada komputer yang memonopoli jaringan, karena setiap komputer mempunyai hak akses yang sama terhadap *token*.
- b. Data mengalir dalam satu arah sehingga terjadinya *collision* dapat dihindarkan.

Kekurangan :

- a. Apabila ada satu komputer dalam *ring* yang gagal berfungsi, maka akan mempengaruhi keseluruhan jaringan.
- b. Sulit untuk mengatasi kerusakan di jaringan yang menggunakan topologi *ring*.
- c. Menambah atau mengurangi komputer akan mengacaukan jaringan.
- d. Sulit untuk melakukan konfigurasi ulang.<sup>[6]</sup>

... yang dapat dilihat sebagai salah satu bentuk ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...  
 ... yang berkaitan dengan ...

Kontribusi ...  
 dengan kontribusi ...  
 yang ada akan ...  
 aplikasi yang ...  
 diindikasikan ...  
 digunakan untuk ...  
 dimana tidak ...

- Kelompok
- a. Tidak ada ...
- b. Data ...
- Kelompok
- a. ...
- b. ...
- c. ...
- d. ...

#### 4. *Tree Network (Jaringan Pohon)*

Pada jaringan pohon, terdapat beberapa tingkatan simpul (*node*). Pusat atau simpul yang lebih tinggi tingkatannya, dapat mengatur simpul lain yang lebih rendah tingkatannya. Data yang dikirim perlu melalui simpul pusat terlebih dahulu. Misalnya untuk bergerak dari komputer dengan *node-3* ke komputer *node-7* seperti halnya pada gambar, data yang ada harus melewati *node-3*, 5 dan *node-6* sebelum berakhir pada *node-7*. Keunggulan jaringan model pohon seperti ini adalah, dapat terbentuknya suatu kelompok yang dibutuhkan pada setiap saat.

Sebagai contoh, perusahaan dapat membentuk kelompok yang terdiri atas terminal pembukuan, serta pada kelompok lain dibentuk untuk terminal penjualan. Adapun kelemahannya adalah, apabila simpul yang lebih tinggi kemudian tidak berfungsi, maka kelompok lainnya yang berada dibawahnya akhirnya juga menjadi tidak efektif. Cara kerja jaringan pohon ini relatif menjadi lambat.

Kelebihan :

- a. Data terpusat secara hirarki sehingga manajemen data lebih baik dan mudah.

Kekurangan :

- a. Komputer di bawahnya tidak dapat dioperasikan apabila jalur pada komputer tingkat atasnya terputus.<sup>[6]</sup>

#### 5. *Topologi Bus*

Pada topologi bus semua komputer (terminal) dihubungkan pada jalur data (bus) yang berbentuk garis lurus (linear). Sehingga, data yang dikirim akan melalui semua terminal pada jalur tersebut. Jika alamat data tersebut sesuai dengan alamat yang dilalui, maka data tersebut akan diterima dan diproses. Namun, jika alamat tidak sesuai, maka data akan diabaikan oleh terminal yang dilalui dan pencarian alamat akan diteruskan hingga ditemukan alamat yang sesuai.

Kelebihan :

- a. Hemat kabel



- b. Tidak membutuhkan kendali pusat
- c. Penambahan dan pengurangan terminal dapat dilakukan tanpa mengganggu operasi yang berjalan
- d. Layout kabel sederhana.

Kekurangan :

- a. Kecepatan akan menurun bila jumlah user (pemakai) bertambah
- b. Jika terjadi gangguan atau kerusakan pada salah satu lokasi (titik) dalam jaringan maka hal itu akan mempengaruhi jaringan secara keseluruhan. <sup>[6]</sup>

## 2.9 Sistem Operasi *Linux*

*Linux* adalah sistem operasi yang pertama kali diciptakan oleh Linus Torvalds pada tahun 1991. *Linux* bersifat *open source* yang *free* dibawah lisensi GPL (*GNU Public License*), yang merupakan sistem operasi 32-64bit, dan dapat dijalankan pada berbagai platform baru. *Linux* itu sendiri adalah sistem operasi *multi-user* yang efisien, selain itu juga *linux* mampu mengerjakan lebih dari satu tugas pada waktu yang bersamaan, atau lebih sering kita sebut dengan sebutan *multitasking*.

Sistem operasi *linux* terdiri dari kernel, program sistem, dan beberapa program aplikasi. Kernel merupakan inti dari sistem operasi yang mengatur pengerjaan memori, piranti masukan keluaran, proses-proses, pemakaian file pada file sistem dan lain-lain.

Kernel *linux* terdiri dari beberapa bagian penting, seperti: manajemen proses, manajemen memori, *hardware device drivers*, *filesystem drivers*, manajemen jaringan dan lain-lain. Namun bagian yang terpenting dari kernel ialah manajemen proses dan manajemen memori. Manajemen memori menangani daerah pemakaian memori, daerah swap, bagian-bagian kernel dan untuk *buffer cache*. Manajemen proses menangani pembuatan proses-proses dan penjadwalan proses. Pada bagian dasar kernel berisi *hardware device drivers* untuk setiap jenis hardware yang didukung.



Program sistem dan semua program-program lainnya yang berjalan di atas kernel disebut *user mode*. Perbedaan mendasar antara program sistem dan program aplikasi adalah program sistem dibutuhkan agar suatu sistem operasi dapat berjalan sedangkan program aplikasi adalah program yang dibutuhkan untuk menjalankan suatu aplikasi tertentu. Contoh dari program sistem adalah daemon, sedangkan contoh dari program aplikasi adalah pengolah kata (*word processor*).<sup>[6]</sup>

## **BAB III**

### **PERANCANGAN DAN DESAIN APLIKASI**

#### **3.1 Analisa system**

Analisa system Performansi *Real Time Streaming Protokol* (RTSP) untuk aplikasi video streaming pada Jaringan Mobile IPv4 dengan menggunakan *Video LAN Client* (VLC) meliputi deskripsi umum, fitur aplikasi VLC serta analisa kebutuhan system dimana pemahaman ketiga aspek tersebut sangatlah diperlukan untuk menghasilkan suatu system yang baik sehingga didapatkan analisa yang baik pula, tentunya dengan mengacu pada literatur-literatur yang banyak tersedia mengenai permasalahan dalam analisa tersebut.

##### **3.1.1 Deskripsi Umum**

Analisa Performansi *Real Time Streaming Protokol* (RTSP) untuk aplikasi video streaming pada Jaringan Mobile IPv4 dengan menggunakan *Video LAN Client* (VLC) merupakan analisa terhadap QoS jaringan serta *real time streaming* yang difokuskan pada jaringan Mobile IPv4 dengan menggunakan aplikasi client server VLC yang terinstallasi dengan baik pada server maupun *client* dengan menggunakan *protocol* RTSP.

##### **3.1.2 Fitur Aplikasi *Video LAN Client***

Pada analisa performansi RTSP untuk aplikasi video streaming pada jaringan Mobile IPv4 dengan menggunakan VLC fitur yang digunakan ialah:

###### **1. *Stream***

Pada fitur inilah dilakukan konfigurasi yang difungsikan sebagai video server.

###### **2. *Network Stream***

Pada fitur inilah dilakukan konfigurasi untuk mengakses video yang berada di server sehingga dapat ditampilkan pada perangkat client yang dalam hal ini berupa *Personal Computer* (PC) dan perangkat *mobile*.

### **3.1.3 Analisa Kebutuhan Sistem**

Secara keseluruhan analisa performansi *RTSP* untuk aplikasi video *streaming* pada jaringan Mobile IPv4 dengan menggunakan VLC memiliki kebutuhan perangkat keras sebagai berikut:

1. Komputer Sebagai Server
2. Kabel UTP Cat 5, 5e dan 6
3. Tang Crimping
4. Pin RJ 45
5. LAN Tester
6. Akses Point
7. Smartphone sebagai client

Selain Perangkat keras, pada analisa kali ini juga memerlukan perangkat lunak sebagai berikut:

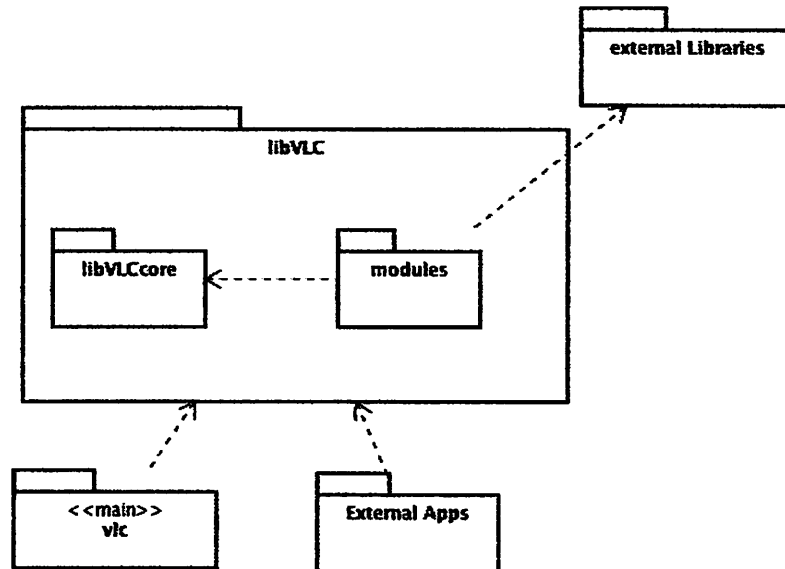
1. Microsoft Windows 7
2. VLC Media Player

## **3.2 Perancangan Sistem**

Perancangan system merupakan langkah awal yang harus dilakukan sebelum dilakukan analisa lebih lanjut mengenai QoS pada jaringan video LAN. Dimana pada perancangan system akan diberikan gambar mengenai arsitektur aplikasi VLC serta diagram bloknya.

### **3.2.1 Arsitektur Aplikasi Video LAN Client**

Berikut yaitu arsitektur aplikasi Video LAN Client yang ditunjukkan pada Gambar 3.1.



Gambar 3.1. Arsitektur VLC

*LibVLCcore* adalah inti utama dari kerangka. Fungsinya ialah memberikan lapisan berorientasi objek ke C, modul pemuatan / pembongkaran dan satu set fungsi abstrak tentang multimedia: masukan, *multiplexing*, *demultiplexing*, *output audio*, *video output*.

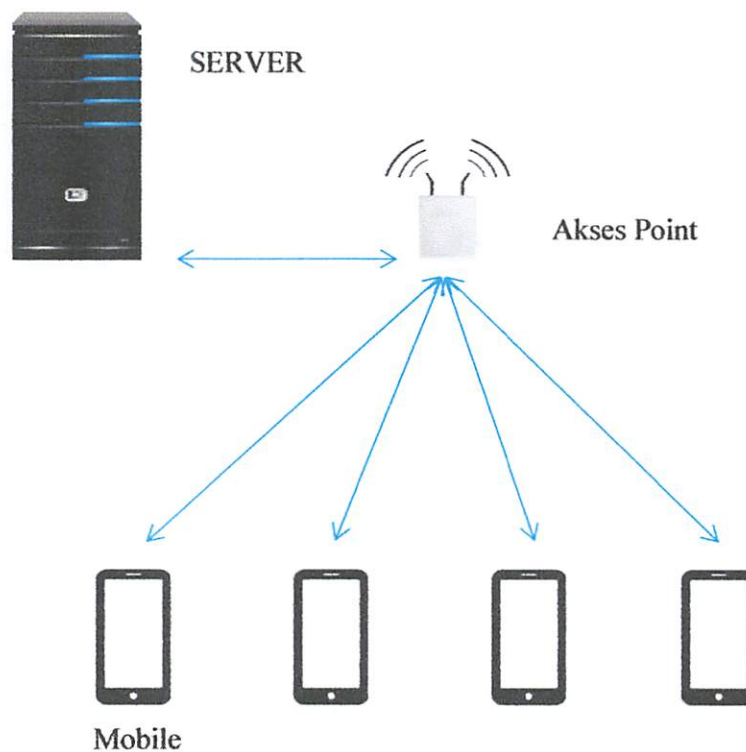
Modul menyediakan fungsi konkret kerangka. Modul dikategorikan sesuai kemampuan mereka. Ada modul untuk manajemen input (berkas, jaringan, cd), modul untuk *codec* (mp3, divx, dll), modul untuk GUI (tekstual, web, telnet, qt berbasis, macosx asli).

*External libraries* merupakan modul-modul yang sewaktu-waktu dapat ditambahkan.

*Vlc (Main)* adalah utama pemain. Ini menginisialisasi *libVLC* dan peluncuran antar muka pengguna.

### 3.2.2 Diagram Blok

Dibawah ini ditunjukkan diagram blok server dimana gambaran untuk server, akses point dan perangkat mobile yang akan digunakan pada pengujian, yang dapat dilihat pada gambar 3.2.



Gambar 3.2. Diagram Blok Sistem

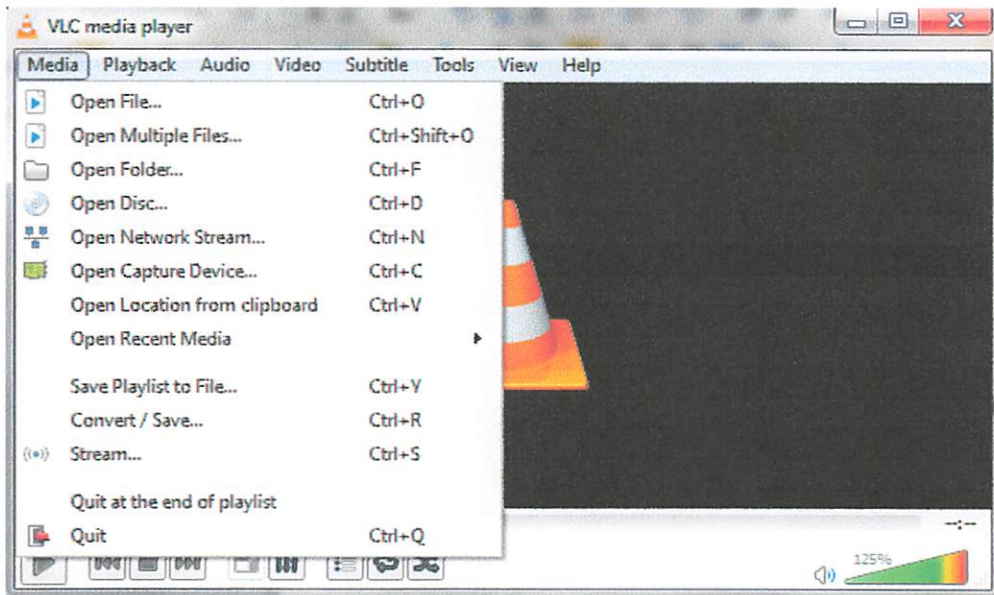
### 3.3 Konfigurasi

Dalam Konfigurasi Analisa system Performansi *Real Time Streaming Protokol* (RTSP) untuk aplikasi video streaming pada jaringan Mobile IPv4 dengan menggunakan *Video LAN Client* (VLC), ada beberapa tahapan yang dilakukan, yaitu:

1. Konfigurasi *Server Side*
2. Konfigurasi *Client Side*
3. Konfigurasi *WireShark*

### 3.3.1 Konfigurasi *Server Side*

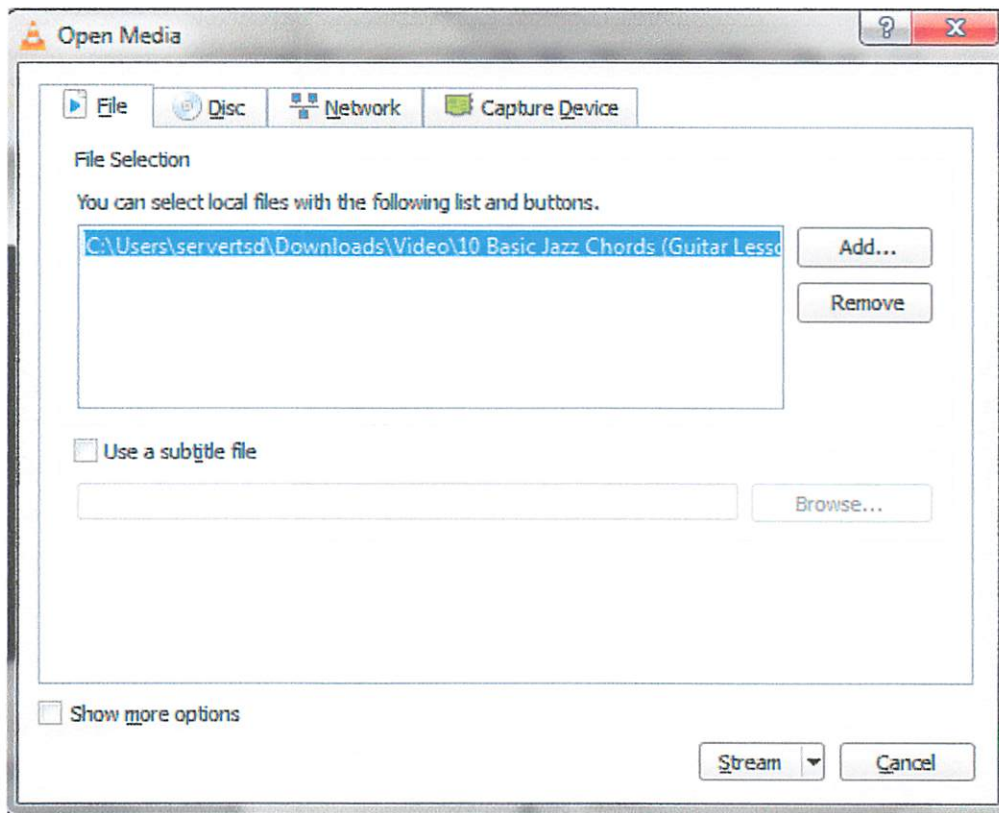
Pada konfigurasi *server side* perlu dilakukan beberapa tahapan konfigurasi yang akan di jelaskan dalam bentuk gambar dan tulisan serta perangkat lunak yang akan digunakan untuk mengambil contoh data yang nantinya akan dianalisa.



Gambar 3.3.Konfigurasi *server* tahap 1.

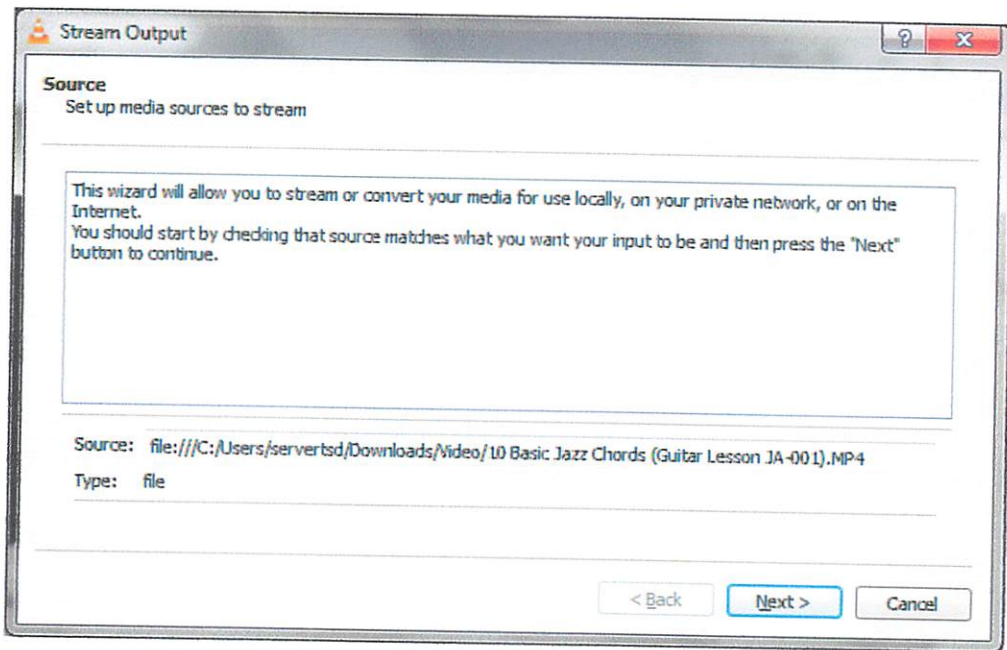
Pada gambar 3.3 menjelaskan tentang tahapan pertama yang dilakukan untuk mengkonfigurasi video Streaming server dengan menggunakan aplikasi *VLC* di mana terlebih dahulu kita memilih menu media lalu pilih menu *stream*.





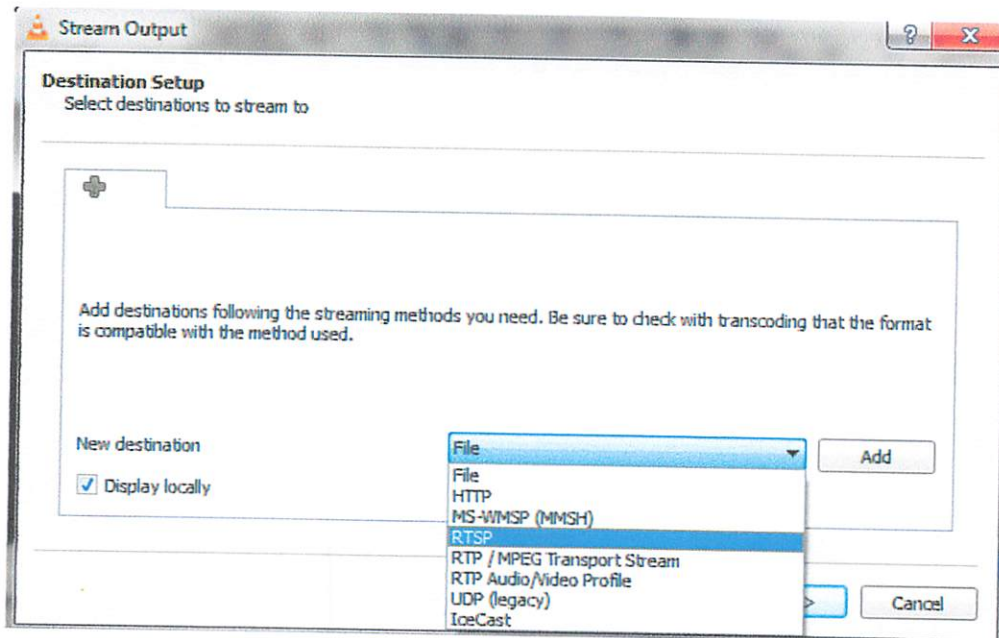
Gambar 3.4. Konfigurasi *server* tahap 2

Pada gambar 3.4 menjelaskan tentang tahapan kedua yang dilakukan untuk mengkonfigurasi video *streaming server* dengan menggunakan aplikasi VLC dimana pada tahapan ini kita memilih menu file, lalu mengambil file video yang akan kita *streaming* dengan cara klik tombol *add*. Lalu kita bisa menambahkan video tersebut.



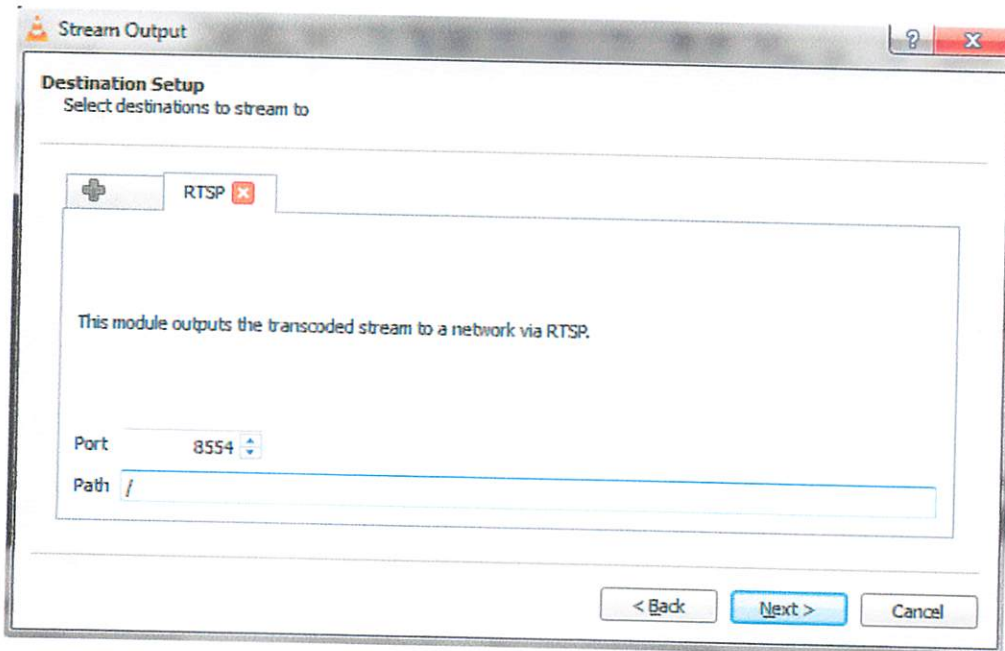
Gambar 3.5. Konfigurasi *server* tahap 3

Pada gambar 3.5 menjelaskan tentang tahapan ketiga yang dilakukan untuk mengkonfigurasi video *streaming server* dengan menggunakan aplikasi VLC. Pada jendela ini kita pilih tombol *next*.



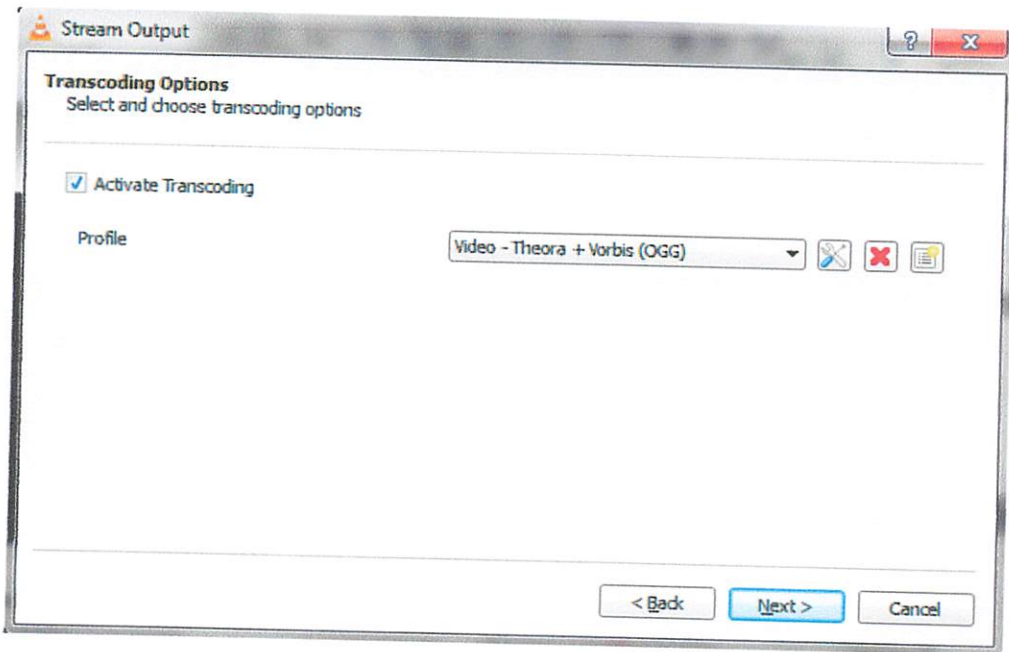
Gambar 3.6. Konfigurasi *server* tahap 4

Pada gambar 3.6 menjelaskan tentang tahapan keempat yang dilakukan untuk mengkonfigurasi video *streaming server* dengan menggunakan aplikasi VLC. Pada jendela ini kita perlu menambahkan destination setup, yang mana kita akan menggunakan RTSP.



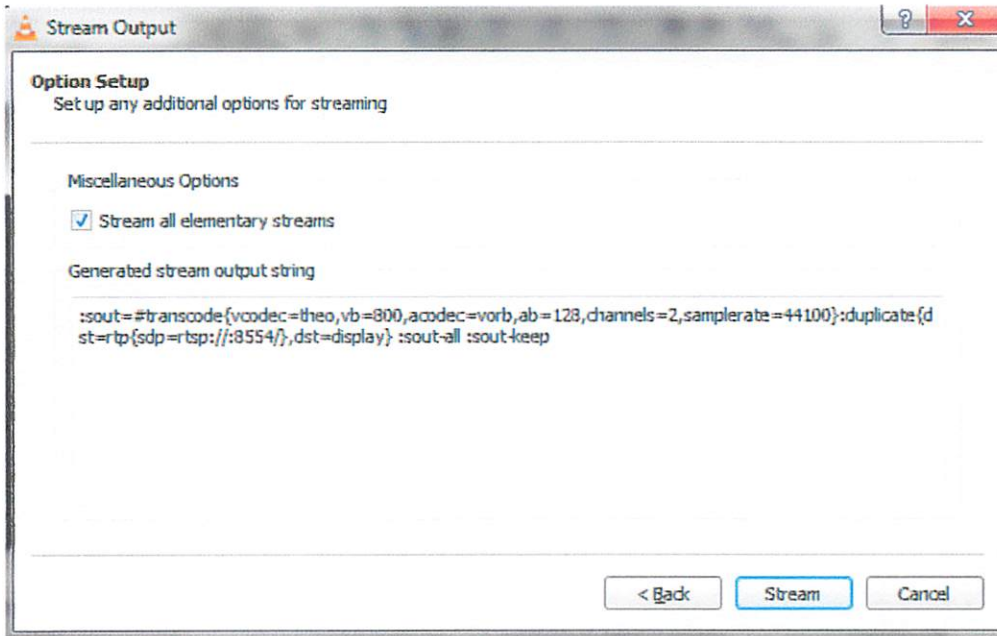
Gambar 3.7. Konfigurasi server tahap 5

Pada gambar 3.7 menjelaskan tentang tahapan kelima yang dilakukan untuk mengkonfigurasi video *streaming server* dengan menggunakan aplikasi VLC. Pada jendela ini kita pilih tombol *next*.



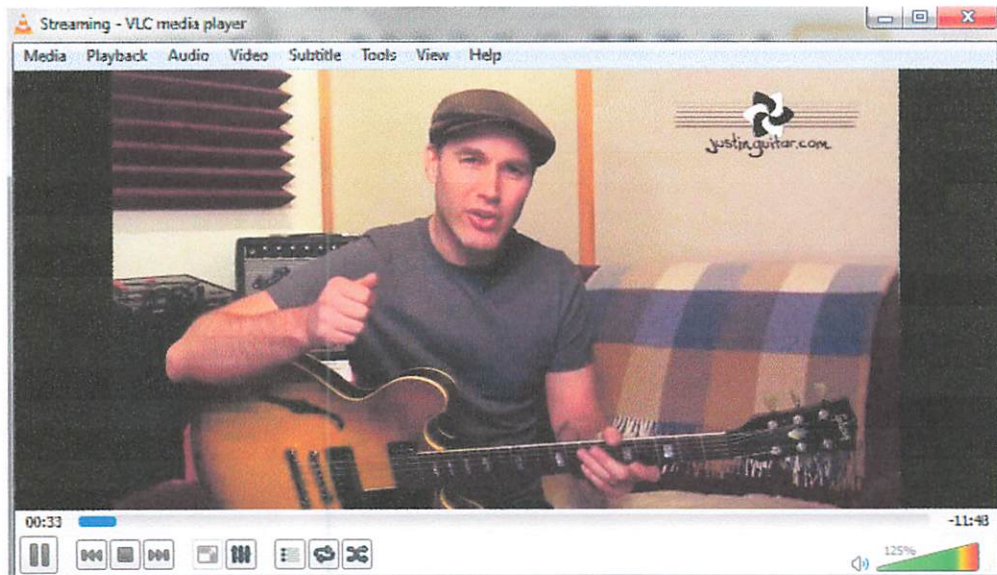
Gambar 3.8. Konfigurasi server tahap 6

Pada gambar 3.8 menjelaskan tentang tahapan keenam yang dilakukan untuk mengkonfigurasi video *streaming server* dengan menggunakan aplikasi VLC. Pada jendela ini kita pilih profile *Video-Theora + Vorbis (VOG)* lalu kita pilih tombol *next*.



Gambar 3.9. Konfigurasi *server* tahap 7

Pada gambar 3.9 menjelaskan tentang tahapan ketujuh yang dilakukan untuk mengkonfigurasi video *streaming server* dengan menggunakan aplikasi VLC. Pada jendela ini kita pilih tombol *Stream*.



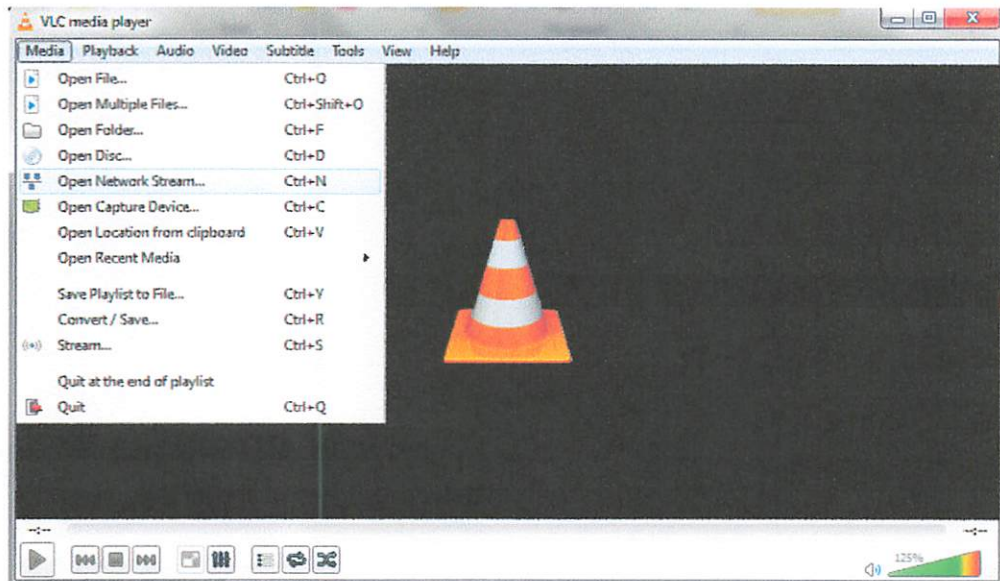
Gambar 3.10. Hasil *Streaming server side*



Pada gambar 3.10 merupakan hasil tampilan video dari sisi server yang telah distreamingkan.

### 3.3.2 Konfigurasi *Client Side*

Pada konfigurasi *Client side* perlu dilakukan beberapa tahapan konfigurasi yang akan di jelaskan dalam bentuk gambar dan tulisan sebagai berikut:



Gambar 3.11. Konfigurasi *Client* tahap 1.

Pada gambar 3.11 menjelaskan tentang tahapan pertama yang dilakukan untuk mengkonfigurasi video *streaming client* dengan menggunakan aplikasi VLC. Pada jendela ini kita pilih menu media lalu selanjutnya pilih *open network stream*.

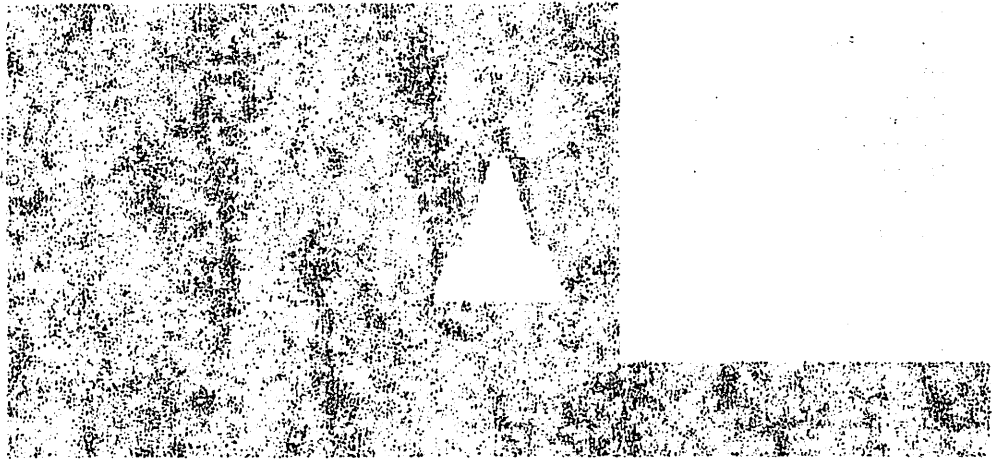
gaya yang lebih baik untuk meningkatkan efisiensi dan produktivitas.

Penelitian ini akan dilakukan

di lingkungan kerja yang berbeda-beda.

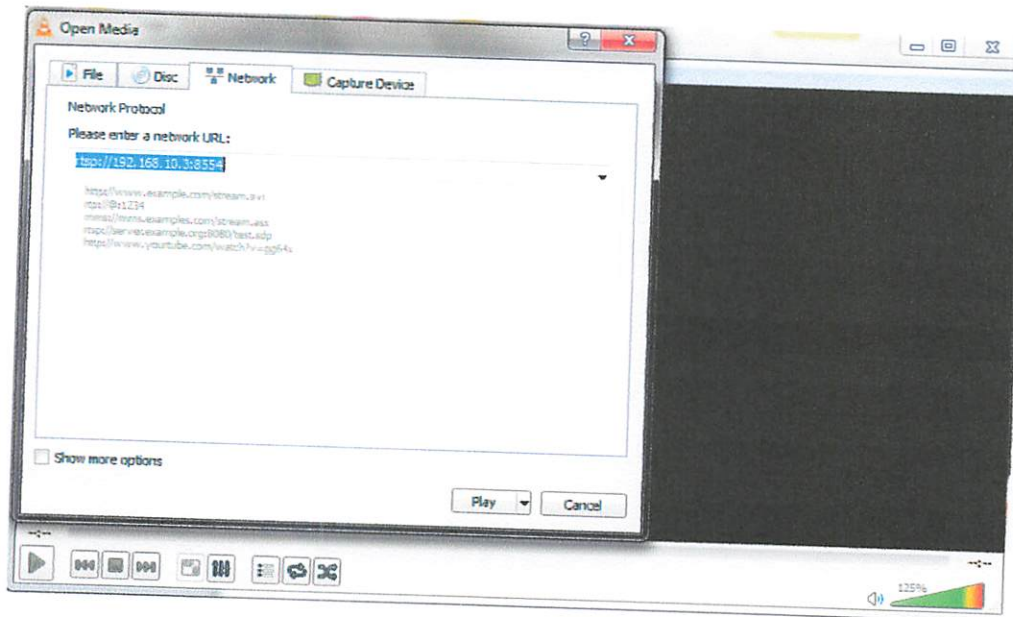
Penelitian ini akan dilakukan di lingkungan kerja yang berbeda-beda.

Penelitian ini akan dilakukan di lingkungan kerja yang berbeda-beda.



Penelitian ini akan dilakukan di lingkungan kerja yang berbeda-beda.

Penelitian ini akan dilakukan di lingkungan kerja yang berbeda-beda. Penelitian ini akan dilakukan di lingkungan kerja yang berbeda-beda. Penelitian ini akan dilakukan di lingkungan kerja yang berbeda-beda. Penelitian ini akan dilakukan di lingkungan kerja yang berbeda-beda. Penelitian ini akan dilakukan di lingkungan kerja yang berbeda-beda.

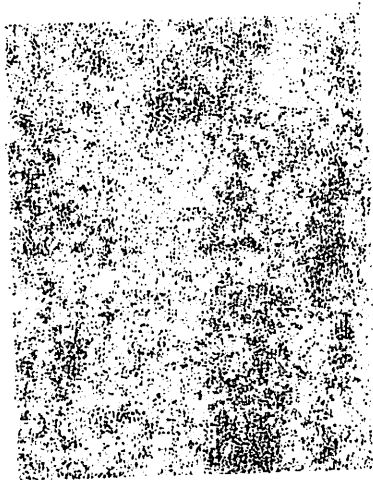


Gambar 3.12. Konfigurasi *Client* tahap 2

Pada gambar 3.12 menjelaskan tentang tahapan kedua yang dilakukan untuk mengkonfigurasi video *streaming client* dengan menggunakan aplikasi VLC. Pada jendela ini kita ketikkan alamat IP komputer server beserta portnya agar video yang ditampilkan pada *computer server* dapat ditampilkan pula dari sisi *client*.



Gambar 3.13. Hasil Video *Streaming* pada *Client*



1970-1971

... ..  
... ..  
... ..  
... ..  
... ..

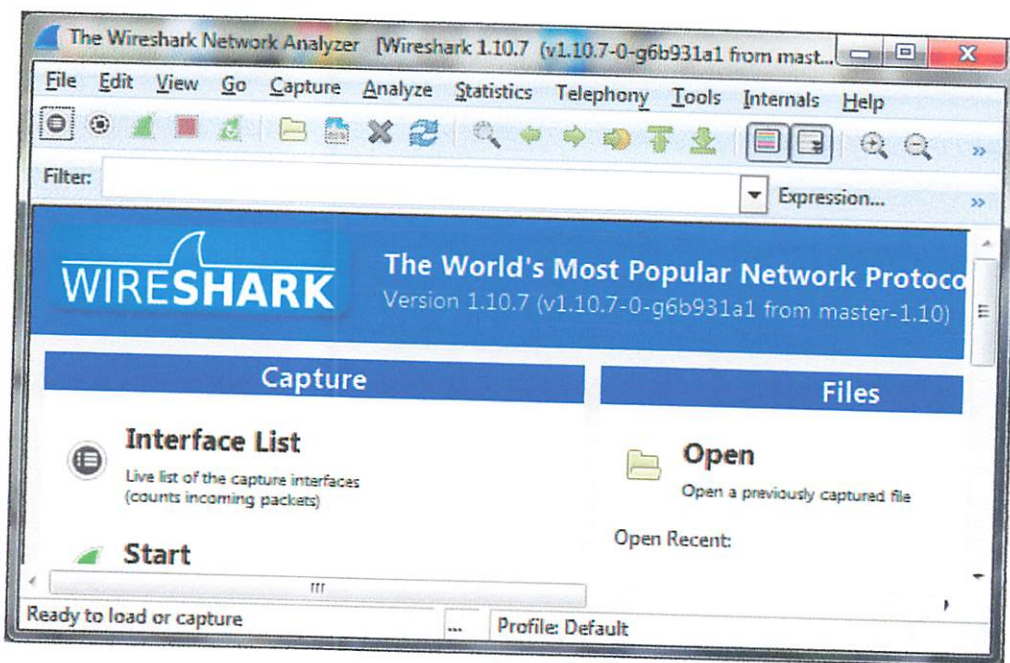


1970-1971

Pada gambar 3.13 merupakan hasil video *streaming* yang diakses oleh *computer client*.

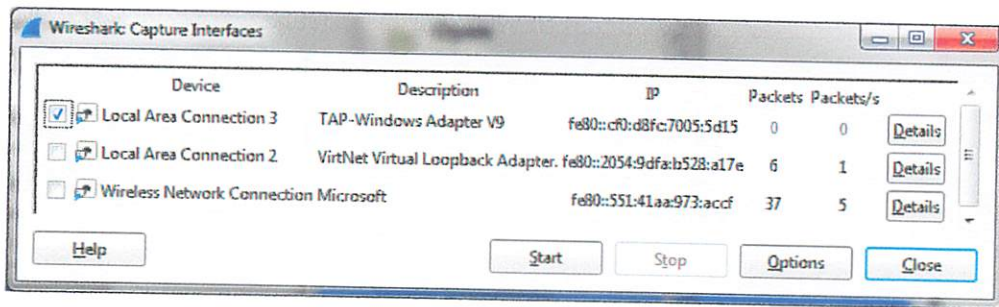
### 3.3.3 Konfigurasi *Wireshark*

Pada konfigurasi *Wireshark* perlu dilakukan beberapa tahapan konfigurasi dimana pada aplikasi *Wireshark* kita akan memanfaatkan beberapa fitur seperti *Summary* serta *Stream analisis* untuk mendapatkan *delay* transmisi maupun *jitter* serta *packet loss* pada transmisi jaringan video *streaming* yang akan di jelaskan dalam bentuk gambar dan tulisan yang ditunjukkan pada Gambar 3.14 sebagai berikut:



Gambar 3.14. Konfigurasi *Wireshark* tahap 1

Pada gambar 3.14 menjelaskan tentang tahapan pertama yang dilakukan untuk mengkonfigurasi aplikasi *Wireshark* yaitu dengan membuka jendela utama aplikasi tersebut.



Gambar 3.15. Konfigurasi *Wireshark* tahap 2

Pada gambar 3.15 menjelaskan tentang tahapan kedua yang dilakukan untuk mengkonfigurasi aplikasi *Wireshark* dengan cara memilih *interface* yang digunakan sebagai media komunikasi pada jaringan LAN. Setelah melalui dua tahapan diatas kita sudah siap melakukan analisa dengan menggunakan fitur-fitur yang di sediakan oleh aplikasi *Wireshark*.

## BAB IV IMPLEMENTASI DAN ANALISIS

### 4.1 Implementasi Sistem

Implementasi sistem adalah tahap penerapan sistem yang telah dibuat dalam tahap perancangan agar sistem tersebut siap untuk dioperasikan. Implementasi sistem untuk analisa performansi *Real Time Streaming Protokol* (RTSP) untuk aplikasi *video streaming* pada jaringan *Mobile IPv4* dengan menggunakan *Video LAN Client* (VLC) dibagi menjadi 2 kategori, yaitu implementasi perangkat lunak dan implementasi perangkat keras.

#### 4.1.1 Implementasi perangkat lunak

Dalam implementasi sistem analisa performansi *Real Time Streaming Protokol* (RTSP) untuk aplikasi *video streaming* pada jaringan *Mobile IPv4* dengan menggunakan *Video LAN Client* (VLC) ini menggunakan beberapa perangkat lunak sebagai berikut:

- a. Sistem Operasi *Windows 7*.
- b. *Video LAN Client* (VLC).
- c. *Wireshark*.

#### 4.1.2 Implementasi perangkat keras

Dalam implementasi sistem analisa performansi *Real Time Streaming Protokol* (RTSP) untuk aplikasi *video streaming* pada jaringan *Mobile IPv4* dengan menggunakan *Video LAN Client* (VLC) ini menggunakan beberapa perangkat keras sebagai berikut:

- a. Satu buah PC untuk *server* dengan spesifikasi minimum :
  - a) Pentium IV 2.6 Ghz.
  - b) 256 MB RAM.
  - c) 4 GB *Harddisk Space*.
- b. *Smartphone* dengan dukungan VLC.
- c. *Akses point*.

- d. Kabel cat 5 + 1 pasang pin RJ 45.
- e. Kabel cat 5e + 1 pasang pin RJ 45.
- f. Kabel cat 6 + 1 pasang pin RJ 45.

## 4.2 Pengujian Sistem

Setelah berhasil di implementasikan, langkah selanjutnya adalah melakukan serangkaian pengujian sistem tersebut serta melakukan beberapa analisa. Pengujian sistem dan analisa dilakukan untuk mengetahui kinerja sitem tersebut serta hasil pengujiannya dapat dianalisa sesuai dengan kebutuhan untuk performansi *Real Time Streaming Protokol* (RTSP) untuk aplikasi video *streaming* pada jaringan *Mobile IPv4* dengan menggunakan *Video LAN Client* (VLC).

### 4.2.1 Pengujian Hasil Pengukuran *Delay* Transmisi

Pengujian *delay* transmisi ialah pengujian untuk mendapatkan waktu yang diperlukan sebuah paket data untuk melintasi suatu media. *Transmission delay* ditentukan oleh kecepatan ( $v$ ) media dan besar paket data.

Berikut ini adalah parameter kualitas sebuah jaringan dilihat dari besarnya *delay* menurut ITU-T dapat dilihat pada tabel 4.1 dibawah ini.

Tabel 4.1 Standarisasi ITU-T *Delay*.

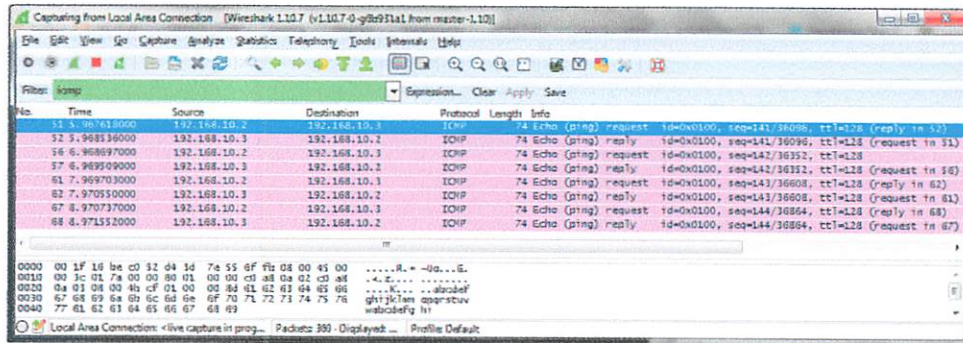
Kategori Delay	Besar Delay
Excellent	< 150 ms
Good	150 – 300 ms
Poor	300– 450 ms
Unnacceptable	>450 ms

#### 1. Pengujian *delay* transmisi menggunakan kabel UTP Cat 5 dan Akses *Point* (WiFi).

Pada pengujian ini media komunikasi dari *server* ke akses *point* menggunakan kabel cat 5 dan dari akses *point* menuju perangkat *mobile* (*smartphone*) dengan menggunakan media *wireless* (WiFi). Berikut akan

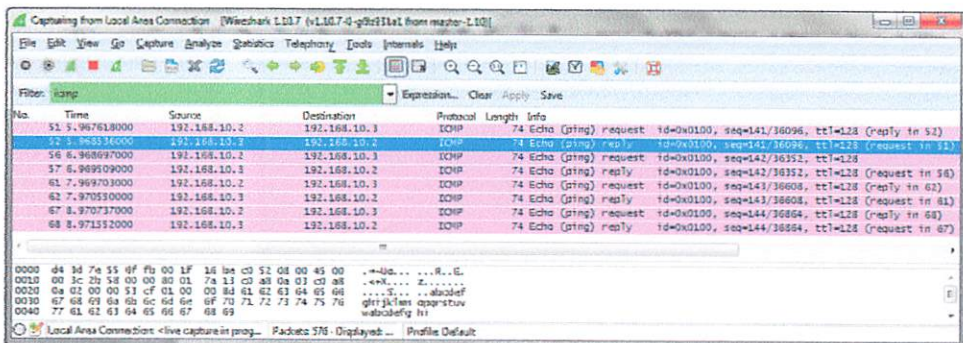


dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi *server*.



Gambar 4.1. *Wireshark Capture ICMP Request 1.*

Nilai 5.967618000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.1 adalah waktu dimana paket ping pertama dikirimkan oleh perangkat *mobile client*.



Gambar 4.2. *Wireshark Capture ICMP Reply 1.*

Nilai 5.968536000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.2 adalah waktu dimana paket ping pertama diterima oleh perangkat *mobile client*.

Dari kedua nilai *request and reply 1* tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$DELAY = \text{Waktu paket diterima (Request 1)} - \text{Waktu Paket dikirim (Reply 1)}$$

Sehingga didapatkan :

$$\text{Delay} = 5.968536000 - 5.967618000 = 0.000918 \text{ s} = 0.918 \text{ ms}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke komputer *server* adalah 0.918 ms

No.	Time	Source	Destination	Protocol	Length	Info
51	5.967618000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=141/36096, ttl=128 (reply in 52)
52	5.968536000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=141/36096, ttl=128 (request in 51)
56	6.968697000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=142/36132, ttl=128
57	6.969509000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=142/36132, ttl=128 (request in 56)
61	7.969703000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=143/36608, ttl=128 (reply in 61)
62	7.970530000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=143/36608, ttl=128 (request in 61)
67	8.970737000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=144/36864, ttl=128 (reply in 68)
68	8.971552000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=144/36864, ttl=128 (request in 67)

Gambar 4.3. Wireshark Capture ICMP Request 2.

Nilai 6.968697000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.3 adalah waktu dimana paket ping kedua dikirimkan oleh perangkat *mobile client*.

No.	Time	Source	Destination	Protocol	Length	Info
51	5.967618000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=141/36096, ttl=128 (reply in 52)
52	5.968536000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=141/36096, ttl=128 (request in 51)
56	6.968697000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=142/36132, ttl=128
57	6.969509000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=142/36132, ttl=128 (request in 56)
61	7.969703000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=143/36608, ttl=128 (reply in 62)
62	7.970530000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=143/36608, ttl=128 (request in 61)
67	8.970737000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=144/36864, ttl=128 (reply in 68)
68	8.971552000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=144/36864, ttl=128 (request in 67)

Gambar 4.4. Wireshark Capture ICMP Reply 2

Nilai 6.969509000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.4 adalah waktu dimana paket ping kedua diterima oleh perangkat *mobile client*.

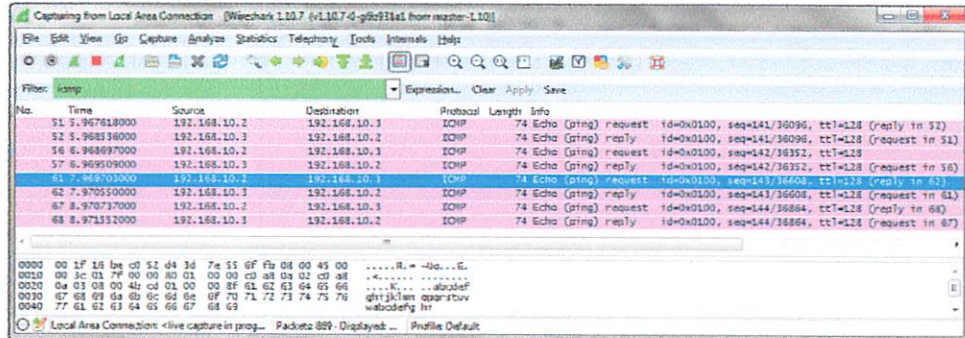
Dari kedua nilai *request and reply* 2 tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 2)} - \text{Waktu Paket dikirim (Reply 2)}$$

Sehingga didapatkan :

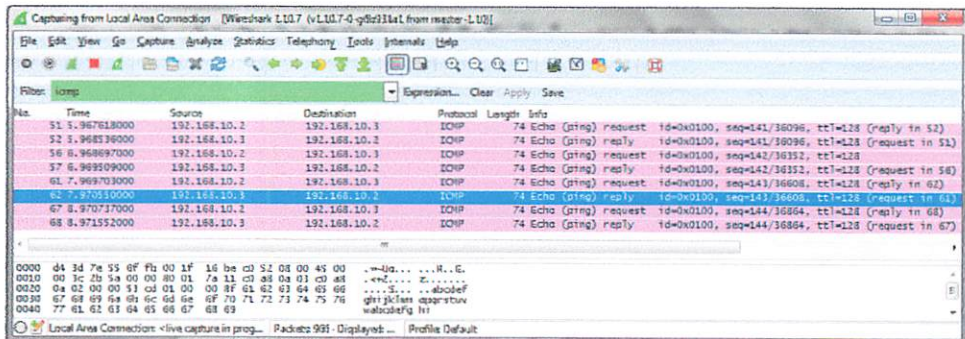
$$Delay = 6.969509000 - 6.968697000 = 0.000812 \text{ s} = 0.812 \text{ ms}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile* ke komputer *server* adalah 0.812 ms.



Gambar 4.5. *Wireshark Capture ICMP Request 3.*

Nilai 7.969703000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.5 adalah waktu dimana paket ping ketiga dikirimkan oleh perangkat *mobile client*.



Gambar 4.6. *Wireshark Capture ICMP Reply 3.*

Nilai 7.970500000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.6 adalah waktu dimana paket ping ketiga diterima oleh perangkat *mobile client*.

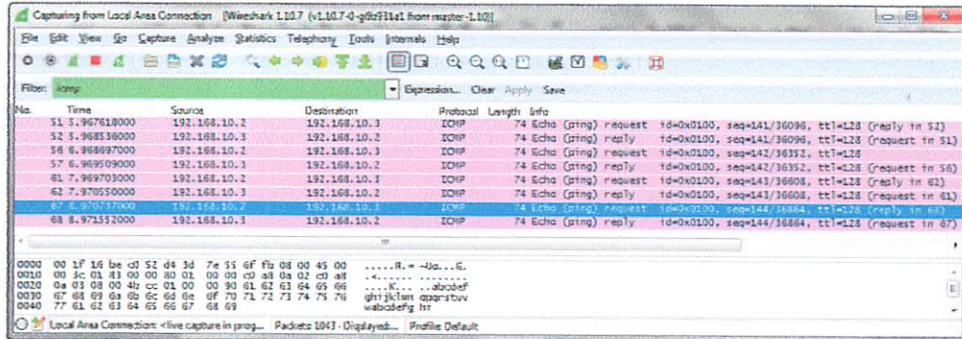
Dari kedua nilai *request and reply 3* tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$DELAY = \text{Waktu paket diterima (Request 3)} - \text{Waktu Paket dikirim (Reply 3)}$$

Sehingga didapatkan :

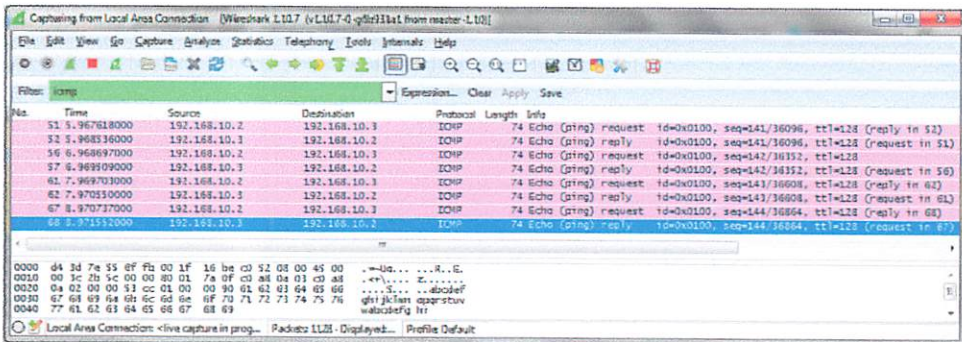
Delay = 7.970550000 - 7.969703000 = 0.000847 s = 0.847 ms.

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke komputer server adalah 0.847 ms.



Gambar 4.7. Wireshark Capture ICMP Request 4.

Nilai 8.970737000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.7 adalah waktu dimana paket ping keempat dikirimkan oleh perangkat *mobile client*.



Gambar 4.8. Wireshark Capture ICMP Reply 4.

Nilai 8.971552000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.8 adalah waktu dimana paket ping keempat diterima oleh perangkat *mobile client*.

Dari kedua nilai *request and reply* 4 tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 4)} - \text{Waktu Paket dikirim (Reply 4)}$$

Sehingga didapatkan :

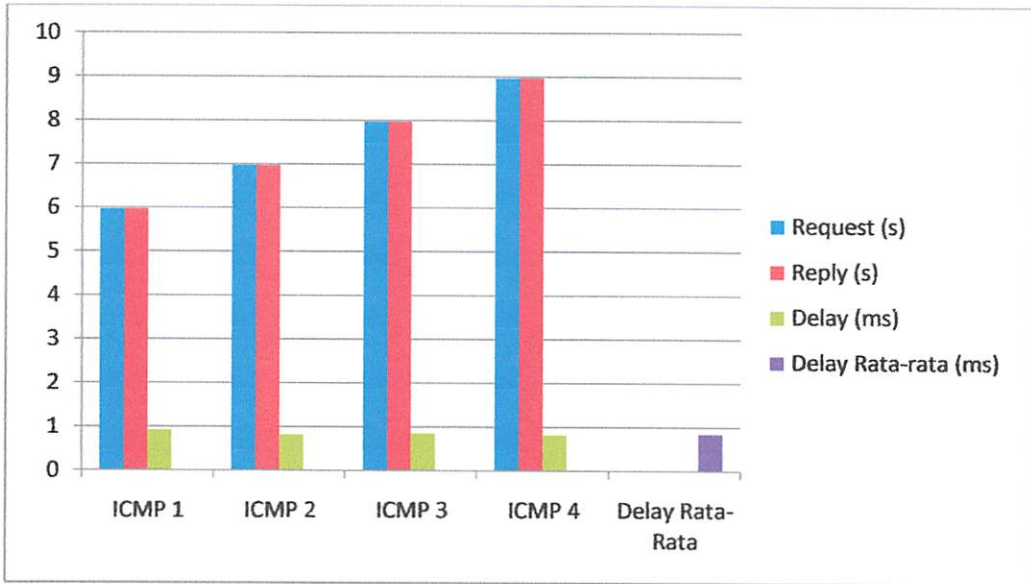
$$\text{Delay} = 8.971552000 - 8.970737000 = 0.000815 \text{ s} = 0.815 \text{ ms.}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke komputer *server* adalah 0.815 ms.

Berikut adalah tabel 4.2 dan grafik 4.1 hasil pengujian *delay* transmisi dari perangkat *mobile* sebagai *client* ke komputer *server*.

Tabel 4.2. Tabel hasil pengujian *delay* transmisi dari perangkat *mobile* sebagai *client* ke komputer *server*.

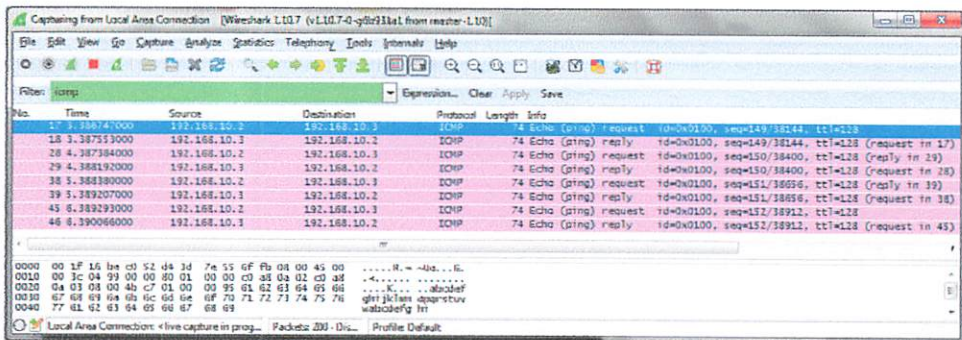
No.	Request (s)	Reply (s)	Delay (s) (replay- request)	Delay (ms) (replay- request)
1	5.967618000	5.968536000	0.000918	0.918 ms
2	6.968697000	6.969509000	0.000812	0.812 ms
3	7.969703000	7.970550000	0.000847	0.847 ms
4	8.970737000	8.971552000	0.000815	0.815 ms
Rata-rata Delay				0.848 ms



Grafik 4.1. Grafik hasil pengujian *delay* transmisi dari perangkat *mobile* sebagai *client* ke komputer *server*.

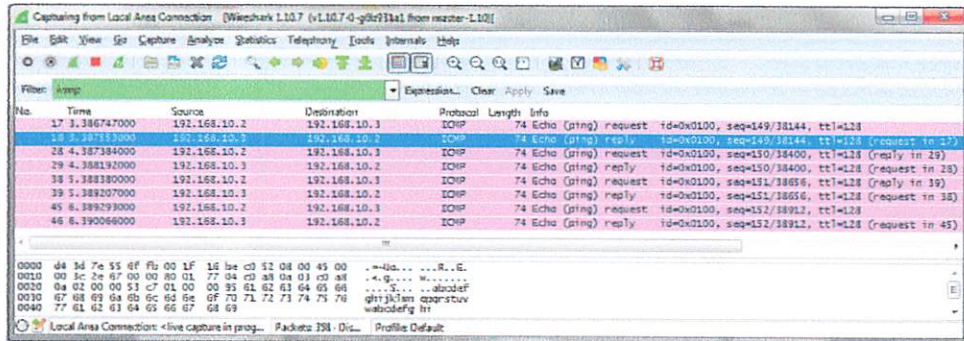
2. Pengujian *delay* transmisi menggunakan kabel UTP Cat 5e dan Akses Point (WiFi).

Pada pengujian ini media komunikasi dari *server* ke akses *point* menggunakan kabel cat 5e dan dari akses *point* menuju perangkat *mobile* (*smartphone*) dengan menggunakan media *wireless* (WiFi). Berikut akan dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi *server*.



Gambar 4.9. *Wireshark Capture ICMP Request 1.*

Nilai 3.386747000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.9 adalah waktu dimana paket ping pertama dikirimkan oleh perangkat *mobile client*.



Gambar 4.10. *Wireshark Capture ICMP Reply 1.*

Nilai 3.387553000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.10 adalah waktu dimana paket ping pertama diterima oleh perangkat *mobile client*.

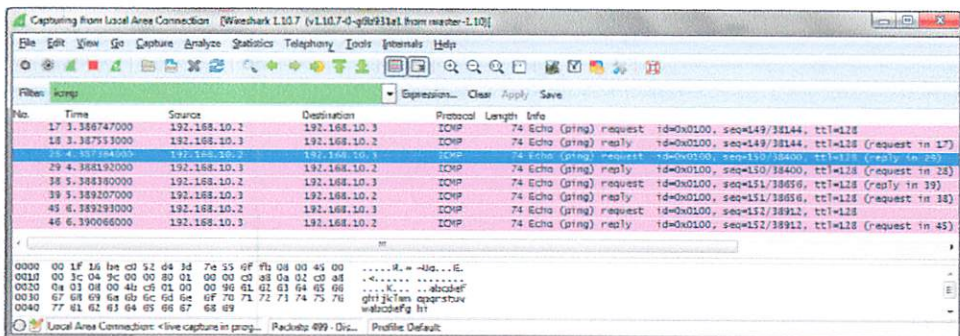
Dari kedua nilai *request and reply 1* tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 1)} - \text{Waktu Paket dikirim (Reply 1)}$$

Sehingga didapatkan :

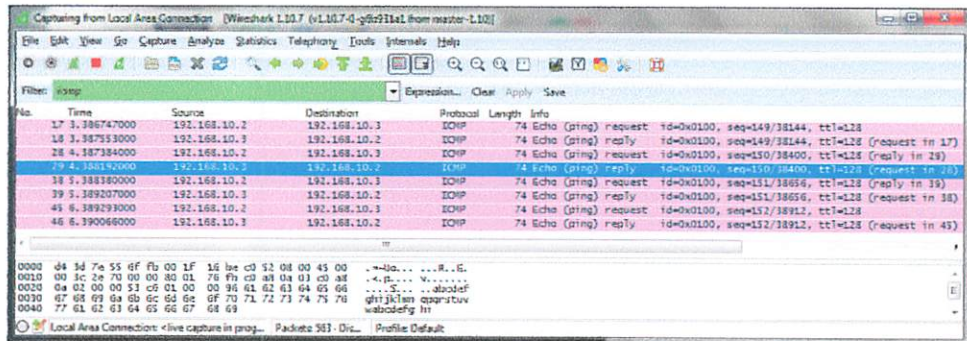
$$\text{Delay} = 3.387553000 - 3.386747000 = 0.000806 \text{ s} = 0.806 \text{ ms.}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 *bytes* dari perangkat *mobile client* ke *server* adalah 0.806 ms.



Gambar 4.11. *Wireshark Capture ICMP Request 2.*

Nilai 4.387384000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.11 adalah waktu dimana paket ping kedua dikirimkan oleh perangkat *mobile client*.



Gambar 4.12. Wireshark Capture ICMP Reply 2

Nilai 4.388192000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.12 adalah waktu dimana paket ping kedua diterima oleh perangkat *mobile client*.

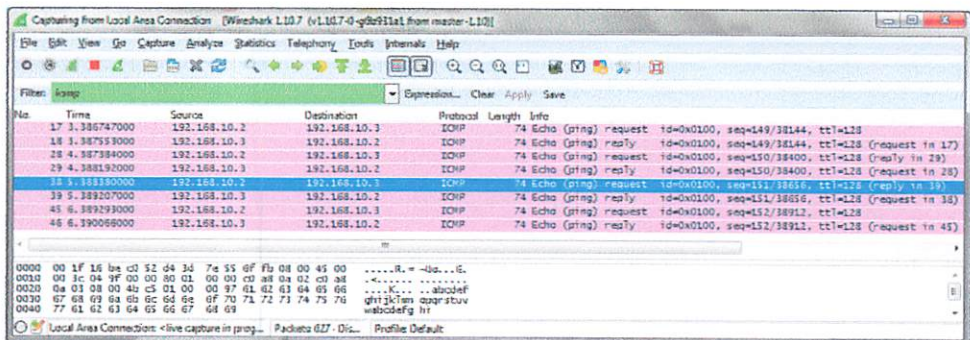
Dari kedua nilai *request and reply 2* tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$DELAY = Waktu\ paket\ diterima\ (Request\ 2) - Waktu\ Paket\ dikirim\ (Reply\ 2)$$

Sehingga didapatkan :

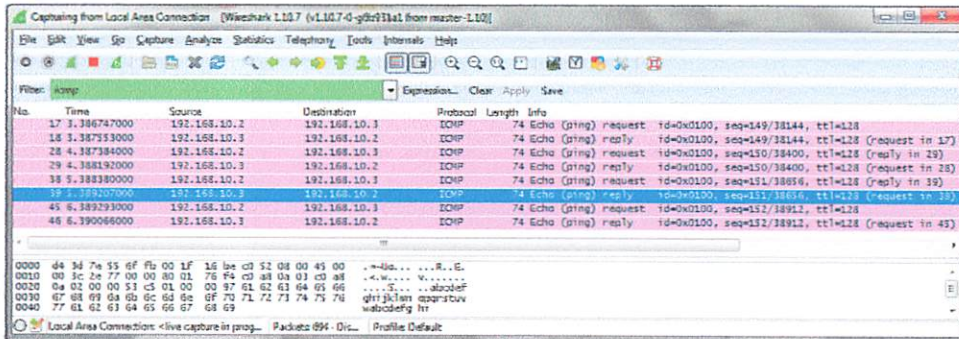
$$Delay = 4.388192000 - 4.387384000 = 0.000808\ s = 0.808\ ms.$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke *server* adalah 0.808 ms.



Gambar 4.13. Wireshark Capture ICMP Request 3.

Nilai 5.388380000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.13 adalah waktu dimana paket ping ketiga dikirimkan oleh perangkat *mobile client*.



Gambar 4.14. Wireshark Capture ICMP Reply 3.

Nilai 5.389207000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.14 adalah waktu dimana paket ping ketiga diterima oleh perangkat *mobile*.

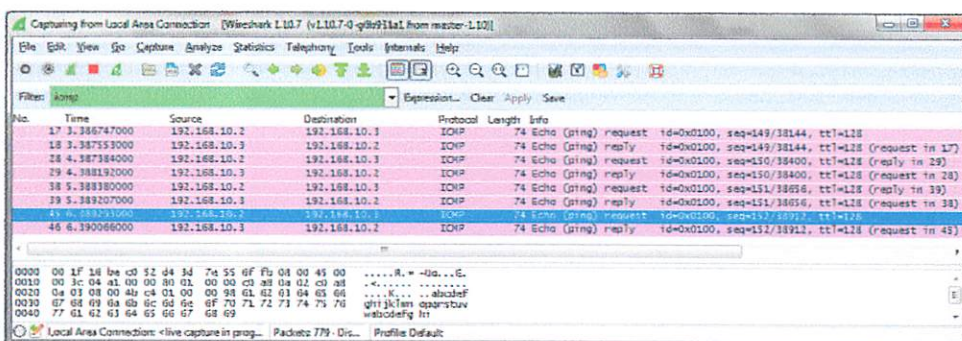
Dari kedua nilai *request and reply* 3 tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 3)} - \text{Waktu Paket dikirim (Reply 3)}$$

Sehingga didapatkan :

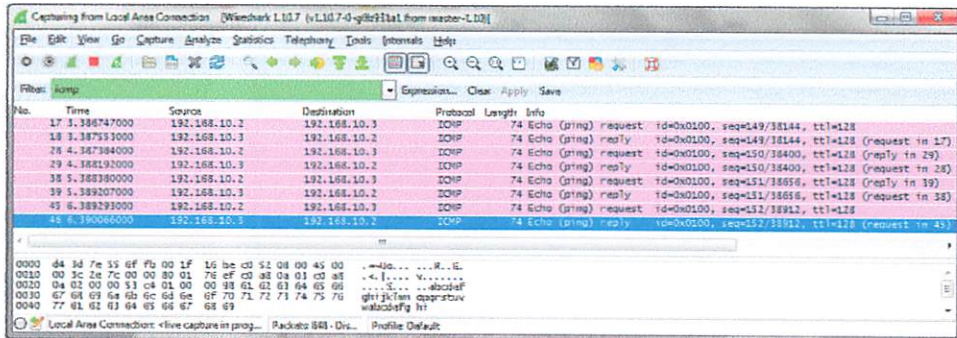
$$\text{Delay} = 5.389207000 - 5.388380000 = 0.000827 \text{ s} = 0.827 \text{ ms.}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke computer *server* adalah 0.827 ms.



Gambar 4.15. Wireshark Capture ICMP Request 4.

Nilai 6.389293000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.15 adalah waktu dimana paket ping keempat dikirimkan oleh perangkat *mobile client*.



Gambar 4.16. *Wireshark Capture ICMP Reply 4.*

Nilai 6.390066000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.16 adalah waktu dimana paket ping keempat diterima oleh perangkat *mobile client*.

Dari kedua nilai *request-reply* 4 tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 4)} - \text{Waktu Paket dikirim (Reply 4)}$$

Sehingga didapatkan :

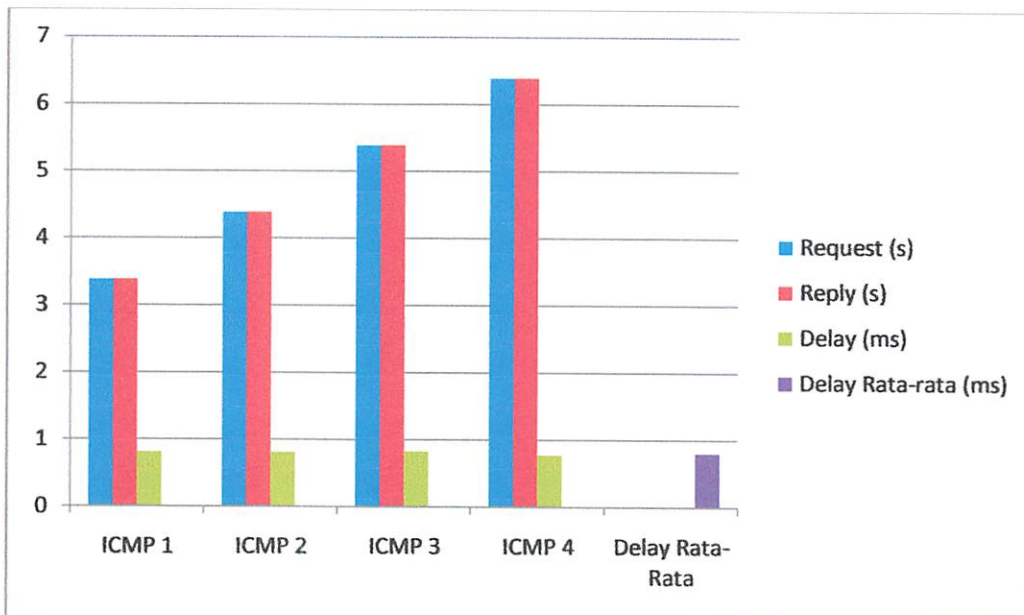
$$\text{Delay} = 6.390066000 - 6.389293000 = 0.000773 \text{ s} = 0.773 \text{ ms.}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke komputer server adalah 0.773 ms.

Berikut adalah tabel 4.3 dan grafik 4.2 hasil pengujian *delay* transmisi dari perangkat *mobile client* ke komputer server :

Tabel 4.3. Tabel hasil pengujian *delay* transmisi dari perangkat *mobile client* ke *computer server*.

No.	Request (s)	Reply (s)	Delay (s) (replay- request)	Delay (ms) (replay- request)
1	3.386747000	3.387553000	0.000806	0.806 ms
2	4.387384000	4.388192000	0.000808	0.808 ms
3	5.388380000	5.389207000	0.000827	0.827 ms
4	6.389293000	6.390066000	0.000773	0.773 ms
Rata-rata Delay				0.8035 ms

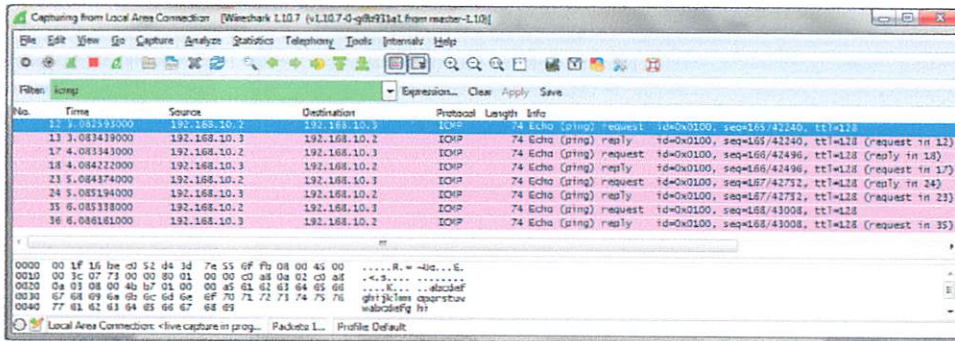


Grafik 4.2. Grafik hasil pengujian *delay* transmisi dari perangkat *mobile client* ke komputer *server*.

### 3. Pengujian *delay* transmisi menggunakan kabel UTP Cat 6 dan Akses Point (WiFi).

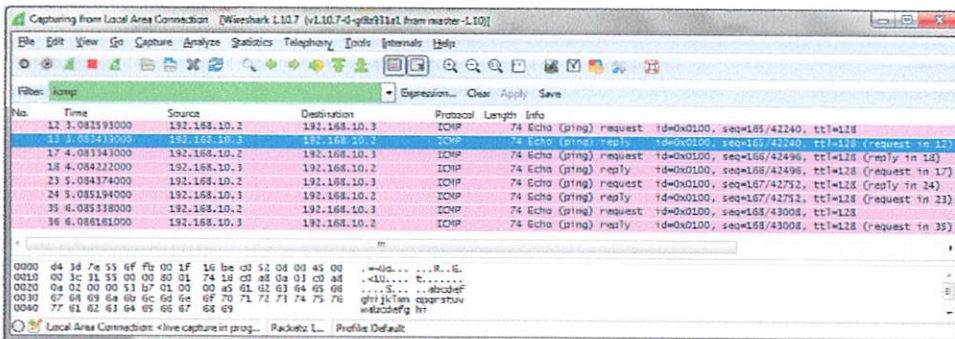
Pada pengujian ini media komunikasi dari *server* ke akses *point* menggunakan kabel cat 6 dan dari akses *point* menuju perangkat *mobile* (*smartphone*) dengan menggunakan media *wireless* (WiFi). Berikut akan

dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi *server*.



Gambar 4.17. *Wireshark Capture ICMP Request 1.*

Nilai 3.082593000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.17 adalah waktu dimana paket ping pertama dikirimkan oleh perangkat *mobile client*.



Gambar 4.18. *Wireshark Capture ICMP Reply 1.*

Nilai 3.083439000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.18 adalah waktu dimana paket ping pertama diterima oleh perangkat *mobile client*.

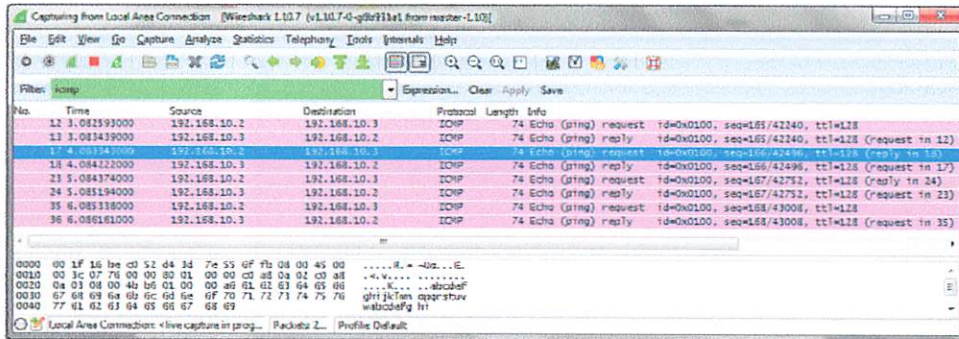
Dari kedua nilai *request and reply 1* tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 1)} - \text{Waktu Paket dikirim (Reply 1)}$$

Sehingga didapatkan :

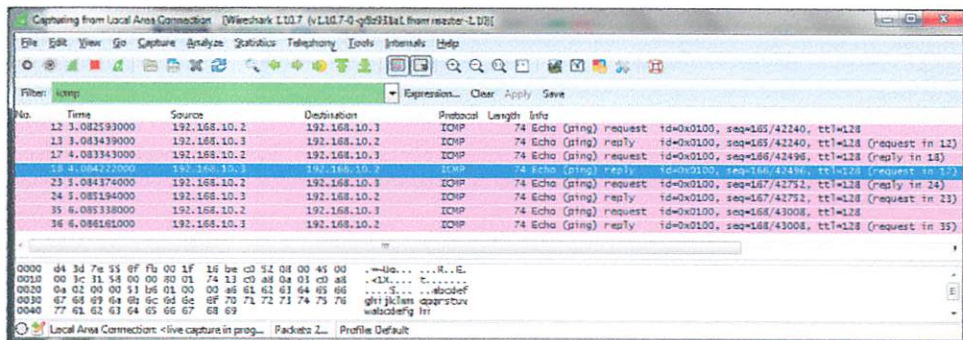
$$\text{Delay} = 3.083439000 - 3.082593000 = 0.000846 \text{ s} = 0.846 \text{ ms}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke server adalah 0.846 ms.



Gambar 4.19. Wireshark Capture ICMP Request 2

Nilai 4.083343000 seconds yang bermarka biru yang ditunjukkan pada gambar 4.19 adalah waktu dimana paket ping kedua dikirimkan oleh perangkat *mobile client*.



Gambar 4.20. Wireshark Capture ICMP Reply 2

Nilai 4.084222000 seconds yang bermarka biru yang ditunjukkan pada gambar 4.20 adalah waktu dimana paket ping kedua diterima oleh perangkat *mobile client*.

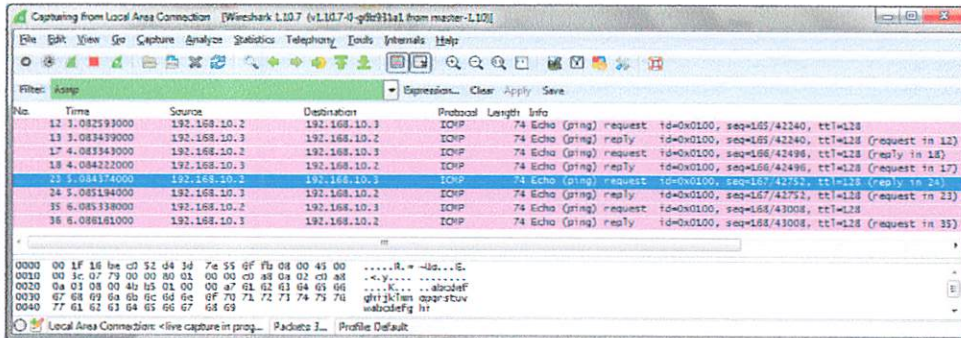
Dari kedua nilai *request and reply* 2 tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 2)} - \text{Waktu Paket dikirim (Reply 2)}$$

Sehingga didapatkan :

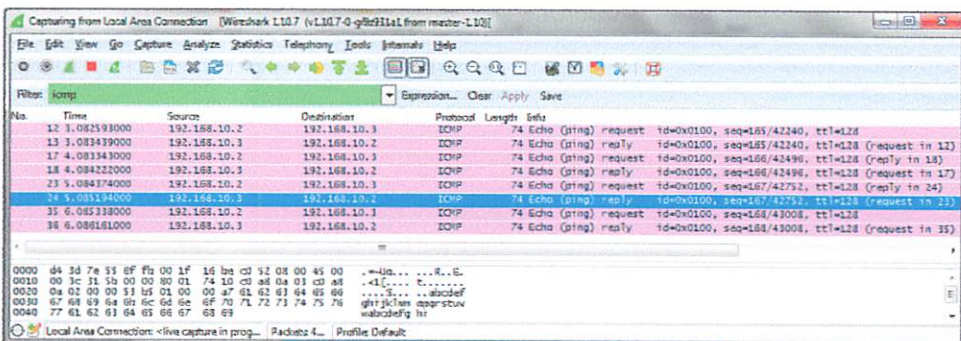
$$\text{Delay} = 4.084222000 - 4.083343000 = 0.000879 \text{ s} = 0.879 \text{ ms}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke komputer server adalah 0.879 ms.



Gambar 4.21. Wireshark Capture ICMP Request 3.

Nilai 5.084374000 *seconds* yang bermarga biru yang ditunjukkan pada gambar 4.21 adalah waktu dimana paket ping ketiga dikirimkan oleh perangkat *mobile client*.



Gambar 4.22. Wireshark Capture ICMP Reply 3

Nilai 5.085194000 *seconds* yang bermarga biru yang ditunjukkan pada gambar 4.22 adalah waktu dimana paket ping ketiga diterima oleh perangkat *mobile client*.

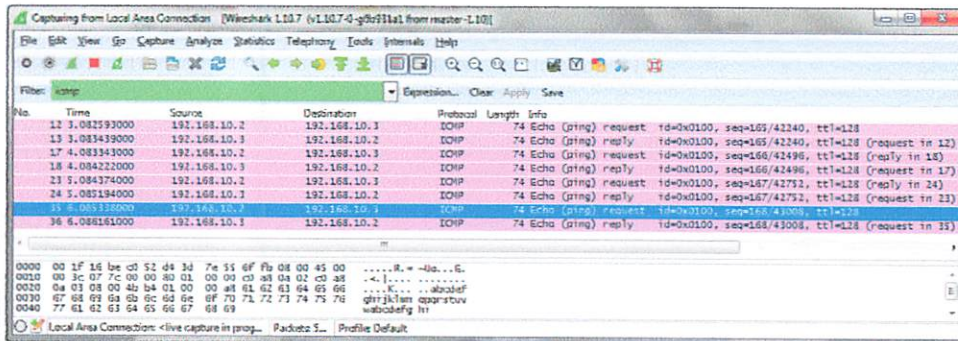
Dari kedua nilai *request and reply* 3 tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 3)} - \text{Waktu Paket dikirim (Reply 3)}$$

Sehingga didapatkan :

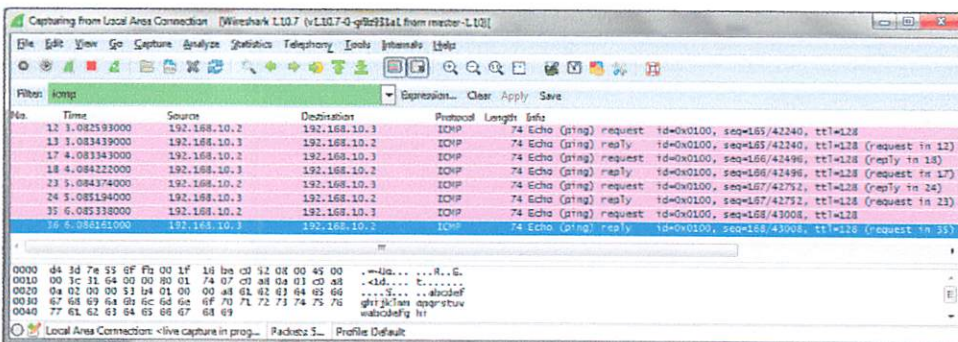
$$\text{Delay} = 5.085194000 - 5.084374000 = 0.000820 \text{ s} = 0.820 \text{ ms.}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari *computer client* ke server adalah 0.820 ms.



Gambar 4.23. Wireshark Capture ICMP Request 4.

Nilai 6.085338000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.23 adalah waktu dimana paket ping keempat dikirimkan oleh perangkat *mobile client*.



Gambar 4.24. Wireshark Capture ICMP Reply 4 menggunakan kabel cat 5e

Nilai 6.086161000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.24 adalah waktu dimana paket ping keempat diterima oleh perangkat *mobile client*.

Dari kedua nilai *request and reply* 4 tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 4)} - \text{Waktu Paket dikirim (Reply 4)}$$

Sehingga didapatkan :

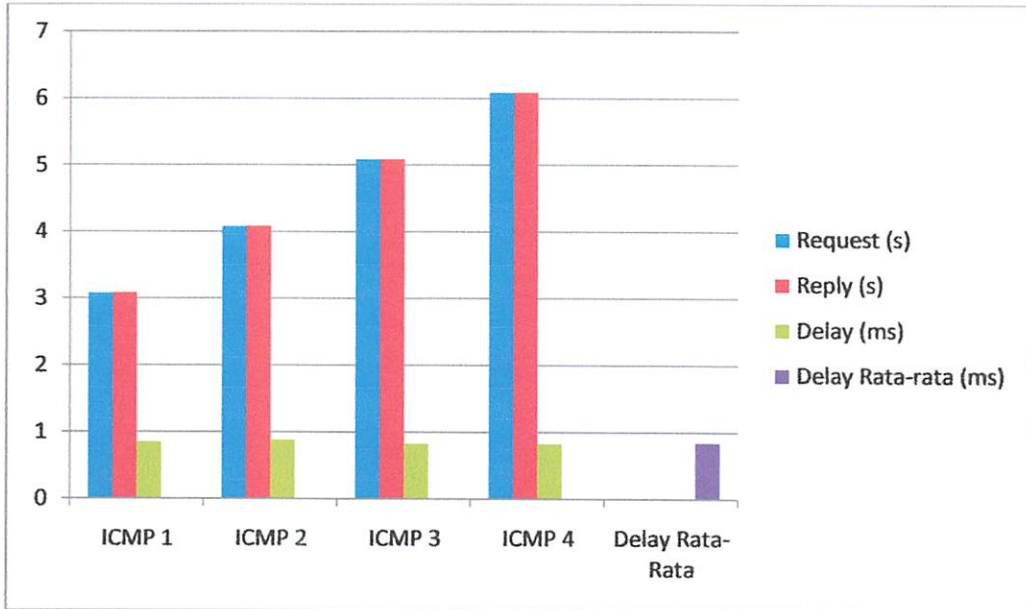
$$\text{Delay} = 6.086161000 - 6.085338000 = 0.000823 \text{ s} = 0.823 \text{ ms.}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke server adalah 0.823 ms.

Berikut adalah table 4.4 dan grafik 4.3 hasil pengujian *delay* transmisi dari perangkat *computer client* ke computer *server* menggunakan kabel cat 6:

Tabel 4.4. Tabel hasil pengujian *delay* transmisi dari computer *client* ke computer *server* menggunakan kabel cat 6.

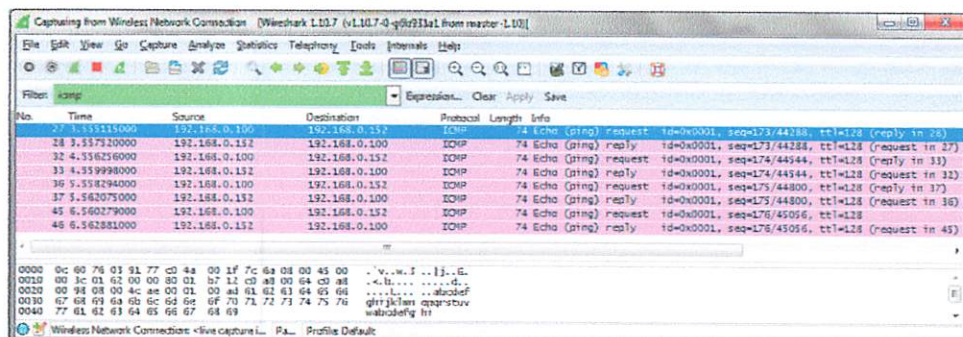
No.	Request (s)	Reply (s)	Delay (s) (replay- request)	Delay (ms) (replay- request)
1	3.082593000	3.083439000	0.000846	0.846 ms
2	4.083343000	4.084222000	0.000879	0.879 ms
3	5.084374000	5.085194000	0.000820	0.820 ms
4	6.085338000	6.086161000	0.000823	0.823 ms
Rata-rata Delay				0.842 ms



Grafik 4.3. Grafik *Delay* Transmisi dari perangkat computer *client* ke komputer *server* menggunakan kabel cat 6.

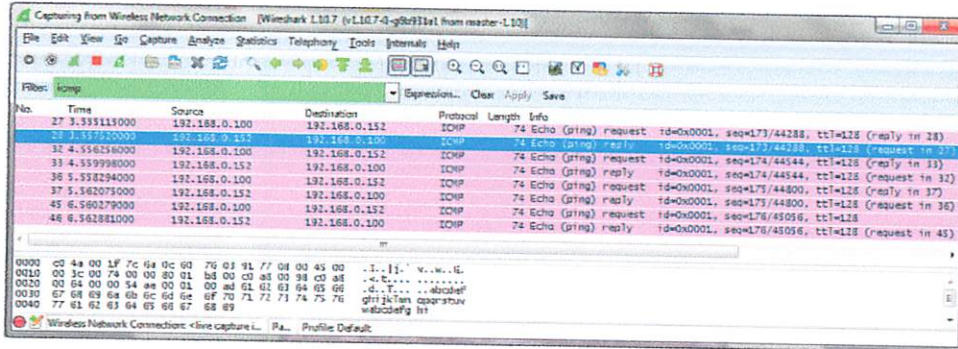
#### 4. Pengujian delay transmisi menggunakan WiFi (*Ad hoc Peer to peer*).

Pada pengujian dengan menggunakan WiFi, *server* menggunakan laptop sedangkan dari sisi *client* menggunakan perangkat *mobile* (*smartphone*) yang sudah terinstall VLC, komunikasi kedua perangkat ini menggunakan *peer to peer ad hoc* dimana kedua perangkat tersambung langsung dengan menggunakan media nirkabel (WiFi). Berikut tampilan *Wireshark* yang diambil dari sisi *server*.



Gambar 4.25. *Wireshark* Capture ICMP Request 1.

Nilai 3.555115000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.25 adalah waktu dimana paket ping pertama dikirimkan oleh perangkat *mobile client*.



Gambar 4.26. *Wireshark Capture ICMP Reply 1.*

Nilai 3.557520000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.26 adalah waktu dimana paket ping pertama diterima oleh perangkat *mobile client*.

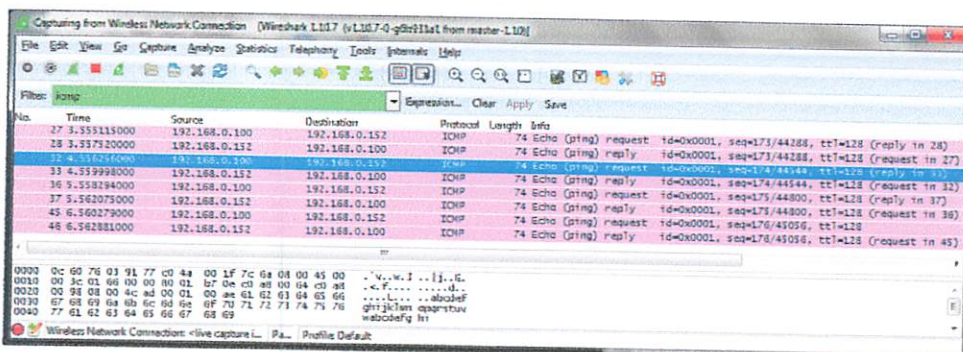
Dari kedua nilai *request and reply 1* tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 1)} - \text{Waktu Paket dikirim (Reply 1)}$$

Sehingga didapatkan :

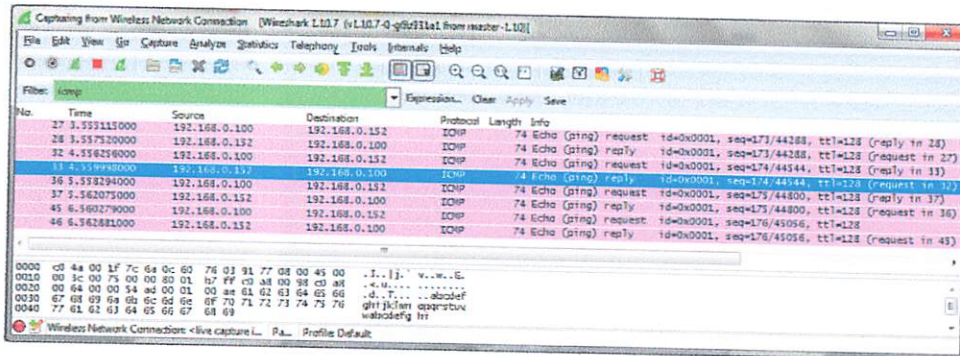
$$\text{Delay} = 3.557520000 - 3.555115000 = 0.002405 \text{ s} = 2.405 \text{ ms}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke komputer server adalah 2.405 ms.



Gambar 4.27. *Wireshark Capture ICMP Request 2*

Nilai 4.556256000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.27 adalah waktu dimana paket ping kedua dikirimkan oleh perangkat *mobile client*.



Gambar 4.28. Wireshark Capture ICMP Reply 2

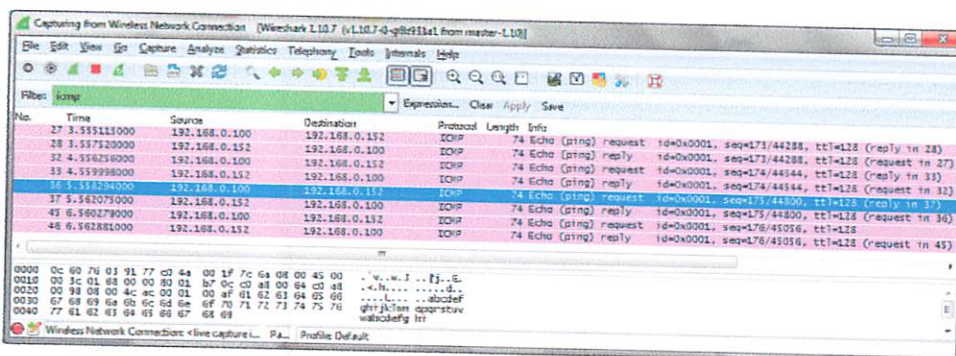
Nilai 4.559998000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.28 adalah waktu dimana paket ping kedua diterima oleh perangkat *mobile client*. Dari kedua nilai *request and reply 2* tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 2)} - \text{Waktu Paket dikirim (Reply 2)}$$

Sehingga didapatkan :

$$\text{Delay} = 4.559998000 - 4.556256000 = 0.003742 \text{ s} = 3.742 \text{ ms}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke *server* adalah 3.742 ms.



Gambar 4.29. Wireshark Capture ICMP Request 3

Nilai 5.558294000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.29 adalah waktu dimana paket ping ketiga dikirimkan oleh perangkat *mobile client*.

No.	Time	Source	Destination	Protocol	Length	Info
27	3.558115000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request id=0x0001, seq=L73/44288, ttl=128 (reply in 28)
28	3.557520000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=L73/44288, ttl=128 (request in 27)
32	4.558256000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request id=0x0001, seq=L74/44544, ttl=128 (request in 31)
33	4.559998000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=L74/44544, ttl=128 (request in 32)
36	5.558294000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request id=0x0001, seq=L75/44800, ttl=128 (request in 37)
37	5.560279000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=L75/44800, ttl=128 (request in 36)
43	6.560279000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request id=0x0001, seq=L76/45056, ttl=128 (request in 45)
46	6.561881000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=L76/45056, ttl=128 (request in 45)

Gambar 4.30. *Wireshark Capture ICMP Reply 3*

Nilai 5.562075000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.30 adalah waktu dimana paket ping ketiga diterima oleh perangkat *mobile client*.

Dari kedua nilai *request and reply 3* tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$DELAY = \text{Waktu paket diterima (Request 3)} - \text{Waktu Paket dikirim (Reply 3)}$$

Sehingga didapatkan :

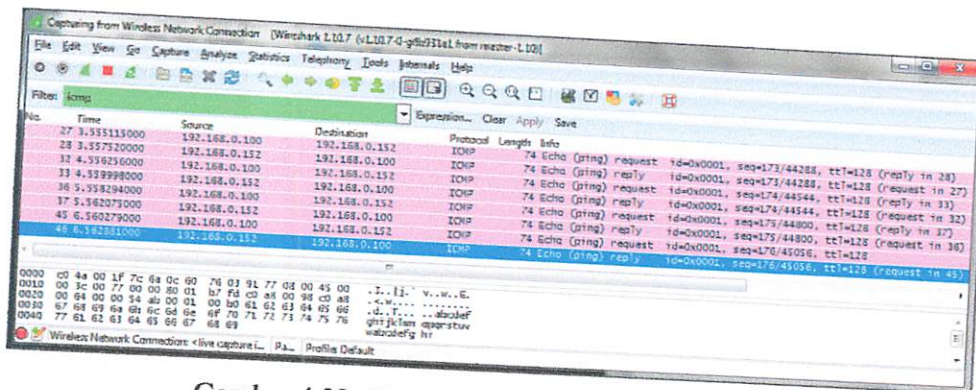
$$Delay = 5.562075000 - 5.558294000 = 0.003781 \text{ s} = 3.781 \text{ ms}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke komputer server adalah 3.781 ms.

No.	Time	Source	Destination	Protocol	Length	Info
27	3.558115000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request id=0x0001, seq=L73/44288, ttl=128 (reply in 28)
28	3.557520000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=L73/44288, ttl=128 (request in 27)
32	4.558256000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request id=0x0001, seq=L74/44544, ttl=128 (request in 31)
33	4.559998000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=L74/44544, ttl=128 (request in 32)
36	5.558294000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request id=0x0001, seq=L75/44800, ttl=128 (request in 37)
37	5.560279000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=L75/44800, ttl=128 (request in 36)
41	6.562075000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request id=0x0001, seq=L76/45056, ttl=128 (request in 45)
46	6.561881000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=L76/45056, ttl=128 (request in 45)

Gambar 4.31. *Wireshark Capture ICMP Request 4*

Nilai 6.560279000 *seconds* yang bermarka biru adalah waktu dimana paket ping keempat dikirimkan oleh perangkat *mobile client*.



Gambar 4.32. Wireshark Capture ICMP Reply 4

Nilai 6.562881000 *seconds* yang bermarka biru yang ditunjukkan pada gambar 4.32 adalah waktu dimana paket ping keempat diterima oleh perangkat *mobile client*.

Dari kedua nilai *request and reply* 4 tersebut kita dapat menghitung *delay* dengan rumus sbb:

$$\text{DELAY} = \text{Waktu paket diterima (Request 4)} - \text{Waktu Paket dikirim (Reply 4)}$$

Sehingga didapatkan :

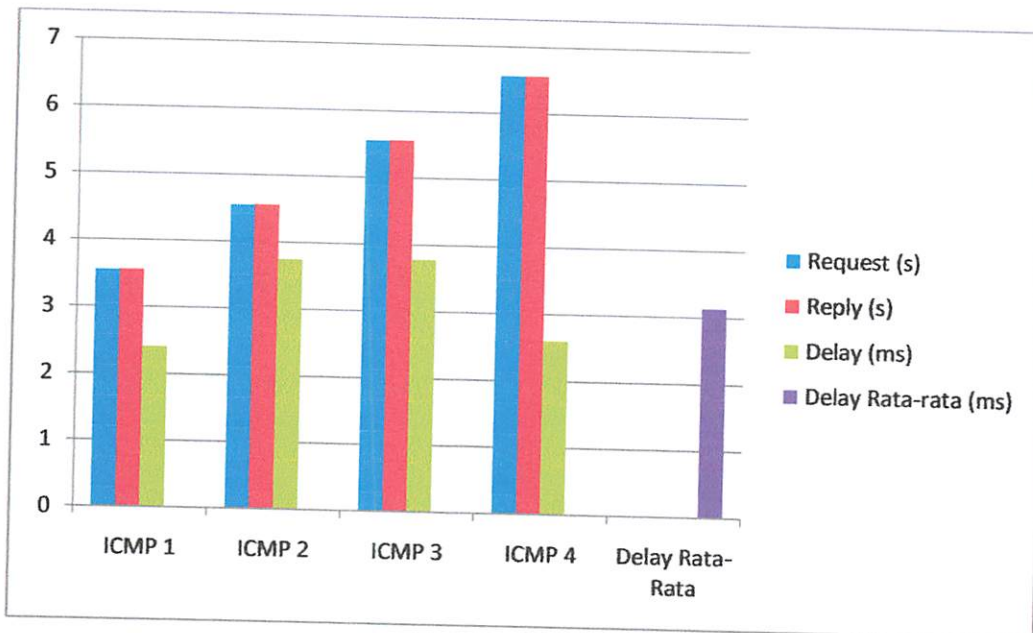
$$\text{Delay} = 6.562881000 - 6.560279000 = 0.002602 \text{ s} = 2.602 \text{ ms}$$

Sehingga *delay* untuk mengirimkan 1 paket 32 bytes dari perangkat *mobile client* ke server adalah 2.602 ms.

Berikut adalah table 4.5 dan grafik 4.4 hasil pengujian *delay* transmisi dari perangkat *mobile client* ke *computer server* menggunakan WiFi secara *ad hoc peer to peer* :

Tabel 4.5. Tabel hasil pengujian *delay* transmisi dari perangkat *mobile client* ke komputer *server* menggunakan WiFi.

No.	Request (s)	Reply (s)	Delay (s) (replay- request)	Delay (ms) (replay- request)
1	3.555115000	3.557520000	0.002405	2.405 ms
2	4.556256000	4.559980000	0.003742	3.742 ms
3	5.558294000	5.562075000	0.003781	3.781 ms
4	6.560279000	6.562881000	0.002602	2.602 ms
Rata-rata Delay				3.1325 ms



Grafik 4.4. Grafik *Delay* Transmisi dari perangkat *mobile client* ke komputer *server* menggunakan WiFi secara *ad hoc peer to peer*.

#### 4.2.2 Pengujian Hasil Pengukuran *Throughput*

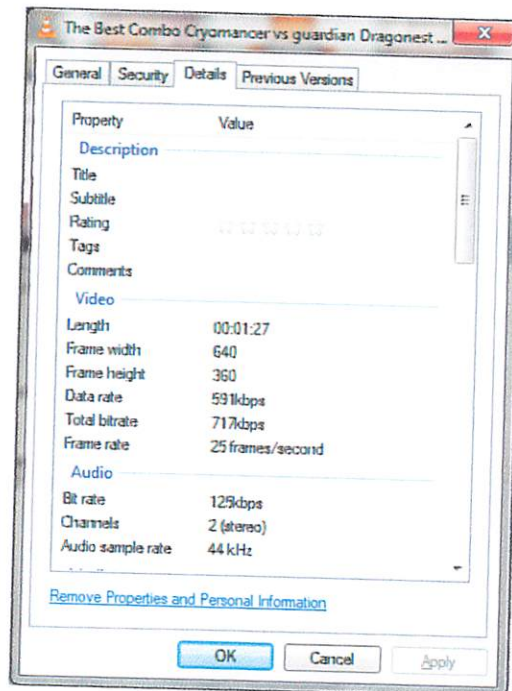
Pada bagian ini akan dibahas tentang analisa *throughput*. Dimana *throughput* adalah kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. Pada umumnya *throughput* selalu dikaitkan dengan *bandwidth*. Karena *throughput* memang bisa disebut juga dengan *bandwidth* dalam kondisi yang sebenarnya. *Bandwidth* lebih bersifat *fix* sementara *throughput* sifatnya adalah dinamis tergantung trafik yang sedang terjadi. Berikut diberikan Rumus *throughput* :

**Rumus *throughput* = Jumlah data yang dikirim / Waktu pengiriman data**

Pada pengukuran kali ini akan digunakan sampling dari 3 file video dengan data *bitrate* sebagai berikut:

1. *Bitrate Video 1* : 717 Kbps.

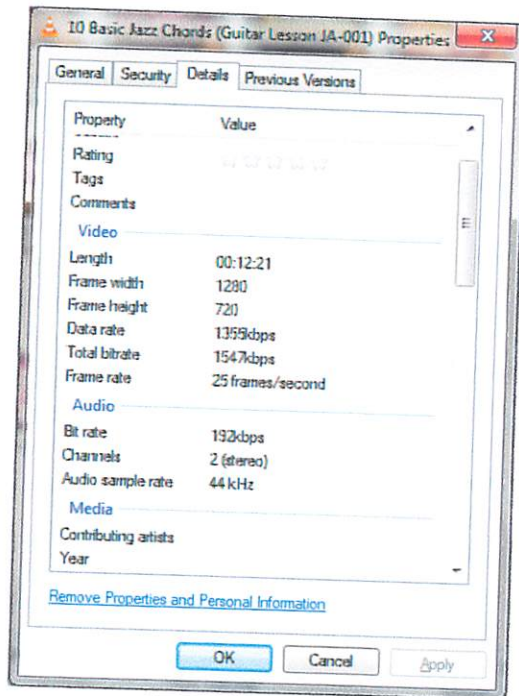
Berikut adalah *properties* pada pengujian *bitrate* video dengan kecepatan 717 Kbps ditunjukkan dalam gambar 4.33.



Gambar 4.33. *video 1 properties*.

2. *Bitrate Video 2: 1547 Kbps.*

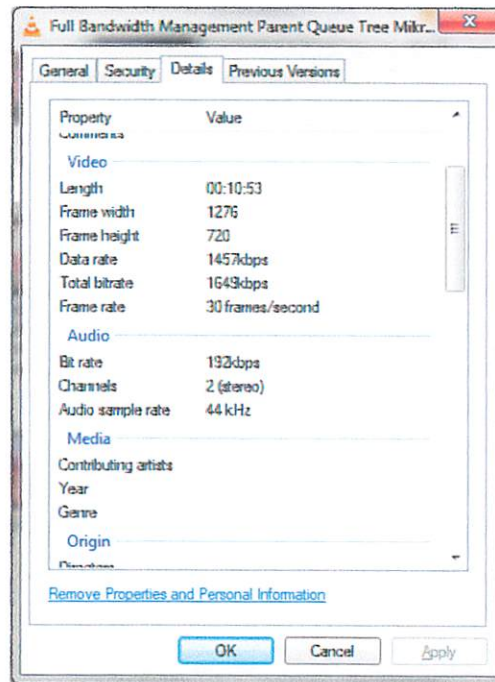
Berikut adalah *properties* pada pengujian *bitrate* video dengan kecepatan 1547 Kbps ditunjukkan dalam gambar 4.34.



Gambar 4.34. *video 2 properties.*

3. *Bitrate Video 3: 1649 Kbps*

Berikut adalah *properties* pada pengujian *bitrate* video dengan kecepatan 1649 Kbps ditunjukkan dalam gambar 4.35.



Gambar 4.35. video 3 properties.

##### 5. Pengujian hasil pengukuran *throughput* menggunakan kabel UTP Cat 5 dan Akses Point (WiFi).

Pada pengujian ini media komunikasi dari *server* ke akses *point* menggunakan kabel cat 5 dan dari akses point menuju perangkat *mobile (smartphone)* dengan menggunakan media *wireless* (WiFi). Berikut akan dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi *server* untuk mendapatkan nilai *throughput*.

Aplikasi pendukung yang dipergunakan untuk menganalisis *throughput* menggunakan kabel UTP cat 5 ialah *Wireshark* dengan cara klik tabs *statistic* lalu klik *summary*, sehingga didapatkan hasil seperti yang ditunjukkan pada data gambar 4.36.

Display					
Display filter:	none				
Ignored packets:	0 (0.000%)				
Traffic	◀ Captured	◀ Displayed	◀ Displayed %	◀ Marked	◀ Marked %
Packets	268	268	100.000%	0	0.000%
Between first and last packet:	4.520 sec				
Avg. packets/sec	59.289				
Avg. packet size	693.399 bytes				
Bytes	185831	185831	100.000%	0	0.000%
Avg. bytes/sec	41110.605				
Avg. MBit/sec	0.329				

Gambar 4.36. *Throughput video 1.*

Dalam menghitung *throughput* berdasarkan hasil pengujian pada gambar 4.36 menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\textit{throughput} = \textit{Jumlah data yang dikirim} / \textit{Waktu pengiriman data}$$

Nilai variabel yang didapatkan:

$$\textit{Throughput simulasi} = 0.329 \text{ Mbps} = 329 \text{ Kbps}$$

$$\textit{Jumlah data yang dikirim} = 185831 \text{ Byte}$$

$$\textit{Waktu pengiriman data} = 4.530 \text{ sec}$$

Sehingga diperoleh *throughput* analisa sebagai berikut:

$$\begin{aligned} \textit{throughput} &= \textit{Jumlah data yang dikirim} / \textit{Waktu pengiriman data} \\ &= 185831 / 4.520 = 41110.605 \text{ Bytes/sec} = 0.329 \text{ Mbit/sec} \\ &= 329 \text{ Kbps.} \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned} \% \textit{ error} &= \frac{(\textit{throughput simulasi} - \textit{throughput analisis})}{\textit{throughput simulasi}} \times 100\% \\ &= ((329 - 329) / 329) \times 100\% \\ &= 0\% \end{aligned}$$

Display					
Display filter:	none				
Ignored packets:	0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	321	321	100.000%	0	0.000%
Between first and last packet	3.631 sec				
Avg. packets/sec	88.397				
Avg. packet size	849.564 bytes				
Bytes	272710	272710	100.000%	0	0.000%
Avg. bytes/sec	75098.836				
Avg. MBit/sec	0.601				

Gambar 4.37. *Througput video 2.*

Dalam menghitung *throughput* berdasarkan hasil pengujian pada gambar 4.37 menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\textit{throughput} = \textit{Jumlah data yang dikirim} / \textit{Waktu pengiriman data}$$

Nilai variabel yang didapatkan:

$$\textit{Throughput simulasi} = 0.601 \text{ Mbps} = 601 \text{ Kbps}$$

$$\textit{Jumlah data yang dikirim} = 272710 \text{ Byte}$$

$$\textit{Waktu pengiriman data} = 3.631 \text{ sec}$$

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned} \textit{throughput} &= \textit{Jumlah data yang dikirim} / \textit{Waktu pengiriman data} \\ &= 272710 / 3.631 = 75098.839 \text{ Bytes/sec} = 0.601 \text{ Mbit/sec} \\ &= 601 \text{ Kbps.} \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned} \% \textit{ error} &= \frac{(\textit{throughput simulasi} - \textit{throughput analisis})}{\textit{throughput simulasi}} \times 100\% \\ &= \frac{(601 - 601)}{601} \times 100\% \\ &= 0\% \end{aligned}$$

Display					
Display filter:					
Ignored packets:	none 0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	526	526	100.000%	0	0.000%
Between first and last packet	5.304 sec				
Avg. packets/sec	99.164				
Avg. packet size	895.447 bytes				
Bytes	471005	471005	100.000%	0	0.000%
Avg. bytes/sec	88796.189				
Avg. MBit/sec	0.710				

Gambar 4.38. *Throughput video 3.*

Untuk perhitungan analisis matematis pada *throughput video 3* yang ditunjukkan dalam gambar 4.38 diatas diperoleh parameter sebagai berikut:

Throughput simulasi = 0.710 Mbps = 710 Kbps

Jumlah data yang dikirim = 471005 Byte.

Waktu pengiriman data = 5.304 sec.

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned}
 \textit{throughput} &= \textit{Jumlah data yang dikirim} / \textit{Waktu pengiriman data} \\
 &= 471005 / 5.304 = 88796.189 \textit{ Bytes/sec} = 0.710 \textit{ Mbit/sec} \\
 &= 710 \textit{ Kbps.}
 \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned}
 \% \textit{ error} &= \frac{(\textit{throughput simulasi} - \textit{throughput analisis})}{\textit{throughput simulasi}} \times 100\% \\
 &= ((710 - 710) / 710) \times 100\% \\
 &= 0 \%
 \end{aligned}$$

Untuk hasil analisis perbandingan nilai *bitrate* dan hasil perhitungan matematis dari parameter yang didapatkan pada saat simulasi dengan menggunakan media transmisi berupa kabel cat 5 dapat ditunjukkan pada tabel 4.6.

Tabel 4.6. Perbandingan *bitrate* dengan *throughput bitrate* pada Kabel UTP CAT 5.

Video	Bitrate (Kbps)	Throughput Analisis (Kbps)	Throughput Simulasi (Kbps)	%Error
Video 1	717	329	329	0
Video 2	1547	601	601	0
Video 3	1649	710	710	0

6. Pengujian hasil pengukuran *throughput* menggunakan kabel UTP Cat 5e dan Akses Point (WiFi).

Pada pengujian ini media komunikasi dari *server* ke akses *point* menggunakan kabel cat 5e dan dari akses *point* menuju perangkat *mobile* (*smartphone*) dengan menggunakan media *wireless* (WiFi). Berikut akan dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi *server* untuk mendapatkan nilai *throughput*.

Aplikasi pendukung yang dipergunakan untuk menganalisis *throughput* menggunakan kabel UTP cat 5e ialah *Wireshark* dengan cara klik tabs *statistic* lalu klik *summary*, sehingga didapatkan hasil yang ditunjukkan dalam gambar 4.39.

Display					
Display filter:	none				
Ignored packets:	0 (0.000%)				
Traffic	◀ Captured	◀ Displayed	◀ Displayed %	◀ Marked	◀ Marked %
Packets	274	274	100.000%	0	0.000%
Between first and last packet	4.030 sec				
Avg. packets/sec	67.994				
Avg. packet size	694.836 bytes				
Bytes	190385	190385	100.000%	0	0.000%
Avg. bytes/sec	47244.868				
Avg. MBit/sec	0.378				

Gambar 4.39. *Throughput* video 1.

Tabel 4.6. Perbandingan hasil dengan wawancara diuraikan dalam Tabel 4.6

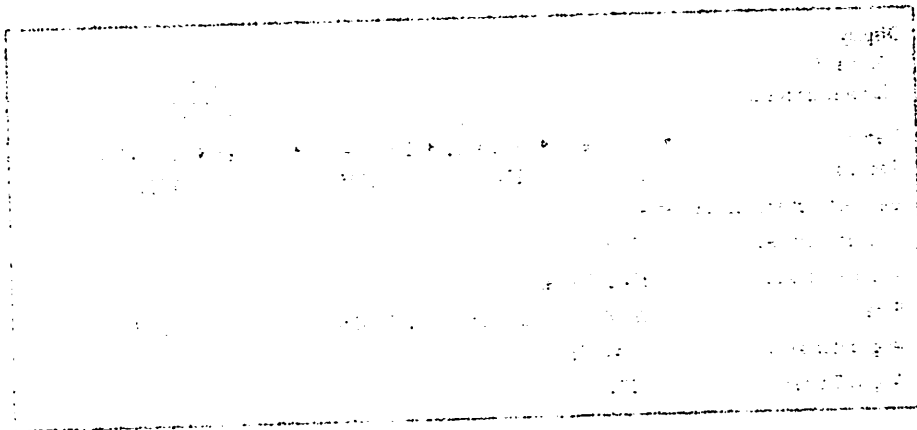
TABLE

Video	Wawancara	Wawancara	Wawancara
Video 1	77	75	75
Video 2	124	106	101
Video 3	104	110	110

6. Bagaimana hasil pengajaran Wawancara menggunakan Tabel 4.6

Car dan Alan (2017)

Hasil pengujian ini media komunikasi yang secara ke arah lain  
 menggunakan tabel car dan alan akan menjadi bagian dari media  
 (car alan) dengan menggunakan media tersebut (Wawancara) akan  
 digunakan hasil tulisan dan beberapa gambar Wawancara yang  
 digunakan dan diambil dari car dan alan untuk membuat nilai Wawancara  
 Aplikasi pembelajaran yang digunakan untuk meningkatkan wawasan  
 menggunakan tabel 4.6 car dan alan Wawancara dengan car dan alan Wawancara  
 dan klik wawancara sehingga digunakan hasil yang digunakan dalam Wawancara



Gambar 4.7. Wawancara (Car dan Alan)

Untuk perhitungan analisis matematis pada *throughput video 1* yang ditunjukkan pada gambar 4.39 diatas diperoleh parameter sebagai berikut:

Throughput simulasi = 0.378 Mbps = 378 Kbps

Jumlah data yang dikirim = 190385 Byte.

Waktu pengiriman data = 4.030 sec.

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned} \text{throughput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\ &= 190385 / 4.030 = 47244.868 \text{ Bytes/sec} = 0.378 \text{ Mbit/sec} \\ &= 378 \text{ Kbps.} \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned} \% \text{ error} &= \frac{(\text{throughput simulasi} - \text{throughput analisis})}{\text{throughput simulasi}} \times 100\% \\ &= \frac{((378 - 378) / 378) \times 100\%}{378} \\ &= 0\% \end{aligned}$$

Display					
Display filter:	none				
Ignored packets:	0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	392	392	100.000%	0	0.000%
Between first and last packet	5.525 sec				
Avg. packets/sec	70.955				
Avg. packet size	768.638 bytes				
Bytes	301306	301306	100.000%	0	0.000%
Avg. bytes/sec	54538.912				
Avg. MBit/sec	0.436				

Gambar 4.40. *Throughput video 2.*

Untuk perhitungan analisis matematis pada *throughput video 2* yang ditunjukkan pada gambar 4.40 diatas diperoleh parameter sebagai berikut:

Throughput simulasi = 0.436 Mbps = 436 Kbps

Jumlah data yang dikirim = 301306 Byte.

Waktu pengiriman data = 5.525 sec.

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned}
 \text{Throughput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\
 &= 301306 / 5.525 = 54538.912 \text{ Bytes/sec} = 0.436 \text{ Mbit/sec} \\
 &= 436 \text{ Kbps.}
 \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned}
 \% \text{ error} &= \frac{(\text{throughput simulasi} - \text{throughput analisis})}{\text{throughput simulasi}} \times 100\% \\
 &= ((436 - 436) / 436) \times 100\% \\
 &= 0 \%
 \end{aligned}$$

Display						
Display filter:						
Ignored packets:						none
						0 (0.000%)
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	322	322	100.000%	0	0.000%	
Between first and last packet 3.186 sec:						
Avg. packets/sec	101.072					
Avg. packet size	895.165 bytes					
Bytes	288243	288243	100.000%	0	0.000%	
Avg. bytes/sec	90476.264					
Avg. MBit/sec	0.724					

Gambar 4.41. *Throughput video 3.*

Untuk perhitungan analisis matematis pada *throughput video 3* dalam gambar 4.41 diatas diperoleh parameter sebagai berikut:

Throughput simulasi = 0.724 Mbps = 724 Kbps

Jumlah data yang dikirim = 288243 Byte.

Waktu pengiriman data = 3.186 sec.

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned}
 \text{throughput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\
 &= 288243 / 3.186 = 90476.264 \text{ Bytes/sec} = 0.724 \text{ Mbit/sec} \\
 &= 724 \text{ Kbps.}
 \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned} \% \text{ error} &= \frac{(\text{throughput simulasi} - \text{throughput analisis})}{\text{throughput simulasi}} \times 100\% \\ &= ((724 - 724) / 724) \times 100\% \\ &= 0\% \end{aligned}$$

Untuk hasil analisis perbandingan nilai *bitrate* dan hasil perhitungan matematis dari parameter yang didapatkan pada saat simulasi dengan menggunakan media transmisi berupa kabel cat 5e dapat ditunjukkan pada tabel 4.7.

Tabel 4.7. Perbandingan *bitrate* dengan *throughput bitrate* pada Kabel UTP CAT 5e.

Video	Bitrate (Kbps)	Througput (Kbps)	% Error
Video 1	717	378	0
Video 2	1547	436	0
Video 3	1649	724	0

#### 7. Pengujian hasil pengukuran *throughput* menggunakan kabel UTP Cat 6 dan Akses Point (WiFi).

Pada pengujian ini media komunikasi dari *server* ke akses *point* menggunakan kabel cat 6 dan dari akses *point* menuju perangkat *mobile* (*smartphone*) dengan menggunakan media *wireless* (WiFi). Berikut akan dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi *server* untuk mendapatkan nilai *throughput*.

Aplikasi pendukung yang dipergunakan untuk menganalisis *throughput* menggunakan kabel UTP cat 6 ialah *Wireshark* dengan cara klik *tabs statistic* lalu klik *summary*, sehingga didapatkan hasil sebagai berikut:

այս առաջին թիվը կապակցված է ընդհանուր թվի հետևի բաժնի հետ։  
 Գրանցված առանձին թիվը 1111 թիվը կապակցված է ընդհանուր թվի հետ։  
 Երկրորդ բաժնի հետ կապակցված է ընդհանուր թվի հետևի բաժնի հետ։  
 Գրանցված թիվը կապակցված է ընդհանուր թվի հետևի բաժնի հետ։  
 (Համարակառուցող) գրանցված թիվը կապակցված է ընդհանուր թվի հետևի բաժնի հետ։  
 Երկրորդ բաժնի հետ կապակցված է ընդհանուր թվի հետևի բաժնի հետ։  
 Որոշում է կայացնել հետևյալ կերպով։

1. Գրանցված թիվը կապակցված է ընդհանուր թվի հետևի բաժնի հետ։

ԱՊՐԵՆ	1010	111	11
ԱՊՐԵՆ	1111	111	11
ԱՊՐԵՆ	111	111	11
ԱՊՐԵՆ	1111111111	1111111111	1111111111

ՇՆՆ.26

Երկրորդ բաժնի հետ կապակցված է ընդհանուր թվի հետևի բաժնի հետ։

Երկրորդ բաժնի հետ կապակցված է ընդհանուր թվի հետևի բաժնի հետ։  
 Երկրորդ բաժնի հետ կապակցված է ընդհանուր թվի հետևի բաժնի հետ։  
 Երկրորդ բաժնի հետ կապակցված է ընդհանուր թվի հետևի բաժնի հետ։

$$\begin{aligned}
 &= 1111 \\
 &= (1111 - 1111) \times 1000
 \end{aligned}$$

$$\text{Երկրորդ բաժնի հետ կապակցված է ընդհանուր թվի հետևի բաժնի հետ։} \times 1000$$

Երկրորդ բաժնի հետ կապակցված է ընդհանուր թվի հետևի բաժնի հետ։

Display filter:		Ignored packets:				none
		Captured	Displayed	Displayed %	Marked	Marked %
Traffic		11833	11833	100.000%	0	0.000%
Packets						
Between first and last packet		151.172 sec				
Avg. packets/sec		78.275				
Avg. packet size		951.049 bytes				
Bytes		11253762	11253762	100.000%	0	0.000%
Avg. bytes/sec		74443.485				
Avg. MBit/sec		0.596				

Gambar 4.42. *Througput video 1.*

Untuk perhitungan analisis matematis pada *througput video 1* yang ditunjukkan pada gambar 4.42 diatas diperoleh parameter sebagai berikut:

Throughput simulasi = 0.596 Mbps = 596 Kbps

Jumlah data yang dikirim = 11253762 Byte.

Waktu pengiriman data = 151.172 sec.

Sehingga didapatkan througput sebagai berikut:

$$\begin{aligned}
 \text{througput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\
 &= 11253762 / 151.172 = 74443.485 \text{ Bytes/sec} = 0.596 \text{ Mbit/sec} \\
 &= 596 \text{ Kbps.}
 \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned}
 \% \text{ error} &= \frac{(\text{througput simulasi} - \text{througput analisis})}{\text{througput simulasi}} \times 100\% \\
 &= ((596 - 596) / 596) \times 100\% \\
 &= 0 \%
 \end{aligned}$$

Display					
Display filter:					
Ignored packets: none 0 (0.000%)					
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	108806	108806	100.000%	0	0.000%
Between first and last packet	730.668 sec				
Avg. packets/sec	148.913				
Avg. packet size	1016.667 bytes				
Bytes	110619502	110619502	100.000%	0	0.000%
Avg. bytes/sec	151395.014				
Avg. MBit/sec	1.211				

Gambar 4.43. *Througput video 2.*

Untuk perhitungan analisis matematis pada *througput video 2* yang ditunjukkan pada gambar 4.43 diatas diperoleh parameter sebagai berikut:

Throughput simulasi = 1.211 Mbps = 1211 Kbps

Jumlah data yang dikirim = 110619502 Byte.

Waktu pengiriman data = 730.66 sec.

Sehingga didapatkan *througput* sebagai berikut:

$$\begin{aligned}
 \text{througput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\
 &= 110619502 / 730.66 = 151395.01 \text{ Bytes/sec} = 1.211 \text{ Mbit/sec} \\
 &= 1211 \text{ Kbps.}
 \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned}
 \% \text{ error} &= \frac{(\text{througput simulasi} - \text{througput analisis})}{\text{througput simulasi}} \times 100\% \\
 &= ((1211 - 1211) / 1211) \times 100\% \\
 &= 0 \%
 \end{aligned}$$

$$= 0 \text{ руб}$$

$$= (1511 - 1511) \times 1511 \times 100\%$$

$$\text{№ 51101} = \frac{\text{Проектная стоимость} - \text{Проектная стоимость}}{\text{Проектная стоимость}} \times 100\%$$

Итого от реализации при продаже готовой продукции:

$$= 1511 \text{ руб}$$

$$= 1511 \times 100\% = 151100\% \text{ продаж} = 1511 \text{ руб}$$

Проектная = Цена при продаже готовой продукции

Зеркало от реализации проектной готовой продукции:

$$\text{Матрица проектная цена} = 15000 \text{ руб}$$

$$\text{Проектная стоимость} = 15000 \times 100\%$$

$$\text{Цена при продаже} = 1511 \text{ руб} = 1511 \text{ руб}$$

Цена при продаже готовой продукции

Цена при продаже готовой продукции

Цена при продаже готовой продукции

№	Наименование	Единица измерения	Количество	Цена	Сумма
1	Матрица проектная	шт	1	15000	15000
2	Проектная стоимость	%	100	15000	15000
3	Цена при продаже	руб	1	1511	1511
4	Итого				1511

Display					
Display filter:					
Ignored packets: none 0 (0.000%)					
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	57041	57041	100.000%	0	0.000%
Between first and last packet	325.270 sec				
Avg. packets/sec	175.365				
Avg. packet size	1038.771 bytes				
Bytes	59252555	59252555	100.000%	0	0.000%
Avg. bytes/sec	182163.937				
Avg. MBit/sec	1.457				

Gambar 4.44. *Throughput video 3.*

Untuk perhitungan analisis matematis pada *throughput video 3* yang ditunjukkan pada gambar 4.44 diperoleh parameter sebagai berikut:

Throughput simulasi = 1.457 Mbps = 1457 Kbps

Jumlah data yang dikirim = 59252555 Byte.

Waktu pengiriman data = 325.270 sec.

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned}
 \text{throughput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\
 &= 59252555 / 325.27 = 182163.94 \text{ Bytes/sec} = 1.457 \text{ Mbit/sec} \\
 &= 1457 \text{ Kbps.}
 \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned}
 \% \text{ error} &= \frac{(\text{throughput simulasi} - \text{throughput analisis})}{\text{throughput simulasi}} \times 100\% \\
 &= ((1457 - 1457) / 1457) \times 100\% \\
 &= 0 \%
 \end{aligned}$$

Untuk hasil analisis perbandingan nilai *bitrate* dan hasil perhitungan matematis dari parameter yang didapatkan pada saat simulasi dengan menggunakan media transmisi berupa kabel cat 6 dapat ditunjukkan pada tabel 4.8.

Year	Production (kg)	Price (Rp)	Revenue (Rp)
1987	1427	1427	2037229
1988	1827	1827	3337229
1989	1827	1827	3337229
1990	1827	1827	3337229
1991	1827	1827	3337229
1992	1827	1827	3337229
1993	1827	1827	3337229
1994	1827	1827	3337229
1995	1827	1827	3337229
1996	1827	1827	3337229
1997	1827	1827	3337229
1998	1827	1827	3337229
1999	1827	1827	3337229
2000	1827	1827	3337229
2001	1827	1827	3337229
2002	1827	1827	3337229
2003	1827	1827	3337229
2004	1827	1827	3337229
2005	1827	1827	3337229
2006	1827	1827	3337229
2007	1827	1827	3337229
2008	1827	1827	3337229
2009	1827	1827	3337229
2010	1827	1827	3337229
2011	1827	1827	3337229
2012	1827	1827	3337229
2013	1827	1827	3337229
2014	1827	1827	3337229
2015	1827	1827	3337229
2016	1827	1827	3337229
2017	1827	1827	3337229
2018	1827	1827	3337229
2019	1827	1827	3337229
2020	1827	1827	3337229
2021	1827	1827	3337229
2022	1827	1827	3337229

Table 1.1: Production and Revenue Data

Untuk perhitungan analisis regresi linier digunakan rumus sebagai berikut:

$$T = \frac{\sum Y}{n} = \frac{1427}{1} = 1427$$

$$T^2 = 1427^2 = 2037229$$

$$W = \frac{\sum XY}{n} = \frac{2037229}{1} = 2037229$$

$$W^2 = 2037229^2 = 4150298229$$

$$T = \frac{\sum Y}{n} = \frac{1427}{1} = 1427$$

$$T^2 = 1427^2 = 2037229$$

$$W = \frac{\sum XY}{n} = \frac{2037229}{1} = 2037229$$

$$W^2 = 2037229^2 = 4150298229$$

Maka didapatkan nilai error sebagai berikut:

$$e = \frac{\sum Y^2 - \frac{(\sum Y)^2}{n}}{n-1} = \frac{2037229 - \frac{1427^2}{1}}{1-1} = 0$$

Untuk hasil analisis regresi linier dapat digunakan rumus sebagai berikut:

$$a = \frac{\sum Y - T \cdot \sum X}{n - \sum X} = \frac{1427 - 1427 \cdot 1}{1 - 1} = 0$$

Tabel 4.8. Perbandingan *bitrate* dengan *throughput bitrate* pada Kabel UTP CAT 6.

Video	Bitrate (Kbps)	Througput Analisis (Kbps)	Throughput Simulasi (Kbps)	% Error
Video 1	717	596	596	0
Video 2	1547	1211	1211	0
Video 3	1649	1457	1457	0

8. Pengujian hasil pengukuran *throughput* menggunakan WiFi (*Ad hoc*) *peer to peer*.

Pada pegujian dengan menggunakan WiFi, *server* menggunakan laptop sedangkan dari sisi *client* menggunakan perangkat *mobile* (*smartphone*) yang sudah terinstall VLC, komunikasi kedua perangkat ini menggunakan *peer to peer ad hoc* dimana kedua perangkat tersambung langsung dengan menggunakan media nirkabel (WiFi). Berikut tampilan *Wireshark* yang diambil dari sisi *server* untuk mendapatkan hasil *throughput*.

Aplikasi pendukung yang dipergunakan untuk menganalisis *throughput* menggunakan WiFi ialah *Wireshark* dengan cara klik tabs *statistic* lalu klik *summary*, sehingga didapatkan hasil sebagai berikut:

Display					
Display filter:	none				
Ignored packets:	0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	79	79	100.000%	0	0.000%
Between first and last packet	1.601 sec				
Avg. packets/sec	49.332				
Avg. packet size	860.975 bytes				
Bytes	68017	68017	100.000%	0	0.000%
Avg. bytes/sec	42473.855				
Avg. MBit/sec	0.340				

Gambar 4.45. *Throughput video 1*

Untuk perhitungan analisis matematis pada *throughput video 1* yang ditunjukkan pada gambar 4.45 diperoleh parameter sebagai berikut:

Throughput simulasi = 0.340 Mbps = 340 Kbps

Jumlah data yang dikirim = 68017 Byte.

Waktu pengiriman data = 1.601 sec.

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned} \text{throughput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\ &= 68017 / 1.601 = 42473.855 \text{ Bytes/sec} = 0.340 \text{ Mbit/sec} \\ &= 340 \text{ Kbps.} \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned} \% \text{ error} &= \frac{(\text{throughput simulasi} - \text{throughput analisis})}{\text{throughput simulasi}} \times 100\% \\ &= ((340 - 340) / 340) \times 100\% \\ &= 0\% \end{aligned}$$

Display					
Display filter:	none				
Ignored packets:	0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	394	394	100.000%	0	0.000%
Between first and last packet	4.809 sec				
Avg. packets/sec	81.924				
Avg. packet size	834.987 bytes				
Bytes	328985	328985	100.000%	0	0.000%
Avg. bytes/sec	68405.779				
Avg. MBit/sec	0.547				

Gambar 4.46. *Throughput video 2.*

Untuk perhitungan analisis matematis pada *throughput video 2* yang ditunjukkan pada gambar 4.46 diperoleh parameter sebagai berikut:

Throughput simulasi = 0.547 Mbps = 547 Kbps

Jumlah data yang dikirim = 328985 Byte.

Waktu pengiriman data = 4.809 sec.

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned}
 \text{throughput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\
 &= 328985 / 4.809 = 68405.779 \text{ Bytes/sec} = 0.547 \text{ Mbit/sec} \\
 &= 547 \text{ Kbps.}
 \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\begin{aligned}
 \% \text{ error} &= \frac{(\text{throughput simulasi} - \text{throughput analisis})}{\text{throughput simulasi}} \times 100\% \\
 &= ((547 - 547) / 547) \times 100\% \\
 &= 0 \%
 \end{aligned}$$

Display					
Display filter:	none				
Ignored packets:	0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	326	326	100.000%	0	0.000%
Between first and last packet	3.514 sec				
Avg. packets/sec	92.779				
Avg. packet size	895.914 bytes				
Bytes	292068	292068	100.000%	0	0.000%
Avg. bytes/sec	83122.397				
Avg. MBit/sec	0.665				

Gambar 4.47. *Throughput video 3.*

Untuk perhitungan analisis matematis pada *throughput video 3* yang ditunjukkan pada gambar 4.47 diperoleh parameter sebagai berikut:

$$\text{Throughput simulasi} = 0.665 \text{ Mbps} = 665 \text{ Kbps}$$

$$\text{Jumlah data yang dikirim} = 292068 \text{ Byte.}$$

$$\text{Waktu pengiriman data} = 3.514 \text{ sec.}$$

Sehingga didapatkan *throughput* sebagai berikut:

$$\begin{aligned}
 \text{Throughput} &= \text{Jumlah data yang dikirim} / \text{Waktu pengiriman data} \\
 &= 292068 / 3.514 = 83122.397 \text{ Bytes/sec} = 0.665 \text{ Mbit/sec} \\
 &= 665 \text{ Kbps.}
 \end{aligned}$$

Maka didapatkan nilai % error sebagai berikut :

$$\% \text{ error} = \frac{(\text{throughput simulasi} - \text{throughput analisis})}{\text{throughput simulasi}} \times 100\%$$

$$\begin{aligned} \text{Therapies} &= \text{jumlah data yang dikurasi} \times \text{rata-rata panjang waktu} \\ &= 12888 \times 0.002 = 0.0257776 \text{ hari} \\ &= 217 \text{ hari} \end{aligned}$$

Jika dibedakan nilai error sebagai berikut :

$$\begin{aligned} \text{error} &= \frac{(\text{Therapies simulasi} - \text{jumlah analisis})}{\text{Therapies simulasi}} \\ &= \frac{(217 - 243) \times 100\%}{217} \\ &= 0.12 \end{aligned}$$

No	Uraian	Waktu
1	Analisis data	0.002
2	Therapies simulasi	0.002
3	Therapies simulasi	0.002
4	Therapies simulasi	0.002
5	Therapies simulasi	0.002
6	Therapies simulasi	0.002
7	Therapies simulasi	0.002
8	Therapies simulasi	0.002
9	Therapies simulasi	0.002
10	Therapies simulasi	0.002
11	Therapies simulasi	0.002
12	Therapies simulasi	0.002
13	Therapies simulasi	0.002
14	Therapies simulasi	0.002
15	Therapies simulasi	0.002
16	Therapies simulasi	0.002
17	Therapies simulasi	0.002
18	Therapies simulasi	0.002
19	Therapies simulasi	0.002
20	Therapies simulasi	0.002
21	Therapies simulasi	0.002
22	Therapies simulasi	0.002
23	Therapies simulasi	0.002
24	Therapies simulasi	0.002
25	Therapies simulasi	0.002
26	Therapies simulasi	0.002
27	Therapies simulasi	0.002
28	Therapies simulasi	0.002
29	Therapies simulasi	0.002
30	Therapies simulasi	0.002
31	Therapies simulasi	0.002
32	Therapies simulasi	0.002
33	Therapies simulasi	0.002
34	Therapies simulasi	0.002
35	Therapies simulasi	0.002
36	Therapies simulasi	0.002
37	Therapies simulasi	0.002
38	Therapies simulasi	0.002
39	Therapies simulasi	0.002
40	Therapies simulasi	0.002
41	Therapies simulasi	0.002
42	Therapies simulasi	0.002
43	Therapies simulasi	0.002
44	Therapies simulasi	0.002
45	Therapies simulasi	0.002
46	Therapies simulasi	0.002
47	Therapies simulasi	0.002
48	Therapies simulasi	0.002
49	Therapies simulasi	0.002
50	Therapies simulasi	0.002

Tabel 4.17. Waktu simulasi

Untuk perhitungan analisis manufaktur ini menggunakan nilai yang ditunjukkan

pada gambar 4.17 di bawah ini sebagai berikut :

$$\text{Therapies simulasi} = 0.002 \times 12888 = 217 \text{ hari}$$

$$\text{jumlah data yang dikurasi} = 12888 \text{ hari}$$

$$\text{Waktu perhitungan data} = 217 \text{ hari}$$

sehingga didapatkan therapies sebagai berikut :

$$\begin{aligned} \text{Therapies} &= \text{jumlah data yang dikurasi} \times \text{rata-rata panjang waktu} \\ &= 12888 \times 0.002 = 0.0257776 \text{ hari} \\ &= 217 \text{ hari} \end{aligned}$$

Jika dibedakan nilai error sebagai berikut :

$$\text{error} = \frac{(\text{Therapies simulasi} - \text{jumlah analisis})}{\text{Therapies simulasi}} \times 100\%$$

$$= ((665 - 665) / 665) \times 100\%$$

$$= 0 \%$$

Untuk hasil analisis perbandingan nilai *bitrate* dan hasil perhitungan matematis dari parameter yang didapatkan pada saat simulasi dengan menggunakan media transmisi berupa WiFi dapat ditunjukkan pada tabel 4.8.

Berikut yaitu tabel 4.9 yang menunjukkan perbandingan *bitrate* dengan *throughput bitrate* menggunakan WiFi.

Tabel 4.9. Perbandingan *bitrate* dengan *throughput bitrate* menggunakan WiFi

Video	Bitrate (Kbps)	Throughput Analisis (Kbps)	Throughput Simulasi (Kbps)	% Error
Video 1	717	340	340	0
Video 2	1547	547	547	0
Video 3	1649	665	665	0

Kesimpulan hasil pengujian dari tabel 4.6, 4.7, 4.8 dan 4.9 ialah perbandingan *throughput* dan nilai *error* yang paling besar terdapat bila menggunakan media komunikasi *wireless peer to peer*, nilai *error* terbesar kedua ialah dengan menggunakan kombinasi cat 5 + akses poin, kemudian di ikuti kombinasi cat 5e + akses poin serta kombinasi cat 6 dan akses point. Dimana pada hasil pengujian diatas menempatkan penggunaan media transmisi kombinasi cat 6 + WiFi berada pada kualitas terbaik dibandingkan dengan menggunakan media transmisi yang lain.

#### 4.2.3 Pengujian Hasil Pengukuran *PacketLoss* dan *Jitter*

Paket *loss* dapat disebabkan oleh sejumlah faktor, mencakup penurunan signal dalam media jaringan, melebihi batas saturasi jaringan, paket yang *corrupt* yang menolak untuk transit, kesalahan *hardware* jaringan.

Dalam menghitung *packet loss* berdasarkan hasil pengujian menggunakan aplikasi *Wireshark* menggunakan persamaan sebagai berikut:

$$\text{Paket Loss} = \frac{\text{Paket Transmite} - \text{Paket Receive}}{\text{Paket Transmite}} \times 100\%$$

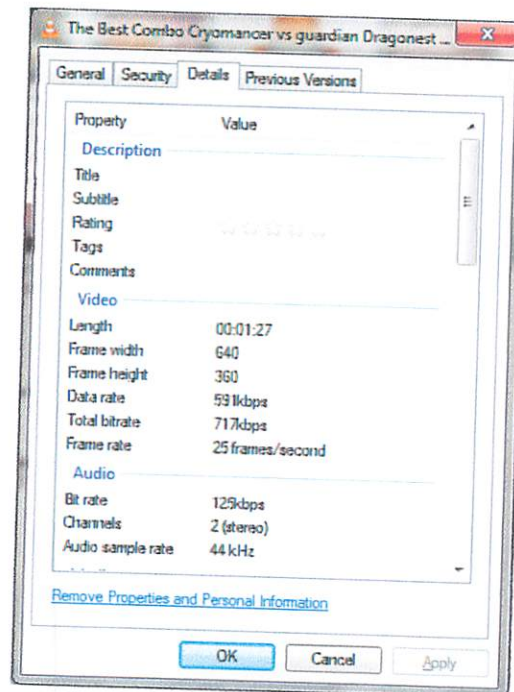
*Jitter* ialah perbedaan waktu kedatangan dari suatu paket ke penerima dengan waktu yang diharapkan. *Jitter* dapat menyebabkan *sampling* di sisi penerima menjadi tidak tepat sasaran, sehingga informasi menjadi rusak., *jitter* dapat dihitung dengan menggunakan persamaan seperti berikut:

$$\text{Jitter} = (\text{Total variasi delay}) / (\text{Total paket yang diterima} - 1)$$

Pada pengukuran *packetloss* dan *Jitter* kali ini akan digunakan *sampling* dari 3 file *video* dengan *data bitrate* sebagai berikut:

1. *Bitrate Video 1* : 717 Kbps.

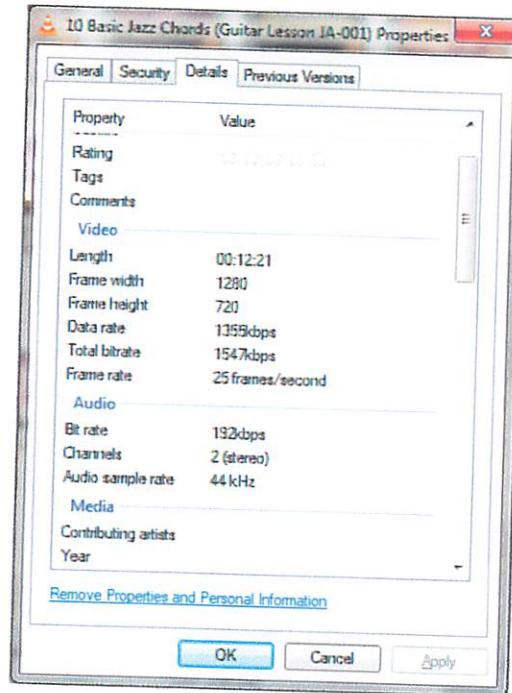
Berikut adalah *properties* pada pengujian *bitrate video* dengan kecepatan 717 Kbps ditunjukkan dalam gambar 4.48.



Gambar 4.48. video 1 properties.

2. *BitrateVideo 2*: 1547 Kbps.

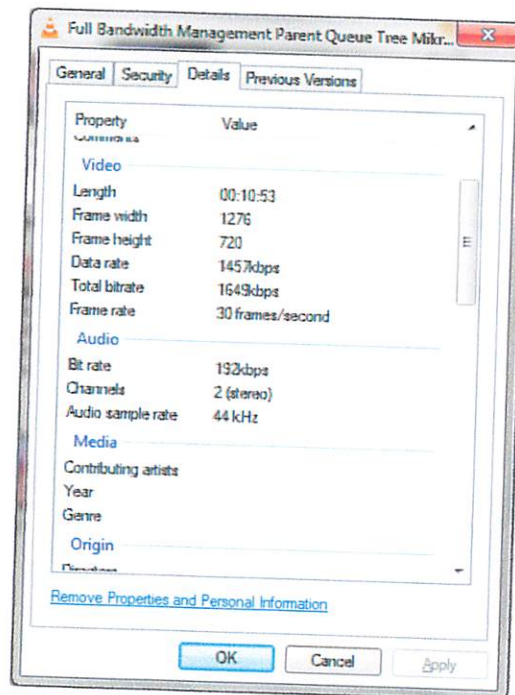
Berikut adalah *properties* pada pengujian *bitrate video* dengan kecepatan 1547 Kbps ditunjukkan dalam gambar 4.49.



Gambar 4.49. *video 2 properties*.

3. *BitrateVideo 3*: 1649 Kbps.

Berikut adalah *properties* pada pengujian *bitrate video* dengan kecepatan 1649 Kbps ditunjukkan dalam gambar 4.50.



Gambar 4.50. video 3 properties.

#### 9. Menggunakan kabel UTP Cat 5 dan akses point (WiFi)

Pada pengujian ini media komunikasi dari *server* ke akses *point* menggunakan kabel cat 5 dan dari akses *point* menuju perangkat *mobile* (*smartphone*) dengan menggunakan media *wireless* (WiFi). Berikut akan dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi *server* untuk mendapatkan nilai *packetloss* serta *jitter*.

Aplikasi pendukung yang dipergunakan untuk menganalisis *throughput* dan *jitter* menggunakan Kabel UTP Cat 5 ialah *Wireshark* dengan cara membuka *capture interface* lalu lakukan *decode* dari UDP ke RTP lalu klik tab *statistic* kemudian RTP, kemudian *Stream analysis*, sehingga didapatkan hasil sebagai berikut:

Packet	Sequence	Delta (ms)	Filtered Jitter (ms)	Skew (ms)	IP BW (kbps)	Marker	Status
1	16378	0.00	0.00	0.00	3.69	SET	[ Ok ]
2	16379	25.97	0.01	0.15	7.38	SET	[ Ok ]
3	16380	30.41	0.28	-4.14	11.06	SET	[ Ok ]
4	16381	21.65	0.54	0.33	14.75	SET	[ Ok ]
7	16382	28.88	0.68	-2.43	18.44	SET	[ Ok ]
8	16383	29.99	0.88	-6.30	22.13	SET	[ Ok ]
10	16384	20.03	1.20	-0.21	25.82	SET	[ Ok ]
12	16385	30.00	1.37	-4.08	29.50	SET	[ Ok ]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms, Mean jitter = 0.00 ms  
 Max skew = -94.25 ms  
 Total RTP packets = 1316 (expected 1316) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 34.35 s (1 ms clock drift, corresponding to 90002 Hz (+0.00%))

Gambar 4.51. Analisa *packetloss* dan *jitter* video 1.

Berdasarkan hasil *capture* pada gambar 4.51 didapatkan parameter sebagai berikut:

Max delta : 0 ms.

Packet transmite : 1316.

Packet receive : 1316.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmite} - \text{Paket Receive}}{\text{Paket Transmite}} \times 100\% \\
 &= \frac{1316 - 1316}{1316} = \frac{0 - 0}{1316} \times 100\% \\
 &= \frac{0}{1316} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (1316 - 1) \\
 &= 0 / 1315 \\
 &= 0 \text{ ms}
 \end{aligned}$$

Packet	Sequence	Delta (ms)	Filtered Jitter (ms)	Skew (ms)	IP BW (kbps)	Marker	Status
216	5353	0.00	0.00	0.00	3.69	SET	[Ok]
221	5354	25.85	0.02	0.27	7.38	SET	[Ok]
222	5355	27.30	0.09	-0.91	11.06	SET	[Ok]
224	5356	25.75	0.11	-0.54	14.75	SET	[Ok]
226	5357	25.13	0.16	0.45	18.44	SET	[Ok]
227	5358	26.87	0.20	-0.29	22.13	SET	[Ok]
233	5359	25.00	0.26	0.84	25.82	SET	[Ok]
234	5360	27.01	0.30	-0.05	29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -1316.75 ms.  
 Total RTP packets = 1652 (expected 1652) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 43.13 s (126 ms clock drift, corresponding to 90.262 Hz (+0.29%))

Gambar 4.52. Analisa *packetloss* dan *jitter* video 2.

Berdasarkan hasil *capture* pada gambar 4.52 didapatkan parameter sebagai berikut:

Max delta : 0 ms.

Packet transmute : 1652.

Packet receive : 1652.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmute} - \text{Paket Receive}}{\text{Paket Transmute}} \times 100\% \\
 &= \frac{1652 - 1652}{1652} = \frac{0 - 0}{1652} \times 100\% \\
 &= \frac{0}{1652} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned} \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)). \\ &= 0 / (1652 - 1). \\ &= 0 / 1651. \\ &= 0 \text{ ms.} \end{aligned}$$

The screenshot shows the 'Wireshark RTP Stream Analysis' window. It displays a table of RTP packets with columns for Packet, Sequence, Delta (ms), Filtered Jitter (ms), Skew (ms), IP BW (kbps), Marker, and Status. The status for all packets is '[Ok]'. Below the table, summary statistics are provided: Max delta = 0.00 ms, Max jitter = 0.00 ms, Mean jitter = 0.00 ms, Max skew = -1117.18 ms, Total RTP packets = 1848 (expected 1848), Lost RTP packets = 0 (0.00%), Sequence errors = 0, and Duration 48.24 s (94 ms clock drift, corresponding to 90176 Hz (+0.20%)).

Packet	Sequence	Delta (ms)	Filtered Jitter (ms)	Skew (ms)	IP BW (kbps)	Marker	Status
1	37	0.00	0.00	0.00	3.69	SET	[Ok]
2	38	20.74	0.34	5.38	7.38	SET	[Ok]
3	39	27.05	0.37	4.45	11.06	SET	[Ok]
4	40	27.75	0.45	2.82	14.75	SET	[Ok]
15	41	25.02	0.49	3.92	18.44	SET	[Ok]
20	42	25.30	0.51	4.74	22.13	SET	[Ok]
21	43	27.23	0.55	3.64	25.82	SET	[Ok]
22	44	25.50	0.55	4.28	29.50	SET	[Ok]

Gambar 4.53. Analisa *packet loss* dan *jitter* video 3.

Berdasarkan hasil *capture* pada gambar 4.53 didapatkan parameter sebagai berikut:

Max delta : 0 ms.  
 Packet transmute : 1848.  
 Packet receive : 1848.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned} \text{Paket Loss} &= \frac{\text{Paket Transmute} - \text{Paket Receive}}{\text{Paket Transmute}} \times 100\% \\ &= \frac{1848 - 1848}{1848} = \frac{0 - 0}{1848} \times 100\% \\ &= \frac{0}{1848} \times 100\% \end{aligned}$$

$$= 0\%$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned} \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\ &= 0 / (1848 - 1) \\ &= 0 / 1847 \\ &= 0 \text{ ms} \end{aligned}$$

Dari hasil pengujian dan analisa *Packetloss* dan *Jitter* dari *video* 1, 2 dan 3 dengan menggunakan kabel cat 5 dirangkum pada tabel 4.10.

Tabel 4.10. Hasil pengujian *Packetloss* dan *Jitter* menggunakan kabel CAT 5

Video	Bitrate (Kbps)	Packet loss	Jitter (ms)
Video 1	717	0	0
Video 2	1547	0	0
Video 3	1649	0	0

#### 10. Menggunakan kabel UTP Cat 5e dan akses point (WiFi)

Pada pengujian ini media komunikasi dari server ke akses *point* menggunakan kabel cat 5e dan dari akses *point* menuju perangkat *mobile* (*smartphone*) dengan menggunakan media *wireless* (WiFi). Berikut akan dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi server untuk mendapatkan nilai *jitter* dan *packetlossnya*.

Aplikasi pendukung yang dipergunakan untuk menganalisis *throughput* dan *jitter* menggunakan Kabel UTP Cat 5e ialah *wireshark* dengan cara membuka *capture interface* lalu lakukan decode dari UDP ke RTP lalu

Dalam menghitung luas beraturan dari segiempat, luas malar  
-Apakah Wawasan dapat digunakan persamaan sebagai berikut

$$\begin{aligned} \text{Jitter} &= (\text{variasi delay}) / \text{bit rate} \times 100 \\ &= 0.11833 \times 100 \\ &= 11.833\% \\ &= 11.83\% \end{aligned}$$

Dari hasil pengujian dan hasil analisis yang telah dilakukan pada tabel 4.10  
dapat disimpulkan bahwa hasil pengujian pada tabel 4.10

Tabel 4.10 Hasil pengujian Wawasan dan Wawasan menggunakan Kabel CAT 5

Video 1	Video 2	Video 3	Wawasan	Wawasan
11.833	11.833	11.833	11.833	11.833
11.833	11.833	11.833	11.833	11.833
11.833	11.833	11.833	11.833	11.833

10. Menggunakan Kabel FTP Cat 5e dan akses point (WIFI)

Pada pengujian ini media komunikasi yang akan digunakan  
menggunakan kabel cat 5e dan akses point yang digunakan adalah  
(wawasan) dengan menggunakan media akses WIFI. Hasil yang  
diperoleh dari pengujian Wawasan yang digunakan sebagai  
dijelaskan dan diuraikan dari hasil pengujian yang telah dilakukan  
berikut ini:

Hasil pengujian yang diperoleh dari pengujian ini  
dijelaskan dan diuraikan dari hasil pengujian yang telah dilakukan  
berikut ini:

klik *tab statistic* kemudian RTP, kemudian *Stream analysis*, sehingga didapatkan hasil sebagai berikut:

Analysing stream from 192.168.10.3 port 50451 to 192.168.10.2 port 54934 SSRC = 0x677A963E

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
34	30674	0.00	0.00	0.00	3.69	SET	[Ok]
35	30675	2.65	1.47	23.47	7.38	SET	[Ok]
36	30676	2.85	2.83	46.75	11.06	SET	[Ok]
37	30677	2.89	4.11	69.93	14.75	SET	[Ok]
38	30678	2.77	5.31	93.33	18.44	SET	[Ok]
39	30679	2.63	6.44	116.82	22.13	SET	[Ok]
40	30680	2.53	7.51	140.37	25.82	SET	[Ok]
41	30681	2.62	8.51	163.87	29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = 15314.91 ms.  
 Total RTP packets = 2028 (expected 2028) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 37.64 s (13663 ms clock drift, corresponding to 122678 Hz (-36.31%))

Gambar 4.54. Analisa *packet loss* dan *jitter* video 1.

Berdasarkan hasil *capture* pada gambar 4.54 didapatkan parameter sebagai berikut:

Max delta : 0 ms  
 Packet transmite : 2028  
 Packet receive : 2028

Dalam menghitung *packet loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmite} - \text{Paket Receive}}{\text{Paket Transmite}} \times 100\% \\
 &= \frac{2028 - 2028}{2028} = \frac{0 - 0}{2028} \times 100\% \\
 &= \frac{0}{2028} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (2028 - 1) \\
 &= 0 / 2028 \\
 &= 0 \text{ ms}
 \end{aligned}$$

Wireshark: RTP Stream Analysis

Forward Direction | Reversed Direction

Analyzing stream from 192.168.10.3 port 49373 to 192.168.10.2 port 59838 SSRC = 0x805A5C20

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
1	5269	0.00	0.00	0.00	3.69	SET	[Ok]
2	5270	27.81	0.11	-1.65	7.38	SET	[Ok]
3	5271	25.92	0.11	-1.49	11.06	SET	[Ok]
4	5272	25.15	0.16	-0.51	14.75	SET	[Ok]
5	5273	26.77	0.20	-1.17	18.44	SET	[Ok]
6	5274	25.21	0.24	-0.25	22.13	SET	[Ok]
13	5275	26.76	0.26	-0.88	25.82	SET	[Ok]
14	5276	25.60	0.28	-0.37	29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -8.86 ms.  
 Total RTP packets = 1479 (expected 1479) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 33.61 s (-0 ms clock drift, corresponding to 89999 Hz (-0.00%))

Buttons: Save payload..., Save as CSV..., Refresh, Jump to, Graph, Player, Next non-Ok, Close

Gambar 4.55. Analisa *packet loss* dan *jitter* video 2.

Berdasarkan hasil capture pada gambar 4.55 didapatkan parameter sebagai berikut:

Packet transmite : 1479.

Packet receive : 1479.

Dalam menghitung *packet loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmite} - \text{Paket Receive}}{\text{Paket Transmite}} \times 100\% \\
 &= \frac{1479 - 1479}{1479} = \frac{0 - 0}{1479} \times 100\% \\
 &= \frac{0}{1479} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (1479 - 1) \\
 &= 0 / 1478 \\
 &= 0 \text{ ms}
 \end{aligned}$$

Wireshark RTP Stream Analysis

Analysing stream from 192.168.10.3 port 51857 to 192.168.10.2 port 55862 SSRC = 0x8809027C

Packet	Sequence	Delta(ms)	Filtered Jitter(rms)	Skew(rms)	IP BW(kbps)	Marker	Status
1	42150	0.00	0.00	0.00	3.69	SET	[ Ok ]
2	42151	29.25	0.20	-3.13	7.38	SET	[ Ok ]
3	42152	21.80	0.45	1.19	11.06	SET	[ Ok ]
4	42153	28.11	0.55	-0.80	14.75	SET	[ Ok ]
5	42154	25.85	0.53	-0.52	18.44	SET	[ Ok ]
8	42155	24.01	0.63	1.59	22.13	SET	[ Ok ]
9	42156	26.92	0.64	0.79	25.82	SET	[ Ok ]
10	42157	28.11	0.73	-1.20	29.50	SET	[ Ok ]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 rms. Mean jitter = 0.00 rms.  
 Max skew = -880.44 ms.  
 Total RTP packets = 1620 (expected 1620) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 42.29 s (-31 ms clock drift, corresponding to 89934 Hz (-0.07%))

Buttons: Save payload..., Save as CSV..., Refresh, Jump to, Graph, Player, Next non-Ok, Close

Gambar 4.56. Analisa *packet loss* dan *jitter* video 3.

Berdasarkan hasil capture pada gambar 4.56 didapatkan parameter sebagai berikut:

Max delta : 0 ms.

Packet transmute : 1620.

Packet receive : 1620.

Dalam menghitung *paket loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmute} - \text{Paket Receive}}{\text{Paket Transmute}} \times 100\% \\
 &= \frac{1620 - 1620}{1620} = \frac{0 - 0}{1620} \times 100\% \\
 &= \frac{0}{1620} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (1620 - 1) \\
 &= 0 / 1619 \\
 &= 0 \text{ ms}
 \end{aligned}$$

Dari hasil pengujian dan analisa *Packet loss* dan *Jitter* dari video 1, 2 dan 3 dengan menggunakan kabel cat 5e dirangkum pada tabel 4.11.

Tabel 4.11. Hasil pengujian *Packetloss* dan *Jitter* menggunakan kabel CAT 5e.

Video	Bitrate (Kbps)	Packet loss	Jitter
Video 1	717	0	0
Video 2	1547	0	0
Video 3	1649	0	0

#### 11. Menggunakan kabel UTP Cat 6 dan akses point (WiFi)

Pada pengujian ini media komunikasi dari server ke akses *point* menggunakan kabel cat 5e dan dari akses point menuju perangkat *mobile* (*smartphone*) dengan menggunakan media *wireless* (WiFi). Berikut akan dijelaskan hasil analisa dari beberapa *capture* gambar *Wireshark* yang dijalankan dan diambil dari sisi server untuk mendapatkan nilai *jitter* dan *packetloss*.

Aplikasi pendukung yang dipergunakan untuk menganalisis *throughput* dan *jitter* menggunakan Kabel UTP Cat 6 ialah *Wireshark* dengan cara membuka *capture interface* lalu lakukan *decode* dari UDP ke RTP lalu klik *tab statistic* kemudian RTP, kemudian *Stream analysis*, sehingga didapatkan hasil sebagai berikut:

Dalam penelitian ini, berfokus pada penelitian dengan metode kuantitatif dengan menggunakan instrumen sebagai berikut:

$$\begin{aligned} \text{Kuesioner} &= (\text{Variabel dalam Society receiver} - \text{V}) \\ &= 0,7 \times 1000 - 1 \\ &= 0,7 \times 1000 \\ &= 0,700 \end{aligned}$$

Dari hasil penelitian dan analisis kuesioner yang telah dilakukan pada tabel 11.1 dan 11.2 dengan menggunakan tabel chi-square sebagai berikut:

Tabel 11.1 Hasil penelitian kuesioner dan analisis menggunakan tabel chi-square

Video	Video 1	Video 2	Video 3
Video 1	10	15	10
Video 2	10	15	10
Video 3	10	15	10

11.1. Penggunaan Kabel UTP Cat 5 dan akses point (WIFI)

Pada penelitian ini media komunikasi dari server ke akses point menggunakan kabel cat 5 dan dari akses point media perantara video (komputer) dengan menggunakan media server (WIFI). Hal ini akan dijelaskan hasil analisis dari beberapa video yang sudah direkam yang diambil dari server untuk mendapatkan nilai video dan kuesioner.

Aplikasi pendukung yang digunakan untuk menganalisis hasilnya yaitu menggunakan tabel chi-square dengan cara membaca video yang telah dilakukan oleh chi-square dan kuesioner yang sudah dilakukan dengan cara membaca kuesioner yang sudah dilakukan sebagai berikut:

Wireshark: RTP Stream Analysis

Forward Direction | Reversed Direction

Analyzing stream from 192.168.10.3 port 54314 to 192.168.10.2 port 50184 SSRC = 0xE3AA3F2D

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
1	8737	0.00	0.00	0.00	3.69	SET	[Ok]
2	8738	23.13	0.19	2.99	7.38	SET	[Ok]
12	8739	25.00	0.25	4.11	11.06	SET	[Ok]
13	8740	26.87	0.28	3.36	14.75	SET	[Ok]
23	8741	28.16	0.39	1.33	18.44	SET	[Ok]
24	8742	23.83	0.51	3.63	22.13	SET	[Ok]
34	8743	30.38	0.74	-0.64	25.82	SET	[Ok]
35	8744	21.64	0.97	3.85	29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -25.86 ms.  
 Total RTP packets = 1207 (expected 1207) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 31.51 s (-1 ms clock drift, corresponding to 89998 Hz (-0.00%))

Buttons: Save payload..., Save as CSV..., Refresh, Jump to, Graph, Player, Next non-Ok, Close

Gambar 4.57. Analisa packetloss dan jitter video 1.

Berdasarkan hasil capture pada gambar 4.57 didapatkan parameter sebagai berikut:

Max delta : 0 ms.

Packet transmite : 1207.

Packet receive : 1207.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmite} - \text{Paket Receive}}{\text{Paket Transmite}} \times 100\% \\
 &= \frac{1207 - 1207}{1207} = \frac{0-0}{1207} \times 100\% \\
 &= \frac{0}{1207} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (1207 - 1) \\
 &= 0 / 1206 \\
 &= 0 \text{ ms}
 \end{aligned}$$

Wireshark: RTP Stream Analysis

Forward Direction | Reversed Direction

Analysing stream from 192.168.10.3 port 54314 to 192.168.10.2 port 50184 SSRC = 0xE3AA3F2D

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
1	8737	0.00	0.00	0.00	3.69	SET	[Ok]
2	8738	23.13	0.19	2.99	7.38	SET	[Ok]
12	8739	25.00	0.25	4.11	11.06	SET	[Ok]
13	8740	26.87	0.28	3.36	14.75	SET	[Ok]
23	8741	28.16	0.39	1.33	18.44	SET	[Ok]
24	8742	23.83	0.51	3.63	22.13	SET	[Ok]
34	8743	30.38	0.74	-0.64	25.82	SET	[Ok]
35	8744	21.64	0.97	3.85	29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -26.86 ms.  
 Total RTP packets = 1207 (expected 1207) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 31.51 s (-1 ms clock drift, corresponding to 89998 Hz (-0.00%))

Buttons: Save payload..., Save as CSV..., Refresh, Jump to, Graph, Player, Next non-Ok, Close

Gambar 4.58. Analisa *packetloss* dan *jitter* video 2.

Berdasarkan hasil *capture* pada gambar 4.58 didapatkan parameter sebagai berikut:

Max delta : 0 ms.

Packet transmite : 1207.

Packet receive : 1207.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmite} - \text{Paket Receive}}{\text{Paket Transmite}} \times 100\% \\
 &= \frac{1207 - 1207}{1207} = \frac{0 - 0}{1207} \times 100\% \\
 &= \frac{0}{1652} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (1652 - 1) \\
 &= 0 / 1651
 \end{aligned}$$

= 0 ms

Wireshark RTP Stream Analysis

Analysing stream from 192.168.10.3 port 53200 to 192.168.10.2 port 50192 SSRC = 0xEC88883D

Packet	Sequence	Delta(ms)	Filtered jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
2	64600	0.00	0.00	0.00	3.69	SET	[Ok]
3	64601	23.97	0.13	2.16	7.38	SET	[Ok]
5	64602	24.91	0.20	3.37	11.06	SET	[Ok]
7	64603	25.87	0.21	3.62	14.75	SET	[Ok]
8	64604	26.11	0.19	3.63	18.44	SET	[Ok]
9	64605	25.21	0.24	4.54	22.13	SET	[Ok]
10	64606	27.03	0.28	3.58	25.82	SET	[Ok]
12	64607	26.71	0.30	3.00	29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -1.273.19 ms.  
 Total RTP packets = 1259 (expected 1259) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration: 32.86 s (210 ms clock drift, corresponding to 90574 Hz (+0.64%))

Buttons: Save payload..., Save as CSV..., Refresh, Jump to, Graph, Player, Next non-Ok, Close

Gambar 4.60. Analisa *packet loss* dan *jitter* video 3.

Berdasarkan hasil *capture* pada gambar 4.60 didapatkan parameter sebagai berikut:

Max delta : 0 ms.

Packet transmute : 1259.

Packet receive : 1259.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmute} - \text{Paket Receive}}{\text{Paket Transmute}} \times 100\% \\
 &= \frac{1259 - 1259}{1259} = \frac{0 - 0}{1259} \times 100\% \\
 &= \frac{0}{1259} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned} \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\ &= 0 / (1259 - 1) \\ &= 0 / 1258 \\ &= 0 \text{ ms} \end{aligned}$$

Dari hasil pengujian dan analisa *Packetloss* dan *Jitter* dari video 1, 2 dan 3 dengan menggunakan kabel cat 6 dirangkum pada tabel 4.12.

Tabel 4.12. Hasil pengujian *Packet loss* dan *Jitter* menggunakan kabel CAT 6.

Video	Bitrate (Kbps)	Packet loss	Jitter (ms)
Video 1	717	0	0
Video 2	1547	0	0
Video 3	1649	0	0

## 12. Menggunakan kabel UTP WiFi (*Ad Hoc*) *peer to peer*

Pada pegujian dengan menggunakan WiFi, *server* menggunakan laptop sedangkan dari sisi *client* menggunakan perangkat *mobile (smartphone)* yang sudah terinstall VLC, komunikasi kedua perangkat ini menggunakan *peer to peer ad hoc* dimana kedua perangkat tersambung langsung dengan menggunakan media nirkabel (WiFi). Berikut tampilan *Wireshark* yang diambil dari sisi server untuk mendapatkan hasil nilai *jitter* dan *packetloss*.

Aplikasi pendukung yang dipergunakan untuk menganalisis *throughput* dan *jitter* menggunakan media transmisi WiFi ialah *Wireshark* dengan cara membuka *capture interface* lalu lakukan *decode* dari UDP ke RTP lalu klik tab *statistic* kemudian RTP, kemudian *Stream analysis*, sehingga didapatkan hasil sebagai berikut:

Ukuran matriksnya. Untuk perhitungan hasil pengujian, proses ini akan dilakukan dengan menggunakan program sebagai berikut:

$$\begin{aligned}
 \text{Filter} &= (\text{Vektor delay}) \cdot \text{Bentuk receive} - 1 \\
 &= 0 \cdot (1 \cdot 2 \cdot 3 \cdot 4 \cdot 5) \\
 &= 0 \cdot 120 \\
 &= 0 \text{ ms}
 \end{aligned}$$

Dari hasil pengujian dan analisis Waveform dan Video 1, 2 dan 3 dengan menggunakan tabel dan grafik pada tabel 4.12.

Tabel 4.12. Hasil pengujian Akras dan VCR menggunakan tabel CAT 6.

Video (Frame)	Frame (ms)	Frame (ms)	Frame (ms)
Video 1	0	0	0
Video 2	0	0	0
Video 3	0	0	0

12. Menggunakan tabel 4.12 dan hasil pengujian...

Hasil pengujian dengan menggunakan WCR secara menggunakan tabel dan grafik dan ini akan menggunakan bentuk grafik (waveform) yang sudah tertera di CD. Untuk hasil pengujian ini menggunakan proses yang akan diuraikan berikut ini dengan menggunakan diagram menggunakan media digital (WIFI) berikut tampilan Waveform yang diambil dari server untuk mendapatkan hasil nilai video dan wave. Analisis perbandingan yang digunakan untuk membandingkan wave dan video menggunakan media digital WIFI yang Waveform dengan cara membaca cuplikan wave dan frame dari UDP ke RTP. Untuk nilai statistik kegunaan RTP kegunaan Waveform sehingga didapatkan hasil sebagai berikut:

Wireshark: RTP Stream Analysis

Forward Direction | Reversed Direction

Analysing stream from 192.168.10.3 port 55100 to 192.168.10.2 port 55874 SSRC = 0x24185249

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
1	27697	0.00	0.00	0.00			
2	27698	25.94	0.05	-0.81	3.69	SET	[Ok]
3	27699	27.00	0.10	-1.69	7.38	SET	[Ok]
4	27700	24.99	0.17	-0.57	11.06	SET	[Ok]
5	27701	30.11	0.41	-4.56	14.75	SET	[Ok]
6	27702	21.89	0.64	-0.32	18.44	SET	[Ok]
7	27703	27.02	0.65	-1.22	22.13	SET	[Ok]
8	27704	32.18	1.00	-7.28	25.82	SET	[Ok]
					29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -2237.35 ms.  
 Total RTP packets = 8577 (expected 8577) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 224.03 s (-159 ms clock drift, corresponding to 89936 Hz (-0.07%))

Buttons: Save payload..., Save as CSV..., Refresh, Jump to, Graph, Player, Next non-Ok, Close

Gambar 4.60. Analisa *packetloss* dan *jitter* video 1.

Berdasarkan hasil capture pada gambar 4.60 didapatkan parameter sebagai berikut:

Max delta : 0 ms.  
 Packet transmite : 8577.  
 Packet receive : 8577.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmite} - \text{Paket Receive}}{\text{Paket Transmite}} \times 100\% \\
 &= \frac{8577 - 8577}{8577} = \frac{0 - 0}{8577} \times 100\% \\
 &= \frac{0}{8577} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (8577 - 1) \\
 &= 0 / 8576
 \end{aligned}$$

= 0 ms

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
3	23396	0.00	0.00	0.00	3.69	SET	[Ok]
4	23397	26.37	0.02	-0.25	7.38	SET	[Ok]
6	23398	23.96	0.15	1.91	11.06	SET	[Ok]
7	23399	28.02	0.26	0.02	14.75	SET	[Ok]
8	23400	24.99	0.31	1.15	18.44	SET	[Ok]
9	23401	26.03	0.30	1.24	22.13	SET	[Ok]
10	23402	26.07	0.28	1.29	25.82	SET	[Ok]
16	23403	32.90	0.69	-5.49	29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -61.31 ms.  
 Total RTP packets = 1234 (expected 1234) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 32.21 s (0 ms clock drift, corresponding to 90001 Hz (+0.00%))

Gambar 4.61. Analisa *packetloss* dan *jitter* video 2.

Berdasarkan hasil *capture* pada gambar 4.61 didapatkan parameter sebagai berikut:

Max delta : 0 ms.

Packet transmute : 1234.

Packet receive : 1234.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmute} - \text{Paket Receive}}{\text{Paket Transmute}} \times 100\% \\
 &= \frac{1234 - 1234}{1234} = \frac{0 - 0}{1234} \times 100\% \\
 &= \frac{0}{1234} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (1234 - 1) \\
 &= 0 / 1233 \\
 &= 0 \text{ ms}
 \end{aligned}$$

Wireshark RTP Stream Analysis

Forward Direction | Reversed Direction

Analyzing stream from 192.168.10.3 port 65043 to 192.168.10.2 port 54290 SSRC = 0x790221CF

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
1	5699	0.00	0.00	0.00	3.69	SET	[Ok]
2	5700	27.03	0.06	-0.90	7.38	SET	[Ok]
3	5701	27.07	0.11	-1.85	11.06	SET	[Ok]
13	5702	24.92	0.18	-0.65	14.75	SET	[Ok]
14	5703	23.01	0.29	-2.54	18.44	SET	[Ok]
15	5704	25.43	0.31	-1.84	22.13	SET	[Ok]
16	5705	24.58	0.39	-0.30	25.82	SET	[Ok]
17	5706	30.01	0.61	-4.18	29.50	SET	[Ok]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -3071.16 ms.  
 Total RTP packets = 865 (expected 865) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 22.57 s (887 ms clock drift, corresponding to 93538 Hz (+3.93%))

Buttons: Save payload..., Save as CSV..., Refresh, Jump to, Graph, Player, Next non-Ok, Close

Gambar 4.62. Analisa *packetloss* dan *jitter* video 3.

Berdasarkan hasil *capture* pada gambar 4.62 didapatkan parameter sebagai berikut:

Max delta : 0 ms.  
 Packet transmute : 865.  
 Packet receive : 865.

Dalam menghitung paket *loss* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Paket Loss} &= \frac{\text{Paket Transmute} - \text{Paket Receive}}{\text{Paket Transmute}} \times 100\% \\
 &= \frac{865 - 865}{865} = \frac{0 - 0}{865} \times 100\% \\
 &= \frac{0}{865} \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dalam menghitung *jitter* berdasarkan hasil pengujian diatas menurut *software Wireshark* dapat digunakan persamaan sebagai berikut:

$$\begin{aligned}
 \text{Jitter} &= (\text{Variasi delay} / (\text{Packet receive} - 1)) \\
 &= 0 / (865 - 1) \\
 &= 0 / 864 \\
 &= 0 \text{ ms}
 \end{aligned}$$

Dari hasil pengujian dan analisa *Packetloss* dan *Jitter* dari *video 1, 2 dan 3* dengan menggunakan WiFi dirangkum pada tabel 4.13.

Tabel 4.13. Hasil pengujian *Packetloss* dan *Jitter* menggunakan WiFi

Video	Bitrate (Kbps)	Packet loss	Jitter
Video 1	717	0	0
Video 2	1547	0	0
Video 3	1649	0	0

Kesimpulan dari tabel 4.10, 4.11, 4.12, dan 4.13 ialah nilai *packet loss* dan *jitter* pada masing-masing pengujian video ialah 0, begitu pula pengujian pada masing-masing media transmisi mulai dari cat 5, cat 5e, cat 6 dan WiFi. Sehingga bisa dikatakan kualitas jaringan yang digunakan untuk *streaming* video sangatlah baik.

## BAB V PENUTUP

### 5.1 Kesimpulan

Dari hasil pengujian, pengambilan data dan analisa yang dilakukan, maka dapat diambil beberapa kesimpulan :

1. Perbandingan *throughput* dan nilai error yang paling besar terdapat bila menggunakan media komunikasi *wireless peer to peer*, nilai error terbesar kedua ialah dengan menggunakan kombinasi cat 5 + akses poin, kemudian di ikuti kombinasi cat 5e + akses poin serta kombinasi cat 6 dan akses point. Dimana pada hasil pengujian diatas menempatkan penggunaan media transmisi kombinasi cat 6 + WiFi berada pada kualitas terbaik dibandingkan dengan menggunakan media transmisi yang lain.
2. Nilai *packet loss* dan *jitter* pada masing-masing pengujian video ialah 0, begitu pula pengujian pada masing-masing media transmisi mulai dari cat 5, cat 5e, cat 6 dan WiFi. Sehingga bisa dikatakan kualitas jaringan yang digunakan untuk *streaming* video sangatlah baik. Hasil dari percobaan ini menyatakan *jitter* masih memenuhi standar untuk *Quality of Service*. Hal ini terjadi karena adanya *buffer* yang digunakan oleh aplikasi *real-time* video.
3. Untuk hasil pengukuran *delay* transmisi mulai dari menggunakan kabel cat 5, cat 5e, cat 6 serta WiFi *ad hoc* didapatkan hasil yang sangat memuaskan. Dimana menurut Standarisasi ITU-T *Delay*, pengujian *delay* transmisi yang dilakukan masuk dalam kategori excellent, karena nilai yang didapatkan kurang dari 150 ms.

## **5.2 Saran-saran**

Berikut ini adalah saran yang diberikan untuk pengembangan selanjutnya.

- 1 Untuk pengembangan selanjutnya dapat ditambahkan beberapa jenis format video yang lain seperti MKV, MPEG, AVI, 3GP dsb.
- 2 Pengujian aplikasi video streaming ini tidak hanya menggunakan aplikasi player VLC tetapi juga menggunakan aplikasi player yang lain. Sehingga bisa didapatkan hasil pengujian dan analisa yang lebih beragam.
- 3 Untuk mendapatkan performa yang optimal pada sistem ini, maka spesifikasi dari server yang digunakan akan sangat mempengaruhi. Semakin tinggi spesifikasi server yang digunakan, maka kinerja yang dihasilkan juga semakin optimal.

**DAFTAR PUSTAKA**

- [1]. Dharma Oetomo, Budi Sutedjo. *Konsep dan Perancangan Jaringan Komputer*. Yogyakarta ; Andi offset. 2003.
- [2]. Kristanto Andri. *Jaringan Komputer*. Yogyakarta : Graha Ilmu. 2003.
- [3]. Schaum's. *Computer Networking*. ERLANGGA. 2004.
- [4]. Drew, Heywood. *Konsep dan Penerapan TCP/IP*. Andi Yogyakarta. 2008.
- [5]. <http://yudidjohan.wordpress.com/2010/10/17/konfigurasi-vlc-sebagai-server-streaming-embeded-on-website/> , diakses tanggal 12 juli 2014
- [6]. <http://staff.unud.ac.id/~linawati/wp-content/uploads/2011/07/proses-video-streaming-dengan-protocol-rtsp.pdf>
- [7]. <http://wiki.wireshark.org/>, diakses tanggal 15 juli 2014
- [8]. Kurniawan, Agus. *Panduan analisis dan investigasi paket data jaringan menggunakan wireshark*, Yogyakarta; Andi offset 2010.



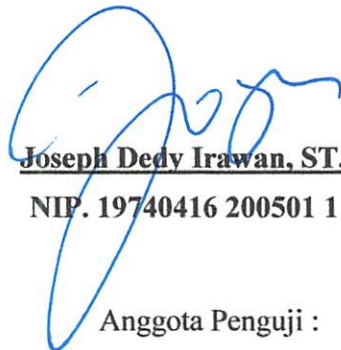


BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Balqis Fitri Anna  
NIM : 09.18.086  
Jurusan : Teknik Informatika S-1  
Judul : **Analisa Performansi Real Time Streaming Protokol (RTSP)  
untuk Aplikasi Video Streaming pada Jaringan Mobile IPv4 dengan  
Menggunakan Video LAN Client (VLC)**  
Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :  
Hari : Selasa  
Tanggal : 18 Februari 2014  
Nilai : A

Panitia Ujian Skripsi :

**Ketua Majelis Penguji**

  
**Joseph Dedy Irawan, ST, MT**  
NIP. 19740416 200501 1 002

Anggota Penguji :

**Penguji I**



**Sandy Nataly M, S.Kom**  
NIP.Y. 1030800418

**Penguji II**



**Suryo Adi Wibowo, ST, MT**  
NIP.P. 1031000438



## FORMULIR PERBAIKAN SKRIPSI

Nama : Balqis Fitri Anna  
NIM : 09.18.086  
Jurusan : Teknik Informatika S-1  
Judul : **Analisa Performansi Real Time Streaming Protokol (RTSP)  
untuk Aplikasi Video Streaming pada Jaringan Mobile IPv4 dengan  
Menggunakan Video LAN Client (VLC)**

No	Penguji	Perbaikan	Paraf
1	Penguji I	1. Tambahkan analisa menggunakan wifi dan peer to peer 2. Perbaiki tulisan istilah asing dengan cetak miring 3. Perbaiki penulisan daftar pustaka	
2	Penguji II	1. Hitungan analisis Jitter, delay, packet loss, throughput 2. Tampilkan hasil pengujian simulasi cat 5, cat 5e, cat 6 dan wifi 3. Rumus matematis pada parameter yang diuji	

Anggota Penguji :

**Penguji I**

**Sandy Nataly M, S.Kom**

NIP. P. 1030800418

**Penguji II**

**Suryo Adi Wibowo, ST, MT.**

NIP.P. 1031000438

Mengetahui :

**Dosen Pembimbing I**

**Ir. Yudi Limpraptono, MT.**

NIP.Y. 1039500274

**Dosen Pembimbing II**

**Michael Ardita, ST, MT.**

NIP.P. 1031000434








## FORMULIR BIMBINGAN SKRIPSI

Nama : Balqis Fitri Anna

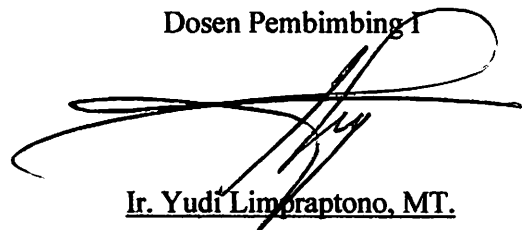
NIM : 09.18.086

Judul : Analisa Performansi Real Time Streaming Protokol (RTSP) untuk Aplikasi Video Streaming pada Jaringan Mobile IPv4 dengan Menggunakan Video LAN Client (VLC)

No.	Tanggal	Uraian	Paraf Pembimbing
1	3 / 2013 Sept	Proposal	
2	13 / 2013 oktober	Bab I , Bab II	
3	29 / 2014 Januari	Bab III , Bab IV , Bab V	
4	25 / 2014 Januari	Makalah Seminar hasil	
5.	18 / 2014 Feb	Laporan 4 / Kompre	

Malang, Januari 2014

Dosen Pembimbing I



Ir. Yudi Limpraptono, MT.

NIP.Y. 1039500274

## FORMULIR BIMBINGAN SKRIPSI

Nama : Balqis Fitri Anna

NIM : 09.18.086

Judul : Analisa Performansi Real Time Streaming Protokol (RTSP) untuk Aplikasi Video Streaming pada Jaringan Mobile IPv4 dengan Menggunakan Video LAN Client (VLC)

No.	Tanggal	Uraian	Paraf Pembimbing
1.	2 / 2013 sept	Proposal	Ar
2.	21 / 2013 Okt	Definisi parameter performansi	Ar
3.	1 / 2013 Nov	Konsep Delay, Jitter, Throughput	Ar
4.	13 / 2014 Jan	Konsep pengujian	Ar
5.	16 / 2014 Jan	Analisa hasil Wireshark	Ar
6.	26 / 2014 Jan	Makalah seminar	Ar
7.	17 / 2014 Feb	Laporan u/ kompre	Ar

Malang, Januari 2014

Dosen Pembimbing II



Michael Ardita, ST, MT.

NIP.P. 1031000434

Malang, 21 Oktober 2013

Lampiran : 1(Satu) berkas

Perihal : Ketersediaan sebagai Pembimbing Skripsi

Kepada : Yth. Bpk/Ibu Ir. Yudi Limpraptono, MT  
Dosen Pembina Prodi Teknik Informatika S-1  
Institut Teknologi Nasional  
MALANG

Yang bertanda tangan dibawah ini:

Nama : BALQIS FITRIANA  
Nim : 0918086  
Prodi : Teknik Informatika S-1

Dengan ini mengajukan permohonan, kiranya bapak bersedia menjadi Dosen Pembimbing Utama / ~~Pendamping~~ \*), untuk penyusunan Skripsi dengan judul (Proposal Terlampir) :

**Analisa Performansi Real Time Streaming Protokol ( RTSP ) Untuk  
Aplikasi Video Streaming Pada Jaringan Mobile IPv4 Dengan  
Menggunakan Video Lan Client ( VLC )**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik. Demikian permohonan kami dan atas kesediaan bapak kami sampaikan terima kasih.

Prodi T. Informatika S-1  
Ketua,



Joseph Dedy Irawan, ST., MT.  
NIP : 197404162005021002

Hormat Kami,

BALQIS FITRIANA

Form S-3a

**PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI**

Sesuai permohonan dari mahasiswa/i :

Nama : BALQIS FITRIANA

Nim : 0918086

Program Studi : Teknik Informatika

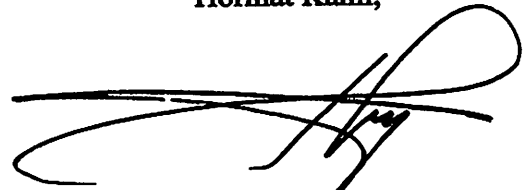
Dengan ini menyatakan bersedia / tidak bersedia \*) membimbing skripsi dari mahasiswa tersebut dengan judul :

**Analisa Performansi Real Time Streaming Protokol ( RTSP ) Untuk  
Aplikasi Video Streaming Pada Jaringan Mobile IPv4 Dengan  
Menggunakan Video Lan Client ( VLC )**

Demikian Surat Pernyataan ini kami buat agar dipergunakan seperlunya.

Malang, 22 - 10 - 2013

Hormat Kami,



Ir. Yudi Limpraptono, MT  
NIP.Y.1039500274

**Catatan :**

Setelah disetujui agar formulir ini diserahkan mahasiswa/i yg bersangkutan kepada Jurusan untuk diproses lebih lanjut

\*) coret yang tidak perlu

Form S-3b



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 21 Oktober 2013

Nomor : ITN-70/INF/TA/2013

Lampiran : ---

Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Ir. Yudi Limpraptono, MT  
Dosen Pembina Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : BALQIS FITRIANA  
Nim : 0918086  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**21 OKTOBER 2013 S/D 21 MARET 2014**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua,



Joseph Dedy Prawan, ST., MT.

NIP : 197404162005021002

Form S-4a

Malang, 21 Oktober 2013

Lampiran : 1(Satu) berkas  
Perihal : Ketersediaan sebagai Pembimbing Skripsi

Kepada : Yth. Bpk/Ibu Michael Ardita, ST.MT  
Dosen Pembina Prodi Teknik Informatika S-1  
Institut Teknologi Nasional  
MALANG

Yang bertanda tangan dibawah ini:

Nama : BALQIS FITRIANA  
Nim : 0918086  
Prodi : Teknik Informatika S-1

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing Utama / Pendamping \*), untuk penyusunan Skripsi dengan judul (Proposal Terlampir) :

**Analisa Performansi Real Time Streaming Protokol ( RTSP ) Untuk  
Aplikasi Video Streaming Pada Jaringan Mobile IPv4 Dengan  
Menggunakan Video Lan Client ( VLC )**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik. Demikian permohonan kami dan atas kesediaan bapak kami sampaikan terima kasih.



Prodi T. Informatika S-1  
Ketua,  
Joseph Dedy Irawan, ST., MT.  
NIP : 197404162005021002

Hormat Kami,

BALQIS FITRIANA

Form S-3a

**PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI**

Sesuai permohonan dari mahasiswa/i :

Nama : BALQIS FITRIANA

Nim : 0918086

Program Studi : Teknik Informatika

Dengan ini menyatakan bersedia / tidak bersedia \*) membimbing skripsi dari mahasiswa tersebut dengan judul :

**Analisa Performansi Real Time Streaming Protokol ( RTSP ) Untuk Aplikasi Video Streaming Pada Jaringan Mobile IPv4 Dengan Menggunakan Video Lan Client ( VLC )**

Demikian Surat Pernyataan ini kami buat agar dipergunakan seperlunya.

Malang, 22-10-2013

Hormat Kami,



Michael Ardita, ST.MT  
NIP.P.1031000434

**Catatan :**

Setelah disetujui agar formulir ini diserahkan mahasiswa/i yg bersangkutan kepada Jurusan untuk diproses lebih lanjut  
\*) coret yang tidak perlu

Form S-3b



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 21 Oktober 2013

Nomor : ITN-70/INF/TA/2013  
Lampiran : ---  
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Michael Ardita, ST.MT  
Dosen Pembina Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : BALQIS FITRIANA  
Nim : 0918086  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**21 OKTOBER 2013 S/D 21 MARET 2014**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua,

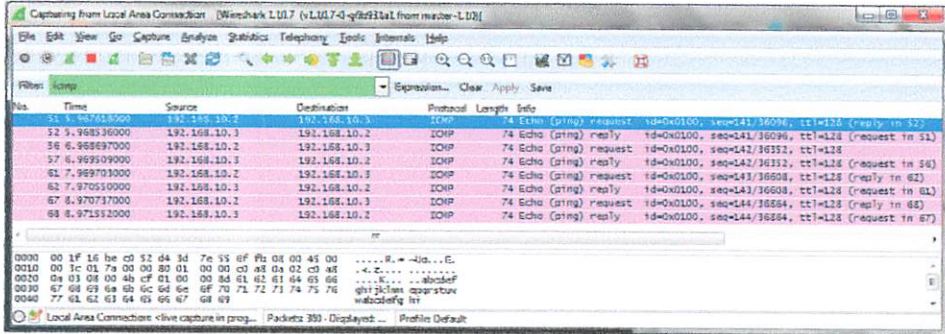


Joseph Dedy Irawan, ST., MT.  
NIP 7197404162005021002

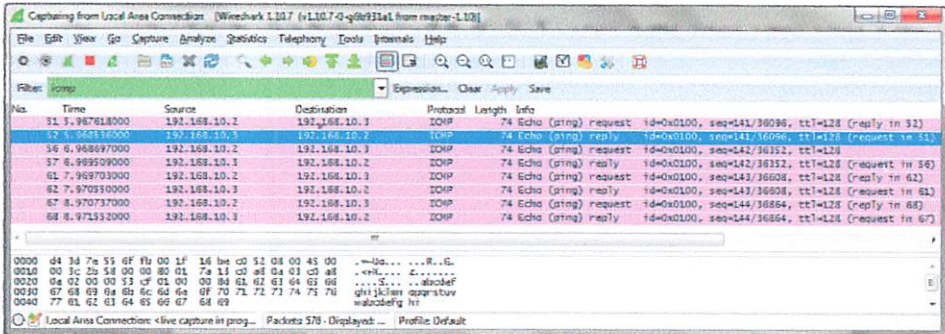
Form S-4a

# LAMPIRAN

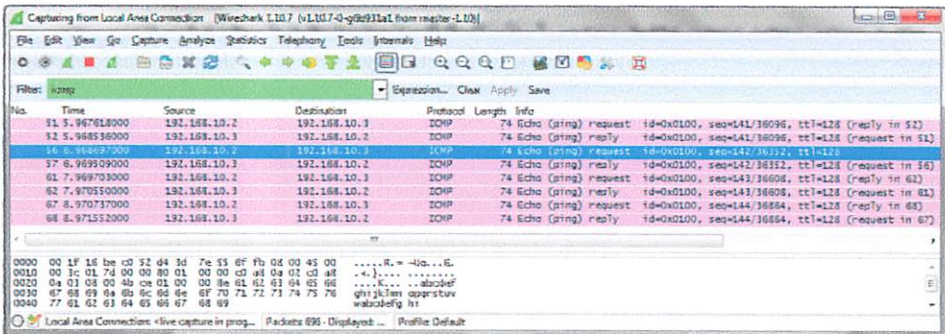
## 1. Wireshark Capture ICMP Request 1.



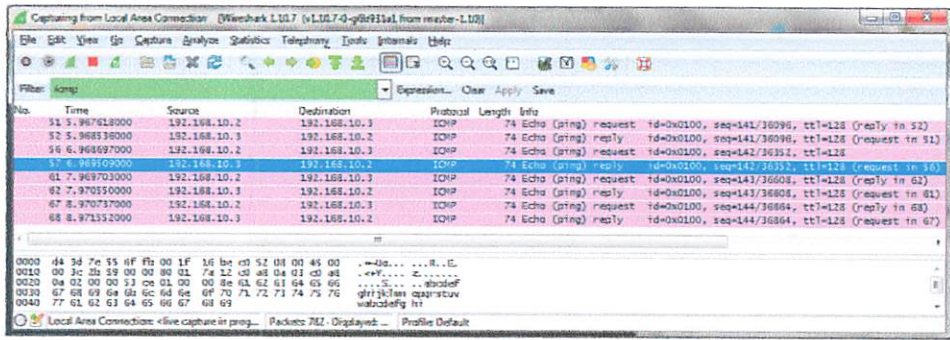
## 2. Wireshark Capture ICMP Reply 1.



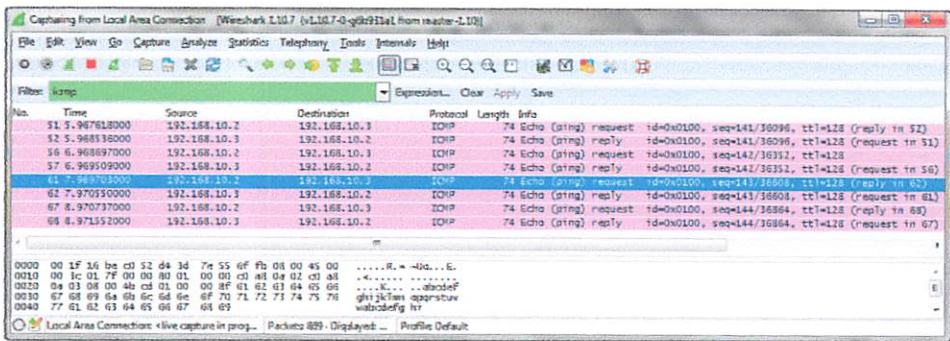
## 3. Wireshark Capture ICMP Request 2.



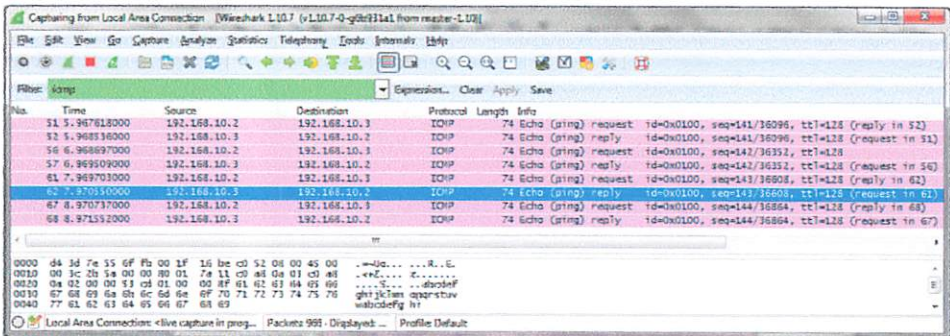
#### 4. Wireshark Capture ICMP Reply 2



#### 5. Wireshark Capture ICMP Request 3.



#### 6. Wireshark Capture ICMP Reply 3.



## 7. Wireshark Capture ICMP Request 4.

Wireshark capture showing ICMP Request 4. The packet list shows a request from 192.168.10.2 to 192.168.10.3 at time 57.969703000. The packet bytes pane shows the ICMP Echo (ping) request structure.

No.	Time	Source	Destination	Protocol	Length	Info
51	5.987818000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=141/36096, ttl=128 (reply in 52)
52	5.988360000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=141/36096, ttl=128 (request in 51)
56	6.988697000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=142/36112, ttl=128
57	6.989309000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=142/36112, ttl=128 (request in 56)
61	7.969703000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=143/36608, ttl=128 (reply in 62)
62	7.970500000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=143/36608, ttl=128 (request in 61)
67	8.970370000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=144/36864, ttl=128 (reply in 68)
68	8.971332000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=144/36864, ttl=128 (request in 67)

## 8. Wireshark Capture ICMP Reply 4.

Wireshark capture showing ICMP Reply 4. The packet list shows a reply from 192.168.10.3 to 192.168.10.2 at time 67.897037000. The packet bytes pane shows the ICMP Echo (ping) reply structure.

No.	Time	Source	Destination	Protocol	Length	Info
51	5.987818000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=141/36096, ttl=128 (reply in 52)
52	5.988360000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=141/36096, ttl=128 (request in 51)
56	6.988697000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=142/36112, ttl=128
57	6.989309000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=142/36112, ttl=128 (request in 56)
61	7.969703000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=143/36608, ttl=128 (reply in 62)
62	7.970500000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=143/36608, ttl=128 (request in 61)
67	8.970370000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=144/36864, ttl=128 (reply in 68)
68	8.971332000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=144/36864, ttl=128 (request in 67)

## 9. Wireshark Capture ICMP Request 1.

Wireshark capture showing ICMP Request 1. The packet list shows a request from 192.168.10.2 to 192.168.10.3 at time 17.3186747000. The packet bytes pane shows the ICMP Echo (ping) request structure.

No.	Time	Source	Destination	Protocol	Length	Info
17	3.186747000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=149/38144, ttl=128
18	3.187533000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=149/38144, ttl=128 (request in 17)
28	4.187384000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=150/38400, ttl=128 (reply in 29)
29	4.188192000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=150/38400, ttl=128 (request in 28)
38	5.188180000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=151/38656, ttl=128 (reply in 39)
39	5.189107000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=151/38656, ttl=128 (request in 38)
45	6.189193000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request id=0x0100, seq=152/38912, ttl=128
46	6.190080000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply id=0x0100, seq=152/38912, ttl=128 (request in 45)

## 10. Wireshark Capture ICMP Reply 1.

0000 44 3d 7e 55 ff 00 1f 16 bc c0 52 08 00 45 00 ... ..R...E.  
0010 00 3c 2e 67 00 00 80 01 77 04 c0 a8 0a 01 c0 a8 ... ..g...v.....  
0020 0a 01 08 00 00 51 c7 01 00 00 96 81 62 61 64 65 66 ... ..s... ..abcklef  
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghtjklm nopqrstuv  
0040 77 61 62 63 64 65 66 67 68 69 wxyzdefgh

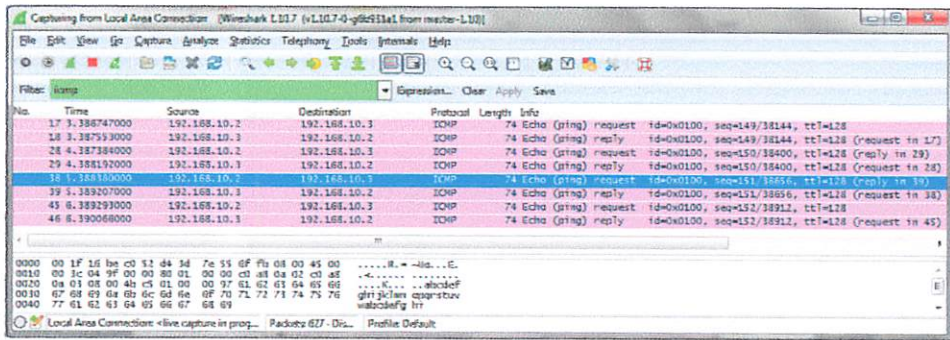
## 11. Wireshark Capture ICMP Request 2.

0000 00 1f 16 bc c0 52 04 3d 7e 55 ff 0b 08 00 45 00 ... ..R...-No...E.  
0010 00 3c 2e 67 00 00 80 01 00 00 c0 a8 0a 02 c0 a8 ... ..g... ..v.....  
0020 0a 01 08 00 00 4b c6 01 00 00 96 81 62 61 64 65 66 ... ..s... ..abcklef  
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghtjklm nopqrstuv  
0040 77 61 62 63 64 65 66 67 68 69 wxyzdefgh

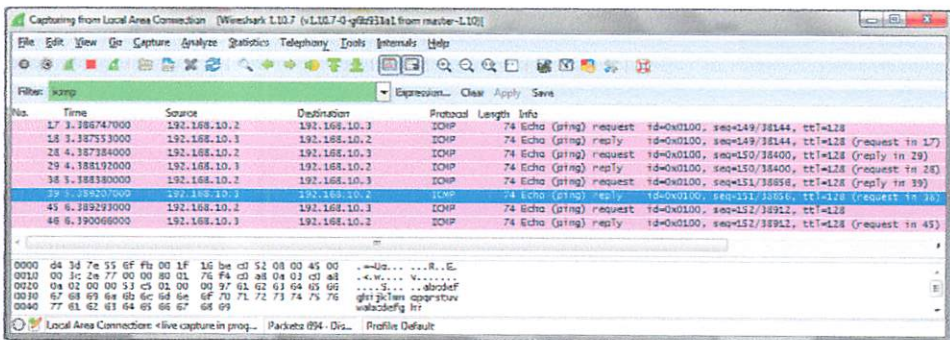
## 12. Wireshark Capture ICMP Reply 2

0000 44 3d 7e 55 ff 00 1f 16 bc c0 52 08 00 45 00 ... ..R...-No...E.  
0010 00 3c 2e 70 00 00 80 01 77 04 c0 a8 0a 03 c0 a8 ... ..g... ..v.....  
0020 0a 01 08 00 00 53 c8 01 00 00 96 81 62 61 64 65 66 ... ..s... ..abcklef  
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghtjklm nopqrstuv  
0040 77 61 62 63 64 65 66 67 68 69 wxyzdefgh

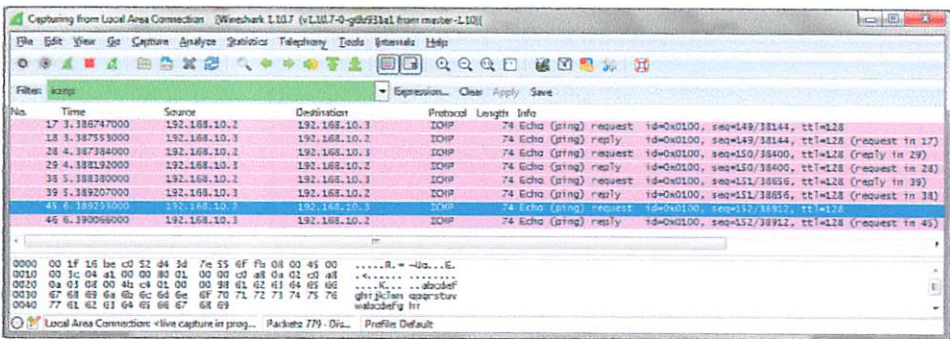
### 13. Wireshark Capture ICMP Request 3.



### 14. Wireshark Capture ICMP Reply 3.



### 15. Wireshark Capture ICMP Request 4.



## 16. Wireshark Capture ICMP Reply 4.

Local Area Connection: <live capture in prog... Packets: 848 - Dis... Profile: Default

No.	Time	Source	Destination	Protocol	Length	Info
17	3.38757000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=149/38144, ttl=128
18	3.38753000	192.168.10.3	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=149/38144, ttl=128 (request in 17)
19	4.387394000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=150/38400, ttl=128 (request in 19)
20	4.388192000	192.168.10.3	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=150/38400, ttl=128 (request in 19)
21	5.388180000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=151/38656, ttl=128 (request in 21)
22	5.389207000	192.168.10.3	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=151/38656, ttl=128 (request in 21)
23	6.389293000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) request id=0x0100, seq=152/38912, ttl=128 (request in 23)
24	6.390091000	192.168.10.3	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=152/38912, ttl=128 (request in 23)

```

0000 44 3d 7e 55 6f fb 00 1f 16 bc c0 52 08 00 45 00  ....R...R..E.
0010 00 3c 2e 7c 00 00 80 01 76 ef c0 a8 04 01 c0 a8  ..K...K.....
0020 0a 02 00 00 53 c4 11 00 00 98 61 62 63 64 65 66  ...K...abodef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmopqrstuv
0040 77 61 62 63 64 65 66 67 68 69                      wxyzdefghij
  
```

## 17. Wireshark Capture ICMP Request 1.

Local Area Connection: <live capture in prog... Packets: 1... Profile: Default

No.	Time	Source	Destination	Protocol	Length	Info
17	3.083133000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) request id=0x0100, seq=165/42240, ttl=128
18	4.083343000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=165/42240, ttl=128 (request in 17)
19	4.084222000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) request id=0x0100, seq=166/42496, ttl=128 (request in 19)
20	5.084174000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=166/42496, ttl=128 (request in 19)
21	5.085194000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) request id=0x0100, seq=167/42752, ttl=128 (request in 21)
22	6.085338000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=167/42752, ttl=128 (request in 21)
23	6.086161000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) request id=0x0100, seq=168/43008, ttl=128 (request in 23)
24	6.086161000	192.168.10.3	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=168/43008, ttl=128 (request in 23)

```

0000 00 1f 16 bc c0 52 04 3d 7e 55 6f fb 08 00 45 00  ....R...R...E.
0010 00 3c 2e 7c 00 00 80 01 76 ef c0 a8 04 01 c0 a8  ..K...K.....
0020 0a 02 00 00 53 c4 11 00 00 98 61 62 63 64 65 66  ...K...abodef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmopqrstuv
0040 77 61 62 63 64 65 66 67 68 69                      wxyzdefghij
  
```

## 18. Wireshark Capture ICMP Reply 1.

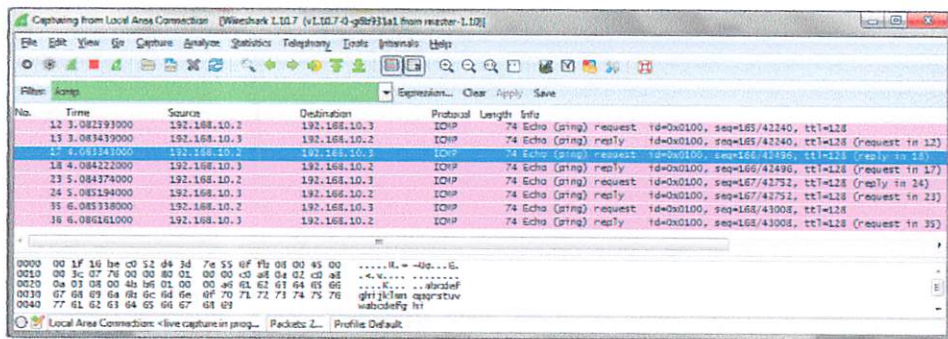
Local Area Connection: <live capture in prog... Packets: 1... Profile: Default

No.	Time	Source	Destination	Protocol	Length	Info
17	3.083133000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) request id=0x0100, seq=165/42240, ttl=128
18	4.083343000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=165/42240, ttl=128 (request in 17)
19	4.084222000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) request id=0x0100, seq=166/42496, ttl=128 (request in 19)
20	5.084174000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=166/42496, ttl=128 (request in 19)
21	5.085194000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) request id=0x0100, seq=167/42752, ttl=128 (request in 21)
22	6.085338000	192.168.10.2	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=167/42752, ttl=128 (request in 21)
23	6.086161000	192.168.10.1	192.168.10.2	ICMP	74	Echo (ping) request id=0x0100, seq=168/43008, ttl=128 (request in 23)
24	6.086161000	192.168.10.3	192.168.10.1	ICMP	74	Echo (ping) reply id=0x0100, seq=168/43008, ttl=128 (request in 23)

```

0000 44 3d 7e 55 6f fb 00 1f 16 bc c0 52 08 00 45 00  ....R...R..E.
0010 00 3c 2e 7c 00 00 80 01 76 ef c0 a8 04 01 c0 a8  ..K...K.....
0020 0a 02 00 00 53 c4 11 00 00 98 61 62 63 64 65 66  ...K...abodef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ghijklmopqrstuv
0040 77 61 62 63 64 65 66 67 68 69                      wxyzdefghij
  
```

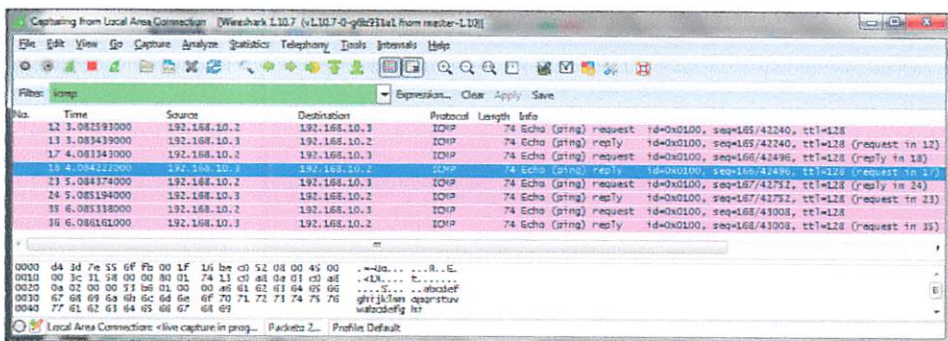
## 19. Wireshark Capture ICMP Request 2



The screenshot shows a Wireshark capture of ICMP traffic. The packet list pane shows several ICMP Echo (ping) requests and replies. The selected packet is an ICMP Echo (ping) request with sequence number 16542240 and TTL 128. The packet bytes pane shows the raw data of the ICMP request, including the type (8), code (0), checksum, identifier (0), and sequence number (16542240).

No.	Time	Source	Destination	Protocol	Length	Info
12	3.082593000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=165/42240, ttl=128
13	3.083439000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=165/42240, ttl=128 (request in 12)
17	4.084223000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=166/42496, ttl=128 (request in 17)
18	4.084223000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=166/42496, ttl=128 (request in 17)
23	5.085194000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=167/42752, ttl=128 (request in 23)
24	5.085194000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=167/42752, ttl=128 (reply in 24)
35	6.085338000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=168/43008, ttl=128
36	6.086161000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=168/43008, ttl=128 (request in 35)

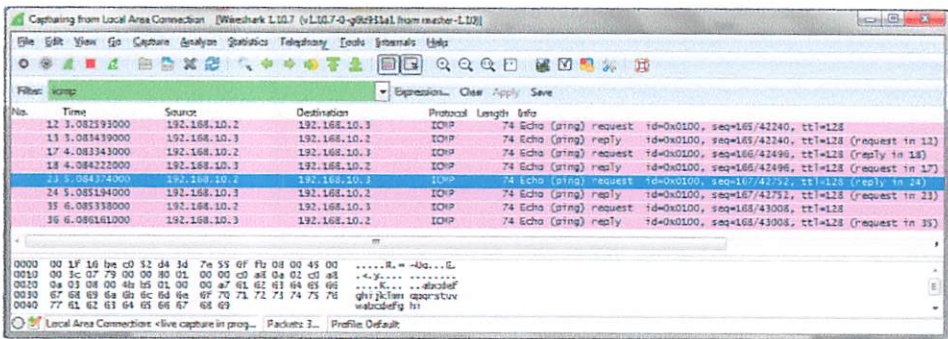
## 20. Wireshark Capture ICMP Reply 2



The screenshot shows a Wireshark capture of ICMP traffic. The packet list pane shows several ICMP Echo (ping) requests and replies. The selected packet is an ICMP Echo (ping) reply with sequence number 16542240 and TTL 128. The packet bytes pane shows the raw data of the ICMP reply, including the type (0), code (0), checksum, identifier (0), and sequence number (16542240).

No.	Time	Source	Destination	Protocol	Length	Info
12	3.082593000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=165/42240, ttl=128
13	3.083439000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=165/42240, ttl=128 (request in 12)
17	4.083439000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=166/42496, ttl=128 (reply in 18)
18	4.084223000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=166/42496, ttl=128 (request in 17)
23	5.085194000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=167/42752, ttl=128 (request in 23)
24	5.085194000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=167/42752, ttl=128 (reply in 24)
35	6.085338000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=168/43008, ttl=128
36	6.086161000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=168/43008, ttl=128 (request in 35)

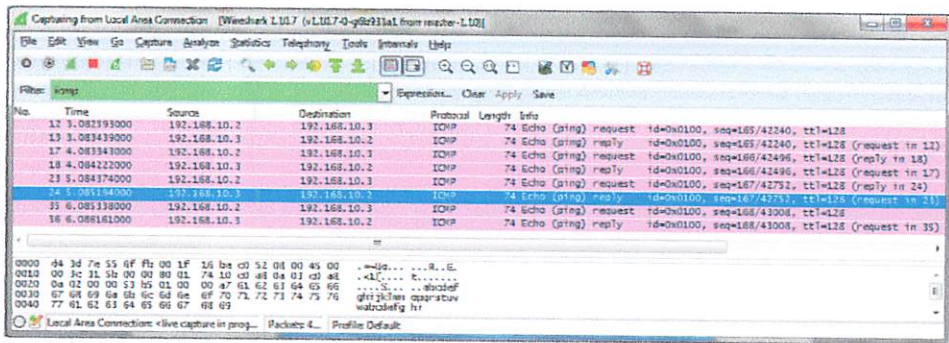
## 21. Wireshark Capture ICMP Request 3.



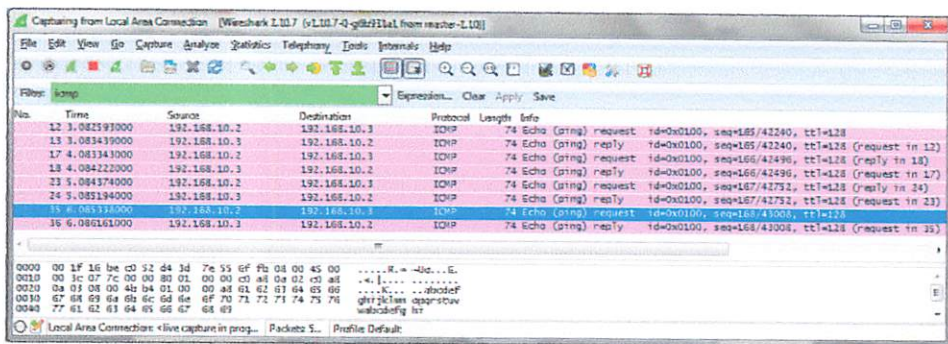
The screenshot shows a Wireshark capture of ICMP traffic. The packet list pane shows several ICMP Echo (ping) requests and replies. The selected packet is an ICMP Echo (ping) request with sequence number 16542240 and TTL 128. The packet bytes pane shows the raw data of the ICMP request, including the type (8), code (0), checksum, identifier (0), and sequence number (16542240).

No.	Time	Source	Destination	Protocol	Length	Info
12	3.082593000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=165/42240, ttl=128
13	3.083439000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=165/42240, ttl=128 (request in 12)
17	4.083439000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=166/42496, ttl=128 (reply in 18)
18	4.084223000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=166/42496, ttl=128 (request in 17)
23	5.085194000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=167/42752, ttl=128 (request in 23)
24	5.085194000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=167/42752, ttl=128 (reply in 24)
35	6.085338000	192.168.10.2	192.168.10.3	ICMP	74	Echo (ping) request 16=0x0100, seq=168/43008, ttl=128
36	6.086161000	192.168.10.3	192.168.10.2	ICMP	74	Echo (ping) reply 16=0x0100, seq=168/43008, ttl=128 (request in 35)

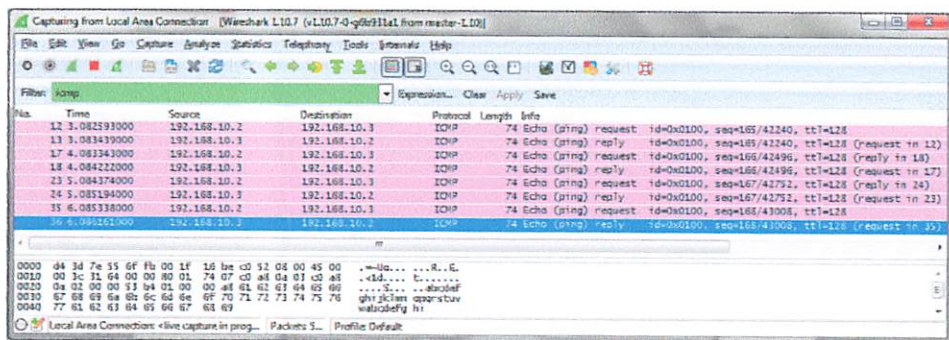
## 22. Wireshark Capture ICMP Reply 3



## 23. Wireshark Capture ICMP Request 4.



## 24. Wireshark Capture ICMP Reply 4 menggunakan kabel cat 5e



## 25. Wireshark Capture ICMP Request 1.

The screenshot shows a Wireshark capture of network traffic. The packet list pane displays several ICMP Echo (ping) requests and replies. The selected packet is an ICMP Echo (ping) request, No. 25, at time 3.557320000. The details pane shows the ICMP Echo (ping) request with sequence number 17344288 and ttl=128. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
25	3.557320000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) request 1d=0x0001, seq=173/44288, ttl=128 (request in 24)
26	3.557320000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) reply 1d=0x0001, seq=173/44288, ttl=128 (request in 27)
31	4.559998000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) request 1d=0x0001, seq=174/44544, ttl=128 (request in 32)
32	4.559998000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) reply 1d=0x0001, seq=174/44544, ttl=128 (request in 31)
36	5.562940000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request 1d=0x0001, seq=175/44800, ttl=128 (request in 37)
37	5.562940000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply 1d=0x0001, seq=175/44800, ttl=128 (request in 36)
45	6.560279000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request 1d=0x0001, seq=176/45056, ttl=128 (request in 43)
46	6.562881000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply 1d=0x0001, seq=176/45056, ttl=128 (request in 45)

## 26. Wireshark Capture ICMP Reply 1.

The screenshot shows a Wireshark capture of network traffic. The packet list pane displays several ICMP Echo (ping) requests and replies. The selected packet is an ICMP Echo (ping) reply, No. 26, at time 3.557320000. The details pane shows the ICMP Echo (ping) reply with sequence number 17344288 and ttl=128. The packet bytes pane shows the raw data in hexadecimal and ASCII.

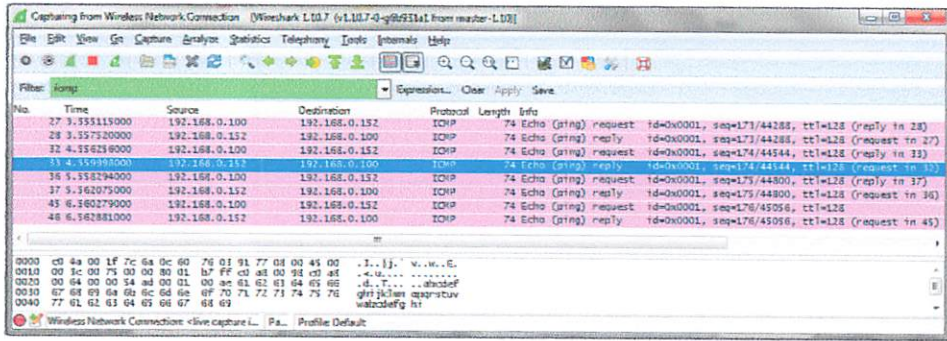
No.	Time	Source	Destination	Protocol	Length	Info
27	3.551130000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request 1d=0x0001, seq=173/44288, ttl=128 (reply in 28)
28	3.557320000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply 1d=0x0001, seq=173/44288, ttl=128 (request in 27)
31	4.559998000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) request 1d=0x0001, seq=174/44544, ttl=128 (request in 32)
32	4.559998000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) reply 1d=0x0001, seq=174/44544, ttl=128 (request in 31)
36	5.562940000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request 1d=0x0001, seq=175/44800, ttl=128 (request in 37)
37	5.562940000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply 1d=0x0001, seq=175/44800, ttl=128 (request in 36)
45	6.560279000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request 1d=0x0001, seq=176/45056, ttl=128 (request in 43)
46	6.562881000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply 1d=0x0001, seq=176/45056, ttl=128 (request in 45)

## 27. Wireshark Capture ICMP Request 2

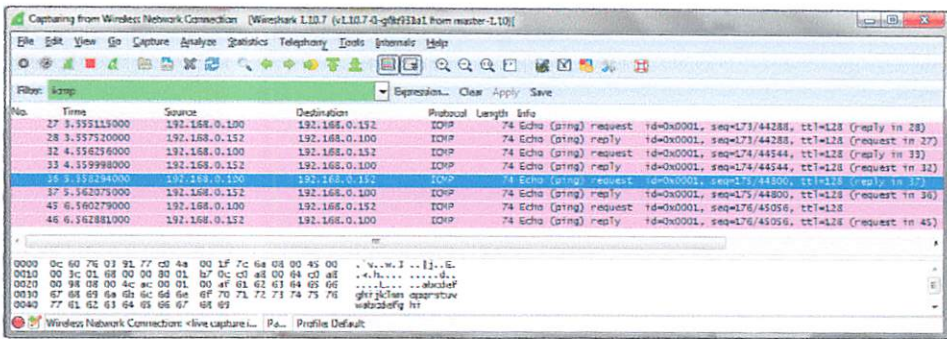
The screenshot shows a Wireshark capture of network traffic. The packet list pane displays several ICMP Echo (ping) requests and replies. The selected packet is an ICMP Echo (ping) request, No. 31, at time 4.559998000. The details pane shows the ICMP Echo (ping) request with sequence number 17444544 and ttl=128. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
27	3.551130000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request 1d=0x0001, seq=173/44288, ttl=128 (reply in 28)
28	3.557320000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply 1d=0x0001, seq=173/44288, ttl=128 (request in 27)
31	4.559998000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) request 1d=0x0001, seq=174/44544, ttl=128 (reply in 32)
32	4.559998000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) reply 1d=0x0001, seq=174/44544, ttl=128 (request in 31)
36	5.562940000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request 1d=0x0001, seq=175/44800, ttl=128 (request in 37)
37	5.562940000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply 1d=0x0001, seq=175/44800, ttl=128 (request in 36)
45	6.560279000	192.168.0.100	192.168.0.152	ICMP	74	Echo (ping) request 1d=0x0001, seq=176/45056, ttl=128 (request in 43)
46	6.562881000	192.168.0.152	192.168.0.100	ICMP	74	Echo (ping) reply 1d=0x0001, seq=176/45056, ttl=128 (request in 45)

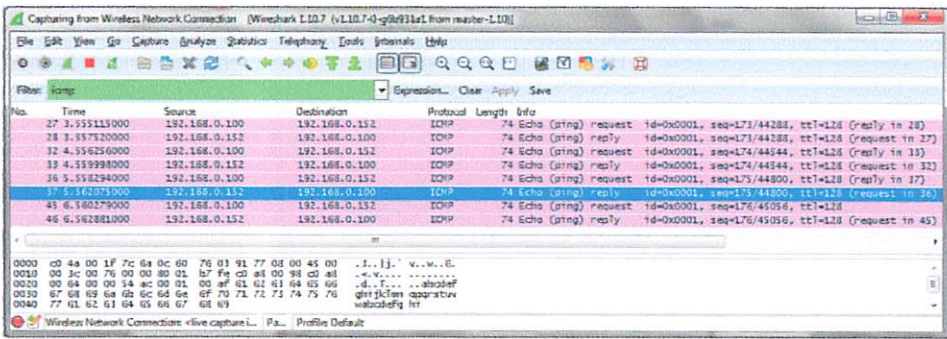
## 28. Wireshark Capture ICMP Reply 2



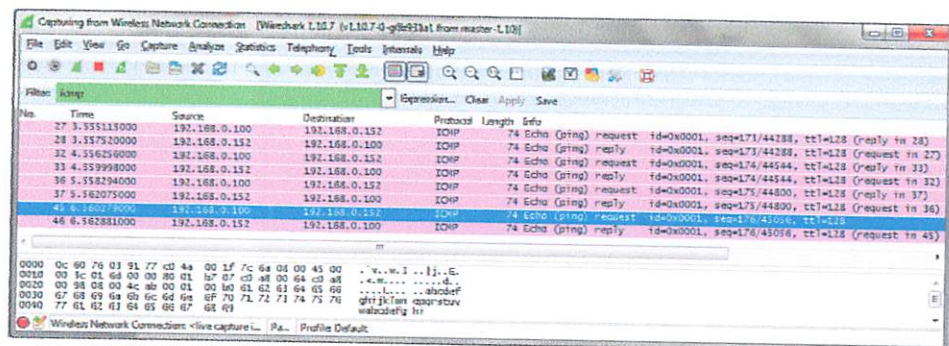
## 29. Wireshark Capture ICMP Request 3



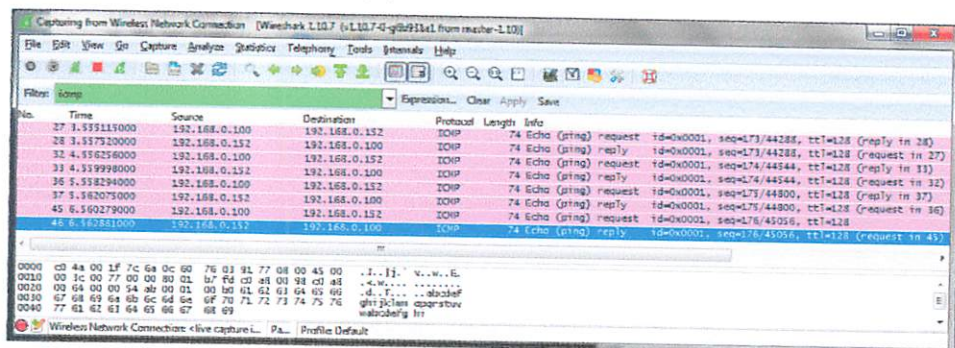
## 30. Wireshark Capture ICMP Reply 3



### 31. Wireshark Capture ICMP Request 4



### 32. Wireshark Capture ICMP Reply 4



### 33. Throughput video 1.

Display	
Display filter:	none
Ignored packets:	0 (0.000%)
Traffic	◀ Captured ▶ Displayed ▶ Displayed % ▶ Marked ▶ Marked %
Packets	268 268 100.000% 0 0.000%
Between first and last packet	4.520 sec
Avg. packets/sec	59.289
Avg. packet size	693.399 bytes
Bytes	185831 185831 100.000% 0 0.000%
Avg. bytes/sec	41110.605
Avg. MBit/sec	0.329

### 34. Throughput video 2.

Display						
Display filter:	none					
Ignored packets:	0 (0.000%)					
Traffic	◀ Captured	◀ Displayed	◀ Displayed %	◀ Marked	◀ Marked %	
Packets	321	321	100.000%	0	0.000%	
Between first and last packet 3.631 sec						
Avg. packets/sec	88.397					
Avg. packet size	849.564 bytes					
Bytes	272710	272710	100.000%	0	0.000%	
Avg. bytes/sec	75098.836					
Avg. MBit/sec	0.601					

### 35. Throughput video 3.

Display						
Display filter:	none					
Ignored packets:	0 (0.000%)					
Traffic	◀ Captured	◀ Displayed	◀ Displayed %	◀ Marked	◀ Marked %	
Packets	526	526	100.000%	0	0.000%	
Between first and last packet 5.304 sec						
Avg. packets/sec	99.164					
Avg. packet size	895.447 bytes					
Bytes	471005	471005	100.000%	0	0.000%	
Avg. bytes/sec	88796.189					
Avg. MBit/sec	0.710					

### 36. Throughput video 1.

Display						
Display filter:	none					
Ignored packets:	0 (0.000%)					
Traffic	◀ Captured	◀ Displayed	◀ Displayed %	◀ Marked	◀ Marked %	
Packets	274	274	100.000%	0	0.000%	
Between first and last packet 4.030 sec						
Avg. packets/sec	67.994					
Avg. packet size	694.836 bytes					
Bytes	190385	190385	100.000%	0	0.000%	
Avg. bytes/sec	47244.868					
Avg. MBit/sec	0.378					

37. Throughput video 2.

Display						
Display filter:		none				
Ignored packets:		0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	392	392	100.000%	0	0.000%	
Between first and last packet 5.525 sec						
Avg. packets/sec	70.955					
Avg. packet size	768.638 bytes					
Bytes	301306	301306	100.000%	0	0.000%	
Avg. bytes/sec	54538.912					
Avg. MBit/sec	0.436					

38. Throughput video 3.

Display						
Display filter:		none				
Ignored packets:		0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	322	322	100.000%	0	0.000%	
Between first and last packet 3.186 sec						
Avg. packets/sec	101.072					
Avg. packet size	895.165 bytes					
Bytes	288243	288243	100.000%	0	0.000%	
Avg. bytes/sec	90476.264					
Avg. MBit/sec	0.724					

39. Throughput video 1.

Display						
Display filter:		none				
Ignored packets:		0 (0.000%)				
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	11833	11833	100.000%	0	0.000%	
Between first and last packet 151.172 sec						
Avg. packets/sec	78.275					
Avg. packet size	951.049 bytes					
Bytes	11253762	11253762	100.000%	0	0.000%	
Avg. bytes/sec	74443.485					
Avg. MBit/sec	0.596					

#### 40. Throughput video 2.

Display						
Display filter:	none					
Ignored packets:	0 (0.000%)					
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	108806	108806	100.000%	0	0.000%	
Between first and last packet	730.668 sec					
Avg. packets/sec	148.913					
Avg. packet size	1016.667 bytes					
Bytes	110619502	110619502	100.000%	0	0.000%	
Avg. bytes/sec	151395.014					
Avg. MBit/sec	1.211					

#### 41. Throughput video 3.

Display						
Display filter:	none					
Ignored packets:	0 (0.000%)					
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	57041	57041	100.000%	0	0.000%	
Between first and last packet	325.270 sec					
Avg. packets/sec	175.365					
Avg. packet size	1038.771 bytes					
Bytes	59252555	59252555	100.000%	0	0.000%	
Avg. bytes/sec	182163.937					
Avg. MBit/sec	1.457					

#### 42. Throughput video 1.

Display						
Display filter:	none					
Ignored packets:	0 (0.000%)					
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	79	79	100.000%	0	0.000%	
Between first and last packet	1.601 sec					
Avg. packets/sec	49.332					
Avg. packet size	860.975 bytes					
Bytes	68017	68017	100.000%	0	0.000%	
Avg. bytes/sec	42473.855					
Avg. MBit/sec	0.340					

#### 43. Throughput video 2.

Display						
Display filter:						
Ignored packets: none 0 (0.000%)						
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	394	394	100.000%	0	0.000%	
Between first and last packet 4.809 sec						
Avg. packets/sec	81.924					
Avg. packet size	834.987 bytes					
Bytes	328985	328985	100.000%	0	0.000%	
Avg. bytes/sec	68405.779					
Avg. MBit/sec	0.547					

#### 44. Throughput video 3.

Display						
Display filter:						
Ignored packets: none 0 (0.000%)						
Traffic	Captured	Displayed	Displayed %	Marked	Marked %	
Packets	326	326	100.000%	0	0.000%	
Between first and last packet 3.514 sec						
Avg. packets/sec	92.779					
Avg. packet size	895.914 bytes					
Bytes	292068	292068	100.000%	0	0.000%	
Avg. bytes/sec	83122.397					
Avg. MBit/sec	0.665					

#### 45. Analisa packet loss dan jitter video 1.

Wireshark RTP Stream Analysis

Forward Direction | Reversed Direction

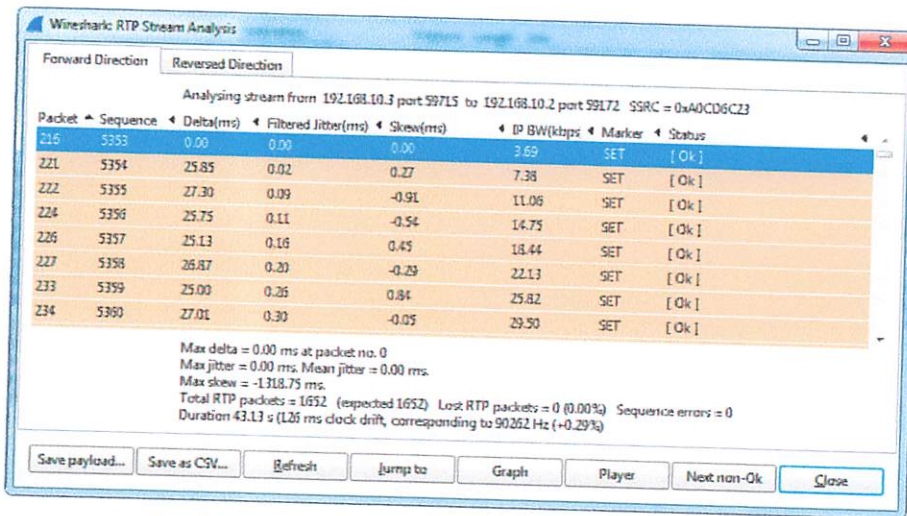
Analysing stream from 192.168.10.3 port 69045 to 192.168.10.2 port 59160 SSRC = 0x635D5EAB

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
1	16378	0.00	0.00	0.00	3.69	SET	[Ok]
2	16379	25.97	0.01	0.15	7.38	SET	[Ok]
3	16380	30.41	0.28	-4.14	11.06	SET	[Ok]
4	16381	21.65	0.54	0.33	14.75	SET	[Ok]
7	16382	28.88	0.68	-2.43	18.44	SET	[Ok]
8	16383	29.99	0.88	-6.30	22.13	SET	[Ok]
10	16384	20.03	1.20	-0.21	25.82	SET	[Ok]
12	16385	30.00	1.37	-4.08	29.50	SET	[Ok]

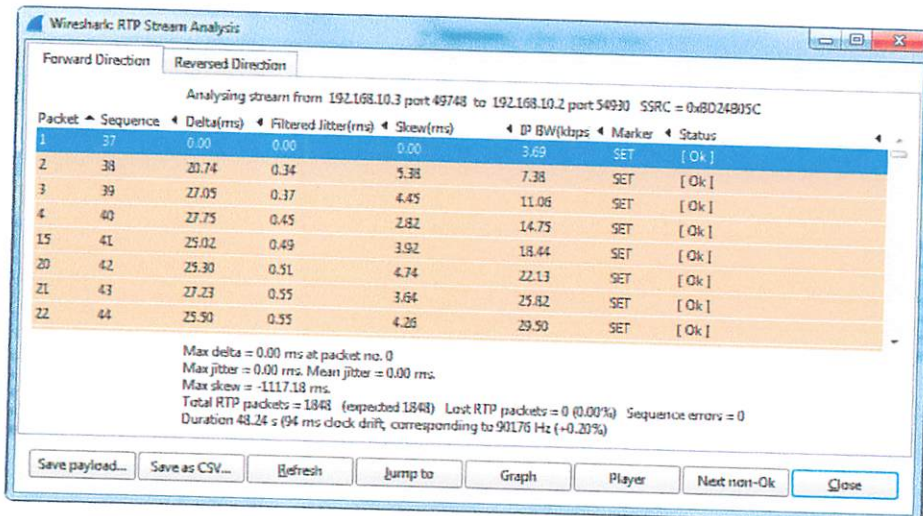
Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -94.25 ms.  
 Total RTP packets = 1316 (expected 1316) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 34.35 s (1 ms clock drift, corresponding to 90002 Hz (+0.00%))

Save payload... Save as CSV... Refresh Jump to Graph Player Next non-Ok Close

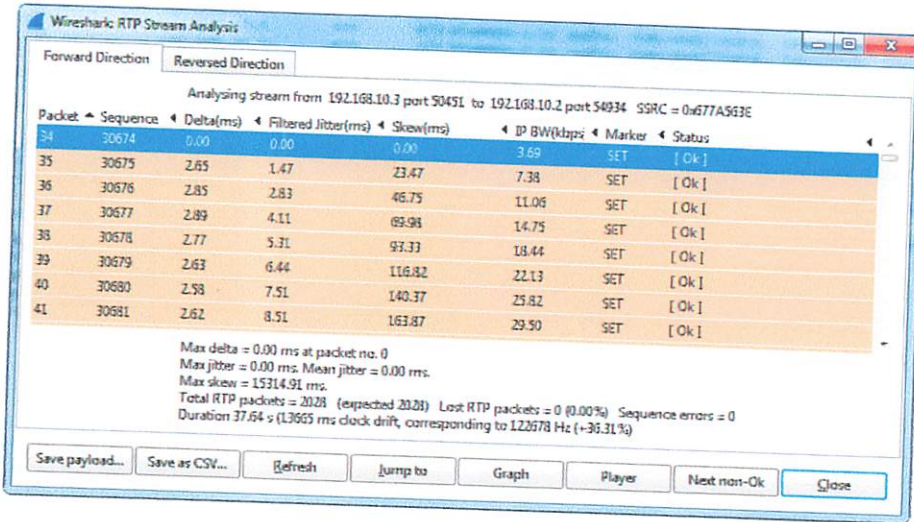
46. Analisa packet loss dan jitter video 2.



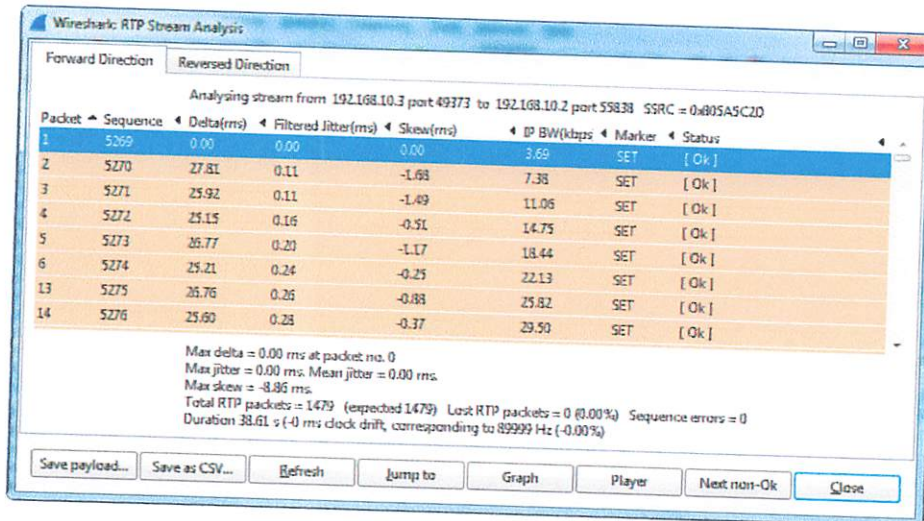
47. Analisa packet loss dan jitter video 3.



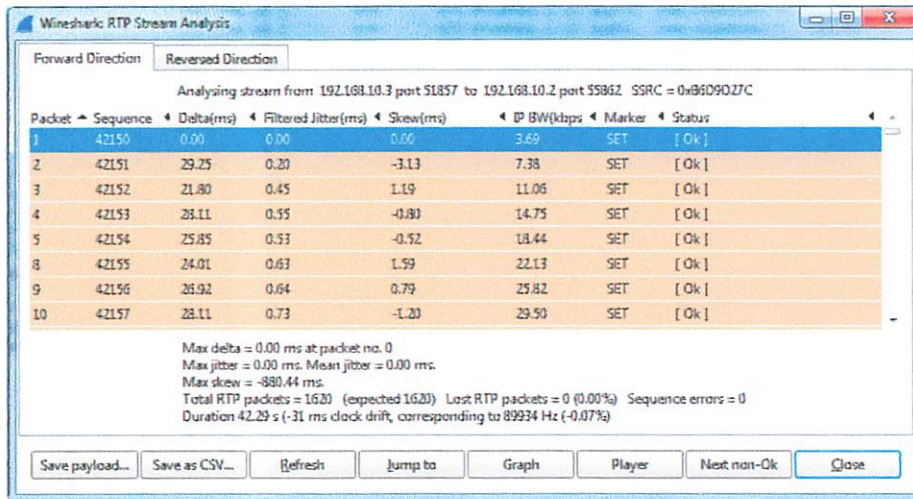
48. Analisa packet loss dan jitter video 1.



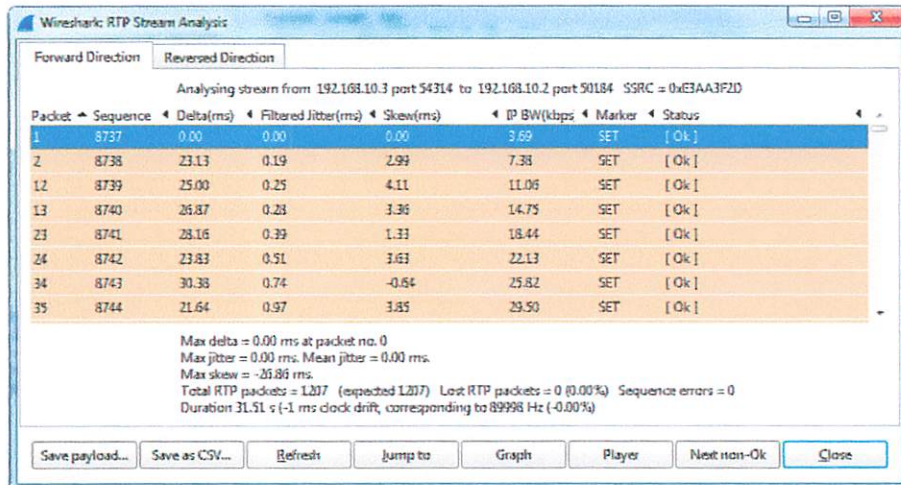
49. Analisa packet loss dan jitter video 2.



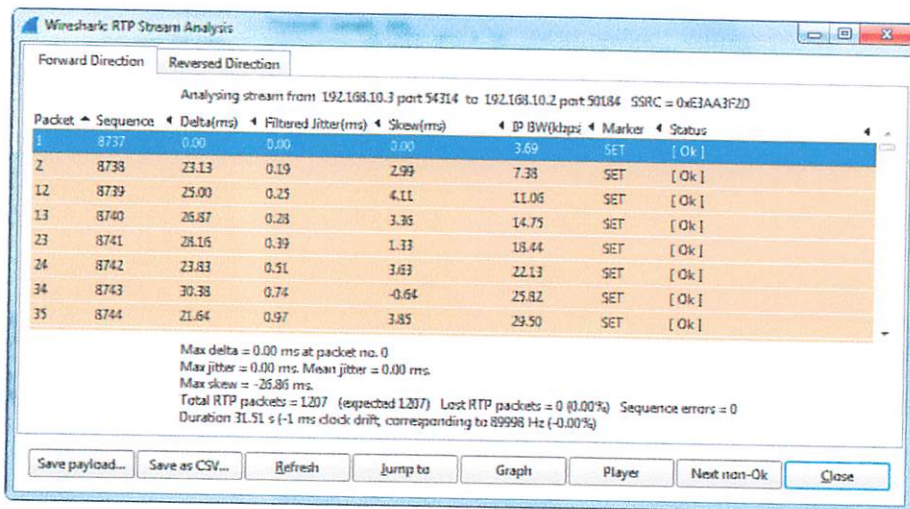
50. Analisa packet loss dan jitter video 3.



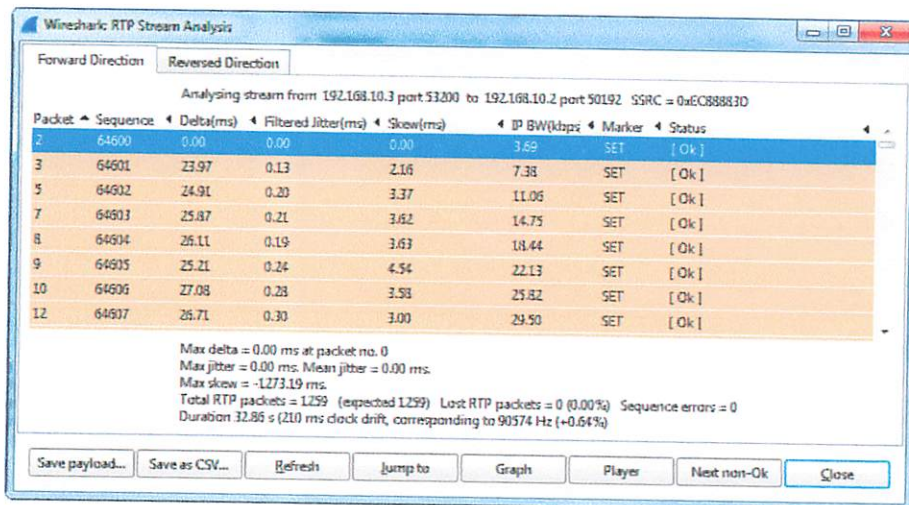
51. Analisa packet loss dan jitter video 1.



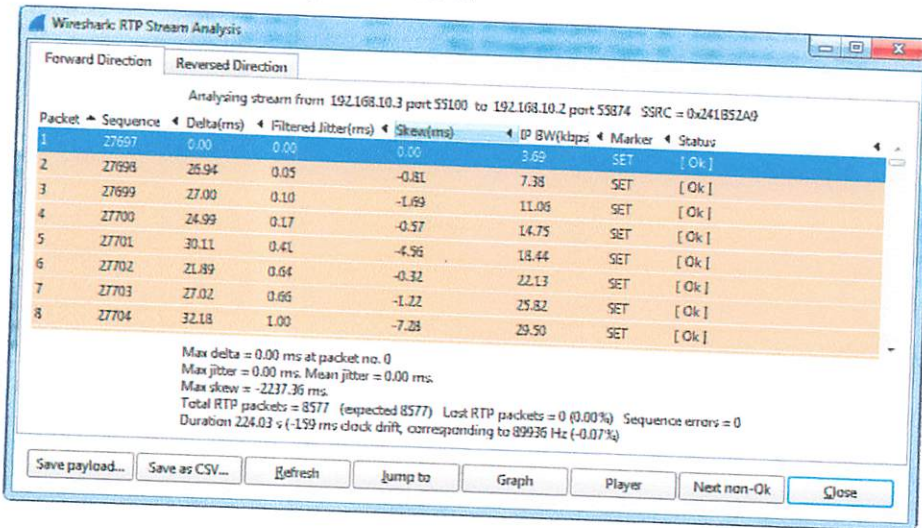
## 52. Analisa packet loss dan jitter video 2.



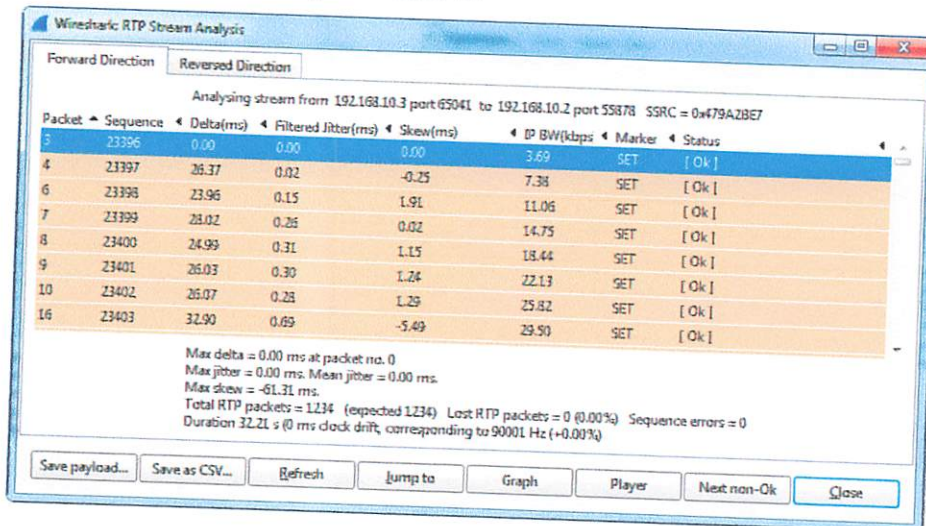
## 53. Analisa packet loss dan jitter video 3.



54. Analisa packet loss dan jitter video 1.



55. Analisa packet loss dan jitter video 2.



56. Analisa packet loss dan jitter video 3.

Wireshark: RTP Stream Analysis

Forward Direction    Reversed Direction

Analysing stream from 192.168.10.3 port 69043 to 192.168.10.2 port 54250 SSRC = 0x790221CF

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	DP BW(kbps)	Marker	Status
1	5699	0.00	0.00	0.00	3.69	SET	[ Ok ]
2	5700	27.03	0.06	-0.90	7.38	SET	[ Ok ]
3	5701	27.07	0.11	-1.85	11.06	SET	[ Ok ]
13	5702	24.92	0.18	-0.45	14.75	SET	[ Ok ]
14	5703	28.01	0.29	-2.54	18.44	SET	[ Ok ]
15	5704	25.43	0.31	-1.84	22.13	SET	[ Ok ]
16	5705	24.58	0.39	-0.30	25.82	SET	[ Ok ]
17	5706	30.01	0.61	-4.18	29.50	SET	[ Ok ]

Max delta = 0.00 ms at packet no. 0  
 Max jitter = 0.00 ms. Mean jitter = 0.00 ms.  
 Max skew = -3071.16 ms.  
 Total RTP packets = 865 (expected 865) Lost RTP packets = 0 (0.00%) Sequence errors = 0  
 Duration 22.57 s (887 ms clock drift, corresponding to 93538 Hz (+3.93%))

Save payload...    Save as CSV...    Refresh    Jump to    Graph    Player    Next non-Ok    Close