

BAB IV

HASIL DAN PEMBAHASAN

4.1 Umum

Pada penelitian ini akan dilakukan beberapa tahap untuk mendapatkan hasil, dimulai dengan perancangan konfigurasi kamera *Action*, pemancar, penerima, *Blackmagic Intensity*, *modem router*, laptop yang digunakan sebagai *server* dan *client*. Setelah perancangan selesai, dilakukan pengambilan data pengaruh perubahan resolusi video dari kamera *action* pada jaringan lokal. Penelitian ini menggunakan *modem router* sebagai akses point untuk menghubungkan *server* ke *client* dilihat dari parameter *delay*, *throughput*, dan *packet loss*. Untuk jarak yang diamati antara pemancar dan penerima 20 m, jarak *router* ke *client* 10m dengan pengaruh resolusi video yang diuji yaitu 240p, 360p, 480p, dan 720p. Sedangkan pengambilan data dilakukan dari sisi *client* dengan menggunakan *software packet analyzer Wireshark* dengan media *wireless*. Jaringan yang diamati adalah jaringan yang menghubungkan dari *server* ke *client*.

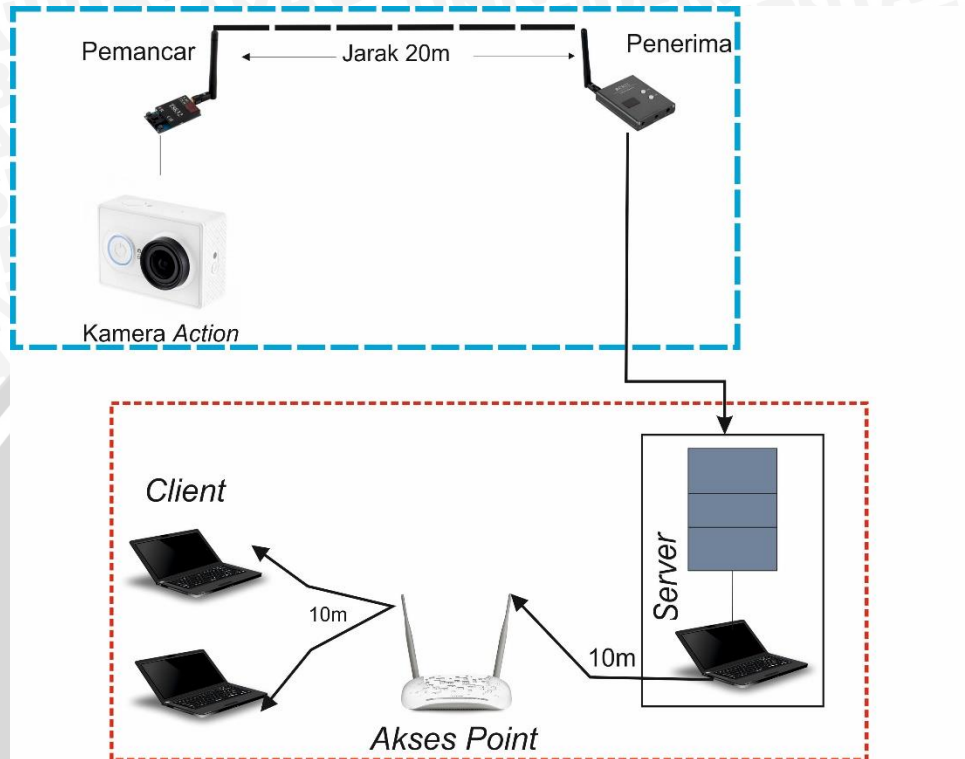
4.2 Pembahasan

Tahapan pembahasan yang dilakukan antara lain : Konfigurasi pada kamera *Action*, *Blackmagic Intensity Shuttle*, *modem router*, instalasi perangkat lunak di sisi *server*, instalasi perangkat lunak di sisi *client* dan pengambilan data menggunakan *software packet analyzer Wireshark*.

4.2.1 Konfigurasi Jaringan

Konfigurasi jaringan menggunakan perangkat yang terdiri dari Kamera *Action Xioami Yi*, pemancar TS832, penerima RC832, *Blackmagic Intensity Shuttle* sebagai alat *capture* yang digunakan agar bisa dilihat langsung oleh *server*, *modem router* sebagai akses *point* pada jaringan, satu buah PC laptop sebagai *server* yang terhubung

pada *Blackmagic Intensity Shuttle* dan dua buah laptop sebagai pengguna. Gambar 4.1 merupakan gambar konfigurasi jaringan.



Gambar 4.1. Konfigurasi Jaringan.

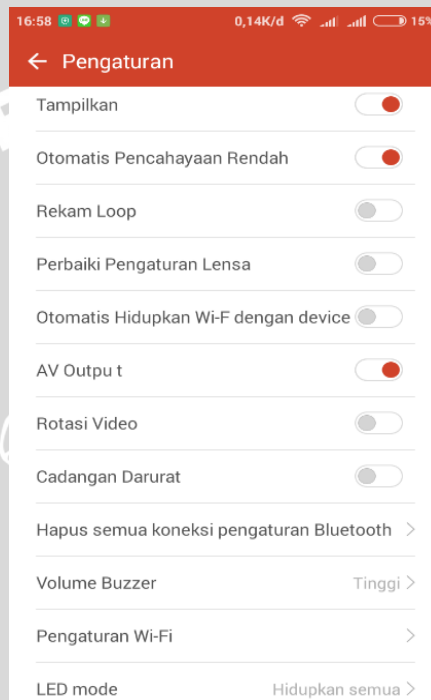
Pada Gambar 4.1 Kamera *Action* terhubung dengan pemancar menggunakan *FPV Transmission Cable* dan ditransmisikan dengan *mode Analog Video (AV)* melalui udara, diterima oleh penerima AV yang akan *dicapture* oleh *Blackmagic Intensity Shuttle* dan di *streaming*kan oleh *server* menuju *client* melalui media *wireless*. Variasi *client* yang digunakan adalah 1 *client* dan 2 *client*. Untuk konfigurasi IP yang digunakan dijelaskan pada Tabel 4.1.

Tabel 4.1. Konfigurasi IP

Alamat IP	Keterangan
192.168.10.1	Server
192.168.10.5	PC Client 1
192.168.10.6	PC Client 2

4.2.2 Konfigurasi Perangkat Kamera Action

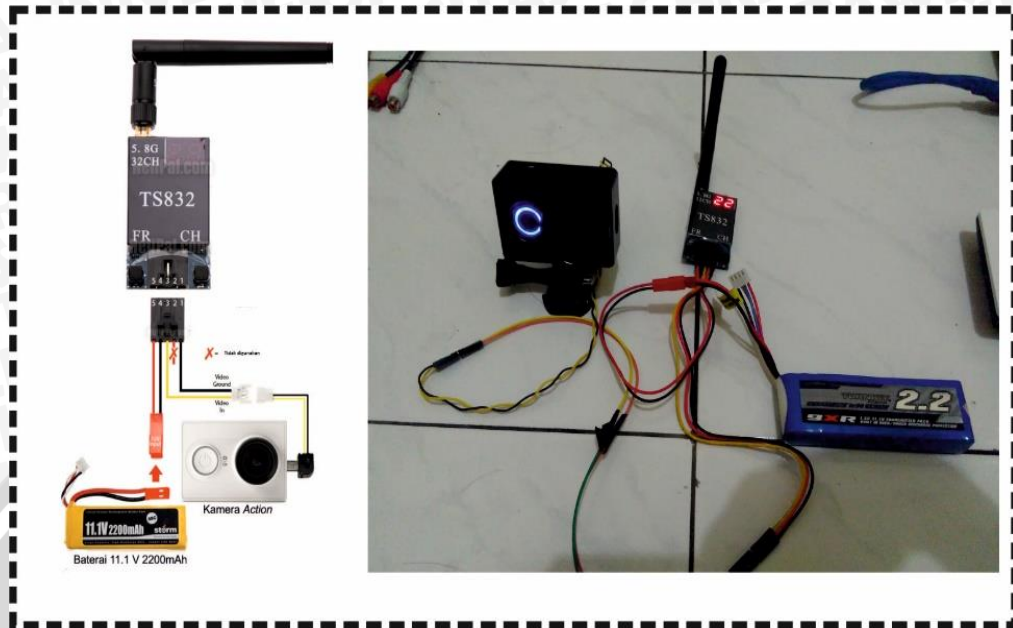
Kamera Action *Xiaomi Yi* diatur agar dapat mengirimkan video *mode AV* dengan cara masuk ke Pengaturan yang sudah dikoneksikan dengan *wifi* melalui *gadget*, kemudian klik *Device Setting*, hidupkan *AV output* pada pengaturan. Pengaturan ini hanya memerlukan koneksi dengan *gadget* sekali saja dapat dilihat pada gambar 4.2.



Gambar 4.2. Konfigurasi Kamera Action

4.2.3 Konfigurasi pada Pemancar TS832

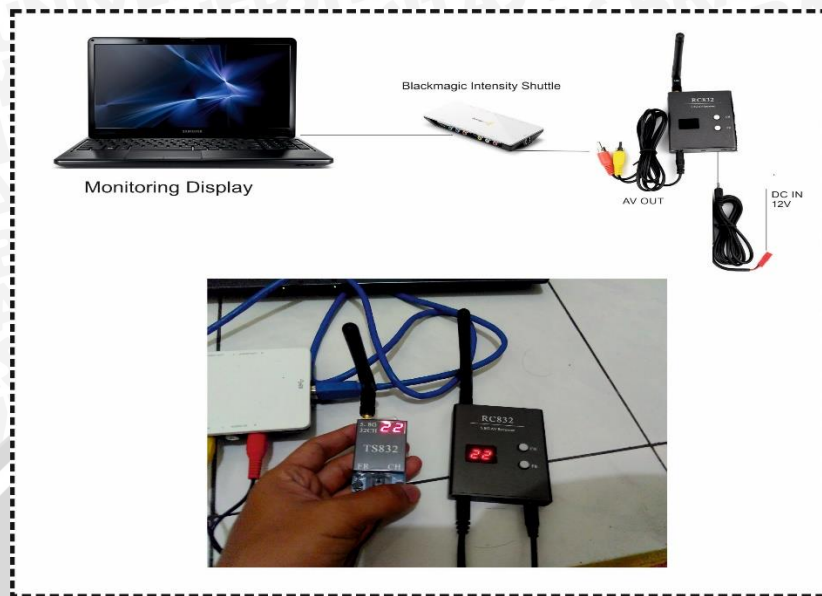
Pemancar berfungsi sebagai alat untuk mengirimkan video yang berupa *AV Output* keluaran dari kamera *Action* yang akan diteruskan kepada penerima *AV*. Pemancar yang digunakan membutuhkan input daya 12 volt yang diambil dari baterai *Lipo 3S*. *Frekuensi* dan *Channel* yang diinginkan diatur dan disamakan pada pemancar maupun penerima dapat dilihat pada Gambar 4.3



Gambar 4.3 Konfigurasi Pemancar TS832

4.2.4 Konfigurasi Penerima RC832

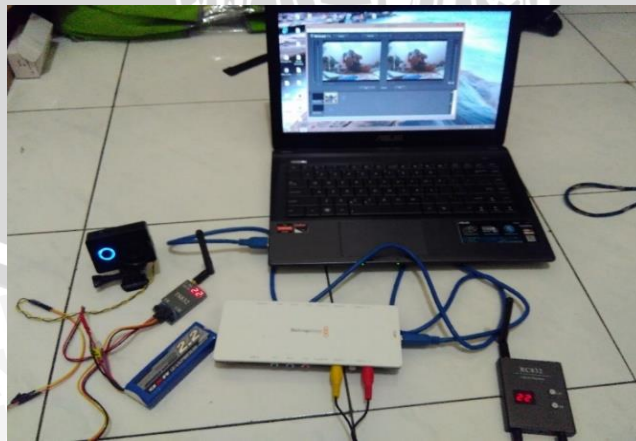
Penerima RC832 berfungsi sebagai penerima *Analog Video (AV)* yang menghubungkan data video yang dihasilkan kamera ke *Blackmagic Intensity Shuttle*. Banyak pilihan *frekuensi* dan *channel* yang ingin digunakan. *Frekuensi* dan *channel* pada penerima harus sama dengan *frekuensi* dan *channel* pada Pemancar. Terdapat 2 *output* dari penerima: AV Out 1 yang digunakan pada penelitian ini dan Av Out 2 jika ingin digunakan untuk penerima dari pemancar yang lain. Pada Penerima menggunakan input 12 volt sumber PLN. Konfigurasi Penerima bisa dilihat pada gambar 4.4



Gambar 4.4. Konfigurasi pada Penerima RC832

4.2.5 Konfigurasi *Blackmagic Intensity Shuttle*

Blackmagic Intensity Shuttle adalah alat *capture* dengan USB 3.0 yang dapat mengirimkan data lebih cepat 3 sampai 10 kali dibandingkan dengan USB 2.0 sehingga alat ini sangat mendukung dalam pentransferan data dari penerima ke laptop sebagai *server*. *Blackmagic Intensity Shuttle* memiliki banyak masukan yaitu HDMI mendukung *capture* video hingga 1080p, *analog component*, *composite* dan *S-Video*. Konfigurasi *Blackmagic Intensity Shuttle* dapat dilihat pada Gambar 4.5.



Gambar 4.5. Konfigurasi *Blackmagic Intensity Shuttle*.

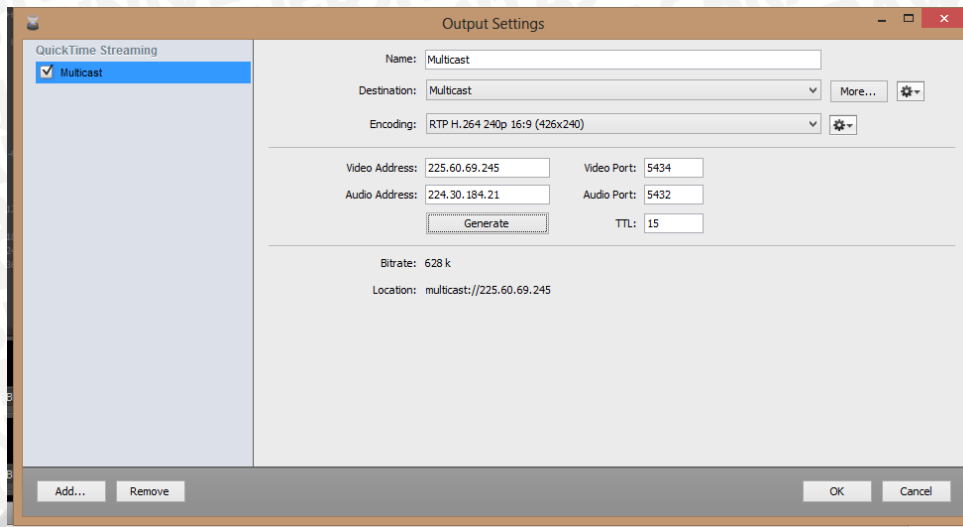
4.2.6 Konfigurasi Laptop sebagai Server

Server berguna untuk meneruskan video yang masuk dari *Blackmagic Intensity Shuttle* ke PC Client yang ingin melihat data video yang dikirimkan kamera *action* secara *live streaming*. Untuk melihat hasil *capture* menggunakan *software Wirecast* yang dapat dilihat pada Gambar 4.6



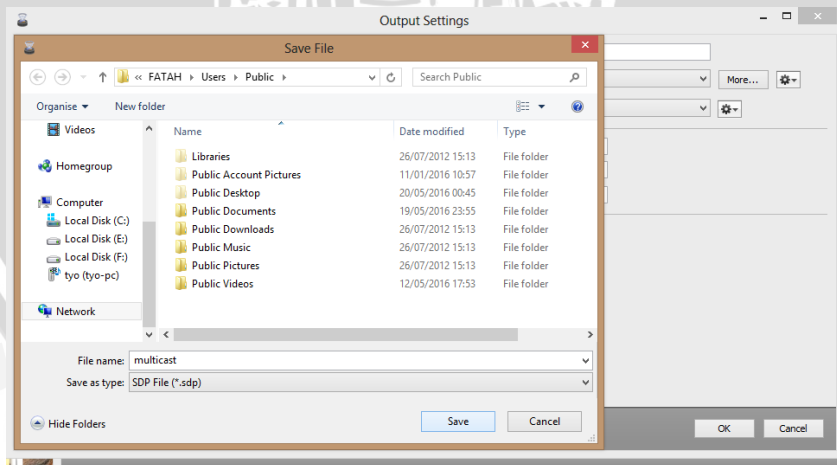
Gambar 4.6. Tampilan *Software Wirecast*

Streaming dilakukan dengan cara *multicast* yaitu proses pengiriman data dari satu titik ke banyak titik yang merupakan bagian dari satu grup tertentu dan yang memang menginginkan data tersebut. Kita bisa menentukan kualitas video yaitu resolusi video yang akan divariasikan pada penelitian ini di menu pengaturan pada *recording*. Konfigurasi pada *Wirecast* bisa dilihat pada Gambar 4.7.



Gambar 4.7. Konfigurasi *Multicast* pada *Wirecast*

Streaming multicast harus menyimpan data berupa SDP (*Session Description Protocol*) file yaitu format untuk mendeskripsikan *streaming media* dan inisialisasi parameter yang digunakan, seperti tipe media, format, *bitrate*, alamat dll. Kemudian kita simpan SDP file di penyimpanan *Network* yang dapat dilihat oleh *PC client 1* dan *PC client 2* sehingga SDP file dapat diakses pada semua *client*. Kemudian kita bisa klik *Stream* untuk meneruskan video yang tampil pada *Wirecast*. Penyimpanan SDP file dapat dilihat pada Gambar 4.8.



Gambar 4.8. Penyimpanan SDP file.

4.2.7 Spesifikasi Perangkat

- Spesifikasi Laptop yang dijadikan server dapat dilihat pada Tabel 4.2

Tabel 4.2 Spesifikasi Laptop sebagai Server.

Spesifikasi	Keterangan
Operating System:	Windows 8 Pro 64-bit
Processor	AMD A8-4500M APU with Radeon(tm) HD Graphics
RAM	4 GB or higher
VGA	AMD Radeon HD 7640G + 7470M Dual Graphics
USB	USB 3.0

- Modem Router

Jaringan Lokal Nirkabel menggunakan jaringan *modem router* TP Link untuk menghubungkan dari *server* ke *client* yang memiliki bandwith 100 Mbps. Server dan client langsung dihubungkan ke jaringan router tersebut dengan IP yang telah dikonfigurasi. Modem Router TP Link dapat dilihat pada Gambar 4.9.



Gambar 4.9. Modem Router TP Link

- PC Client

PC *client* digunakan sebagai sarana meminta layanan *live streaming* dan aplikasi yang digunakan adalah Video LAN Client (VLC) yang diinstal pada PC *client*. Spesifikasi minimal yang dibutuhkan untuk PC *client* ditunjukkan oleh Tabel 4.2

Tabel 4.3. Spesifikasi PC *Client*

Spesifikasi	Keterangan
Processor	Intel Core i3 <i>or higher</i> dan AMD A8 <i>or higher</i>
RAM	2 GB <i>or higher</i>
VGA	Intel HD Display atau ATI Radeon
USB	USB 2.0

- *FPV Transmission Cable*

Jenis kabel ini digunakan untuk menghubungkan kamera action ke pemancar dapat dilihat pada Gambar 4.10



Gambar 4.10 . *FPV Transmission Cable*

Dengan spesifikasi dapat dilihat pada Tabel 4.4

Tabel 4.4. Spesifikasi *FPV Transmission Cable*

AV Out to Transmitter : Black = GND, Yellow= Signal
DC 5V IN : Black = GND, Red = +5V IN
Micro USB to Xiaomi Yi Camera

4.3 Pengambilan dan Analisis Data

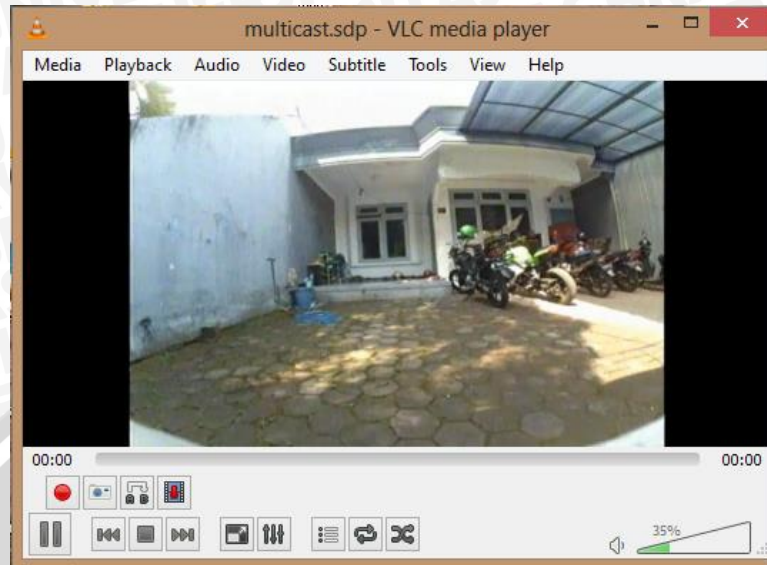
Penelitian ini mempunyai beberapa variabel yaitu dari resolusi video yang digunakan 240p, 360p, 480p dan 720p. Layanan yang digunakan pada penelitian ini yaitu *live streaming* menggunakan kamera *action* yang dikonfigurasi dengan jaringan *wireless* yaitu *modem router* sebagai akses *point* yang menghubungkan *server* untuk meneruskan video hasil *capture* dari kamera *action* ke *client* secara langsung. Dari konfigurasi tersebut akan diambil data sebanyak sepuluh kali yang kemudian hasil dari data tersebut akan di rata-rata.

Pada proses pengambilan data dilaksanakan dengan menggunakan *packet analyzer Wireshark* yang terletak pada sisi *client*. *Packet analyzer Wireshark* melewati proses *capturing*, membaca dan menganalisis aliran data yang melewati *interface client*. *Interface* pada *client* melalui jaringan *modem router TP Link*. Proses *capture* data pada jaringan akan berlangsung satu menit.

Setelah didapatkan data dari proses *capturing* data, data kemudian dianalisis menjadi aliran RTP. Ketika data sudah menjadi aliran RTP, maka diperoleh data sesuai dengan parameter yang dibutuhkan pada penelitian ini antara lain *delay*, *throughput*, dan *packet loss*.

4.4 Hasil Eksperimen

Pada pengambilan data ini *client* akan meminta *layanan live streaming* dan *server* akan mengatur besar *resolusi* video yang akan dikirimkan. Gambar 4.10 merupakan hasil eksperimen pengambilan data dengan layanan *live streaming* di sisi *client*. Penelitian ini dilakukan pada tanggal 23-25 Mei 2016 jam 8.00 pagi sampai jam 15.00.



Gambar 4.11. Pengaruh *Resolusi* Video Terhadap Layanan *Live Streaming* dengan Kamera *Action*

Gambar 4.11 menunjukkan hasil tayangan *live streaming* dari kamera *action* yang *distreaming*kan oleh *server* kepada *client*. *Resolusi* video mempengaruhi ukuran video dan *bitrate* video yang akan *distreaming*kan melalui jaringan. Pemilihan jenis *resolusi* video yang tepat akan memberikan kualitas yang sangat baik pada jaringan tersebut.

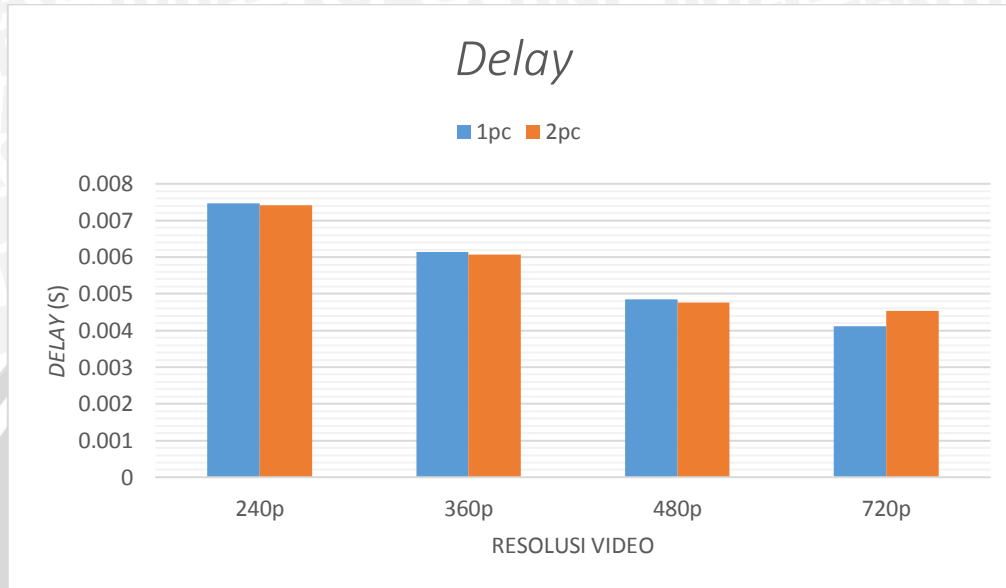
4.4.1 Delay

Delay menunjukkan waktu tunda yang dibutuhkan untuk mengirimkan data dari *server* hingga sampai ke *client*. Gambar 4.12 merupakan hasil pengamatan pengaruh *resolusi* video terhadap *delay* pada layanan *live streaming* dengan variasi jumlah *client* yaitu PC 1 dan PC 2. Tabel 4.5 merupakan hasil *delay* rata-rata yang diambil data sebanyak sepuluh kali percobaan.

Tabel 4.5 Pengaruh *resolusi* video terhadap variasi jumlah *client* dan *delay*

<i>Resolusi</i>	1 Client	2 Client
240p	0.007472	0.007416
360p	0.006148	0.006077
480p	0.004849	0.004757
720p	0.004112	0.004537

Kemudian jika Tabel 4.5. direpresentasikan ke dalam bentuk grafik hubungan antara besar resolusi video dengan *delay* yang diterima ditunjukkan pada Gambar 4.12 .



Gambar 4.12. Pengaruh *Resolusi* Video terhadap *Delay*

Berdasarkan hasil penelitian menunjukkan nilai *delay* yang berubah-ubah sesuai dengan resolusinya, semakin besar resolusi maka semakin kecil nilai *delay* yang didapatkan. Hal ini disebabkan karena jumlah paket data lebih besar dalam resolusi yang semakin besar tetapi waktu pengambilan setiap resolusi adalah sama. Jumlah client mempengaruhi besarnya *delay*, pada resolusi 240p,360p dan 480p *delay* 1 client lebih besar daripada 2 client, sedangkan pada resolusi 720p *delay* 2 client lebih besar daripada 1 client. Kualitas *Live Streaming* dalam penelitian ini adalah baik karena sesuai dengan standar ITU G-114 (*delay* < 100 ms)

4.4.2 Throughput

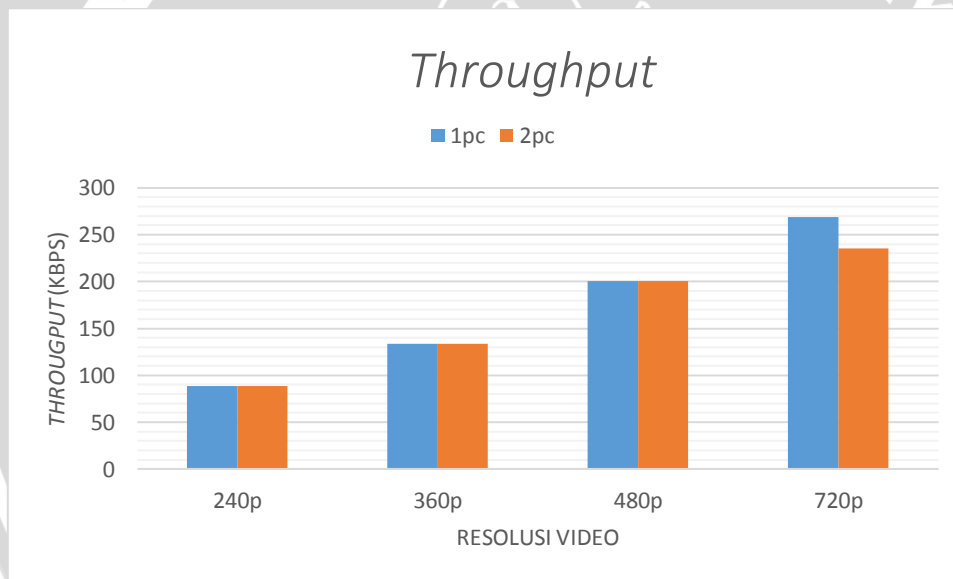
Throughput menunjukkan kecepatan penerimaan paket data yang diterima pengguna dengan keadaan benar terhadap waktu *transmisi* yang dibutuhkan dari sumber data pengirim ke penerima. *Throughput* memiliki satuan *byte/detik*. Hasil pengamatan pengaruh resolusi video terhadap *throughput* pada penelitian ditunjukkan

oleh Tabel 4.6 dan Gambar 4.13. Tabel 4.6 merupakan hasil rata-rata yang diambil data sebanyak sepuluh kali percobaan.

Tabel 4.6. Pengaruh resolusi video terhadap variasi jumlah *client* dan *Throughput*

Resolusi	1 Client	2 Client
240p	88.7	88.6
360p	133.4	133.6
480p	200.3	200.8
720p	268.6	235.2

Kemudian jika Tabel 4.5 direpresentasikan ke dalam bentuk grafik hubungan antara besar *resolusi* video dengan *throughput* ditunjukkan pada Gambar 4.13



Gambar 4.13. Pengaruh *Resolusi Video* terhadap *Throughput*

Berdasarkan hasil pengamatan, ditunjukkan adanya perubahan nilai *throughput* pada *resolusi* video yang semakin besar pada jaringan lokal *wireless*. Pada Tabel 4.5 menunjukkan nilai *throughput* semakin besar pada *resolusi* yang lebih besar. Pada variasi *client* menunjukkan hasil yang sama dimana nilai *throughput* semakin besar ketika *resolusi* video semakin besar, tetapi pada resolusi 720p *throughput* pada 1 *client* lebih besar daripada 2 *client*.

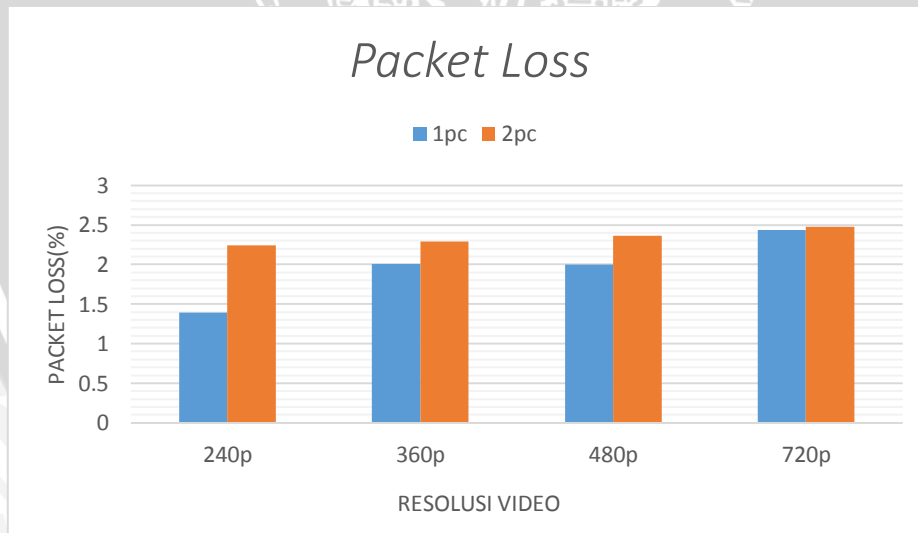
4.4.3 Packet Loss

Packet Loss pada penelitian menunjukkan nilai *rasio* dalam persen paket yang hilang dengan jumlah paket keseluruhan yang dikirimkan dari *server* ke *client*. Hasil pengamatan pengaruh resolusi video terhadap *packet loss* pada penelitian ditunjukkan oleh Tabel 4.7. dan Gambar 4.14. Tabel 4.7 merupakan hasil rata-rata yang diambil data sebanyak sepuluh kali percobaan.

Tabel 4.7. Pengaruh *resolusi* video terhadap variasi jumlah *client* dan *Packet Loss*

Resolusi	1 Client	2 Client
240p	1.39	2.24
360p	2.01	2.29
480p	2	2.36
720p	2.44	2.48

Kemudian jika Tabel 4.7 direpresentasikan ke dalam bentuk grafik hubungan antara besar kapasitas kanal dengan *packet loss* ditunjukkan pada Gambar 4.14.



Gambar 4.14. Pengaruh Resolusi Video terhadap *Packet Loss*

Pada pengambilan data menggunakan Wireshark, dapat dianalisis bahwa nilai *packet loss* dipengaruhi oleh besarnya *resolusi* video dan jumlah *client* dalam suatu layanan. Nilai *packet loss* terbesar didapat ketika 2 *client* mengakses *live streaming* pada resolusi video 720p.

4.5 Hasil Melalui Perhitungan Teoritis

Pada bagian ini dihasilkan parameter jaringan yaitu *delay end-to-end*, *packet loss* dan *throughput* melalui perhitungan dengan literatur yang ada, standar perangkat atau standar yang telah ditetapkan.

4.5.1 Spesifikasi Paket Live Streaming

Spesifikasi paket video streaming yang digunakan adalah sebagai berikut :

1. Pada penelitian ini menggunakan jenis *codec*, yaitu *codec* H.264 untuk video
Video codec adalah H.264/MPEG-4 AVC dengan *delay codec* 50 ms.
2. *Frame rate video streaming* yang dipergunakan adalah 30fps atau 33ms
3. Protokol: RTP, UDP dan IP

Panjang *Header* menurut Wikipedia.:

4. Panjang *header* RTP = 12 byte
5. Panjang *header* IP = 20 byte
6. Panjang *header* UDP = 8 byte
7. Probabilitas *bit error* = 10^{-10}
8. Waktu pengiriman total data = 60 s
9. Resolusi video dan *bitrate* yang digunakan bisa dilihat pada *software*

Wirecast:

- a. 240p (426x240), 628 Kbps
- b. 360p (640x360), 983 Kbps
- c. 480p (854x480), 1513 Kbps
- d. 720p (1280x720), 2563 Kbps

Untuk menghitung besar *payload video* pada layanan *video streaming* tiap *frame* ditentukan persamaan 2.1 dan 2.2, untuk video 240p

$$R_{240p} = \text{Bitrate}_{\text{video}} \times \text{frame rate}$$

$$= (628 \cdot 10^3 \text{bps}) \times (0,033) = 20724 \text{ bit}$$

Payload video kemudian diencodekan menggunakan *codec* H.264/ AVC MPEG-4 dan. Jumlah *payload* disegmentasi berdasarkan *payload* yang ditentukan oleh

codec. Sehingga jumlah nilai paket untuk *video* dapat diperoleh dengan menggunakan persamaan (2-3) :

$$P_V = \frac{P_{LV}}{P_{LV \max}} = \frac{20724}{254 \times 8} = 10.1988189 \approx 11 \text{ paket}$$

Kemudian jumlah *payload audio* dan *video* ditambah dengan *header* NALU, RTP, UDP dan IP menggunakan persamaan :

$$\begin{aligned} \text{Paket data video} &= P_{LV} + (P_V \times (\text{NALU} + H_{RTP} + H_{UDP} + H_{IP})) \\ &= 20724 + (11 \times (8+96+64+160)) \text{ bit} \\ &= 24069 \text{ bit} \end{aligned}$$

Sehingga jumlah besar paket data aplikasi *video streaming* yang ditransmisikan pada sistem ditentukan menggunakan persamaan (2-7) didapat:

$$\begin{aligned} W_{data} &= P_{data \text{ audio}} + P_{data \text{ video}} = (0+24069) \text{ bit} = 24069 \text{ bit} \\ W_{data} &= 3008 \text{ byte} \end{aligned}$$

4.5.2 Delay End-to-End

- $t_{end \text{ to end}} = t_{codec} + t_{net}$
- Analisis *Delay Jaringan*

$$t_{net} = t_{proses} + t_{prop} + t_{trans} + t_w$$

1) *Delay Proses* (t_{proses})

Pada layer transportasi paket data *audio* dan data *video* mengalami perubahan *format* dengan adanya penambahan *header* RTP dan UDP sehingga format data menjadi segmen, kemudian besar data ditambah *header* RTP (persamaan 2-11):

$$\begin{aligned} W_{message} &= W_{data} + \text{Header}_{RTP} \\ &= (3008 + 12) \text{ byte} \end{aligned}$$

$$W_{message} = 3020 \text{ byte}$$

Kemudian menambahkan *header* UDP. Pada protokol UDP memiliki panjang maksimal segmen (MSS UDP) yaitu sebesar 1640 *byte*, segmen data difragmentasi.

$$N_{\text{segmen}} = \frac{3020}{1640}$$

$$N_{\text{segmen}} = 1.841860716 \approx 2 \text{ segmen}$$

Dienkapsulasi adalah proses panjang maksimal data UDP di fragmentasi, dalam perhitungan UDP, terdapat 3 buah segmen yang berisi data 1460 *byte* dan 1 segmen berisi data $0,9009 \times 1460 = 1328,454 \text{ byte}$, total segmen adalah 4 segmen UDP.

Kemudian proses dienkapsulasi (2-13):

$$\begin{aligned} W_{\text{segmen}} &= W_{\text{message}} + (N_{\text{segmen}} \times \text{Header}_{\text{UDP}}) \\ &= 3020 + (2 \times 8) \end{aligned}$$

$$W_{\text{segmen}} = 3036 \text{ byte}$$

Kemudian proses penambahan *header IP* (20 *byte*) pada segmen UDP sehingga menjadi *datagram IP* :

$$\begin{aligned} W_{\text{datagram}} &= W_{\text{segmen}} + \text{Header}_{\text{IP}} \\ &= 3036 + 20 \end{aligned}$$

$$W_{\text{datagram}} = 3056 \text{ byte}$$

Proses selanjut yang dilakukan adalah *datagram IP* dienkapsulasi menjadi *frame ethernet* yaitu menambahkan *header ethernet* 18 *byte* pada *layer network* dengan menggunakan protokol *ethernet*

$$\begin{aligned} W_{\text{frame ethernet}} &= W_{\text{datagram}} + \text{Header}_{\text{ethernet}} \\ &= 3056 + 18 \text{ byte} \end{aligned}$$

$$W_{\text{frame ethernet}} = 3074 \text{ byte}$$

Dari *frame ethernet* mengalami penambahan *header MAC* 6 *byte*

$$\begin{aligned} W_{\text{frame total}} &= W_{\text{frame ethernet}} + \text{Header}_{\text{MAC}} \\ &= 3074 \text{ byte} + 6 \text{ byte} \end{aligned}$$

$$W_{\text{frame total}} = 3080 \text{ byte}$$

Perhitungan *delay* enkapsulasi dan deenkapsulasi adalah sebagai berikut,

- Jumlah paket yang dikirim (N) = 3080 *byte*

- *Average Packet Size* dari Wireshark :

Resolusi	Average Packet Size (byte)
240p	634
360p	823
480p	973
720p	1100

- Waktu pengiriman total data = 60 s

Perhitungan kecepatan pemrosesan pada terminal pengirim adalah sebagai berikut,

$$C_{proses} = \frac{3080 \times 634}{60} \times 8$$

$$C_{proses} = 260310.8021 \text{ bps}$$

Jadi, perhitungan *delay* enkapsulasi adalah sebagai berikut:

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

$$t_{enc} = \frac{(20 + 6 + 8 + 12 + 18) \times 8 \text{ bit}}{260310.8021}$$

$$t_{enc} = 0.00196688 \text{ sekon}$$

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

$$t_{dec} = \frac{(20 + 6 + 8 + 12 + 18) \times 8 \text{ bit}}{260310.8021}$$

$$t_{decc} = 0.00196688 \text{ sekon}$$

Sehingga didapatkan *delay* proses adalah persamaan sebagai berikut :

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

$$t_{proses} = 0.00196688 + 0.00196688$$

$$t_{proses} = 0.003933759s$$

Tabel 4.8 berikut merupakan hasil nilai *delay* proses pada masing-masing kapasitas kanal.

Tabel 4.8. Delay Proses

No	Resolusi Video	Delay proses (s)
1	240p	0.003933759
2	360p	0.001950764
3	480p	0.001074656
4	720p	0.000563282

2) *Delay* Propagasi (t_{prop})

$$t_{prop} = \frac{d_{max}}{v} = \frac{10}{3 \times 10^8} = 3,33 \times 10^{-8} \text{ s}$$

Tabel 4.9. berikut merupakan hasil nilai *delay* propagasi pada masing-masing resolusi video.

Tabel 4.9. Delay Propagasi

No	Resolusi Video	Delay Propagasi (s)
1	240p	$3,33 \times 10^{-8}$
2	360p	$3,33 \times 10^{-8}$
3	480p	$3,33 \times 10^{-8}$
4	720p	$3,33 \times 10^{-8}$

3) *Delay* Transmisi (t_{trans})

Panjang *header* (L'), didapatkan

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

$$L' = (20 + 6 + 8 + 12 + 18) \times 8 \text{ bit}$$

$$L' = 512 \text{ bit}$$

Sedangkan diketahui dari data primer sebagai berikut:

- Panjang paket data (L) 240p = 634 bytes
- Kapasitas Kanal (B) = 100 Mbps

Sehingga, perhitungan *delay* transmisi adalah sebagai berikut:

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

$$t_t = \frac{(634 + 512)}{100000000}$$

$$t_t = 0.001146 \text{ sekon}$$

Tabel 4.10. berikut merupakan hasil nilai *delay* transmisi pada masing-masing resolusi video.

Tabel 4.10. Delay Transmisi

No	Resolusi Video	Delay Transmisi(s)
1	240p	0.001146
2	360p	0.001335
3	480p	0.001485
4	720p	0.001612

4) *Delay* Antrean (t_w)

Delay antrian adalah lamanya waktu yang diperlukan paket data untuk diproses pada suatu perangkat, terkait dengan pengiriman dan penerimaan paket data. Perhitungan kecepatan kedatangan paket pada *server* (λ_p) dan kecepatan pelayanan *server* (μ) adalah sebagai berikut:

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

$$\lambda_p = \frac{3080}{60}$$

$$\lambda_p = 51.32310768 \text{ paket/sekon}$$

Sedangkan,

$$\mu = \frac{c}{L_t} \qquad L_t = \frac{w}{N \text{ paket video}} = \frac{3080}{11} = 295 \text{ byte/paket}$$

$$\mu = \frac{10^6}{295}$$

$$\mu = 3389.830508 \text{ paket/sekon}$$

Perhitungan nilai faktor utilitas(ρ) menggunakan persamaan 2.12 adalah sebagai berikut:

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

$$\rho = \frac{51.32310768}{3389.830508}$$

$$\rho = 0.015140317$$

Jadi, perhitungan *delay* antrian adalah sebagai berikut:

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

$$t_w = \frac{1}{3389.830508(1 - 0.015140317)}$$

$$t_w = 0.000299535 \text{ sekon}$$

Tabel 4.11. berikut merupakan hasil nilai *delay* antrean pada masing-masing resolusi video untuk 2 client

Tabel 4.11. Delay Antrean

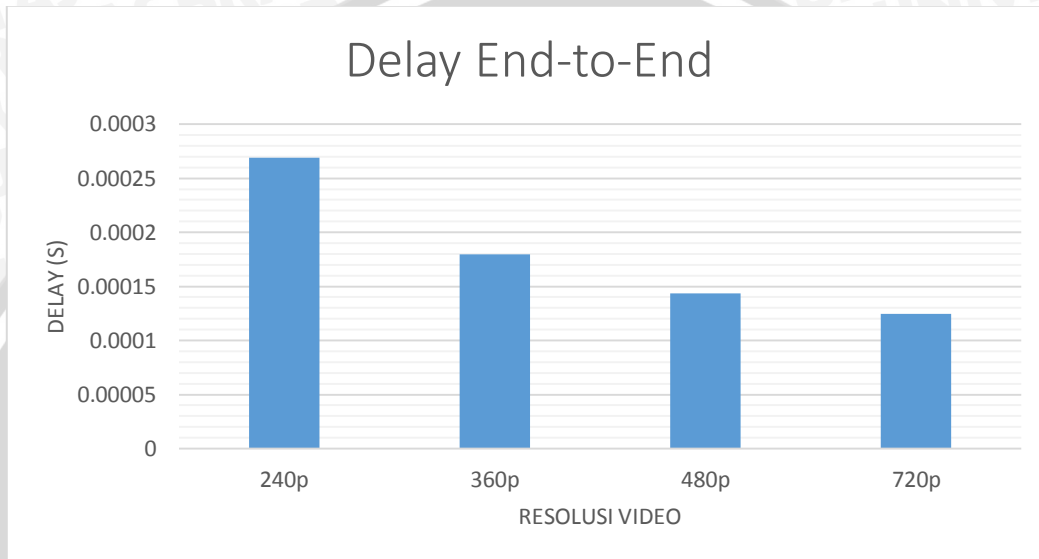
No	Resolusi Video	Delay Antrian (s)
1	240p	0.000299535
2	360p	0.000302105
3	480p	0.000306052
4	720p	0.000314144

Dari data hasil perhitungan *delay* proses, *delay* propagasi, *delay* transmisi dan *delay* antrean maka didapatkan hasil *delay end to end* seperti pada Tabel 4.12. berikut :

Tabel 4.12. Delay End-to-End

No	Resolusi Video	Delay End-to-End (s)
1	240p	0.000268966
2	360p	0.000179395
3	480p	0.000143287
4	720p	0.000124473

Kemudian jika Tabel 4.11. direpresentasikan ke dalam bentuk grafik hubungan antara besar kapasitas kanal dengan *delay delay end to end* yang diterima ditunjukkan pada Gambar 4.15



Gambar 4.15. Pengaruh Resolusi Video Terhadap *Delay End-to-End*

Dari hasil perhitungan diatas, dapat dianalisis bahwa :

1. *Delay* jaringan dalam perhitungan di atas meliputi *delay* proses yang dipengaruhi oleh kecepatan pemrosesan pada pengirim dan penerima, *delay* transmisi yang dipengaruhi oleh kecepatan pemrosesan di setiap node, *delay* propagasi yang dipengaruhi jarak antar node dan media propagasi yang digunakan, serta *delay* antrean yang dipengaruhi oleh faktor utilitas pada jaringan.
2. *Delay end to end* juga dipengaruhi oleh besar resolusi video yang diberikan oleh server. semakin besar resolusi video maka nilai *delay* semakin kecil sehingga waktu keterlambatan pengiriman data dari server menuju client relatif kecil.

4.5.3 Probabilitas Packet Loss

Mencari nilai *payload video* melalui persamaan (2.14)

$$\begin{aligned}
 P_{LV} &= B_{\text{CODEC}} \times \text{frame rate} \\
 &= (628 \cdot 10^3 \text{bps}) \times (30 \cdot 10^3) = 18840 \text{bit} \\
 &= 18840/8 = 2355 \text{byte}
 \end{aligned}$$

Mencari nilai probabilitas *payload* pada layanan *live streaming*.

$$\begin{aligned}
 \rho &= P_{\text{size}} \cdot \rho_b \\
 &= 2355 \times (10^{-10}) \\
 &= 0.0002355
 \end{aligned}$$

Untuk mengetahui probabilitas *packet loss* pada jaringan wireless, dapat ditentukan oleh tipe modulasi yang digunakan pada akses point. Pada penelitian ini, modulasi yang digunakan adalah modulasi 64-QAM mempunyai 6 bit untuk tiap symbol maka jumlah $n=64$ dengan bandwidth kanal (B_w)=3,75 MHz maka nilai bit ratenya.

$$\begin{aligned}
 R &= B_w \text{ kanal} \times \log_2 n \\
 R &= (3,75 \times 10^6) \times \log_2 64 \\
 R &= 22,5 \times 10^6 \text{bps} = 42 \text{ Mbps}
 \end{aligned}$$

Nilai probabilitas packet loss dapat dicari dengan persamaan :

$$\rho_{\text{tot}} = 1 - [(1 - \rho_{\text{net}})(1 - \rho_{\text{vid}})]$$

Dengan ρ_{net} merupakan probabilitas bit error pada akses point dan ρ_{vid} adalah probabilitas *payload* layanan *live streaming* dengan nilai $1,5104 \times 10^{-3}$, maka ρ_{net} dapat dicari dengan persamaan :

$$P_{\text{network}} = \frac{e^{-\frac{E_b}{N_o}}}{2 \sqrt{\pi \left(\frac{E_b}{N_o}\right)}}$$

Untuk mencari nilai $\frac{E_b}{N_o}$ maka terlebih dahulu mencari nilai N_o dan P_r .

$$\begin{aligned}
 \text{Diketahui} &: T_o = 290^\circ \text{K} \\
 &T_{\text{sys}} = 49^\circ \text{C} = 322^\circ \text{K}
 \end{aligned}$$

$$F = \frac{T_o + T_{\text{sys}}}{T_o} = \frac{290 + 322}{290} = 2,11$$

$$NF = 10 \log F = 10 \log 2,11 = 3,24$$

$$\begin{aligned} N_o &= 10 \log kT + 10 \log B + NF \\ &= 10 \log 1,38 \times 10^{-23} \times 290 + 10 \log 2.4 \times 10^9 + 3.24 \\ &= -105,96 \text{ dBm} \end{aligned}$$

N_o merupakan noise level pada akses point dengan nilai -105.96 dBm atau $2.54 \times 10^{-11} \text{ mW}$ dan nilai P_r merupakan daya terima yang didapat oleh penerima sebesar -17.7 dBm atau 0.01699 mW . Untuk mendapatkan nilai $\frac{E_b}{N_o}$ maka perlu dicari nilai SNR dengan menggunakan persamaan 2.15.

$$SNR = 10 \log \frac{P_r}{N_o}$$

$$SNR = 10 \log \frac{0.01699 \text{ mW}}{2.54 \times 10^{-11} \text{ mW}}$$

$$SNR = 88.25 \text{ dB}$$

E_b/N_o dengan panjang jarak akses point ke user 10 m dan bandwidth 2400 MHz. SNR untuk modulasi 64-QAM adalah sebesar 88.25. R adalah bit rate dengan nilai 42 Mbps, sehingga E_b/N_o dapat dihitung dengan persamaan:

$$\frac{E_b}{N_o} = SNR - 10 \log \frac{B}{R}$$

$$\frac{E_b}{N_o} = 88.25 - 10 \log \frac{2.4 \times 10^9}{42 \times 10^6}$$

$$\frac{E_b}{N_o} = 38.22 \text{ dB}$$

Dan $P_{network}$ bisa dihitung dengan rumus :

$$P_{network} = \frac{e^{\frac{E_b}{N_o}}}{2 \sqrt{\pi \left(\frac{E_b}{N_o}\right)}}$$

$$P_{network} = \frac{e^{-38.22}}{2\sqrt{3,14(38.22)}}$$

$$P_{network} = 0.11 \times 10^{-17}$$

Sehingga probabilitas *packet loss* pada layanan *live streaming* pada akses point modem router dengan modulasi 64-QAM dapat dihitung dengan persamaan berikut.

$$\begin{aligned} \rho_{tot} &= 1 - [(1 - \rho_{net})(1 - \rho_{via})] \\ &= 1 - [(1 - 0.11 \times 10^{-17})(1 - 0.0002355)] \\ &= 0.0002355 \end{aligned}$$

Dan nilai prosentase *packet loss* yang didapat adalah sebagai berikut:

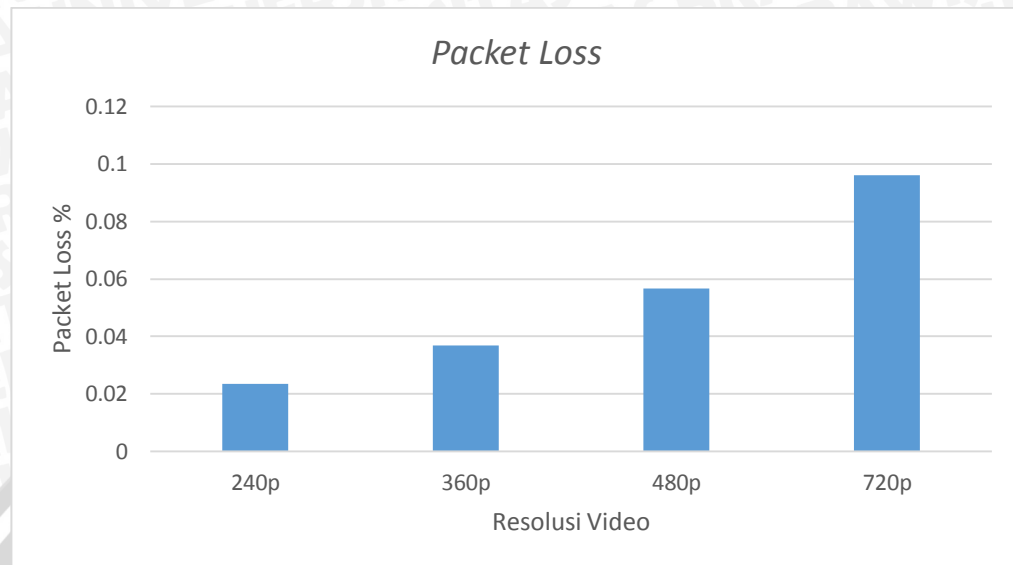
$$\begin{aligned} \rho_{tot}(\%) &= \rho_{tot} \times 100 \% \\ &= 0.0002355 \times 100 \% \\ &= 0.02355 \% \end{aligned}$$

Dari perhitungan, maka didapatkan nilai *packet loss* pada Tabel 4.13. untuk masing-masing resolusi video yaitu :

Tabel 4.13. Probabilitas *Packet loss*

No	Resolusi Video	Packet Loss (%)
1	240p	0.02355
2	360p	0.0368625
3	480p	0.0567375
4	720p	0.0961125

Kemudian jika Tabel 4.13. direpresentasikan ke dalam bentuk grafik hubungan antara besar *resolusi* video dengan probabilitas *packet loss* yang diterima ditunjukkan pada Gambar 4.16.



Gambar 4.16. Pengaruh Resolusi Video Terhadap *Packet Loss*

Dari hasil perhitungan diatas, dapat dianalisis bahwa :

1. *Packet loss* pada perhitungan didapatkan nilai dengan pengaruh resolusi video yaitu semakin besar resolusi video maka *Packet loss* semakin besar. Hal ini juga dipengaruhi nilai Eb/No.
2. Pada hasil perhitungan packet loss didapatkan nilai kategori bagus menurut standar TIPHON yaitu dengan standar 0-3 %

4.5.4 Throughput

Perhitungan *throughput* secara teoritis menurut Persamaan 2.1 adalah sebagai berikut:

- Perhitungan *throughput* pada resolusi 240p, diketahui:

Width: 426 *Frame Rate*: 30

Height: 240 *Motion Rank*: 1

$$\lambda_{FL} = (\text{Width} \times \text{Height} \times \text{Frame Rate} \times 0,07 \times \text{Motion rank}) + \lambda_{FL\text{Audio}}$$

$$\begin{aligned} \lambda_{FL} &= 426 \times 240 \times 30 \times 0,07 \times 1 + 0 \\ &= 214,704 \text{ kbps} \end{aligned}$$

- Perhitungan *throughput* pada resolusi 360p, diketahui:

Width: 640 *Frame Rate*: 30

Height: 360 *Motion Rank*: 1

$$\lambda_{FL} = (\textit{Width} \times \textit{Height} \times \textit{Frame Rate} \times 0,07 \times \textit{Motion rank}) + \lambda_{FL\textit{Audio}}$$

$$\begin{aligned} \lambda_{FL} &= 640 \times 360 \times 30 \times 0,07 \times 1 \\ &= 483,840 \text{ kbps} \end{aligned}$$

- Perhitungan *throughput* pada resolusi 480p, diketahui:

Width: 854 *Frame Rate*: 30

Height: 480 *Motion Rank*: 1

$$\lambda_{FL} = (\textit{Width} \times \textit{Height} \times \textit{Frame Rate} \times 0,07 \times \textit{Motion rank}) + \lambda_{FL\textit{Audio}}$$

$$\begin{aligned} \lambda_{FL} &= 854 \times 480 \times 30 \times 0,07 \times 1 \\ &= 860,832 \text{ kbps} \end{aligned}$$

- Perhitungan *throughput* pada resolusi 720p , diketahui:

Width: 1280 *Frame Rate*: 30

Height: 720 *Motion Rank*: 1

$$\lambda_{FL} = (\textit{Width} \times \textit{Height} \times \textit{Frame Rate} \times 0,07 \times \textit{Motion rank}) + \lambda_{FL\textit{Audio}}$$

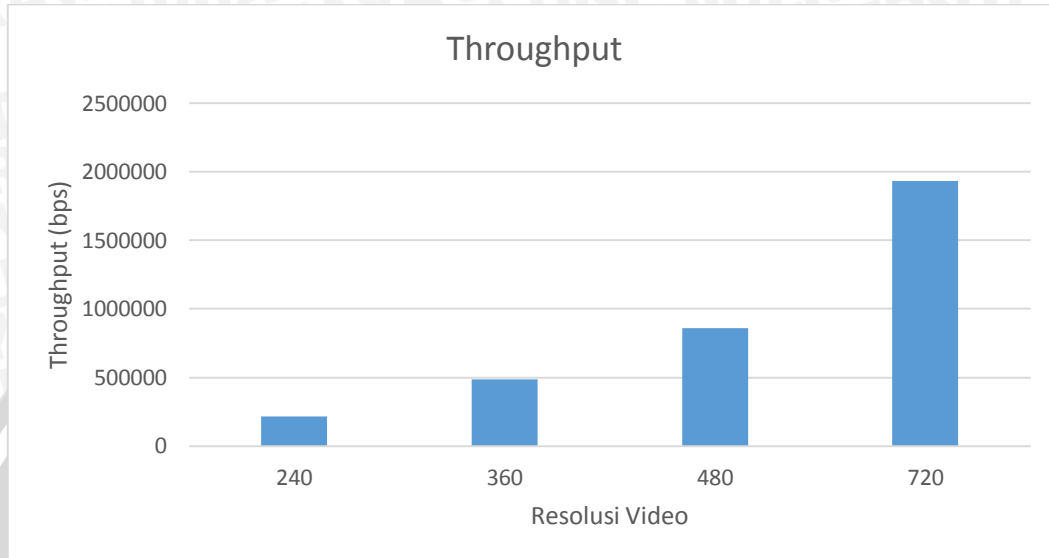
$$\begin{aligned} \lambda_{FL} &= 1280 \times 720 \times 30 \times 0,07 \times 1 \\ &= 1935,360 \text{ kbps} \end{aligned}$$

Dari perhitungan, maka didapatkan nilai *throughput* pada Tabel 4.14. untuk masing-masing resolusi video yaitu :

Tabel 4.14. *Throughput*

No	Resolusi Video	Througput (bps)
1	240p	214704
2	360p	483840
3	480p	860832
4	720p	1935360

Kemudian jika Tabel 4.14. direpresentasikan ke dalam bentuk grafik hubungan antara besar resolusi video dengan *throughput* yang diterima ditunjukkan pada Gambar 4.17.



Gambar 4.17. Pengaruh Resolusi Video Terhadap *Throughput*

Dari perhitungan *throughput* di atas, dapat dianalisis bahwa :

1. *Throughput* dipengaruhi oleh *resolusi* video yang digunakan
2. Semakin besar resolusi video maka nilai *throughput* semakin besar, hal ini mempengaruhi kecepatan akses *point* dalam pengiriman data.

4.6 Hasil dan Analisa

1. Pada analisis simulasi *delay* menggunakan *packet analyser Wireshark* didapatkan nilai tertinggi *delay* yaitu pada *resolusi* 240p, sedangkan nilai *delay* terkecil pada *resolusi* yang paling tinggi yaitu 720p pada 1 *client*. Antara 1 *client* dan 2 *client* pada resolusi 240p,360p,480p nilai *delay* 1 *client* lebih besar daripada 2 *client*, tetapi pada resolusi 720p *delay* 2 *client* lebih besar daripada 1 *client*. Pada pengamatan semakin besar *resolusi* maka nilai *delay* yang terjadi semakin kecil. Nilai *delay* yang semakin kecil disebabkan karena jumlah paket

yang *capture* semakin besar sedangkan waktu yang digunakan selama pengambilan data adalah sama. Nilai *delay* dalam penelitian ini memiliki kualitas yang baik sesuai dengan ITU-T G.114 karena nilai *delay* berkisar antara 0- 150 ms.

2. Pada analisis *throughput* menggunakan *packet analyser Wireshark* dan perhitungan didapatkan perbandingan nilai *throughput* dan *resolusi* menunjukkan, semakin besar *resolusi* maka *throughput* yang dihasilkan akan semakin besar. Hal ini disebabkan jumlah *frame* dan total *bit resolusi* video. Jumlah *frame* dan total *bit* pada *resolusi* yang besar menunjukkan nilai yang besar pula, hal ini yang mengakibatkan nilai *throughput* semakin besar. Jumlah *client* juga mempengaruhi besarnya *throughput*, pada *resolusi* 240p, 360p, dan 480p nilai *throughput* cenderung sama, tetapi pada *resolusi* 720p nilai *throughput* pada 1 *client* lebih besar daripada 2 *client*. Semakin besar *throughput* yang didapatkan maka semakin optimal transmisi data yang dilakukan dalam jaringan.
3. Pada analisis simulasi *Packet loss* menggunakan *packet analyzer Wireshark* didapatkan nilai untuk *packet loss* terendah adalah pada *resolusi* 240p, sedangkan *packet loss* terbesar adalah pada *resolusi* 720p. Antara 1 *client* dan 2 *client* jika dirata-rata secara keseluruhan maka nilai *packet loss* 2 *client* lebih tinggi daripada 1 *client*. Nilai *packet loss* pada *resolusi* yang lebih besar menunjukkan nilai yang besar juga. Hal ini disebabkan oleh kemungkinan jumlah paket yang besar akan kehilangan paket yang besar juga. Dengan *resolusi* video 720p yang merupakan *packet loss* terbesar tapi masih tergolong dikategorikan baik menurut TIPHON yaitu dibawah 3%.