

ANALISIS PERFORMA RANCANG BANGUN SISTEM LIVE STREAMING MENGGUNAKAN SOFTWARE ENCODER VMIX

Rizal Priyambudi^{1*}, Stanislaus Jiwanda Pinasthika²

^{1,2}Universitas Negeri Yogyakarta; Jalan Colombo No.1 Karangmalang Yogyakarta 55281; 0274-586168

Received: 14 Juli 2024
Accepted: 31 Juli 2024
Published: 7 Agustus 2024

Keywords:
Live Streaming, Encoder, Vmix, QoS.

Correspondent Email:
rizalpriyambudi@uny.ac.id

Abstrak. Live streaming merupakan teknologi yang memungkinkan distribusi konten audio dan video secara real-time melalui internet. Pada penelitian ini, akan dilakukan pengujian performansi Quality of Service (QoS) dalam penerapan live streaming pada sistem broadcasting Televisi menggunakan software encoder Vmix. Pengujian dilakukan dengan memberikan variasi nilai bitrate sebesar 500 kbps, 750 kbps, 1000 kbps, dan 1250 kbps selama proses upload video menuju server. kemudian selama proses download konten video dari server, akan diberikan variasi jumlah pengguna dengan jumlah satu, dua, dan tiga. Hasil pengujian menunjukkan bahwa bitrate 1250 kbps memberikan nilai QoS terbaik dengan delay berkisar 5.177 ms, jitter 8.635 ms, packet loss 0.422%, dan throughput 1429.206 kbit. Untuk pengujian proses download, dua pengguna aktif memberikan kualifikasi QoS terbaik dengan nilai delay 5.0988 ms, jitter 8.639 ms, packet loss 0.6022 %, dan throughput 1604.9635 kbit.

Abstract. Live streaming is a technology that allows the distribution of audio and video content in real-time over the internet. In this study, Quality of Service (QoS) performance testing will be conducted in the implementation of live streaming on Television broadcasting systems using Vmix encoder software. Testing is done by giving variations in bitrate values of 500 kbps, 750 kbps, 1000 kbps, and 1250 kbps during the video upload process to the server. Then, during the video content download process from the server, variations in the number of clients will be given with one, two, and three clients. The test results show that a bitrates of 1250 kbps provides the best QoS value with a delay of around 5.177 ms, jitter of 8.635 ms, packet loss of 0.422%, and throughput of 1429.206 kbit. For the download process test, two active clients provide the best QoS qualification with a delay value of 5.0988 ms, jitter of 8.639 ms, packet loss of 0.6022%, and throughput of 1604.9635 kbit.

1. PENDAHULUAN

Layanan *Live Streaming* merupakan teknologi multimedia yang sudah sering dijumpai oleh pengguna internet dimana berbagai jenis konten video dapat diakses secara langsung melalui platform yang berbeda-beda[1]. Mulai dari sosial media seperti youtube, facebook, dan Instagram sudah mempunyai fasilitas untuk melakukan proses *streaming*. Fasilitas *streaming* di sosial media diberikan secara gratis kepada penggunanya, namun

para penggunan juga diberi aturan dalam menggunakan fasilitas tersebut sesuai dengan aturan pengguna (*user policy*) yang sudah ditetapkan masing-masing sosial media yang menyediakan fasilitas *streaming*. Disamping keberadaan sosial media, tedapat juga perusahaan yang bergerak di bidang CDN (*Content Delivery Network*) yang memberikan jasa berlangganan *server* untuk keperluan *streaming* dengan tarif yang sudah ditentukan[2][3].

Dalam penerapannya, konsep *live streaming* yang digunakan oleh Perusahaan besar dibidang *broadcasting* diproses pada suatu MCR (*Master Control Room*) yang memuat berbagai perangkat yang mendukung proses transmisi siaran. Secara umum, MRC terbagi menjadi bagian yang terdiri dari CG (*Character Generator*), CCU (*Camera Control Unit*), Video Server, Vision Mixer/Switcher, DVE (*Digital Effect Video*), Clear Com, VTR (*Video tape Recording*), Patch panel, dan Audio Mixer. Semua komponen dalam MCR tentunya membutuhkan biaya yang besar untuk direalisasikan sehingga tidak semua kalangan mampu untuk membangun fasilitas tersebut.

Berdasarkan pada kondisi tersebut, penulis menawarkan solusi lain dengan menerapkan konsep *live streaming* berbasis perangkat *digital* yang terpasang pada perangkat komputer sehingga akan lebih hemat dalam realisasi sistem dan juga berdampak pada mobilitas yang lebih fleksibel karena berukuran lebih kecil dibandingkan dengan sistem *live streaming* yang berbasis MCR konvensional maupun yang berbasis mobil SNG (*Satellite News Gathering*).

Dalam membangun sistem streaming, protokol *streaming* diperlukan untuk melakukan proses distribusi konten audio dan video. Penentuan jenis protokol yang digunakan disesuaikan dengan jenis *codec* dan *media player* yang akan digunakan untuk menampilkan konten video. salah satu protokol yang sering digunakan untuk melakukan distribusi file audio dan video dalam proses streaming adalah RTMP (*Real Time Messaging Protocol*). RTMP merupakan protokol yang berbasis TCP yang memelihara koneksi persisten dan memungkinkan komunikasi dengan latensi rendah dan mampu untuk diterima oleh decoder yang bersifat *embed player* pada layanan website[3], [4], [5], [6]. Karena memiliki kemampuan untuk diterima oleh *decoder* yang terintegrasi dengan website, RTMP dipilih untuk diterapkan dalam sistem kerja *live streaming* yang akan dibuat.

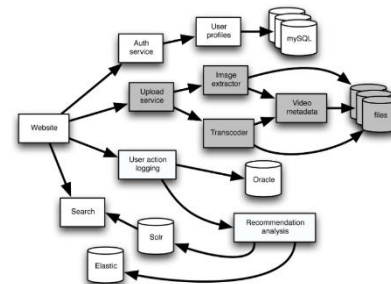
Vmix merupakan sebuah perangkat lunak *broadcasting* yang mampu untuk memberikan layanan siaran konten secara langsung dan juga berbasis rekaman yang dilengkapi dengan fitur *live editing* selama proses siaran berjalan[7]. Sistem streaming yang akan dibuat memanfaatkan software vMix sebagai peralatan *editing* sekaligus *encoder*. Untuk memulai proses siaran langsung, digunakan fasilitas *encoder* pada vMix untuk memulai proses *upload* konten video menuju *server* yang selanjutnya akan diteruskan agar dapat dinikmati oleh pengguna[8]. Untuk dapat menikmati video secara *streaming*, digunakan *video player* yang dapat diakses secara online melalui *website*. Untuk keperluan input, video yang digunakan berasal dari

kamera studio yang akan dikirim secara *wireless* menggunakan *audio video transmitter*. Input akan diterima oleh *software* vMix untuk kemudian diolah dengan fasilitas *live broadcast* sehingga siap untuk ditayangkan secara langsung. Tentunya dalam melakukan pengujian layanan *streaming* yang membutuhkan konektivitas untuk pengiriman data dalam jumlah besar diperlukan adanya pengukuran QoS (*Quality of Service*) untuk dapat mengetahui kemampuan sistem yang sudah dirancang[9][10].

2. TINJAUAN PUSTAKA

2.1. Video Streaming

Video streaming merupakan teknologi pengiriman data baik video maupun audio dalam bentuk yang telah dikompresi melalui jaringan internet untuk ditampilkan pada suatu video player secara realtime yang bekerja seperti pada gambar 2.1. Terdapat beberapa tipe video streaming yang meliputi webcast yang dapat ditampilkan secara langsung (*live*) dan Video on Demand (VOD) yang melakukan penyimpanan video didalam server terlebih dahulu sebelum ditampilkan.



Gambar 2.1. Aliran Data Streaming

2.2. Vmix Live Video Streaming

Vmix merupakan *software* video *mixer* yang memungkinkan melakukan produksi tv hingga melakukan penyiaran konten. vMix menawarkan beberapa keunggulan dengan mampu melakukan aktifitas penyiaran yang umumnya dilakukan oleh perangkat-perangkat yang terdapat didalam MCR. Karena berbentuk *software* dan mampu melakukan pekerjaan dari kumpulan perangkat didalam MCR, maka secara mobilitas vMix lebih fleksibel dibandingkan sistem MCR konvensional. vMix mampu memberikan dukungan terhadap *input* video secara bersamaan dari beberapa lokasi yang berbeda, mampu melakukan proses perekaman, *streaming*, serta memberikan *output* pada monitor. Selain itu juga mampu menambahkan efek transisi, *chroma key* dan memberikan *title* secara *live*.

2.3. Nginx dan RTMP Module

Nginx merupakan server HTTP dan proxy yang bersifat *opensource* yang juga bisa dimanfaatkan sebagai proxy IMAP/POP3. Nginx dirancang dan

dikembangkan oleh Igor Sysoev mulai tahun 2002 dan dirilis ke publik pada tahun 2004. Beberapa kelebihan yang ditawarkan adalah, kemudahan dalam melakukan instalasi, dapat beroperasi di banyak platform sistem operasi, mudah dikonfigurasi, memiliki fasilitas keamanan yang cukup lengkap, dapat dikombinasikan dengan modul yang sudah disediakan, konsumsi sumber daya yang tidak boros, dan efisien dalam menampilkan konten yang sifatnya statik. Nginx juga dilengkapi dengan fitur reverse proxy multiple protocols (HTTP, Memcached, PHP-FPM, SCGI, uwsgi), Stream HTTP video (FLV, HDS, HLS, MP4) serta HTTP/2 gateway. Beberapa situs terkenal yang menggunakan Nginx adalah Wordpress, Fastmail, Ohloh, Sourceforge, dan github.

2.4. Quality of Service

Quality of Service (QoS) merupakan sekumpulan metode dan mekanisme yang digunakan untuk menjamin performa suatu jaringan (terutama jaringan internet) dalam penyediaan layanan kepada aplikasi – aplikasi di dalam jaringan tersebut[10]. Quality of Service dilihat dan diukur berdasarkan sudut pandang penyedia layanan. Sedangkan QoE (Quality of Experience) dinilai dari sudut pandang pengguna tersebut. QoS pada bidang telekomunikasi didefinisikan melalui beberapa parameter, yaitu efisiensi bandwidth, throughput, *rasio packet loss*, *bit error rate*, *collision probability*, *jitter* dan *delay*. Terdapat beberapa faktor yang dapat menurunkan nilai QoS, seperti: redaman, distorsi, dan *noise*. Terkait dengan layanan kualitas pada suatu jaringan, dibutuhkan adanya delay dan jitter yang lebih kecil, bahkan jika memungkinkan tidak ada delay dan jitter sama sekali.

3. METODE PENELITIAN

Penelitian dilaksanakan dengan menerapkan metode prototipe dalam pembuatan sistem. Pengembangan sistem melalui beberapa tahapan yang dimulai dengan menentukan spesifikasi dari sistem, Perakitan, dan validasi data. Berikut adalah detail dari setiap tahapan yang dilalui:

3.1. Spesifikasi Sistem

Sistem *live streaming* yang dibuat memiliki beberapa spesifikasi kerja yang akan dijelaskan sebagai berikut.

- 3.1.1. Mampu menerima *input* dari 3 kamera sebagai *source* untuk menampilkan tayangan video yang berbeda secara bersamaan.
- 3.1.2. Menggunakan tambahan relay server untuk dapat membagi koneksi sehingga

live streaming dapat dilakukan melalui jaringan lokal maupun jaringan internet.

- 3.1.3. Output yang diberikan memiliki resolusi 720x576 widescreen dengan aspect ratio 16:9.
- 3.1.4. Menggunakan protokol RTMP untuk melakukan proses upload konten video menuju media server.

3.2. Perancangan Sistem

Live streaming yang dibuat terbagi menjadi empat komponen utama yang mendukung proses penyiaran konten video. Komponen tersebut meliputi *media source*, *live streaming software*, *server streaming*, dan *web streaming*. Setiap komponen tersebut akan dihubungkan dengan Modem GSM untuk mendapatkan akses internet. Diagram blok sistem *streaming* dapat dilihat pada Gambar 3.1.

Komponen pertama adalah *media source* yang terdiri dari tiga buah kamera ENG (*Electronics News Gathering*). Konten yang diterima oleh setiap kamera akan didistribusikan secara wireless menggunakan *AV Sender* sehingga dapat diterima oleh *vmix* (*live video streaming software*). Dikarenakan *AV sender* yang digunakan memiliki kabel dengan tipe RCA (*Radio Corporation of America*), diperlukan adanya *converter* untuk menghubungkan *AV sender* dengan perangkat *live video streaming software* yang berbasis USB sehingga konten video dapat diterima dengan baik untuk dapat dilakukan proses *mixing* dan *streaming*. Detail skema input ditunjukkan Gambar 3.2.

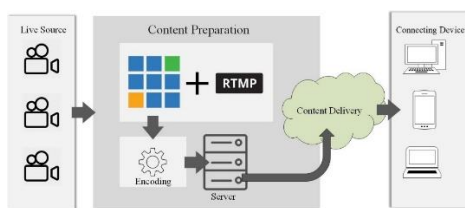
Setelah dilakukan perekaman input video, proses selanjutnya adalah *mixing* menggunakan *software vmix*. Untuk mendukung proses *mixing*, *vmix* dilengkapi dengan perangkat komputer dengan spesifikasi yang sudah ditentukan sehingga fitur *vmix* dapat digunakan secara maksimal. Fitur yang akan digunakan meliputi pengaturan aspect ratio 16:9 dengan resolusi *widescreen* serta penambahan *zoom* baik secara vertikal maupun horizontal dengan skala tertentu sehingga konten yang ditampilkan dapat memenuhi area *video player*. Selain itu, digunakan juga fitur *stream* untuk mengatur alur transmisi video menuju *website*.

Fitur *stream* pada *software vmix* merupakan perwujudan perangkat *encoder* yang melakukan proses *encoding* video sehingga dapat ditransmisikan dalam bentuk bit-bit data untuk kemudian dikirim menuju server streaming yang akan meneruskan konten video menuju *video player* yang sudah disiapkan. dalam proses transmisi video, digunakan protokol RTMP yang merupakan salah satu protokol *streaming* yang berbasis TCP. Penggunaan protokol RTMP ditandai dengan penggunaan RTMP URL yang berisikan alamat IP

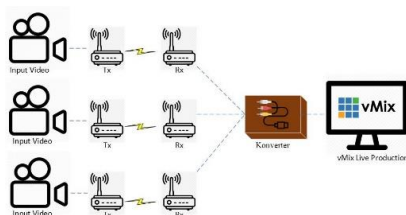
dari *server streaming* dan *Stream key* yang berisi kode verifikasi dalam melakukan transmisi[5]. Penggunaan kode verifikasi tersebut bertujuan melakukan sinkronisasi antara *encoder* dengan *decoder*.

Setelah melalui proses stream dari Vmix, bit-bit data akan diterima oleh *server streaming*. *Server streaming* merupakan sebuah *web server* ataupun aplikasi yang terdapat dalam suatu perangkat *server* yang bertujuan untuk mendistribusikan konten audio dan video secara *real-time* di internet. Dari pilihan *server streaming* yang ada, NGINX merupakan pilihan yang akan digunakan pada penelitian ini. Untuk dapat melakukan distribusi data video, NGINX dilengkapi dengan modul RTMP yang telah disesuaikan dengan protokol *streaming* yang akan digunakan. Untuk dapat menjalankan *web server* tersebut, akan digunakan sistem operasi *raspbian* yang telah terinstall dalam perangkat *single-board circuit raspberry pi 3*[11]. Raspberry pi 3 memiliki kemampuan sebagai *server* bagi konten yang akan ditransmisikan pada bagian *output* dari sistem yang menuju kearah pengguna[12]. Skema *output stream* dapat dilihat secara lebih detail pada Gambar 3.3.

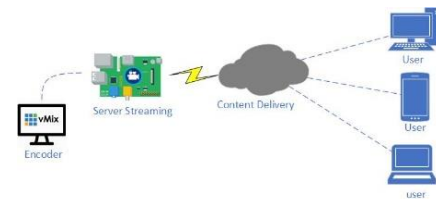
Untuk dapat mengakses video, telah disediakan *website* yang didalamnya terdapat video player yang dapat melakukan proses *decoding* packet data video yang diterima dari *server*. Proses *decoding* yang dilakukan berkebalikan dari proses *encoding* dimana tiap bagian dari packet data yang diterima akan disatukan kembali menjadi suatu file informasi yang utuh sehingga dapat dinikmati oleh pengguna. Detail lengkap dari proses pengiriman konten *streaming* hingga dapat diterima pengguna dapat dilihat pada gambar 3.1.



Gambar 3.1. Diagram Blok Sistem



Gambar 3.2. Skema Input Video



Gambar 3.3. Skema Output Stream

3.3. Validasi Data

Pada penelitian ini, digunakan alat ukur berupa perangkat lunak *network analyzer* yaitu *wireshark* untuk memonitor beberapa parameter seperti *throughput*, *delay*, *jitter*, dan *packet loss* pada jaringan yang digunakan baik untuk *upload* dan *download*[13]. Pengujian kinerja sistem *live streaming* dilakukan untuk mengukur performa jaringan selama proses pengiriman data video dari PC vMix hingga dapat diterima oleh pengguna yang terhubung. Pengiriman data yang dilakukan berupa pemutaran video dengan durasi 2 menit. Dalam melakukan proses pengujian, digunakan perangkat lunak *wireshark* untuk memantau lalu lintas data yang terjadi[14]. Pemantauan jaringan yang dilakukan terbagi menjadi dua tahapan yang tahap pengujian aktivitas *upload* dari *encoder* menuju *server* dengan memberikan variasi *bitrate* dan pengukuran aktivitas *download* dengan adanya variasi jumlah pengguna yang melakukan akses menuju *website streaming*.

Pengujian proses *upload* video dilakukan dengan menjalankan *wireshark* pada perangkat *encoder*. Dari perangkat *encoder* tersebut, data video akan di-upload menuju *server*. Dengan adanya video yang dikirim menuju *server*, menunjukkan adanya komunikasi dua arah antara perangkat *encoder* dengan *server*. Selama proses pengujian, akan dilakukan perubahan nilai dari *bitrate* video dengan empat variasi nilai yang berkisar dari 500 kbps hingga 1250 kbps. Adanya variasi nilai *bitrate* akan berpengaruh terhadap jumlah *packet data* yang dikirimkan selama proses pengiriman video. berdasarkan pada jumlah *packet data* yang terkirim, akan dilakukan analisis terhadap parameter QoS (*Quality of Service*) yang meliputi data *throughput*, *delay*, *jitter*, dan *packet loss* yang nilainya dihitung berdasarkan standar dari TIPHON[15]. Berdasarkan pada hasil perhitungan data, nilai masing-masing parameter akan dikategorikan berdasarkan standar[16].

Parameter *throughput* bertujuan mengukur kinerja jaringan berdasarkan kemampuan dalam melakukan transfer data antara dua titik dari jaringan. Perhitungan *throughput* dilakukan dengan mengikuti persamaan (1). Sebagai pembanding, hasil perhitungan *throughput* akan dibandingkan dari standar QoS yang terdapat pada Tabel 3.1.

$$\text{Throughput} = \frac{\text{Jumlah data yang dikirim}}{\text{waktu pengiriman data}} \quad (1)$$

Tabel 3.1. Kategori *Throughput*

kategori	<i>Throughput</i>
<i>Excellent</i>	76-100 %
<i>Good</i>	51-75 %
<i>Poor</i>	26-50 %
<i>Unacceptable</i>	0-25 %

Pada tahap selanjutnya, proses pengukuran *delay* yang merupakan parameter jaringan yang berfungsi untuk melakukan analisa durasi waktu tempuh dari data untuk dapat dikirimkan dari titik awal transmisi hingga sampai pada bagian penerima. *Delay* dapat dihitung menggunakan persamaan (2). Data nilai *delay* tersebut akan dikategorikan berdasarkan nilai dari Tabel 3.2.

$$\text{Delay} = \text{Waktu paket diterima} - \text{waktu dikirim} \quad (2)$$

Tabel 3.2. Kategori *Delay*

kategori	<i>Delay</i>
<i>Excellent</i>	< 150 ms
<i>Good</i>	150 to 300 ms
<i>Poor</i>	300 to 450 ms
<i>Unacceptable</i>	> 450 ms

Parameter ketiga adalah *packet loss* yang didefinisikan sebagai jumlah kegagalan selama proses transmisi data berlangsung berdasarkan persamaan (3). Data *packet loss* tersebut akan dikelompokkan berdasarkan data yang terdapat pada Tabel 3.3.

$$\text{Packet loss} = \left(\frac{\text{paket dikirim} - \text{paket diterima}}{\text{paket yang dikirim}} \right) \times 100\% \quad (3)$$

Tabel 3.3. Kategori *Packet Loss*

kategori	<i>Packet Loss</i>
<i>Excellent</i>	0 %
<i>Good</i>	1-3 %
<i>Poor</i>	4-15 %
<i>Unacceptable</i>	125 s/d 225 %

pengambilan data diakhiri dengan perhitungan nilai *jitter* yang menunjukkan perbedaan waktu pengiriman dari setiap data transmisi atau dapat diartikan sebagai variasi nilai *delay* yang terjadi. Nilai *jitter* dapat diperoleh dengan mengikuti persamaan (4). Hasil pengamatan *jitter* akan disesuaikan kategorinya berdasarkan informasi pada Tabel 4.4.

$$\text{Jitter} = \frac{\text{total variasi delay}}{\text{Total paket yang diterima} - 1} \quad (4)$$

Tabel 4.4. Kategori *Jitter*

kategori	<i>Jitter</i>
<i>Excellent</i>	0 ms
<i>Good</i>	0 to 75 ms
<i>Poor</i>	76 to 125 ms
<i>Unacceptable</i>	125 to 225 ms

4. HASIL DAN PEMBAHASAN

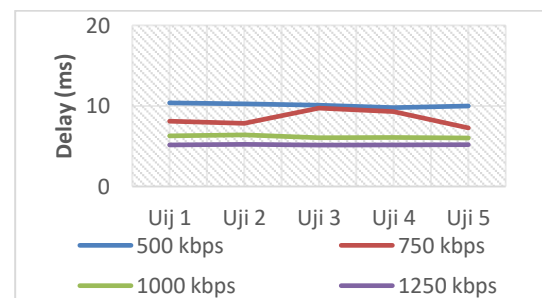
Berdasarkan data yang sudah diperoleh selama proses pengujian, dapat dilakukan analisis terhadap performa dari sistem *live streaming*. Selama proses *streaming* dijalankan, variasi nilai *bitrate* dan penambahan jumlah pengguna memberikan pengaruh terhadap parameter QoS pada saat melakukan proses *upload* maupun *download* konten video.

4.1. Hasil Data Upload

Proses *upload* yang dimaksudkan merupakan waktu ketika *encoder* mengirimkan packet data menuju *server media streaming*. Selama proses *transfer* data tersebut, diperoleh data terkait parameter QoS yang dapat ditampilkan sebagai berikut.

4.1.1. Delay

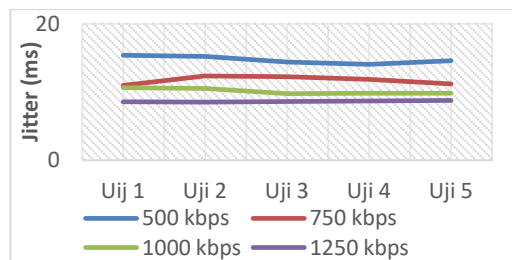
Dari gambar 4.1., dapat disimpulkan bahwa penambahan nilai *bitrate* berbanding terbalik dengan penambahan nilai *delay*. Nilai *bitrate* menunjukkan banyaknya jumlah packet yang dapat ditransfer dari *encoder* menuju *server* setiap detiknya. Oleh karena itu, semakin besar *bitrate* yang digunakan, semakin banyak packet yang dapat ditransfer dalam hitungan detik. Apabila kapasitas packet yang dikirim lebih besar, waktu pengiriman packet dapat dipersingkat sehingga *delay* yang terjadi akan lebih kecil. Terbukti bahwa *bitrate* 500 kbps memiliki *delay* dengan rata-rata 10.103 ms, sedangkan *bitrate* 1250 kbps memberikan waktu *delay* tersingkat dengan nilai rerata 5.177 ms.



Gambar 4.1. Delay Upload

4.1.2. Jitter

Analisis *jitter* yang dilakukan bertujuan untuk mengetahui nilai dari variasi *delay* yang diperoleh oleh sistem *live streaming* selama proses upload berjalan. Untuk mempermudah dalam melakukan analisis, berikut tampilan data *jitter* pada Gambar 4.2. yang ditampilkan dalam bentuk grafik.

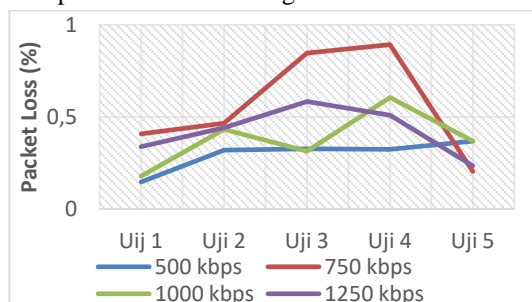


Gambar 4.2. jitter Upload

Analisis *jitter* masih memiliki keterkaitan dengan adanya *delay*. *Jitter* merupakan variasi *delay* yang terjadi dari sistem yang dibuat. Jadi, nilai *jitter* juga dipengaruhi oleh besarnya *bitrate* yang digunakan. Angka *bitrate* 500 kbps menunjukkan nilai *jitter* terbesar selama pengujian. Kondisi tersebut dapat diartikan bahwa variasi *delay* ketika *bitrate* 500 kbps digunakan bernilai tinggi. Dengan perlakuan yang hampir sama ketika melakukan analisis terhadap *delay*, dapat dijelaskan bahwa ketika *bitrate* bernilai 500 kbps, jumlah bit data yang ditransfer memiliki nilai terkecil sehingga proses pengiriman data menuju *server* membutuhkan waktu lebih lama dibandingkan ketika penggunaan *bitrate* 750 kbps, 1000 kbps, dan 1250 kbps. Akibatnya, *delay* semakin besar dan memberikan nilai *jitter* yang meningkat. Dari *delay* tersebut, tentunya memberikan dampak terhadap *jitter* yang juga bernilai semakin besar.

4.1.3. Packet Loss

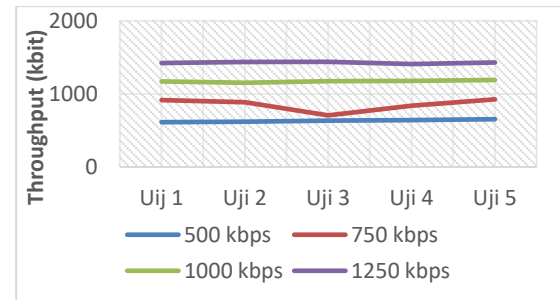
Packet Loss menampilkan banyaknya jumlah packet data yang gagal untuk ditampilkan yang disebabkan hilangnya packet selama proses transmisi data terjadi. Berikut detail data data pengujian *packet loss* pada Gambar 4.3. yang ditampilkan dalam bentuk grafik.



Gambar 4.3. Packet Loss Upload

4.1.4. Throughput

Nilai *throughput* menunjukkan *bandwidth* aktual yang dimiliki sebuah jaringan internet ketika sedang melakukan komunikasi maupun transfer data. Dalam proses upload sistem *live streaming*, diperoleh nilai *throughput* dengan detail pada Gambar 4.4.



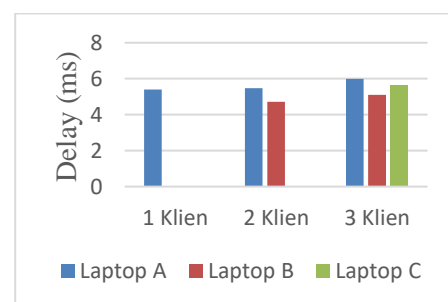
Gambar 4.4. Throughput Upload

4.2. Hasil Data Download

Proses download yang dimaksudkan merupakan aktivitas pengguna yang melakukan akses *streaming* melalui *website* yang telah disiapkan. Jumlah pengguna yang digunakan juga bervariasi dari satu pengguna hingga maksimal tiga pengguna. Dari masing-masing kondisi yang diberikan, akan dilakukan analisis terhadap parameter QoS yang sudah ditentukan.

3.1.1. Delay

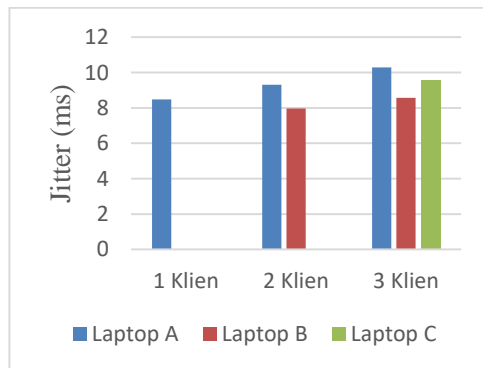
Berdasarkan data hasil perhitungan *delay* pada proses *download*, berikut tampilan data setelah diubah dalam bentuk grafik yang ditampilkan pada Gambar 4.5. *Delay* yang terjadi selama proses download *streaming* mengalami peningkatan yang tidak terlalu signifikan. Peningkatan terjadi ketika diberikan pengguna tambahan di waktu yang bersamaan. Dengan adanya penambahan pengguna, maka beban *traffic* yang dihasilkan akan meningkat dikarenakan harus memberikan akses kepada lebih banyak pengguna. Terbukti bahwa ketika terdapat tiga pengguna yang terhubung secara langsung *delay* menunjukkan nilai yang lebih tinggi dengan rata-rata 5.659 ms.



Gambar 4.5. Delay Download

3.1.2. Jitter

Untuk mengetahui pengaruh penambahan pengguna terhadap *jitter* yang dihasilkan, berikut grafik pengambilan data yang diperoleh ketika melakukan pengujian.



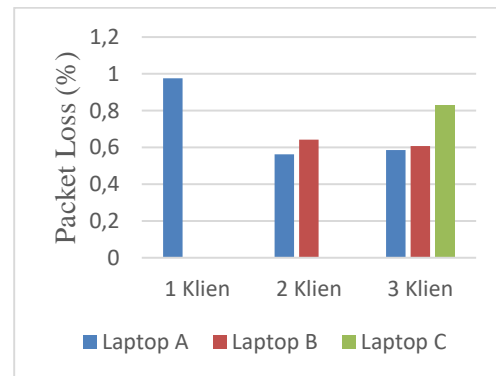
Gambar 4.6. Jitter Download

Nilai *jitter* yang ditunjukkan oleh gambar 4.6. menunjukkan pola yang sama dengan nilai *delay* pada gambar 4.5. hal ini terjadi dikarenakan *jitter* merupakan hasil perhitungan dari variasi *delay* yang ada. Maka dari itu, data *jitter* yang diperoleh dapat dinyatakan mengacu pada data *delay* yang telah didapat sehingga *jitter* akan mengalami peningkatan apabila jumlah pengguna yang ada ditambah. Terbukti bahwa penggunaan tiga pengguna menunjukkan nilai grafik yang lebih tinggi dibandingkan dengan variasi jumlah pengguna yang lain.

3.1.3. Packet Loss

Packet loss yang diperoleh merupakan bentuk kegagalan pengiriman packet data sehingga terdapat sebagian informasi yang tidak dapat ditampilkan. Berikut detail dari analisa *packet loss* pada sistem live streaming yang ditunjukkan pada Gambar 4.7.

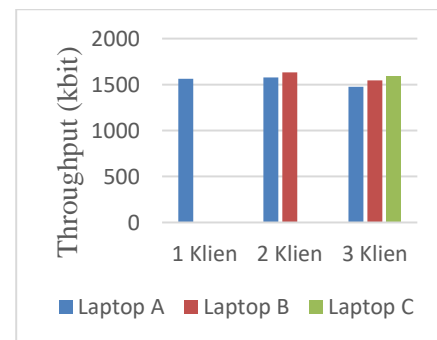
Gambar 4.7. menampilkan nilai packet loss ketika diberikan variasi jumlah pengguna pada sistem *live streaming*. Rata-rata packet loss terbesar diperoleh ketika hanya ada satu pengguna yang melakukan akses *streaming*. Hal tersebut disebabkan seluruh *traffic* yang ada hanya melayani satu pengguna sehingga jumlah packet data yang ditransfer paling banyak dibanding kondisi pengguna yang lain. Jika jumlah *packet data* yang dikirim semakin banyak, maka peluang terjadinya *loss* akan semakin banyak.



Gambar 4.7. Packet Loss Download

3.1.4. Throughput

Ketika melakukan *streaming*, *traffic* yang ada akan terbagi sejumlah pengguna yang terhubung, berdasarkan pembagian *traffic* tersebut, setiap pengguna akan memiliki nilai *throughput* tersendiri untuk melakukan proses *download* konten *streaming*, berikut tampilan data *throughput* yang diperoleh selama proses pengujian.



Gambar 4.8. Throughput Download

Berdasarkan data gambar 4.8, *throughput* dari setiap variasi pengguna yang diberikan memiliki kisaran angka yang sama. Hal ini dikarenakan pada saat proses *download* data video dari *server* menuju pengguna, jumlah bit data yang dikirim memiliki jumlah yang sama untuk setiap pengguna yang terhubung.

5. KESIMPULAN

Dari hasil pengujian dan analisis sistem pada penelitian ini dapat diambil beberapa kesimpulan sebagai berikut:

- Delay* yang diperoleh selama proses *upload* data video menuju *server* memiliki nilai yang berbanding terbalik dengan penambahan *bitrate* yang dilakukan. Ketika *bitrate* yang digunakan memiliki nilai tertinggi yaitu 1250 kbps, *delay* yang dihasilkan

- menunjukkan angka paling kecil dengan nilai rata-rata 4.306 ms.
- b. *Delay* yang diperoleh pada proses *download* data video oleh pengguna memiliki nilai yang berbanding lurus dengan penambahan pengguna. Ketika pengguna yang melakukan akses mencapai angka terbesar yaitu tiga pengguna, *delay* yang dihasilkan menunjukkan nilai rata-rata terbesar yaitu 5.582 ms.
 - c. *Jitter* ketika proses *upload* data video akan semakin mengecil bersamaan dengan penambahan *bitrate* yang dilakukan. hal ini dibuktikan pada saat *bitrate* bernilai 1250 kbps, *jitter* yang dihasilkan memiliki nilai rata-rata 7.015 ms.
 - d. *Jitter* pada saat proses *download* meningkat bersamaan dengan penambahan jumlah pengguna. Terlihat ketika tiga pengguna terhubung, *jitter* menunjukkan angka rata-rata terbesar 9.469 ms.
 - e. *Packet loss* ketika proses *upload* memiliki nilai yang bervariasi dan cenderung naik ketika *bitrate* yang digunakan semakin besar.
 - f. *Packet loss* ketika melakukan proses *download* bernilai kecil ketika jumlah pengguna semakin banyak. Ketika pengguna yang melakukan aktivitas streaming berjumlah tiga, *packet loss* memiliki nilai rata-rata 0.428%
 - g. Nilai *throughput* pada proses *upload* video akan meningkat bersamaan dengan meningkatnya nilai *bitrate*. Hal ini ditunjukkan ketika *bitrate* 1250 kbps, nilai *throughput* berada pada kisaran 15048 kbit.
 - h. *Throughput* ketika proses *download* data video berkisar pada angka 1400-1600 kbit pada masing-masing pengguna dikarenakan jumlah data yang akan diterima oleh masing-masing pengguna dari *server* kurang lebih sama.

Berdasarkan poin kesimpulan yang telah dibahas secara detail tersebut, apabila data yang diperoleh dibandingkan dengan standar yang dikeluarkan oleh TIPHON, maka dapat disimpulkan bahwa sistem yang dirancang masuk dalam kategori *excellent* pada seluruh parameter QoS yang diujikan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] A. Yuhaneef, S. Aulia, and R. Fala, "Analisis Kualitas Jaringan 4g Pada Video Streaming Di Daerah Lubuk Minturun," *JIKO (Jurnal Inform. dan Komputer)*, vol. 8, no. 1, p. 137, Feb. 2024, doi: 10.26798/jiko.v8i1.1068.
- [2] A. M. Risayaf Fawwaz Pradipta, Rendy Munadi, "Analisis Komparasi Performa Content Delivery Network (Cdn) Dalam Implementasi Video On Demand Dan Live Server Berbasis Teknologi Cloud Computing Comparative Analysis Of Content Delivery Network (Cdn) Performance In Video On Demand And Live Server Imple.," *e-Proceeding Eng.*, vol. 8, no. 6, pp. 2639–2649, 2022.
- [3] T. Kumar *et al.*, "Cloud-based video streaming services: Trends, challenges, and opportunities," *CAAI Transactions on Intelligence Technology*, vol. 9, no. 2. John Wiley and Sons Inc, pp. 265–285, Apr. 01, 2024. doi: 10.1049/cit2.12299.
- [4] F. I. Winarto, B. Irawan, R. E. Saputra, F. Teknik, and U. Telkom, "Analisis Performansi Rtmp Live Streaming Server Berbasis Raspberry Pi Untuk Video Surveillance System," *E-proceeding Telkom Univ.*, vol. 3, no. 2, pp. 2163–2171, 2016.
- [5] S. Kirve, R. Waghela, K. More, T. Prasade, and M. Patil, "Live Streaming Website Using WebRTC And Rtmp," *Int. J. Progress. Res. Eng. Manag. Sci.*, vol. 4, no. 4, pp. 978–980, 2024, doi: 10.58257/IJPREMS33374.
- [6] O. Kryvoshei, P. Kamencay, and A. Stech, "Server Implementation Design for Adaptive 3D Video Streaming," *2024 34th Int. Conf. Radioelektronika*, pp. 1–6, 2024, doi: 10.1109/radioelektronika61599.2024.10524068.
- [7] L. yudi Haryanto, A. Hayat, and A. H. Arribathi, "Multicam Studio Design Using Vmix As A Learning Media In SMK Bina Am Ma'mur," *ADI J. Recent Innov.*, vol. 3, no. 1, pp. 1–8, Jun. 2021, doi: 10.34306/ajri.v3i1.331.
- [8] U. M. W. Wahyudi, "Enhancing Student Motivation in Virtual Classrooms: the role of vMix technology," *Indones. J. Curric. Educ. Technol. Stud.*, vol. 11, no. 1, Apr. 2023, doi: 10.15294/ijcets.v11i1.67524.
- [9] E. Rosiska, V. Karnadi, and S. Ani Arnomo, "Analisis Qos Video Streaming Jaringan Wireless (Studi Kasus: Taman Internet Engku

- Putri Batam,” *Comput. Based Inf. Syst. J.*, vol. 08, no. 02, pp. 46–54, 2020, [Online]. Available: <http://ejournal.upbatam.ac.id/index.php/cbishttp://ejournal.upbatam.ac.id/index.php/cbis>
- [10] D. A. Rachman, Y. Muhyidin, and M. A. Sunandar, “Analysis Quality of Service of Internet Network Fiber To the Home Service Pt. Xyz Using Wireshark,” *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3s1, pp. 997–1006, 2023, doi: 10.23960/jitet.v11i3s1.3436.
- [11] B. She, Q. Wang, X. Zhong, Z. Zhang, Z. Qin, and G. Li, “The Design and Implementation of Campus Network Streaming Media Live Video On-Demand System Based on Nginx and FFmpeg,” *J. Phys. Conf. Ser.*, vol. 1631, no. 1, pp. 1–7, 2020, doi: 10.1088/1742-6596/1631/1/012158.
- [12] V. Vatmala, B. Maruddani, A. Diamah, and A. Jaenul, “Analisis Desain Dan Kualitas Layanan Video Streaming Berbasis Raspberry Pi Server Pada Jaringan Wireless Local Area Network (Wlan),” in *9th Applied Business and Engineering Conference*, 2021, pp. 674–686.
- [13] S. Tari, “Analisis Quality Of Service (QoS) Radio Streaming Menggunakan Icecast Pada Wi-Fi 802.11n,” *J. List. Telekomun. Elektron.*, vol. 20, no. 1, pp. 2549–8762, 2023.
- [14] Barawinfried, “Performance Analysis Of Live Streaming Services In The Vision Plus Application With Quality Of Service Parameters Using Wi-Fi 5 And Wi-Fi 6 Network Standards,” Banyumas.
- [15] H. S. Widhiatmoko H.P, Oki Sri Linangkung, “Analisis Kualitas Layanan Jitter Video Streaming Pada Aplikasi Disney+ Hotstar Menggunakan Jaringan Wi-Fi,” *Pros. Semin. Nasional LPPM UNSOED*, vol. 13, no. 1, pp. 286–290, 2024.
- [16] M. R. Hidayat and T. F. Shantony, “Analisis Jaringan Wireless Local Area Network (Wlan) Di Gedung Teknik Elektro Universitas Jenderal Achmad Yani Untuk Layanan Video Live Streaming Youtube Analysis of Wireless Local Area Network (Wlan) At the Electrical Engineering Building of the Univ,” *J. Elektro Telekomun. Terap.*, vol. 8, no. 2, pp. 1080–1090, 2021, doi: 10.25124/jett.v8i2.4145.