



## Analisis Penggunaan Protokol SRT dalam Pengiriman Audio dan Video di TVRI Kalimantan Timur

Kevin Adam Firdaus Halomoan Nasution<sup>1\*</sup>, Muhammad Fahmi<sup>2</sup>, Kusno Harianto<sup>3</sup>

<sup>1-3</sup>Program Studi Sistem Informasi, STMIK Widya Cipta Dharma, Indonesia

Email: [2241024@wicida.ac.id](mailto:2241024@wicida.ac.id), [mfahmi@wicida.ac.id](mailto:mfahmi@wicida.ac.id), [kusno@wicida.ac.id](mailto:kusno@wicida.ac.id)

\*Penulis Korespondensi: [2241024@wicida.ac.id](mailto:2241024@wicida.ac.id)

**Abstract:** *Live streaming has become an increasingly important technology in digital broadcasting, particularly for real-time transmission that requires high quality and low latency. The selection of video and audio transmission protocols plays a crucial role in determining the performance and stability of a broadcast system. Real-Time Messaging Protocol (RTMP) is still widely used; however, it has limitations when operating under unstable network conditions. As an alternative, Secure Reliable Transport (SRT) was developed to improve the reliability of multimedia data transmission through error correction mechanisms and adaptive latency management. This study aims to implement live streaming using the vMix software with the SRT protocol and to analyze its performance by comparing it with the RTMP protocol. The research method employed is an experimental approach, in which both protocols are tested and evaluated based on latency, video and audio quality, connection stability, and resilience to packet loss. The experiments are conducted under varying network conditions to simulate real-time broadcasting scenarios. The results indicate that the SRT protocol outperforms RTMP in terms of lower latency, higher streaming stability, and better data recovery under unstable network conditions. The implementation of SRT in a vMix-based live streaming system can be considered an effective solution for delivering high-quality real-time broadcasts.*

**Keywords:** *Live Streaming; Real-Time Broadcasting; RTMP; SRT; VMix.*

**Abstrak:** *Live streaming merupakan teknologi yang semakin penting dalam penyiaran digital, khususnya untuk kebutuhan siaran real-time yang menuntut kualitas tinggi dan latensi rendah. Pemilihan protokol transmisi video dan audio menjadi faktor krusial dalam menentukan kualitas dan stabilitas siaran. Protokol Real-Time Messaging Protocol (RTMP) masih banyak digunakan, namun memiliki keterbatasan dalam menghadapi kondisi jaringan yang tidak stabil. Sebagai alternatif, Secure Reliable Transport (SRT) dikembangkan untuk meningkatkan keandalan pengiriman data multimedia melalui mekanisme koreksi kesalahan dan manajemen latensi adaptif. Penelitian ini bertujuan untuk mengimplementasikan live streaming menggunakan perangkat lunak vMix dengan protokol SRT serta menganalisis performanya melalui perbandingan dengan protokol RTMP. Metode penelitian yang digunakan adalah metode eksperimen dengan melakukan pengujian langsung terhadap kedua protokol berdasarkan parameter latensi, kualitas video dan audio, stabilitas koneksi, serta ketahanan terhadap packet loss. Pengujian dilakukan pada kondisi jaringan yang bervariasi untuk merepresentasikan situasi siaran real-time secara nyata. Hasil penelitian menunjukkan bahwa protokol SRT mampu memberikan performa yang lebih baik dibandingkan RTMP, terutama dalam hal latensi yang lebih rendah, stabilitas siaran yang lebih tinggi, serta kemampuan pemulihan data yang lebih baik pada kondisi jaringan yang tidak stabil. Dengan demikian, implementasi SRT pada sistem live streaming berbasis vMix dapat menjadi solusi yang efektif untuk menghasilkan siaran real-time berkualitas tinggi.*

**Kata Kunci:** *Live Streaming; RTMP; Siaran Real-Time; SRT; VMix.*

### 1. LATAR BELAKANG

Perkembangan teknologi informasi dan komunikasi telah mendorong meningkatnya kebutuhan akan layanan *live streaming* dalam berbagai bidang, seperti penyiaran televisi, pendidikan, hiburan, dan komunikasi daring (Hermawan & Desmira, 2025). Tonggo dan Irwansyah (2021) menjelaskan bahwa *live streaming* memungkinkan distribusi konten video dan audio secara langsung kepada audiens tanpa jeda waktu yang signifikan sehingga kualitas dan keandalan sistem menjadi faktor utama dalam keberhasilan siaran *real-time*. Salah satu

komponen penting dalam sistem *live streaming* adalah protokol transmisi data yang digunakan untuk mengirimkan konten multimedia melalui jaringan internet (Samain, 2019).

*Real-Time Messaging Protocol* (RTMP) menjadi protokol yang telah lama digunakan dalam sistem *live streaming* karena kemampuannya dalam mengirimkan data audio dan video secara kontinu (Vincze et al., 2023). Namun, RTMP memiliki beberapa keterbatasan, terutama dalam menghadapi kondisi jaringan yang tidak stabil, seperti tingginya latensi, *packet loss*, dan *jitter*. Keterbatasan tersebut dapat menyebabkan penurunan kualitas siaran, gangguan audio, hingga terputusnya koneksi siaran sehingga kurang optimal untuk kebutuhan siaran *real-time* berkualitas tinggi (Sardana et al., 2025).

Seiring dengan perkembangan teknologi *streaming*, *Secure Reliable Transport* (SRT) diperkenalkan sebagai protokol transmisi yang dirancang untuk meningkatkan keandalan dan efisiensi pengiriman data multimedia melalui jaringan internet. SRT memiliki mekanisme pengendalian latensi adaptif, koreksi kesalahan, serta enkripsi data yang memungkinkan siaran tetap stabil meskipun pada kondisi jaringan yang tidak ideal. Penerapan protokol SRT dinilai mampu mengatasi berbagai permasalahan yang sering muncul pada protokol RTMP, khususnya dalam konteks siaran langsung dengan tuntutan kualitas tinggi dan latensi rendah (Dama & Nurohman, 2024).

Jafar et al. (2025) menjelaskan bahwa perangkat lunak vMix sebagai salah satu solusi *live production* dan *streaming* profesional telah mendukung penggunaan berbagai protokol, termasuk RTMP dan SRT. Dengan kemampuan vMix dalam mengelola input video, audio, serta *encoding* secara *real-time*, perangkat lunak ini menjadi platform yang tepat untuk mengimplementasikan dan menganalisis performa kedua protokol tersebut. Oleh karena itu, diperlukan suatu penelitian yang mengkaji implementasi *live streaming* menggunakan vMix dengan protokol SRT serta membandingkannya dengan protokol RTMP guna mengetahui efektivitasnya sebagai solusi siaran *real-time* berkualitas tinggi (Priyambudi & Pinasthika, 2024).

## 2. KAJIAN TEORITIS

### *Live Streaming*

*Streaming* adalah sebuah layanan yang memproses data yang diterima secara langsung tanpa harus menunggu seluruh data selesai dikirim. Layanan *streaming* saat ini mencakup layanan audio dan video. Teknologi *streaming*, yang juga dikenal sebagai *streaming* media, merupakan teknologi untuk menjalankan file (audio atau video) dari server *streaming* (halaman web), baik secara langsung (*live*) maupun rekaman, di mana file tersebut harus terlebih dahulu

dikodekan dengan laju data tertentu yang sesuai untuk transmisi melalui jaringan sesuai dengan kapasitas *bandwidth* pengguna (Azhar et al., 2021).

Dalam implementasinya, teknologi *streaming* memerlukan proses pengkodean file audio dan video dengan berbagai variasi laju data agar dapat menyesuaikan dengan kecepatan jaringan dan sistem akses data. Pengguna dapat langsung melihat atau mendengarkan file audio dan video dari server *streaming* secara *live* sehingga tidak perlu menunggu waktu lama untuk mengakses file berukuran besar. Kualitas file *streaming* bergantung pada besar *bandwidth*, jenis konten file, serta jumlah data yang dapat ditransmisikan per detik saat melewati jaringan (Agustin et al., 2023).

### **VMix**

VMix merupakan perangkat lunak *live production* yang digunakan untuk melakukan *mixing*, *switching*, dan *streaming* video secara *real-time*. vMix mendukung *berVMix* merupakan perangkat lunak video *mixer* yang digunakan untuk mendukung proses produksi hingga penyiaran konten layaknya siaran televisi. Aplikasi ini memiliki berbagai keunggulan karena mampu menjalankan fungsi-fungsi penyiaran yang biasanya dilakukan oleh sejumlah perangkat di dalam ruang *Master Control Room* (MCR). Karena berbasis perangkat lunak dan dapat menggantikan peran beberapa perangkat tersebut, vMix menawarkan fleksibilitas mobilitas yang lebih tinggi dibandingkan dengan sistem MCR konvensional (Hermawan & Desmira, 2025).

Selain itu, Priyambudi dan Pinasthika (2024) mengatakan vMix mendukung penggunaan beberapa sumber input video secara simultan dari lokasi yang berbeda. Perangkat lunak ini juga mampu melakukan perekaman, proses *streaming*, serta menghasilkan output ke monitor. Tidak hanya itu, vMix menyediakan fitur tambahan, seperti efek transisi, *chroma key*, serta penambahan *title* secara langsung (*live*). Bagi format input dan protokol *streaming* sehingga banyak digunakan dalam produksi siaran langsung profesional, baik untuk kebutuhan penyiaran, webinar, maupun acara daring lainnya.

### **Real-Time Messaging Protocol (RTMP)**

RTMP adalah protokol komunikasi yang dikembangkan untuk mengirimkan data audio, video, dan data lainnya secara *real-time* melalui jaringan internet. Protokol ini bekerja dengan koneksi berbasis *Transmission Control Protocol* (TCP) dan banyak digunakan dalam sistem *streaming* konvensional. Namun, RTMP memiliki keterbatasan dalam mengatasi kondisi jaringan yang tidak stabil (Shwetha & Devi, 2021).

### **Secure Reliable Transport (SRT)**

*Secure Reliable Transport* (SRT) merupakan protokol *open-source* yang dirancang untuk pengiriman konten video dan audio berkualitas tinggi dengan latensi rendah. SRT menggunakan mekanisme retransmisi data, pengaturan *buffer* adaptif, serta enkripsi untuk meningkatkan keandalan dan keamanan transmisi sehingga cocok untuk siaran *real-time* pada jaringan yang dinamis (Luu et al., 2025).

### **3. METODE PENELITIAN**

Penelitian ini menggunakan metode eksperimen dengan pendekatan komparatif (*comparative experimental study*). Metode ini bertujuan untuk membandingkan performa dua protokol *live streaming*, yaitu *Real-Time Messaging Protocol* (RTMP) dan *Secure Reliable Transport* (SRT), dalam pengiriman video dan audio secara *real-time* menggunakan perangkat lunak vMix (Priyambudi & Pinasthika, 2024). Putra et al. (2023) menjelaskan bahwa pendekatan komparatif digunakan untuk memperoleh gambaran objektif mengenai perbedaan kinerja kedua protokol berdasarkan parameter teknis yang terukur.

Pengujian dilakukan pada lingkungan dan konfigurasi sistem yang sama sehingga perbedaan hasil yang diperoleh dapat dikaitkan langsung dengan karakteristik masing-masing protokol. Eksperimen dilakukan dengan melakukan percobaan pada tiga kondisi jaringan yang berbeda, yaitu kondisi jaringan normal, kondisi pembatasan *bandwidth* (*bandwidth limit*), dan kondisi simulasi *packet loss*.

Pada kondisi jaringan normal, pengujian dilakukan tanpa adanya gangguan atau pembatasan jaringan untuk mengetahui performa dasar dari masing-masing protokol. Pada kondisi pembatasan *bandwidth*, dilakukan pengaturan kapasitas jaringan untuk mensimulasikan keterbatasan *bandwidth* yang sering terjadi pada jaringan nyata. Sedangkan pada kondisi simulasi *packet loss*, dilakukan penambahan gangguan berupa kehilangan paket untuk mengamati ketahanan protokol dalam menghadapi kondisi jaringan yang tidak stabil.

Setiap percobaan dilakukan dengan mengirimkan data video dan audio secara *real-time* dari pengirim ke penerima menggunakan vMix. Selanjutnya, dilakukan pengukuran terhadap parameter kualitas layanan (*Quality of Service/QoS*) yang meliputi *delay*, *jitter*, *throughput*, dan *packet loss*. Data dianalisis menggunakan Microsoft Excel dan Python untuk hasil yang akurat dan terstruktur. Data hasil pengujian kemudian dibandingkan untuk menentukan protokol yang memiliki performa terbaik pada masing-masing kondisi jaringan.

## Objek dan Lingkup Penelitian

Objek penelitian ini adalah sistem *live streaming* berbasis vMix yang memanfaatkan protokol RTMP dan SRT sebagai media transmisi data video dan audio dengan *routing* statis. Prayudha et al. (2025) menjelaskan statik *routing* merupakan metode pengaturan rute jaringan yang dilakukan secara manual oleh administrator jaringan. Pada metode ini, administrator menentukan tabel *routing* yang berisi jalur yang akan dilalui oleh paket data berdasarkan alamat IP tujuan (*IP address*). Rute yang telah ditetapkan pada *routing* statis bersifat tetap dan tidak akan berubah kecuali dilakukan perubahan secara manual oleh administrator.

Proses konfigurasi *routing* statis pada penelitian ini dilakukan menggunakan aplikasi Winbox. Winbox merupakan utilitas berbasis *Graphical User Interface* (GUI) yang digunakan untuk melakukan administrasi sistem operasi RouterOS pada perangkat MikroTik. Aplikasi ini memungkinkan administrator untuk terhubung ke perangkat MikroTik melalui alamat IP maupun MAC *address* dengan menggunakan koneksi kabel Ethernet (Nursobah et al., 2023).

Lingkup penelitian difokuskan pada analisis performa jaringan serta kualitas siaran secara *real-time*. Penelitian ini dilaksanakan di TVRI Kalimantan Timur. Batasan penelitian meliputi penggunaan *encoder live streaming* berbasis vMix dengan konfigurasi yang seragam untuk kedua protokol, yaitu RTMP dan SRT. Konten siaran yang digunakan berupa video dan audio *real-time* dengan durasi serta kualitas yang identik sehingga memungkinkan perbandingan yang objektif. Pengujian difokuskan pada kedua protokol tersebut tanpa melibatkan metode transmisi lainnya. Analisis performa jaringan dilakukan menggunakan Wireshark untuk mengamati parameter-parameter jaringan yang relevan.

## Arsitektur Sistem Penelitian

Arsitektur sistem *live streaming* dalam penelitian ini terdiri dari beberapa komponen utama, yaitu sumber video dan audio, vMix sebagai *encoder*, jaringan internet sebagai media transmisi, server *streaming*, serta klien penerima siaran.

Alur sistem secara umum dimulai dari kamera dan mikrofon yang menangkap video serta audio secara *real-time*. Selanjutnya, vMix melakukan proses encoding dan pengemasan data multimedia sebelum data tersebut dikirim melalui jaringan internet menggunakan protokol RTMP atau SRT. Selama proses transmisi berlangsung, *Wireshark* digunakan untuk melakukan capture dan monitoring paket data. Setelah itu, server *streaming* menerima data dan meneruskannya kepada klien penerima, sehingga klien dapat menampilkan siaran live secara *real-time*. Arsitektur yang sama diterapkan pada kedua protokol untuk memastikan kesetaraan kondisi pengujian.

## Parameter dan Variabel Pengujian

Parameter yang digunakan dalam penelitian ini dibagi menjadi parameter kualitas siaran dan parameter jaringan Priyambudi dan Pinasthika (2024). Parameter tersebut meliputi latensi yang digunakan untuk mengukur waktu tunda antara pengiriman dan penerimaan video serta audio, packet loss yang mengukur persentase paket data yang hilang selama proses transmisi dan dianalisis menggunakan Wireshark, serta jitter yang digunakan untuk mengukur variasi waktu kedatangan paket data yang memengaruhi kestabilan siaran. Selain itu, throughput digunakan untuk mengukur kecepatan transfer data efektif selama proses live streaming. Penelitian ini juga menilai kualitas video berdasarkan kestabilan frame, kejernihan gambar, dan artefak visual, serta kualitas audio berdasarkan kejernihan suara dan sinkronisasi audio-video. Parameter terakhir adalah stabilitas koneksi yang digunakan untuk menilai kontinuitas siaran tanpa buffering maupun terputusnya koneksi.

## Skenario dan Prosedur Pengujian

Prosedur pengujian dalam penelitian ini dilakukan melalui beberapa tahapan yang dimulai dengan menyiapkan sistem live streaming menggunakan vMix dengan konfigurasi encoding yang sama. Selanjutnya, Wireshark diaktifkan untuk melakukan capture lalu lintas jaringan pada *interface* yang digunakan. Simulasi kondisi jaringan tidak stabil dilakukan menggunakan *Clumsy* dengan mengatur parameter packet loss, sedangkan pembatasan bandwidth dilakukan melalui konfigurasi perangkat MikroTik menggunakan aplikasi Winbox guna merepresentasikan kondisi jaringan dengan kapasitas terbatas. Setelah itu, dilakukan siaran live menggunakan protokol RTMP selama durasi tertentu, kemudian hasil pengujian berupa latensi, packet loss, jitter, dan throughput dicatat berdasarkan data dari Wireshark. Siaran RTMP selanjutnya dihentikan dan hasil capture disimpan sebelum langkah pengujian yang sama diulangi menggunakan protokol SRT. Pengujian dilakukan pada beberapa kondisi jaringan, seperti jaringan stabil dan jaringan dengan gangguan bandwidth, kemudian seluruh data hasil pengujian dikumpulkan untuk dianalisis lebih lanjut

## Teknik Pengumpulan Data

Teknik pengumpulan data dilakukan melalui observasi langsung, pengukuran jaringan, dan dokumentasi. Observasi langsung merupakan pengamatan kualitas siaran video dan audio selama *live streaming*. Pengukuran jaringan adalah pengambilan data *packet loss*, *jitter*, *throughput*, dan latensi menggunakan Wireshark. Dokumentasi berupa hasil *capture* Wireshark, log sistem vMix, dan tangkapan layar selama pengujian. Pengukuran parameter jaringan, seperti *jitter* dan *packet loss*, dilakukan menggunakan aplikasi Wireshark yang berfungsi untuk menangkap dan menganalisis paket data selama proses transmisi berlangsung.

Pengukuran *packet loss* pada protokol SRT dilakukan menggunakan statistik yang dihasilkan oleh aplikasi vMix pada sisi penerima karena SRT menggunakan mekanisme retransmisi internal yang tidak dapat diamati secara langsung melalui Wireshark. Sementara itu, pada protokol RTMP yang berbasis *Transmission Control Protocol* (TCP), *packet loss* dianalisis melalui jumlah *retransmission* paket yang terdeteksi pada Wireshark karena mekanisme TCP secara otomatis melakukan pengiriman ulang terhadap paket yang hilang.

#### 4. HASIL DAN PEMBAHASAN

##### Hasil

##### *Uji Latensi Streaming*

Pengujian latensi *streaming* dilakukan untuk mengetahui waktu tunda (*delay*) antara pengiriman dan penerimaan video serta audio pada sistem *live streaming* menggunakan protokol RTMP dan SRT. Pengujian dilakukan pada *bandwidth* yang sama, dengan beberapa kali percobaan untuk masing-masing protokol gabungan. Berikut hasil pengujian latensi *streaming* RTMP dan SRT yang disajikan pada tabel 1 di bawah ini:

**Tabel 1.** Hasil Pengujian Latensi *Streaming* RTMP dan SRT (ms).

Kondisi Jaringan	Protokol	Delay (ms)	Jitter (ms)	Packet Loss (%)	Throughput
Normal	SRT	21,5	7,30	0	1,80
	RTMP	22	9,03	0,01	3,21
Limit Bandwith	SRT	18,5	10,32	2,86	1,81
	RTMP	25	3,04	0,03	3,18
Packet Loss 5%	SRT	20,5	9,89	0,01	1,80
	RTMP	25	3,53	0,03	3,11

Berdasarkan tabel 1, terlihat bahwa protokol SRT dan RTMP menunjukkan karakteristik performa yang berbeda pada berbagai kondisi jaringan. Pada kondisi normal, kedua protokol memiliki nilai *delay* yang relatif serupa, yaitu pada kisaran 21–22 ms, yang menunjukkan bahwa secara jaringan keduanya mampu mentransmisikan data dengan latensi yang rendah. Namun, nilai *jitter* pada SRT lebih tinggi dibandingkan RTMP, yaitu sekitar 10 ms pada SRT dan 9 ms pada RTMP. Hal ini disebabkan oleh penggunaan protokol *User Datagram Protocol* (UDP) pada SRT yang tidak memiliki mekanisme *buffering*, seperti TCP pada RTMP sehingga variasi *delay* antar paket lebih terlihat secara langsung. Sementara itu, nilai *packet loss* pada kedua protokol tergolong sangat kecil, yaitu 0% pada SRT dan sekitar 0,01% pada RTMP sehingga keduanya mampu menjaga keandalan transmisi pada kondisi jaringan normal.

Pada kondisi pembatasan *bandwidth*, perbedaan karakteristik kedua protokol menjadi lebih jelas. SRT menunjukkan peningkatan nilai *packet loss* hingga 2,89% serta *jitter* yang mencapai 10,32 ms, yang menandakan adanya gangguan pada jaringan. Meskipun demikian, nilai *throughput* SRT tetap stabil di sekitar 1,81 Mbps. Sebaliknya, RTMP menunjukkan nilai *packet loss* yang sangat kecil, yaitu sekitar 0,03%, serta *jitter* yang lebih rendah sebesar 3,04 ms. Namun, kondisi ini dipengaruhi oleh mekanisme *retransmission* pada protokol TCP yang digunakan RTMP, di mana paket yang hilang akan dikirim ulang sehingga nilai *packet loss* terlihat kecil dan *jitter* menjadi lebih stabil. Meskipun demikian, proses *retransmission* tersebut dapat menyebabkan penundaan pengiriman data yang tidak sepenuhnya tercermin dalam nilai delay rata-rata.

Pada kondisi simulasi *packet loss* sebesar 5%, SRT tetap menunjukkan performa yang stabil dengan nilai *delay* sekitar 20,5 ms, *jitter* sekitar 9,90 ms, serta *packet loss* yang sangat kecil, yaitu 0,01%. Hal ini menunjukkan bahwa SRT mampu menangani gangguan jaringan secara adaptif tanpa menyebabkan peningkatan *delay* yang signifikan. Di sisi lain, RTMP kembali menunjukkan nilai *packet loss* yang rendah, yaitu sekitar 0,03%, serta *jitter* yang relatif kecil sebesar 3,53 ms dengan *delay* sekitar 25 ms. Namun, rendahnya nilai *packet loss* pada RTMP tidak sepenuhnya mencerminkan kondisi jaringan yang sebenarnya karena adanya mekanisme *retransmission* pada TCP yang menutupi kehilangan paket dengan pengiriman ulang.

Secara keseluruhan, protokol SRT menunjukkan keunggulan dalam merepresentasikan kondisi jaringan secara lebih nyata serta menjaga kestabilan transmisi pada berbagai kondisi, terutama pada jaringan yang mengalami gangguan. Sementara itu, RTMP cenderung menghasilkan nilai *jitter* dan *packet loss* yang lebih rendah secara numerik, namun hal ini dipengaruhi oleh mekanisme *buffering* dan *retransmission* pada TCP. Dengan demikian, SRT lebih sesuai digunakan untuk kebutuhan *live streaming real-time* yang memerlukan kestabilan dan respons terhadap kondisi jaringan, sedangkan RTMP lebih optimal pada kondisi jaringan yang stabil.

### Uji Packet Loss

Pengujian *packet loss* dilakukan menggunakan aplikasi vMix dan Wireshark untuk mengukur tingkat kehilangan paket data selama proses *live streaming* berlangsung. Perhitungan *packet loss* dilakukan menggunakan rumus berikut:

$$Packet\ Loss = \left( \frac{Paket\ Dikirim - Paket\ Diterima}{Paket\ Dikirim} \right) \times 100 \dots\dots(i)$$

Hasil pengujian *packet loss* RTMP dan SRT disajikan pada tabel 2 berikut:

**Tabel 2.** Hasil Pengujian *Packet Loss* RTMP dan SRT (%).

Kondisi Jaringan	RTMP (%)	SRT (%)
Normal	0,01	0
<i>Limit Bandwidth</i>	0,03	2,86
<i>Packet Loss 5%</i>	0,03	0,01

Hasil pengujian pada tabel 2 menunjukkan bahwa terdapat perbedaan karakteristik *packet loss* antara protokol RTMP dan SRT pada berbagai kondisi jaringan. Pada kondisi normal, RTMP mengalami *packet loss* sebesar 0,01%, sedangkan SRT tidak mengalami *packet loss* (0%). Hal ini menunjukkan bahwa kedua protokol mampu menjaga keutuhan data dengan sangat baik pada kondisi jaringan yang stabil, dengan SRT menunjukkan performa yang sedikit lebih baik.

Pada kondisi pembatasan *bandwidth*, RTMP menunjukkan nilai *packet loss* yang relatif kecil, yaitu sebesar 0,03%, sedangkan SRT mengalami peningkatan *packet loss* hingga 2,89%. Secara numerik, RTMP terlihat lebih unggul, namun hal ini dipengaruhi oleh penggunaan protokol TCP yang memiliki mekanisme *retransmission*. Mekanisme tersebut memungkinkan paket yang hilang dikirim ulang sehingga nilai *packet loss* yang terukur menjadi kecil, meskipun pada kenyataannya terjadi gangguan dalam proses transmisi data.

Hal ini diperkuat oleh hasil observasi selama pengujian, di mana pada kondisi pembatasan *bandwidth*, transmisi menggunakan RTMP mengalami gangguan berupa tampilan video yang tidak stabil, munculnya *glitch*, serta terputus-putus. Sebaliknya, pada kondisi yang sama, SRT tidak menunjukkan gangguan yang signifikan dan mampu mempertahankan kestabilan tampilan video dan audio.

Pada kondisi simulasi *packet loss* sebesar 5%, RTMP mencatat nilai *packet loss* sebesar 0,03%, sedangkan SRT hanya sebesar 0,01%. Meskipun secara numerik RTMP menunjukkan nilai *packet loss* yang rendah, hasil observasi menunjukkan bahwa kualitas *streaming* tetap mengalami penurunan, seperti munculnya *glitch* dan gangguan pada tampilan video. Hal ini disebabkan oleh mekanisme *retransmission* pada TCP yang menyebabkan keterlambatan pengiriman data. Di sisi lain, SRT tetap mampu menjaga kestabilan kualitas *streaming* tanpa menunjukkan gangguan yang berarti.

Secara keseluruhan, protokol SRT lebih mampu menjaga kualitas transmisi secara konsisten pada berbagai kondisi jaringan, terutama pada kondisi yang tidak stabil. Sementara itu, nilai *packet loss* pada RTMP cenderung tidak sepenuhnya merepresentasikan kondisi jaringan yang sebenarnya karena dipengaruhi oleh mekanisme *retransmission* pada TCP. Oleh

karena itu, dalam konteks *live streaming real-time*, SRT lebih direkomendasikan karena mampu mempertahankan kualitas audio dan video meskipun terjadi gangguan jaringan.

**Uji Jitter**

*Jitter* diukur untuk mengetahui kestabilan waktu kedatangan paket data selama *streaming* berlangsung. Perhitungan *jitter* dilakukan menggunakan rumus berikut:

$$Jitter = \frac{Total\ Variasi\ Delay}{Total\ Paket\ yang\ Diterima - 1} \dots\dots(ii)$$

Berikut hasil pengujian *jitter* RTMP dan SRT yang telah dilakukan, disajikan pada tabel 3 di bawah ini:

**Tabel 3.** Hasil Pengujian *Jitter* RTMP dan SRT (ms).

Kondisi Jaringan	RTMP (ms)	SRT (ms)
Normal	9,03	7,30
Limit Bandwith	3,04	10,32
Packet Loss 5%	3,53	9,89

Hasil pengujian pada tabel 3 menunjukkan bahwa nilai *jitter* pada protokol RTMP dan SRT memiliki karakteristik yang berbeda pada setiap kondisi jaringan. Pada kondisi normal, RTMP menghasilkan nilai *jitter* sebesar 9,03 ms, sedangkan SRT sebesar 7,30 ms. Nilai ini menunjukkan bahwa pada kondisi jaringan yang stabil, SRT memiliki variasi *delay* antar paket yang lebih kecil dibandingkan RTMP sehingga transmisi data cenderung lebih konsisten.

Pada kondisi pembatasan *bandwidth*, nilai *jitter* RTMP sebesar 3,04 ms terlihat lebih rendah dibandingkan SRT yang mencapai 10,32 ms. Secara numerik, RTMP tampak lebih stabil. Namun, nilai tersebut tidak sepenuhnya mencerminkan kondisi jaringan yang sebenarnya. Hal ini disebabkan oleh mekanisme *buffering* dan *retransmission* pada protokol TCP yang digunakan RTMP sehingga variasi waktu kedatangan paket menjadi lebih “halus” secara statistik. Meskipun demikian, hasil observasi menunjukkan bahwa pada kondisi ini kualitas video RTMP mengalami gangguan, seperti munculnya *glitch*, gambar yang tidak stabil, serta sesekali terputus. Sebaliknya, SRT tetap mampu mempertahankan kestabilan tampilan video dan audio meskipun nilai *jitter* lebih tinggi.

Pada kondisi simulasi *packet loss* sebesar 5%, nilai *jitter* RTMP sebesar 3,53 ms kembali terlihat lebih rendah dibandingkan SRT sebesar 9,89 ms. Namun, seperti pada kondisi sebelumnya, nilai *jitter* RTMP yang rendah tidak sepenuhnya mencerminkan kestabilan transmisi yang sebenarnya. Hasil observasi menunjukkan bahwa kualitas *streaming* pada RTMP tetap mengalami penurunan, ditandai dengan munculnya *glitch*, ketidakstabilan gambar, serta gangguan pada tampilan video akibat proses *retransmission* yang menyebabkan

keterlambatan pengiriman data. Sementara itu, SRT tetap menunjukkan performa yang stabil tanpa gangguan yang signifikan selama proses *streaming* berlangsung.

Secara keseluruhan, meskipun RTMP pada beberapa kondisi menunjukkan nilai *jitter* yang lebih rendah secara numerik, hal tersebut dipengaruhi oleh mekanisme *buffering* dan *retransmission* pada TCP. Di sisi lain, SRT mampu merepresentasikan kondisi jaringan secara lebih nyata dan mempertahankan kualitas *streaming* yang stabil tanpa gangguan visual. Oleh karena itu, dalam konteks *live streaming real-time*, SRT lebih unggul dalam menjaga kestabilan transmisi dibandingkan RTMP.

### **Uji Delay**

*Delay* diukur untuk mengetahui waktu tunda yang terjadi selama proses pengiriman paket data dari pengirim ke penerima pada saat *streaming* berlangsung. Perhitungan *delay* dilakukan menggunakan rumus berikut:

$$\text{Delay} = \text{Waktu Paket Diterima} - \text{Waktu Paket Dikirim} \dots\dots(iii)$$

Berikut hasil pengujian *delay* RTMP dan SRT yang telah dilakukan, kemudian disajikan pada tabel 4 di bawah ini:

**Tabel 4.** Hasil Pengujian *Delay* RTMP dan SRT (ms)

<b>Kondisi Jaringan</b>	<b>RTMP (ms)</b>	<b>SRT (ms)</b>
Normal	22	21,5
<i>Limit Bandwith</i>	25	18,5
<i>Packet Loss 5%</i>	25	20,5

Berdasarkan tabel 4, terlihat bahwa nilai *delay* pada protokol RTMP dan SRT relatif rendah dan berada pada kisaran yang hampir sama pada berbagai kondisi jaringan. Pada kondisi normal, RTMP memiliki *delay* sebesar 22 ms, sedangkan SRT sebesar 21,5 ms. Hal ini menunjukkan bahwa secara jaringan, kedua protokol mampu mentransmisikan data dengan latensi yang rendah dan mendukung kebutuhan komunikasi *real-time*.

Pada kondisi pembatasan *bandwidth*, *delay* pada RTMP meningkat menjadi 25 ms, sementara SRT justru menunjukkan nilai yang lebih rendah, yaitu sebesar 18,5 ms. Hal ini menunjukkan bahwa SRT mampu mempertahankan latensi yang stabil meskipun terjadi gangguan pada kapasitas jaringan. Di sisi lain, RTMP mengalami sedikit peningkatan *delay* yang dipengaruhi oleh mekanisme pengiriman ulang data pada protokol TCP.

Pada kondisi simulasi *packet loss* sebesar 5%, nilai *delay* pada RTMP tetap berada di sekitar 25 ms, sedangkan SRT sebesar 20,5 ms. Hal ini menunjukkan bahwa kedua protokol masih mampu menjaga latensi dalam batas yang relatif rendah, namun SRT tetap menunjukkan performa yang lebih stabil dibandingkan RTMP.

Secara keseluruhan, baik RTMP maupun SRT mampu mempertahankan nilai *delay* yang rendah pada berbagai kondisi jaringan. Namun demikian, SRT cenderung lebih konsisten dalam menjaga kestabilan latensi, terutama pada kondisi jaringan yang mengalami gangguan. Selain itu, meskipun nilai *delay* RTMP secara numerik relatif rendah, hasil observasi menunjukkan bahwa kualitas *streaming* tetap mengalami gangguan, seperti munculnya *glitch* dan ketidakstabilan tampilan. Hal ini disebabkan oleh mekanisme *retransmission* pada TCP yang menyebabkan keterlambatan pengiriman data secara tidak langsung.

**Uji Throughput**

*Throughput* diukur untuk mengetahui tingkat kecepatan data yang berhasil dikirimkan dari pengirim ke penerima dalam satuan waktu selama proses *streaming* berlangsung. Perhitungan *throughput* dilakukan menggunakan rumus berikut:

$$Throughput = \frac{\text{Jumlah Data yang Dikirim}}{\text{Waktu Pengiriman Data}} \dots\dots(iv)$$

Berikut hasil pengujian *throughput* RTMP dan SRT yang disajikan pada tabel 5 di bawah ini:

**Tabel 5.** Hasil Pengujian *Throughput* RTMP dan SRT (Mbps).

Kondisi Jaringan	RTMP (Mbps)	SRT (Mbps)
Normal	3,21	1,80
<i>Limit Bandwidth</i>	3,18	1,81
<i>Packet Loss 5%</i>	3,11	1,80

Berdasarkan tabel 5, terlihat bahwa nilai *throughput* pada protokol RTMP dan SRT menunjukkan perbedaan pada setiap kondisi jaringan. Pada kondisi normal, RTMP menghasilkan *throughput* yang lebih tinggi, yaitu sebesar 3,21 Mbps dibandingkan SRT sebesar 1,80 Mbps. Hal ini menunjukkan bahwa RTMP mampu mengirimkan data dengan laju yang lebih besar pada kondisi jaringan yang stabil.

Pada kondisi pembatasan *bandwidth*, nilai *throughput* pada kedua protokol relatif stabil, di mana RTMP sebesar 3,18 Mbps dan SRT sebesar 1,81 Mbps. Tidak terdapat perubahan yang signifikan dibandingkan kondisi normal, yang menunjukkan bahwa kedua protokol tetap mempertahankan laju pengiriman data meskipun terjadi keterbatasan *bandwidth*. Namun demikian, mekanisme pengiriman data pada masing-masing protokol berbeda, di mana RTMP menggunakan TCP dengan kontrol aliran dan *retransmission*, sedangkan SRT menggunakan pendekatan yang lebih adaptif pada tingkat aplikasi.

Pada kondisi simulasi *packet loss* sebesar 5%, *throughput* RTMP mengalami sedikit penurunan menjadi 3,11 Mbps, sedangkan SRT tetap berada di sekitar 1,80 Mbps. Hal ini menunjukkan bahwa kedua protokol masih mampu menjaga kestabilan laju pengiriman data

meskipun terjadi gangguan jaringan. Namun, meskipun RTMP memiliki nilai *throughput* yang lebih tinggi, hasil observasi menunjukkan bahwa kualitas *streaming* tetap mengalami gangguan, seperti munculnya *glitch* dan terputus-putusnya tampilan. Sebaliknya, SRT tetap mampu mempertahankan kualitas audio dan video yang stabil tanpa gangguan yang signifikan.

Secara keseluruhan, RTMP cenderung menghasilkan *throughput* yang lebih tinggi dibandingkan SRT pada semua kondisi jaringan. Namun, nilai *throughput* yang tinggi tidak selalu berbanding lurus dengan kualitas *streaming* yang dihasilkan. SRT menunjukkan keunggulan dalam menjaga kestabilan transmisi dan kualitas *streaming* secara konsisten, terutama pada kondisi jaringan yang mengalami gangguan.

### **Analisis Kualitas Audio dan Video**

Berdasarkan hasil observasi selama pengujian, kualitas video dan audio pada protokol RTMP menunjukkan adanya gangguan pada berbagai kondisi jaringan. Pada kondisi normal, meskipun nilai *throughput* relatif tinggi dan *delay* berada pada kisaran yang rendah, tampilan video masih menunjukkan ketidakstabilan, seperti munculnya *glitch* dan gangguan visual sesekali. Selain itu, nilai *jitter* yang cukup tinggi juga berpotensi menyebabkan variasi tampilan video, meskipun tidak selalu berdampak langsung pada sinkronisasi audio dan video.

Pada kondisi pembatasan *bandwidth* dan simulasi *packet loss*, kualitas *streaming* RTMP mengalami penurunan yang lebih jelas. Hal ini ditandai dengan munculnya gangguan berupa tampilan video yang tidak stabil, *glitch*, serta sesekali terputus pada audio dan video. Kondisi tersebut dipengaruhi oleh mekanisme *retransmission* pada protokol TCP, di mana paket yang hilang akan dikirim ulang sehingga menyebabkan keterlambatan dalam proses pengiriman data. Meskipun secara numerik nilai *packet loss* dan *jitter* terlihat rendah, mekanisme ini berdampak pada kualitas *streaming* secara *real-time*.

Sebaliknya, protokol SRT menunjukkan performa yang lebih baik dalam menjaga kualitas audio dan video pada berbagai kondisi jaringan. Pada kondisi normal, nilai *delay* yang rendah serta *jitter* yang relatif stabil menghasilkan tampilan video yang lebih halus dan sinkronisasi audio-video yang baik. Pada kondisi pembatasan *bandwidth* dan simulasi *packet loss*, SRT tetap mampu mempertahankan kualitas *streaming* tanpa menunjukkan gangguan yang signifikan, meskipun terjadi fluktuasi jaringan.

Hal ini menunjukkan bahwa protokol SRT memiliki kemampuan adaptasi yang lebih baik terhadap kondisi jaringan yang dinamis melalui mekanisme pengelolaan error dan pengiriman ulang yang lebih efisien pada tingkat aplikasi. Dengan demikian, SRT lebih sesuai digunakan untuk kebutuhan *live streaming real-time* yang menuntut kestabilan, latensi rendah, serta kualitas audio dan video yang konsisten.

## Pembahasan

### Perbandingan RTMP dan SRT

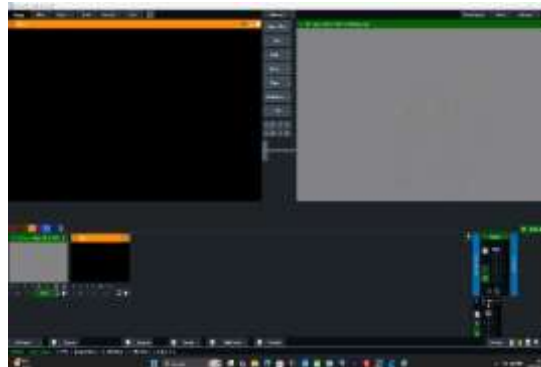
Berdasarkan hasil pengujian dan observasi yang telah dilakukan, terlihat adanya perbedaan performa yang signifikan antara protokol RTMP dan SRT dalam sistem *live streaming* berbasis vMix. Perbedaan tersebut tidak hanya terlihat dari nilai parameter *Quality of Service (QoS)*, seperti *delay*, *jitter*, *throughput*, dan *packet loss*, tetapi juga dapat diamati secara langsung melalui kualitas visual pada hasil *streaming*.



**Gambar 1.** Tampilan Hasil SRT.

Berdasarkan gambar 1, terlihat bahwa penggunaan protokol SRT menghasilkan tampilan video yang stabil dan tidak menunjukkan gangguan yang signifikan selama proses *streaming*. Kualitas gambar terlihat konsisten dengan minim distorsi visual, serta pergerakan video yang relatif halus. Selain itu, sinkronisasi antara audio dan video tetap terjaga dengan baik tanpa adanya keterlambatan yang mencolok.

Hasil observasi ini sejalan dengan data pengujian yang menunjukkan bahwa protokol SRT mampu mempertahankan nilai *delay* yang rendah dan *jitter* yang relatif stabil pada berbagai kondisi jaringan. Meskipun terdapat fluktuasi jaringan, SRT tetap mampu menjaga kualitas transmisi sehingga tampilan video dan audio tidak mengalami gangguan yang berarti.



**Gambar 2.** Tampilan Hasil RTMP.

Sebaliknya, pada penggunaan protokol RTMP, terlihat adanya penurunan kualitas tampilan video pada kondisi tertentu. Berdasarkan gambar 2, tampilan video menunjukkan gangguan berupa distorsi visual, munculnya *noise*, serta berkurangnya detail gambar. Pada beberapa momen, tampilan juga mengalami *freeze* atau tidak menampilkan video secara normal.

Hasil observasi ini sejalan dengan temuan selama pengujian, di mana *streaming* menggunakan RTMP cenderung mengalami gangguan, seperti *glitch*, ketidakstabilan gambar, serta terputus-putus, terutama pada kondisi jaringan yang mengalami pembatasan *bandwidth* dan simulasi *packet loss*. Meskipun nilai parameter QoS, seperti *packet loss* dan *jitter* pada RTMP terlihat relatif kecil secara numerik, kualitas visual yang dihasilkan tidak selalu stabil.

Secara teknis, kondisi tersebut dipengaruhi oleh penggunaan protokol TCP pada RTMP yang menerapkan mekanisme *retransmission*. Mekanisme ini memungkinkan paket yang hilang dikirim ulang sehingga nilai *packet loss* terukur menjadi rendah. Namun, proses pengiriman ulang tersebut menyebabkan keterlambatan dalam aliran data, yang berdampak pada tampilan video secara *real-time*, seperti munculnya *glitch* dan jeda tampilan.

Sebaliknya, SRT menggunakan protokol berbasis *User Datagram Protocol* (UDP) yang dilengkapi dengan mekanisme pengendalian error yang lebih adaptif pada tingkat aplikasi. Pendekatan ini memungkinkan SRT menjaga kontinuitas aliran data tanpa bergantung pada *retransmission* yang dapat menambah *delay* sehingga kualitas video dan audio tetap stabil meskipun kondisi jaringan tidak ideal.

Dengan demikian, dapat disimpulkan bahwa protokol SRT menunjukkan performa yang lebih baik dibandingkan RTMP dalam menjaga kualitas *streaming* secara *real-time*. Hal ini didukung hasil pengujian parameter QoS serta observasi visual, di mana SRT mampu mempertahankan tampilan yang stabil, sedangkan RTMP cenderung mengalami penurunan kualitas pada kondisi jaringan tertentu.

## 5. KESIMPULAN

Berdasarkan hasil penelitian dan analisis yang telah dilakukan terhadap performa protokol *Real-Time Messaging Protocol* (RTMP) dan *Secure Reliable Transport* (SRT) dalam sistem *live streaming* berbasis vMix, dapat disimpulkan bahwa kedua protokol memiliki karakteristik yang berbeda dalam menghadapi kondisi jaringan.

Hasil pengujian menunjukkan bahwa pada parameter *delay*, baik RTMP maupun SRT memiliki nilai yang relatif rendah dan berada pada kisaran yang hampir sama, yaitu sekitar 20–25 ms pada berbagai kondisi jaringan. Hal ini menunjukkan bahwa secara umum kedua

protokol mampu mendukung kebutuhan transmisi data dengan latensi rendah. Namun demikian, SRT cenderung lebih konsisten dalam menjaga kestabilan *delay*, terutama pada kondisi jaringan yang mengalami gangguan.

Pada parameter *jitter*, RTMP secara numerik menunjukkan nilai yang lebih rendah dibandingkan SRT, khususnya pada kondisi pembatasan *bandwidth* dan simulasi *packet loss*. Namun, nilai *jitter* yang rendah pada RTMP tidak sepenuhnya mencerminkan kestabilan transmisi yang sebenarnya. Hal ini disebabkan oleh mekanisme *buffering* dan *retransmission* pada protokol TCP yang digunakan RTMP sehingga variasi *delay* menjadi lebih halus secara statistik. Sebaliknya, SRT menunjukkan nilai *jitter* yang lebih tinggi karena menggunakan protokol *User Datagram Protocol* (UDP), namun nilai tersebut merepresentasikan kondisi jaringan secara lebih nyata.

Pada parameter *packet loss*, RTMP menunjukkan nilai yang relatif kecil pada semua kondisi, sedangkan SRT mengalami peningkatan *packet loss* pada kondisi pembatasan *bandwidth*. Namun demikian, nilai *packet loss* pada RTMP dipengaruhi oleh mekanisme *retransmission* pada TCP yang menyembunyikan kehilangan paket melalui pengiriman ulang sehingga tidak sepenuhnya mencerminkan kondisi jaringan yang sebenarnya.

Pada parameter *throughput*, RTMP menghasilkan nilai yang lebih tinggi dibandingkan SRT pada semua kondisi jaringan. Meskipun demikian, nilai *throughput* yang tinggi tidak selalu berbanding lurus dengan kualitas *streaming* yang dihasilkan.

Berdasarkan hasil observasi visual selama pengujian, *streaming* menggunakan RTMP menunjukkan gangguan kualitas berupa *glitch*, ketidakstabilan tampilan video, serta terputus-putusnya audio dan video, terutama pada kondisi jaringan yang mengalami gangguan. Sebaliknya, SRT mampu mempertahankan kualitas *streaming* yang lebih stabil tanpa menunjukkan gangguan yang signifikan, baik pada kondisi normal maupun saat terjadi pembatasan *bandwidth* dan simulasi *packet loss*.

Secara keseluruhan, protokol SRT lebih andal dalam menjaga kestabilan dan kualitas transmisi multimedia pada kondisi jaringan yang dinamis dan tidak stabil. Sementara itu, RTMP cenderung menghasilkan nilai parameter QoS yang lebih baik secara numerik pada beberapa kondisi, namun tidak selalu diikuti dengan kualitas *streaming* yang stabil. Oleh karena itu, SRT lebih direkomendasikan untuk kebutuhan *live streaming real-time* yang menuntut kestabilan, respons terhadap kondisi jaringan, serta kualitas audio dan video yang konsisten.

## UCAPAN TERIMA KASIH

Terima kasih kepada dosen pembimbing yang telah memberikan arahan, bimbingan, serta masukan yang konstruktif selama proses penelitian berlangsung. Selain itu, terima kasih kepada TVRI Kalimantan Timur yang telah memberikan kesempatan serta dukungan dalam pelaksanaan penelitian di lapangan. Tidak lupa, penulis menghargai dukungan dari keluarga, rekan, dan semua pihak yang tidak dapat disebutkan satu per satu atas doa dan motivasi yang diberikan hingga penelitian ini dapat diselesaikan dengan baik.

## REFERENSI

- Agustin, M., Hermawan, I., Arnaldy, D., Muharram, A. T., & Warsuta, B. (2023). Design of livestream video system and classification of rice disease. *JOIV: International Journal of Informatics Visualization*, 7(1), 139–145. <https://doi.org/10.30630/joiv.7.1.1336>
- Ariyanto, D. (2022). Analisis performa jaringan pada sistem video streaming berbasis internet. *Jurnal Teknologi Informasi dan Komunikasi*, 9(2), 88–96.
- Azhar, R., Santoso, H., & Faisal. (2021). Analisa quality of service menggunakan aplikasi gnum3d sebagai server media streaming. *Jurnal Bumigora Information Technology (BITE)*, 3(1), 45–55. <https://doi.org/10.30812/bite.v3i1>
- Dama, M., & Nurohman, S. (2024). Analisis kualitas layanan (QoS) pada streaming video dan audio menggunakan protokol SRT (Secure Reliable Transport). *Jurnal Ekselesta*, 1(1), 1–6.
- Fadilah, R., & Maulana, H. (2021). Implementasi media streaming menggunakan protokol RTMP pada sistem broadcasting digital. *Jurnal Teknik Informatika*, 14(1), 55–63.
- Hakim, L. (2020). Pengaruh bandwidth terhadap kualitas layanan video streaming. *Jurnal Sistem dan Teknologi Informasi*, 8(3), 101–109.
- Hermawan, A. G., & Desmira. (2025). Analisis kinerja vMix Pro dalam meningkatkan efisiensi produksi dan kualitas tayangan di RRI Banten. *Mars: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer*, 3(5), 220–227. <https://doi.org/10.61132/mars.v3i5.1151>
- Jafar, I., Gunawan, E., & Mandar, G. (2025). Rancang bangun RTMP server berbasis Nginx. *Jurnal Teknologi Informatika (J-TIFA)*, 8(2), 37–41. <https://doi.org/10.37817/ikraith-informatika.v8i3.4361>
- Kurniawan, A., & Saputra, B. (2022). Evaluasi performa protokol komunikasi real-time pada jaringan multimedia. *Jurnal Ilmu Komputer dan Informatika*, 6(2), 75–84.
- Luu, T., Nguyen, Q., Tran, T., Tran, M., Ding, S., Kua, J., & Hoang, T. (2025). Enhancing real-time robot teleoperation with immersive virtual reality in industrial IoT networks. *The International Journal of Advanced Manufacturing Technology*, 6233–6257. <https://doi.org/10.1007/s00170-025-16236-w>
- Mahendra, T. (2023). Monitoring packet loss menggunakan Wireshark pada jaringan multimedia. *Jurnal Informatika Terapan*, 10(1), 44–52.
- Nasrullah, M. (2021). Implementasi secure reliable transport untuk siaran live streaming berbasis internet. *Jurnal Media Informatika*, 5(2), 66–74.

- Nursobah, Aditya, P., & Supriady. (2023). Implementasi jaringan PPPoE dan hotspot server RT/RW net berbasis MikroTik dengan fitur Mikhmon di Adinet Samarinda Seberang. *Jurnal INFORMATIKA*, 13(1), 31–39. <https://doi.org/10.46984/inf-wcd.2204>
- Prasetyo, H., & Nugroho, Y. (2020). Analisis kualitas layanan jaringan multimedia menggunakan parameter QoS. *Jurnal Teknologi dan Sistem Komputer*, 8(4), 211–220.
- Prayudha, A., Fahmi, M., & Fajri, A. (2025). Analisis perbandingan kinerja dynamic routing EIGRP (Enhanced Interior Gateway Routing Protocol) dan OSPF (Open Shortest Path First) (studi kasus: STMIK Widya Cipta Dharma). *Jurnal Inform.*
- Priyambudi, R., & Pinasthika, S. J. (2024). Analisis performa rancang bangun sistem live streaming menggunakan software encoder vMix. *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, 12(3), 3286–3294. <https://doi.org/10.23960/jitet.v12i3.4972>
- Putra, F. P. E., Muslim, F., Hasanah, N., Holipah, Paradina, R., & Alim, R. (2023). Analisis komparasi protokol WebSocket dan MQTT dalam proses push notification. *Jurnal Sistim Informasi dan Teknologi (JSISFOTEK)*, 5(4), 63–72. <https://doi.org/10.60083/jsisfotek.v5i4.325>
- Rahman, F. (2024). Perbandingan performa RTMP dan SRT pada sistem live streaming berbasis cloud. *Jurnal Komputer dan Teknologi Informasi*, 11(2), 91–102.
- Samain, J. (2019). *Improving quality of experience in multimedia streaming by leveraging information-centric networking*.
- Sardana, D., Nayak, P., & Parashar, J. (2025). Stream rider: A developer's perspective on live streaming. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 13(1), 2076–2080. <https://doi.org/10.22214/ijraset.2025.66742>
- Setiawan, E. (2023). Optimasi throughput pada transmisi multimedia real-time. *Jurnal Rekayasa Sistem dan Teknologi Informasi*, 7(3), 150–159.
- Shwetha, M., & Devi, Y. C. R. (2021). Multimedia transmission mechanism for streaming over wireless communication channel. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 12(9), 242–252. <https://doi.org/10.14569/IJACSA.2021.0120928>
- Tonggo, H. L., & Irwansyah. (2021). Mediated Catholic mass during the COVID-19 pandemic: On communication, technology, and spiritual experience. *Jurnal Komunikasi*, 13(1), 20–35. <https://doi.org/10.24912/jk.v13i1.9714>
- Vincze, M., Molnar, B., & Kozlovszky, M. (2023). Real-time network video data streaming in digital medicine. *Computers*, 12, 1–18. <https://doi.org/10.3390/computers12110234>
- Wibowo, A., & Santika, D. (2022). Analisis jitter dan latensi pada layanan video conference berbasis IP. *Jurnal Informatika dan Telekomunikasi*, 9(1), 33–41.
- Yusuf, R. (2021). Implementasi server streaming menggunakan Nginx dan protokol RTMP. *Jurnal Sistem Informasi dan Multimedia*, 4(2), 72–80.