



# J-TIFA

( Jurnal Teknologi Informatika )

| Teknologi Informasi | Jaringan Komputer | Data Mining |



## RANCANG BANGUN RTMP SERVER BERBASIS NGINX

Irsan Jafar<sup>a</sup>, Erwin Gunawan<sup>b</sup>, Gamaria Mandar<sup>c</sup>

<sup>a...c</sup>Teknik Informatika, Universitas Muhamadiyah Maluku Utara, Kota Ternate, Indonesia

email: [irsanjafar123@gmail.com](mailto:irsanjafar123@gmail.com)<sup>a</sup>, email [Erwingunawan@institusi.ac.id](mailto:Erwingunawan@institusi.ac.id)<sup>b</sup>, email [Gamariamandar@institusi.ac.id](mailto:Gamariamandar@institusi.ac.id)<sup>c</sup>

### Abstrak

Pesatnya perkembangan teknologi informasi telah mengubah cara penyampaian informasi dari media konvensional ke media digital berbasis video, khususnya *live streaming*. Namun, ketergantungan pada platform pihak ketiga seringkali menghadapi kendala terkait privasi data, kontrol penuh atas konten, dan biaya berlangganan. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sebuah server *Real-Time Messaging Protocol* (RTMP) mandiri dengan memanfaatkan Nginx dan modul *nginx-rtmp-module*. Metode penelitian yang digunakan meliputi tahap analisis kebutuhan sistem, perancangan arsitektur jaringan, konfigurasi server pada sistem operasi berbasis Linux, serta pengujian performa. Server ini dikonfigurasi untuk menerima *input stream* dari perangkat sumber (seperti OBS Studio) dan mendistribusikannya kembali ke berbagai klien. Selain protokol RTMP, sistem ini juga diintegrasikan dengan protokol HLS (*HTTP Live Streaming*) agar konten dapat diakses secara fleksibel melalui berbagai peramban web dan perangkat seluler. Hasil pengujian menunjukkan bahwa server Nginx-RTMP mampu melakukan transmisi data video dengan *latency* yang rendah dan stabilitas koneksi yang terjaga pada jaringan lokal maupun internet. Implementasi ini memberikan solusi infrastruktur penyiaran yang efisien, berbiaya rendah (*open-source*), dan memberikan kendali penuh kepada administrator dalam mengelola hak akses serta kualitas bit rate video.

**Kata Kunci:** RTMP, Nginx, Live Streaming, HLS, Server, Open Source.

### Abstract

The rapid development of information technology has transformed information delivery from conventional media to video-based digital media, particularly through live streaming. However, reliance on third-party platforms often faces challenges regarding data privacy, full content control, and subscription costs. This research aims to design and implement an independent Real-Time Messaging Protocol (RTMP) server by utilizing Nginx and the *nginx-rtmp-module*. The research methodology includes system requirements analysis, network architecture design, server configuration on a Linux-based operating system, and performance testing. The server is configured to receive input streams from source devices (such as OBS Studio) and redistribute them to various clients. In addition to the RTMP protocol, the system is integrated with the HLS (*HTTP Live Streaming*) protocol to ensure content can be accessed flexibly across various web browsers and mobile devices. The test results indicate that the Nginx-RTMP server is capable of transmitting video data with low latency and maintained connection stability over both local networks and the internet. This implementation provides an efficient, low-cost (*open-source*) broadcasting infrastructure solution and grants administrators full control over managing access rights and video bit rate quality.

**Keywords:** RTMP, Nginx, Live Streaming, HLS, Server, Open Source.

## 1. Pendahuluan

Perkembangan *drone* yang digunakan untuk melakukan *live streaming* telah mengalami kemajuan yang signifikan dalam beberapa tahun terakhir. Dengan kemajuan dalam teknologi penerbangan, kamera, serta konektivitas internet, *drone* kini dapat mengalirkan video secara langsung dengan kualitas tinggi dari sudut pandang yang sebelumnya tidak mungkin dijangkau. teknologi *drone* dalam *live streaming* telah digunakan dalam berbagai bidang dan industri. Berikut adalah beberapa contoh penggunaannya seperti. Acara hiburan dan olahraga. *Drone* digunakan untuk menyiarkan acara seperti konser, festival, pertandingan olahraga, dan acara lainnya. Dengan kemampuannya memberikan sudut pandang unik dari udara, *drone* menawarkan pengalaman menonton yang lebih menarik. Misalnya, dalam pertandingan sepak bola atau balap mobil, *drone* dapat memberikan gambaran udara yang dinamis, sehingga penonton merasa lebih dekat dengan aksi. Transmisi video *real-time* dari *drone* sangat penting untuk berbagai aplikasi karena memungkinkan pengguna untuk melihat dan merespons informasi visual secara langsung saat *drone* beroperasi. Dalam pengawasan, transmisi video *real-time* memungkinkan operator untuk segera melihat kejadian yang mencurigakan atau situasi darurat. hiburan Menciptakan pengalaman imersif, siaran langsung dari udara, dan konten kreatif. pemantauan Memudahkan inspeksi infrastruktur, pertanian presisi, dan pemantauan lingkungan. permasalahan dalam *streaming* video meminimalkan latensi, sehingga ideal untuk aplikasi yang membutuhkan respons cepat seperti pengawasan dan pengendalian *drone*. Sedangkan *Server* Linux memberikan kontrol penuh atas konfigurasi dan optimasi *server* untuk performa *streaming* yang lebih baik dan dapat ditingkatkan untuk menangani banyak koneksi dan *streaming* secara bersamaan. Linux menawarkan fitur keamanan yang kuat dan dapat

dikonfigurasi untuk melindungi transmisi video dari penyadapan.

## 1. Metode Penelitian

Metode penelitian ini menggunakan pendekatan singkat dengan tiga poin utama:

### 1. Perancangan Sistem

menggunakan metode rekayasa perangkat lunak (*software engineering*) dengan pendekatan *Research and Development (R&D)*. Metode ini bertujuan untuk mengembangkan sistem *broadcast server* yang dapat digunakan untuk kebutuhan *live streaming* dan *teleconference* secara mandiri tanpa ketergantungan pada platform pihak ketiga

### 2. Analisis Hasil dan Evaluasi

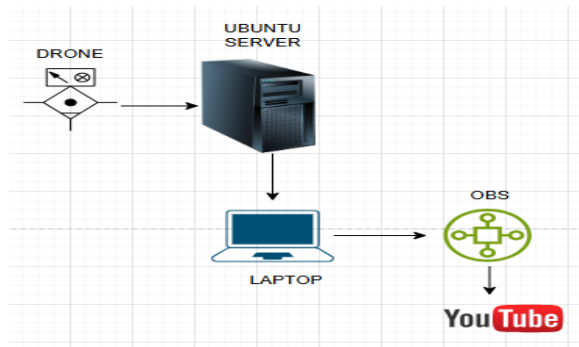
Hasil dari penelitian yang dilakukan menunjukkan bahwa sistem *broadcast server* yang dikembangkan dengan menggunakan *Nginx* dan modul RTMP berhasil berfungsi sesuai dengan tujuan. *Server* ini mampu menjalankan berbagai fitur penting seperti *live streaming*, *teleconference*, perekaman sesi (*archiving*), pemutaran ulang video melalui video *on demand* (VoD), serta melakukan *relay* ke *server* atau *platform* lain. Pengujian yang dilakukan menunjukkan bahwa sistem mampu menjalankan *teleconference* secara stabil dengan 2 hingga 4 partisipan, dan tingkat latensi yang dihasilkan masih berada dalam batas wajar untuk kebutuhan komunikasi *real-time*.

## 2. Penerapan perancangan system

Desain sistem ini bertujuan untuk mengimplementasikan RTMP *server* berbasis *Linux* yang dapat menerima dan men-*streaming* video secara *real-time* dari *drone* ke platform *live streaming*. Desain ini mencakup aspek arsitektur sistem secara keseluruhan, pengaturan perangkat keras dan perangkat lunak, serta interaksi antar komponen dalam sistem.

Arsitektur sistem yang diusulkan terdiri dari beberapa komponen utama yang saling berinteraksi untuk memastikan transmisi video dari *drone* ke *server* RTMP dan kemudian ke platform *live streaming*.

Berikut adalah gambaran umum arsitektur sistem:



### 1. Drone

*Drone* dilengkapi dengan kamera yang dapat menangkap video secara langsung. Video ini kemudian dikodekan oleh encoder video onboard, kemudian dikirimkan menggunakan protokol RTMP ke server yang sudah disiapkan. Drone ini berfungsi sebagai pengirim sumber video dalam sistem.

### 2. RTMP Server

RTMP server diimplementasikan di sistem operasi *Linux* menggunakan perangkat lunak *NGINX* dengan modul RTMP. Server ini bertugas untuk menerima aliran video RTMP dari *drone*, menyalurkannya ke platform *live streaming*, dan mengatur berbagai konfigurasi seperti *streaming* kualitas dan keamanan koneksi. RTMP server juga dapat menyediakan fitur untuk menangani beberapa kanal *streaming* yang berbeda sesuai kebutuhan.

### 3. Broadcast Software

*Broadcast software* dalam hal ini menggunakan *Open Broadcaster Software* (OBS), berperan sebagai aplikasi penghubung antara RTMP server dan sumber video yang akan disalurkan. OBS digunakan oleh operator untuk mengelola aliran video dari *drone*, melakukan pengaturan seperti penambahan *overlay*, transisi, dan pengaturan kualitas video. OBS dapat melakukan konfigurasi aliran video dengan berbagai format dan bitrate yang sesuai dengan spesifikasi RTMP server dan platform *live streaming*

yang digunakan. Meskipun RTMP server dan platform *live streaming* sudah menangani sebagian besar pengaturan, OBS memberikan kontrol tambahan atas tampilan dan kustomisasi aliran video, serta memungkinkan penggunaan fitur tambahan seperti audio *mixing*, pengaturan *webcam*, dan penggabungan berbagai sumber video yang berbeda sebelum disiarkan ke *audiens*.

### 4. Platform Live Streaming

Setelah video diterima oleh RTMP server, server kemudian mengalirkan video ke platform *live streaming* seperti *YouTube*, *Facebook Live*, atau platform kustom yang mendukung RTMP. Platform ini bertugas menampilkan video secara langsung kepada *audiens* dan menyediakan fitur interaksi dengan penonton.

### 3. Hasil dan Pembahasan

Deskripsi Hasil Implementasi Pada bagian ini, akan dijelaskan secara rinci mengenai hasil implementasi sistem RTMP server berbasis *Linux* yang digunakan untuk mendukung *live streaming* dari *drone*. Hasil implementasi ini mencakup pengaturan sistem server, integrasi dengan *drone*, serta pengujian untuk mengukur kinerja dan fungsionalitas sistem secara keseluruhan.

Instalasi dan Konfigurasi RTMP Server pada *Linux*, Implementasi dimulai dengan pemilihan distribusi *Linux* yang digunakan sebagai server, yaitu *Ubuntu 22.04 LTS*, yang dipilih karena kestabilan dan kompatibilitasnya dengan perangkat lunak RTMP yang digunakan. Server RTMP diatur menggunakan *NGINX* yang telah dipasang dengan modul RTMP untuk mendukung protokol RTMP.

Proses instalasi melibatkan langkah-langkah berikut:

a) Melakukan *update* dan *upgrade* melalui perintah:  
`#sudo apt update & sudo apt upgrade -y`

b) Setelah proses *update* selesai dilanjutkan dengan instalasi *Nginx* beserta modul RTMP menggunakan perintah:

`#sudo apt install nginx libnginx-mod-rtmp -y`

```

root@areebultahost:~# sudo apt install nginx libnginx-mod-rtmp -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fontconfig-dejavu-core libfontconfig1 libgd3 libhybi0 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fop nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fontconfig-dejavu-core libfontconfig1 libgd3 libhybi0 libjpeg-turbo8 libjpeg8
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-rtmp
  libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx nginx-common nginx-core
0 upgraded, 18 newly installed, 0 to remove and 0 not upgraded.
Need to get 2562 kB of archives.
After this operation, 8313 kB of additional disk space will be used.

```

- c) Konfigurasi Server: Konfigurasi file *NGINX* disesuaikan untuk menerima *stream* RTMP dari *drone*, termasuk pengaturan *port* dan nama aplikasi *stream*. Berikut adalah potongan konfigurasi *server* RTMP yang digunakan

```

$ sudo nano /etc/nginx/nginx.conf

```

*Drone* yang digunakan dalam pengujian ini adalah *DJI Mavic Air*, yang dilengkapi dengan kamera 4K. Proses pengaturan agar *drone* dapat mengirimkan video secara langsung ke *server* RTMP melibatkan beberapa langkah teknis, termasuk konfigurasi koneksi jaringan. Konfigurasi *Custom* RTMP pada *DJI Mavic Air* diatur seperti berikut

## Kesimpulan

beberapa tantangan utama yang perlu diperhatikan.

1. Pertama, stabilitas koneksi jaringan menjadi faktor krusial, karena *drone* sering beroperasi di area terbuka yang mungkin memiliki sinyal internet atau *Wi-Fi* yang tidak stabil, sehingga dapat otomatis agar *stream* tidak terputus saat salah satu jaringan bermasalah. Sebagai langkah perekaman lokal di *drone* agar video tetap tersimpan meskipun koneksi terputus.
2. Kedua, pengelolaan *bandwidth* dan *bitrate* video juga menjadi tantangan, karena kualitas video dari *drone* biasanya tinggi (HD atau lebih), yang memerlukan *bandwidth* besar; jika tidak disesuaikan dengan kapasitas jaringan, akan menyebabkan *delay*

atau gangguan pada transmisi data. Untuk mengatasi tantangan pengelolaan *bandwidth* dan *bitrate* video pada *drone*, terapkan *adaptive bitrate streaming* (ABR) untuk menyesuaikan kualitas video dengan kapasitas jaringan secara otomatis. dengan menggunakan *codec* video efisien seperti H.264 atau H.265 untuk kompresi yang optimal, mengurangi penggunaan *bandwidth* tanpa mengorbankan kualitas. Implementasikan pengelolaan *bandwidth* dinamis dengan protokol seperti SRT atau RTSP untuk menyesuaikan *transmisi* video berdasarkan kondisi jaringan *real-time*. Sebagai cadangan, aktifkan perekaman lokal di *drone* untuk menyimpan video jika terjadi gangguan pada koneksi.

Things (IoT) dan otomasi sistem kendali perangkat elektronik.

## Referensi

- Anggelina I Diwi1, R Rumani M2,, & Ida Wahidah3. (2014). Analisis Kualitas Layanan Video Live Streaming. *Buletin Pos dan Telekomunikasi*, Vol.12 No.3 September 2014 : 207 - 216, 207-216.
- Jordan Andrean, Andri Johanri, & Onno W purbo. (2015). Membangun Server Streaming Dengan Nginx-RTMP. 2015 *Hacking Straming*, 1-117.
- Muhamad Hamdan Rifai 1,, Budhi Irawan, S.Si, M.T.2, & , Randy Erfa Saputra, S.T, M.T.3. (2016). ANALISIS PERFORMANSI. *e-Proceeding of Engineering : Vol.3, No.2 Agustus 2016 | Page 2269*.
- nyoman buda, & putu agus yogi subawa. (2020). analisis kinerja server elearning untuk pembelajaran synchronous berbasis big blue buttondalam mendukung pembelajaran di masa covid-19. issn 2598-7542 i e-issn 25989650, 78-84.
- tommy andreas susanto1,, henry novianus palit2,, & agustinus noertjahyana3. (n.d.). pengembangan video broadcasting server untuk live.

- winarto, f. i., & irawan, b. (2016, juni). *live streaming server berbasis raspberry pi untuk video surveillance system*
- susanto, t. a., & palit, h. n (2018, mei). *pengembangan video broadcasting server untuk live streaming menggunakan nginx dan rtmp dengan studi kasus teleconference*
- pambudi, d. t., & sabara, m. a. (2020). *membangun server rtmp streaming di smp negeri 1 balapulang menggunakan ubuntu 16.04.*