

**PERANCANGAN VIDEO SERVER STREAMING MENGGUNAKAN NGINX DAN
INTERFACE MENGGUNAKAN ADOBE HDS UNTUK VIDEO LIVE
STREAMING DI GEDUNG P FAKULTAS TEKNIK ELEKTRO TELKOM
UNIVERSITY**

**THE DESIGN OF STREAMING VIDEO SERVER USING NGINX AND AND
INTERFACE USING ADOBE HDS FOR LIVE VIDEO STREAMING ON P
BUILDING FACULTY OF ELECTRICAL ENGINEERING TELKOM
UNIVERSITY**

Algimar Mochammad Firdaus¹, R. Rumani M.², Randy Erfa Saputra³

^{1,2,3} Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

¹gigi.algi@gmail.com, ²rumani@telkomuniversity.ac.id, ³resaputra@telkomuniversity.ac.id

Abstrak

Penggunaan *Live Video Streaming* saat ini sangat diperlukan untuk penyampaian informasi yang dapat dimanfaatkan untuk berbagai bidang. Perancangan *live video streaming* diharapkan akan mempermudah perpindahan informasi dan komunikasi di lingkungan kampus Telkom University. Dengan adanya *live video streaming*, maka penyampaian informasi bisa secara cepat dan tepat sampai dan mengurangi resource dan waktu yang dibutuhkan untuk menyampaikan perintah maupun informasi penting. Selain itu, perancangan *live video streaming* juga dimaksudkan untuk mengoptimalkan jaringan *wireless* yang telah ada. *Live Video Streaming* adalah perangkat lunak berbasis *web* yang bertugas sebagai penyedia informasi dalam bentuk *video* terhadap klien, yang bertugas menyediakan sarana agar *video* dapat di-broadcast. *Web video server* pada umumnya melakukan perekaman, penyimpanan, dan menjalankan berbagai *video* tanpa adanya pengurangan kualitas *video*, sehingga dapat dikatakan *web video server* yang baik adalah yang dapat mengantarkan *video* tanpa adanya degradasi kualitas sama sekali ketika sampai pada penerima informasi (client). Pada kasus ini Nginx adalah Video Server yang bersifat open source, handal, memiliki banyak fitur, memakan resource yang rendah dan dapat melakukan streaming dalam jumlah yang besar dengan tingkat kegagalan yang rendah. Aplikasi berbasis web ini menggunakan Nginx, Adobe HDS dan PHP.

Kata kunci: *web server, NginX, live video streaming, PHP, web-based application*

Abstract

Use of Live Video Streaming is now indispensable for the delivery of information that can be utilized for various fields. The design of live video streaming is expected to facilitate the echange of information and communication at Telkom University. With the use of live video streaming, the delivery of information can be quickly and precisely arrived, also reduce the resources and time required to deliver orders as well as important information. Live Video Streaming Application is a web-based software that serves as an information provider have the shape of video to the client, which is responsible for providing a way to broadcast video. Web video server generally do the recording, storage, and run a variety of video without any reduction in video quality, so we can say that a good video server is a server that can deliver video without any degradation of quality at all when it arrives to the information recipient (client). In this case Nginx is a Web Server that is open source, reliable, has many features, using low resources also can stream in large numbers with low failure rate, this web-based application using Nginx, Adobe HDS and PHP.

Keywords: *web server, NginX, live video streaming, PHP , web-based application*

1. Pendahuluan

1.1 Latar Belakang

Data *audio visual* menjadi salah satu data yang paling banyak mengisi *traffic* jaringan *internet* pada saat ini ^[1]. Trafik *video* berkembang paling cepat daripada jenis trafik *internet* yang lain, dan secara total dapat bertumbuh sebesar 90% ^[2]. Transfer *video* melalui IP sudah bukan merupakan hal yang aneh dimulai dari berkembangnya protokol UDP. Banyak protokol yang diciptakan untuk mengungguli protokol tersebut untuk memberikan kualitas multimedia yang jauh lebih bagus seiring berkembangnya teknologi audio visual itu sendiri. *Web server* merupakan sebuah sistem komputer yang memproses permintaan (request) melalui HTTP, *basic* dari *network protocol* yang digunakan untuk mendistribusikan informasi didalam *World Wide Web*. Fungsi utama dari *Web server* adalah untuk menyimpan, memproses dan mengirimkan halaman-halaman web kepada klien. Komunikasi antar klien dan server menggunakan HTTP (Hypertext Transfer Protocol). Halaman-halaman web yang dikirimkan kebanyakan berupa dokumen HTML (Hypertext Markup Language). Apache, dan Nginx (dibaca: *engine x*) merupakan salah satu *web server* yang cukup banyak digunakan oleh *website-website* didunia. Pemilihan untuk menggunakan *web server* untuk suatu *website* tidak terlepas dari kelebihan dan kekurangan dari masing-masing *web server*. Apache memiliki kelebihan seperti *open source*, proses instalasi yang mudah, dan juga sebagai *web server* yang paling banyak digunakan diseluruh dunia (per Juni 2013, Apache diperkirakan melayani 54.2% dari *website* yang aktif) ^[5]. Sama halnya dengan Apache, Nginx merupakan *web server* yang juga bersifat *open source*. Selain itu, Nginx memiliki kelebihan seperti performa stabil, mudah dikonfigurasi, dapat lintas platform, dan juga menggunakan *resource* yang sedikit dibanding dengan Apache (terbukti dengan konsumsi RAM yang lebih sedikit pada tingkat performa yang tidak terlalu berbeda, pada perbandingan apache dan nginx) ^[4], serta dibuktikan pada sebuah penelitian bahwa besar *throughput* data *video* yang melalui *web server* Nginx lebih besar daripada Apache, dan lebih besarnya kegagalan pada *web server* Apache dibanding Nginx ^[3]. Telkom University adalah kampus berbasis teknologi informasi, pada kampus ini di bagian Fakultas Teknik Elektro dan Komunikasi dibutuhkan sebuah media penyebar informasi yang relevan dengan perkembangan jaman pada jajaran pegawai di Telkom University terkhusus gedung P dengan memanfaatkan media berbasis *web* dan jaringan yang dapat di akses melalui *laptop* ataupun *smartphone*, sehingga dibutuhkan sebuah *aplikasi* yang mumpuni untuk memenuhi permintaan tersebut. Maka dalam penelitian ini akan digunakan Nginx sebagai *web server* dalam perancangan *aplikasi live video streaming* di gedung P Telkom University karena sifatnya yang *open source* dan penggunaan *resource* yang minimal dengan performa yang sama dengan Apache, dan diharapkan dapat memenuhi kebutuhan sebagai media berbagi informasi antar pegawai, dengan *resource* yang se-minimum mungkin.

1.2 Perumusan Masalah

Permasalahan yang dirumuskan dalam penelitian ini adalah :

1. Bagaimana membuat media yang tepat untuk dapat digunakan sebagai media pertukaran informasi antar pegawai Telkom di gedung P Telkom University?
2. Bagaimana membuat *aplikasi live video streaming* di gedung P Telkom University yang *open source* dengan *resource* yang minimum?
3. Bagaimana membuat *aplikasi live video streaming* yang dapat mengakomodasi keterbatasan *bandwidth* pada jaringan?
4. Bagaimana membuat *aplikasi live video streaming* yang dapat disimpan kedalam basis data agar dapat dilihat lain waktu?

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah diatas, maka tujuan dari penelitian ini adalah :

1. Membuat *aplikasi live video streaming* kampus berbasis *web* sebagai media berbagi informasi antar pegawai pada Telkom University
2. Penerapan Nginx sebagai *web server* dalam perancangan *aplikasi live video streaming* di Gedung P Telkom University
3. Penerapan *aplikasi live video streaming* yang dapat diakses menggunakan semua perangkat yang memiliki *Browser* yang telah mendukung HTML5
4. Membuat *aplikasi live video streaming* dengan fitur VOD (*Video On Demand*) yang terintegrasi dengan sistem basis data agar video dapat disimpan dan dilihat lain waktu

1.4 Metodologi Penelitian

Metodologi penelitian pada pengerjaan tugas akhir ini adalah :

1. *Prepare*: Pada tahap awal ini proses yang dilakukan adalah mempersiapkan segala sesuatu, dimulai dari persiapan kebutuhan aplikasi agar dapat melakukan analisa awal untuk menggunakan NginX sebagai *web server* dalam perancangan *live video streaming* di Telkom University.
2. *Plan*: Dalam tahap ini, yang dilakukan adalah perencanaan *software* apa saja yang digunakan dalam perancangan aplikasi ini. Juga menentukan skenario yang dilakukan untuk melakukan pengujian terhadap aplikasi.
3. *Design*: Dalam tahap ini dilakukan penentuan terhadap fitur-fitur apa saja yang akan dimasukkan kedalam aplikasi, menggunakan protokol apa, menggunakan tools apa saja sehingga dapat merancang sebuah aplikasi yang optimal.
4. *Implement*: Tahap ini adalah tahap dimana sistem mulai dibuat berdasarkan fitur-fitur yang telah direncanakan.
5. *Operate*: Tahap pengujian pada sistem yang telah dibangun, diuji berdasarkan skenario yang telah ditentukan pada tahap *Plan*.
6. *Optimize*: Tahap optimasi ini dilakukan dengan menganalisa kinerja sistem yang sudah dibuat apakah sudah berjalan dengan baik, dan menambahkan atau memperbaiki fitur yang belum ada atau yang belum bekerja secara sempurna sebelumnya.

2. Dasar Teori

2.1 Streaming Video ^{[2][3]}

Streaming video merupakan sebuah teknologi yang memungkinkan pen-transmisian data berupa *audio* dan *video*. Pengkodean dari *video* atau citra bergerak yang bisa digunakan adalah MPEG (Moving Picture Expert Group). Adapun beberapa jenis standar yang digunakan diantaranya : MPEG 1, MPEG 2 , MPEG 3 dan MPEG 4. Adapun beberapa komponen yang digunakan dalam streaming video adalah : video Encoder dan Decoder, Receiver.

2.2 HTTP ^[1]

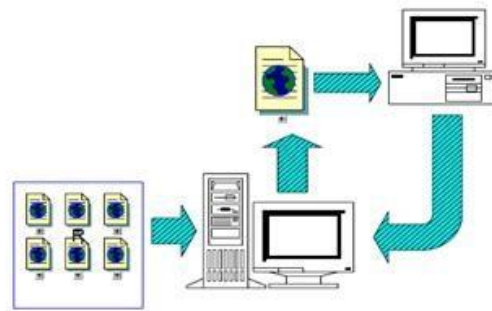
Hypertext Transfer Protocol (HTTP) adalah sebuah protokol aplikasi untuk didistribusikan , dan di kolaborasikan sebagai sistem informasi *hypermedia*. HTTP adalah dasar dari komunikasi data untuk *World Wide Web*. Hypertext terstruktur teks yang menggunakan *link* logis (*hyperlink*) antara *node* yang berisi teks . HTTP adalah protokol untuk bertukar atau saling mengirim *hypertext*. Kelebihan pengantaran paket data melalui HTTP melingkupi kemampuannya untuk menggunakan *web server* konvensional, mudah meng-eksploitasi layanan CDN (*Content Delivery Network*) dan *web caches*, serta mudah melintasi *firewalls*.

2.3 Web Server ^{[4][5]}

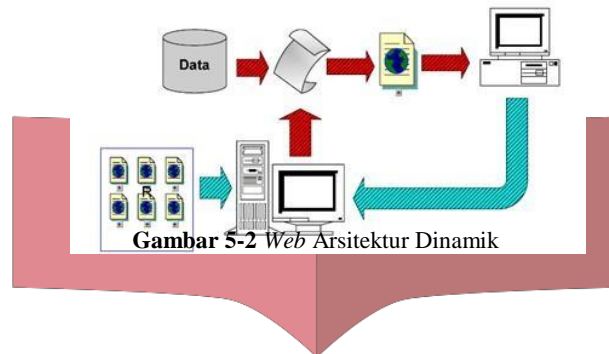
Web server merupakan sebuah system komputer yang memproses permintaan (*request*) melalui HTTP, *basic* dari protokol jaringan yang digunakan untuk mendistribusikan informasi didalam *World Wide Web*. Fungsi utama dari *Web server* adalah untuk menyimpan, memproses dan mengirimkan halaman-halaman *web* kepada klien. Komunikasi antar klien dan *server* menggunakan HTTP (*Hypertext Transfer Protocol*). Halaman-halaman *web* yang dikirimkan kebanyakan berupa dokumen HTML (*Hypertext Markup Language*). Adapun macam-macam *web server* diantaranya adalah sebagai berikut:

- Apache Web Server (<http://www.apache.org>)
- Internet Information Service, IIS (<http://www.microsoft.com/iis>)
- Xitami Web Server (<http://www.xitami.com>)
- Nginx Web Server (<http://www.nginx.org>)

Gambaran umum arsitektur web baik statik maupun dinamik dapat anda lihat pada gambar berikut:



Gambar 5-1 Web Arsitektur Standar



Gambar 5-2 Web Arsitektur Dinamik

2.4 NginX^[6]

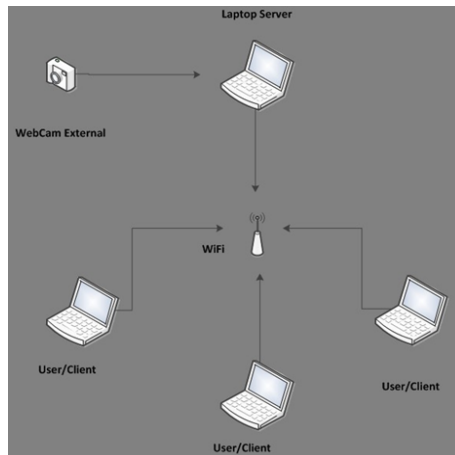
Nginx adalah web gratis, *open-source*, memiliki kinerja tinggi sebagai server HTTP dan *reverse proxy*, serta *server proxy IMAP / POP3*. *Nginx* dikenal untuk kinerja tinggi, stabilitas, kaya akan fitur dan konsumsi sumber daya yang rendah. *Nginx* adalah salah satu dari beberapa *server* yang didesain untuk mengatasi masalah sumber daya. Tidak seperti *server* tradisional, *nginx* tidak bergantung pada *thread* untuk menangani permintaan. Sebaliknya ia menggunakan (*asynchronous*) arsitektur yang lebih efektif menangani *scalable-event*. Arsitektur ini membutuhkan *memory* yang sangat kecil dan yang paling utama adalah beban terhadap *memory* yang dapat diprediksikan. Bahkan jika *server* tidak berharap untuk menangani ribuan permintaan simultan, *server* masih bisa mendapatkan keuntungan dari *NginX* dengan kinerja tinggi dan *footprint* memori yang kecil. *NginX* dapat bekerja dari berbagai skala. dari VPS yang kecil semua dapat berjalan sampai ke kelompok besar server. *Nginx* menyokong beberapa situs visibilitas tinggi, seperti *Netflix*, *Hulu*, *Pinterest*, *CloudFlare*, *Airbnb*, *WordPress.com*, *GitHub*, *SoundCloud*, *Zynga*, *Eventbrite*, *Zappos*, *Media Temple*, *Heroku*, *RightScale*, *Engine Yard* *MaxCDN* dan banyak lainnya.

3. Pembahasan

Dalam penelitian tugas akhir ini, akan dibangun aplikasi *Live Video Streaming* berbasis *Web Browser*. Pengembangan sistem ini didasarkan kebutuhan atas media penyedia informasi antar pegawai dalam kampus Universitas Telkom. Pada kasus ini aplikasi akan diimplementasi di gedung P Fakultas Teknik Elektro Universitas Telkom, aplikasi *Live Video Streaming Server* yang dibangun dapat digunakan sebagai media penyebaran informasi antar pegawai sehingga dapat mempersingkat dan mempermudah penyampaian informasi dalam kampus, maupun *event-event* yang dapat direkam dan disebar kepada mahasiswa maupun dosen.

3.1 Gambaran Umum Aplikasi yang Akan Dibangun

Secara umum perancangan video live streaming dapat dilihat pada gambar sebagai berikut:



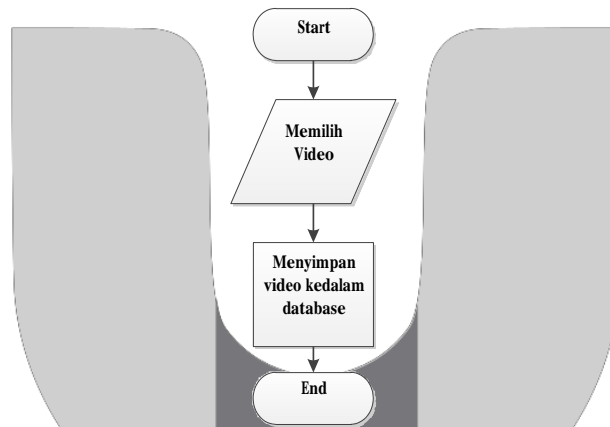
Gambar 1. Gambaran Umum Aplikasi

Berikut adalah penjelasan dari gambaran umum aplikasi *video live streaming*:

1. Objek ditangkap menggunakan *smartphone* Android /*webcam* PC eksternal.
2. Output dari *smartphone* Android/ *webcam* PC eksternal ditransmisikan ke PC *server*.
3. PC *server* melakukan encoding melalui *web server* NginX, *video* dikompresi menjadi format mp4.
4. *Video* dimunculkan pada halaman *web* pengguna.
5. Setelah *streaming* selesai (di-*stop*), *video* akan langsung disimpan di *server*.

3.2 Perancangan Fitur Video On Demand (VOD)

Flowchart Aplikasi



Gambar 2 flowchart mengupload Video (VOD)

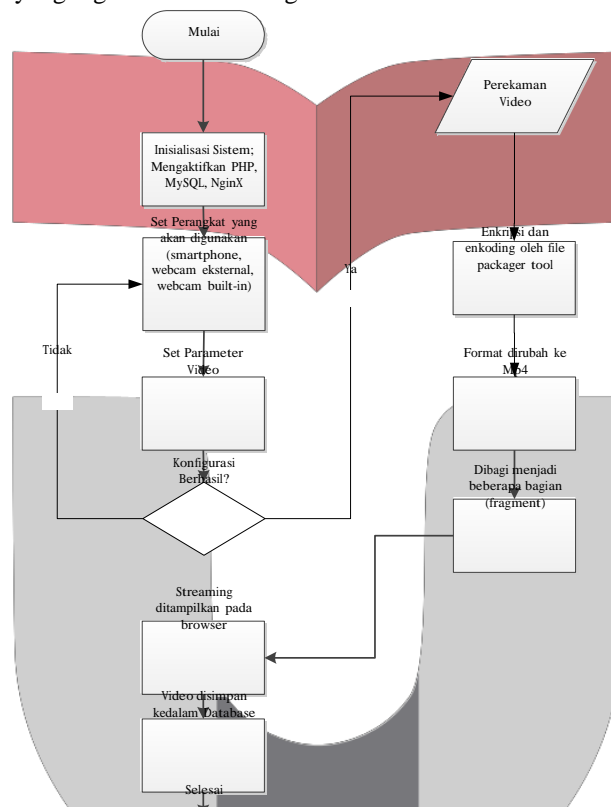


Gambar 3 flowchart menonton Video On Demand

Fitur *Video On Demand*, digambarkan pada flowchart 2 yaitu proses mengupload *video* ke dalam basis data dan flowchart 3 menunjukkan proses menonton *video* pada fitur VOD. Tiap *video* akan memiliki kategori masing-masing, yaitu mahasiswa dan dosen, dengan kategori mahasiswa dapat dilihat oleh pengguna dengan *privilege* mahasiswa maupun *privilege* dosen, dan *video* dengan kategori dosen hanya dapat dilihat oleh pengguna dengan *privilege* dosen. Tujuan dari pembagian kategori terhadap *video* dan pemberian *privilege* terhadap pengguna adalah untuk membatasi *video-video* yang memang ditujukan terhadap dosen agar tidak dapat dilihat oleh mahasiswa. Format *video* yang dapat diupload adalah *video* dengan format mp4, hal ini dilakukan atas pertimbangan bahwa media rekam (berupa *handphone* Android) yang digunakan pada pengujian aplikasi ini memiliki *format output video* mp4, dan ukuran dari format mp4 yang cenderung kecil.

3.3 Perancangan Fitur Live Video Streaming

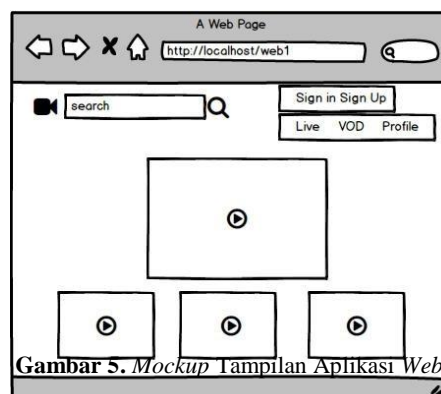
Proses pengembangan aplikasi *live video streaming* ini tergambar dalam Gambar 4 Flowchart aplikasi *live video streaming*. Seperti yang sudah dibahas sebelumnya, untuk mengoptimasi penggunaan *resource* secara minimum, maka *web server* yang digunakan adalah NginX [3].



Gambar 4 Flowchart Aplikasi Live Video Streaming

3.4 Perancangan Keluaran Video pada Web Browser

Rancangan tampilan antar muka Aplikasi Live Video Streaming pada Browser :



Gambar 5. Mockup Tampilan Aplikasi Web

Pada bagian *header* akan berisikan, *logo website*, fitur *search*, fitur *login*, *menu* untuk memilih fitur VOD, *Live Streaming*, dan *profile*, sedangkan pada bagian *content* akan berisikan keluaran *video* untuk *browser* menggunakan *flowplayer*, yang akan digunakan sebagai *output video live streaming* nanti, sedangkan pada bagian *footer* akan berisikan VOD-VOD yang telah disimpan dalam *database* yang dapat dilihat sesuai dengan *privilege user* yang tersedia, *web* dibangun dengan menggunakan bahasa pemrograman PHP, HTML, HTML5 dan *javascript*.

3.5 Implementasi Instalasi Program

Untuk Membuat dan menjalankan aplikasi *live video streaming* ini memerlukan instalasi *program* yaitu :

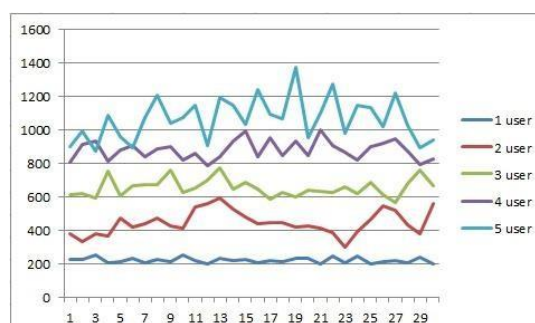
1. Menginstall *Winginx* untuk menjalankan *server apache* dan *database MySQL*, serta untuk dapat menjalankan fungsi-fungsi PHP dalam program.
2. Menginstall *Notepad++* untuk melakukan proses pengkodean dalam aplikasi *live video streaming*.
3. Menginstall *Media Server Configurator* sebagai wadah untuk menjalankan modul-modul bawaan *server NginX*.
4. Menginstall *Live Server Configurator* untuk melakukan konfigurasi terhadap alat-alat yang digunakan ketika melakukan *live video streaming*.
5. Menginstall *Droidcam* pada *windows* (PC yang digunakan sebagai *server*) dan *android* (alat yang digunakan sebagai media rekam) sebagai aplikasi agar dapat menggunakan *smartphone* *android* sebagai media rekam.
6. Menginstall *Mozilla Firefox* atau *browser* yang telah mendukung HTML5 secara penuh untuk menampilkan aplikasi *live video streaming*.

3.6 Pengujian Live Streaming

Untuk mengetahui seberapa besar beban yang ditimbulkan ketika menggunakan fitur *live streaming* pada aplikasi *live video streaming* berbasis *web* ini, maka akan dilakukan pengujian besaran *bandwidth* penggunaan aplikasi. Analisa besaran *bandwidth* dilakukan dengan menggunakan *webcam eksternal* sebagai media rekam untuk menentukan apakah aplikasi dapat menghasilkan gambar yang optimal tanpa mengorbankan *bandwidth* jaringan, selanjutnya mengubah jumlah *user* yang akan mengakses *server*, seberapa besar pengaruh jumlah *user* terhadap *bandwidth* yang digunakan oleh aplikasi ini. Untuk mengukur besar *bandwidth* yang dipakai, maka akan dilakukan proses *live streaming* dengan media rekam yang tidak bergerak (media rekam dibiarkan berdiri dalam posisi yang sama) agar tidak terjadi perbedaan *bitrate* yang terlalu besar diakibatkan oleh perubahan *frame* gambar yang terlalu banyak, dan akan dilakukan pencatatan *output data* dari *server* (dalam kasus ini PC *Laptop*), menggunakan *Network Meter*. Pencatatan akan dilakukan tiap 1 detik, selama 30 detik (30 interval), dengan jumlah *user* yang akan ditambah satu setiap satu kali pengujian telah selesai (pada kasus ini maksimal menggunakan 5 *user*). Nilai *bitrate* rata-rata akan dihitung dengan rumus sebagai berikut :

$$\text{bitrate} \left(\frac{\text{bit}}{30\text{s}} \right) = \left(\frac{b_1 + b_2 + \dots + b_{30}}{30} \right) \quad (1)$$

Dimana b_1 - b_{30} adalah besar *bitrate* pada detik ke- n , dengan perhitungan tersebut maka bisa didapatkan angka rata-rata penggunaan aplikasi *live streaming* dalam jarak waktu 30 detik. Hal ini untuk menentukan apakah aplikasi ini terlalu berat untuk digunakan dalam lingkungan kampus sehingga dapat mempengaruhi dan membebani jaringan kampus atau tidak. *Browser* yang digunakan dalam pengujian *live streaming* ini adalah *Mozilla Firefox*, karena telah mendukung HTML5 dan dapat menggunakan *plug-in video berbasis flash*.



Gambar 6 Grafik Total *Bitrate* dan Rata-Rata *Bitrate* Pengujian

Tabel 1 Jumlah *Bitrate* dan Rata-rata *Bitrate* Pengujian *Live Streaming*

Jumlah User	Total <i>Bitrate</i> (kbit/30s)	Rata-rata <i>Bitrate</i> (kbit/s)
1	6679,1	222,6
2	13420,5	447,4
3	19702,5	656,8
4	26428,1	880,9
5	32063,5	1068,8

Dapat dilihat dari hasil pengujian bahwa rata-rata *bitrate* yang dibutuhkan oleh 5 *user* yang melakukan koneksi kepada server secara bersamaan hanya membutuhkan *bandwidth* sebesar 1068 kbit/s atau 1,043 Mbit/s, *bitrate* sekian tergolong kecil karena kapasitas *bandwidth* dari sebuah jaringan LAN secara umum berkisar antara 100-300 Mbit/s, sehingga secara teori sebuah jaringan LAN secara umum dapat menampung jumlah *user* berkisar antara 475-1438 *user* dalam waktu yang bersamaan. Hasil pengujian aplikasi menunjukkan 1 pengguna memiliki rata-rata penggunaan *bitrate* sebesar 222,6 kbit/s, dengan *delay* sebesar 2-4 detik menurut pengamatan kami. Berdasarkan standar ITU-T G.1010^[8] bahwa *streaming video* satu arah, dengan *bitrate* antara 16-384 kbit/s memiliki standar *delay* kurang dari 10 detik. Sehingga dapat disimpulkan bahwa aplikasi *live video streaming* berbasis *web* ini telah memenuhi standar ITU-T G.1010. Di kampus Telkom University, *bandwidth* yang dimiliki adalah 1 Gbit/s, secara teori jaringan kampus dapat menampung hingga 4908 *user* yang mengakses aplikasi *live streaming* secara bersamaan.

4. Kesimpulan

Web server Nginx dapat diimplementasikan pada aplikasi *live video streaming* berbasis *web* dengan penggunaan *bandwidth* yang relatif kecil dan fungsi yang telah dapat memenuhi kebutuhan sebagai media pertukaran informasi. Besar *bandwidth* yang digunakan ketika melakukan *live video streaming* sebanyak 1 *user* sebesar 222,6 kbit/s dan 5 *user* adalah sebesar 1068,8 kbit/s, dengan menggunakan protokol TCP-*Unicast*, nilai tersebut cenderung kecil karena besar *bandwidth* kampus Telkom University yang cukup besar yaitu 1 Gigabit/s. Dalam kasus pengembangan *live video streaming* yang paling penting adalah bagaimana sistem dapat mengirim gambar *video* tepat sesuai kondisi *real-time* yang ditangkap oleh media rekam tanpa ada gambar yang hilang, kondisi minimal aplikasi dianggap berhasil jika telah dapat menampilkan *live video streaming* dengan sempurna.

Daftar Pustaka

- [1] H. Zhang. *Internet video: The 2011 perspective (keynote talk)*. IWQoS, 2011.
- [2] **HTTP Dynamic Streaming on the Adobe Flash Platform** [Technical White Paper], diakses pada 15-06-2015 pada pukul 02.39 WIB.
- [3] J. Summers, T. Brecht, D. Eager, and B. Wong. **Methodologies for generating HTTP streaming video workloads to evaluate web server performance**. In *5th Annual International Systems and Storage Conference (SYSTOR)*, 2012.
- [4] Kamas Muhammad, 2014. **Apache vs Nginx**. (<http://sokam.or.id/log/592/apache-vs-nginx>, diakses 20 Nopember 2014, Pukul 17.23 WIB).
- [5] Netcraft, 2013. **June 2013 Web Server Survey**. (<http://news.netcraft.com/archives/2013/06/06/june-2013-web-server-survey-3.html>, diakses 25 Nopember 2014, Pukul 01.37 WIB).
- [6] NginX Website. **NginX Wiki's Documentation**. (<https://www.nginx.com/resources/wiki/>, diakses 13 Januari 2015, pukul 13.50 WIB).