

IMPLEMENTASI DAN ANALISIS PERFORMANSI LAYANAN VIDEO STREAMING DAN CCTV BERBASIS RED5 PADA VIRTUALSERVER

Rahmat Juniar Dwiansyah^{*)}, Yuli Christyono, and Adian Fatchur Rochim

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}*email:rahmatjuniar.d@gmail.com*

Abstrak

Bila dilihat dari terus meningkatnya penggunaan internet sekarang ini, suatu aplikasi multimedia menjadi sangat dibutuhkan. Salah satu layanan atau aplikasi yang sangat menarik untuk diterapkan adalah video streaming, beberapa tahun belakangan ini video streaming menjadi salah satu layanan yang paling banyak digunakan oleh kebanyakan orang. Video streaming yang bersifat live streaming juga dapat digunakan untuk memantau suatu lokasi tertentu atau berperan sebagai CCTV. Pada penelitian ini dirancang, diimplementasikan dan dianalisis suatu sistem layanan video streaming dan CCTV yang berbasis Red5 pada virtual server untuk menganalisis kualitas dari layanan tersebut, juga untuk melihat apakah layanan ini sudah layak diterapkan. Dari hasil pengujian dan analisis dapat diketahui bahwa sistem layanan video streaming dan CCTV berbasis Red5 pada virtual server berjalan dengan baik dan layak untuk digunakan. Nilai delay, packet loss dan throughput masih dalam kategori baik dan memenuhi standar berdasarkan ITU-T G1010.

Kata Kunci : Video streaming, CCTV, Red5, ITU-T G1010, virtual server, web.

Abstract

The increase of the use of internet nowadays leads the needs of multimedia application. One of appealing multimedia applications to be utilized in recent years is video streaming that has been used by most of people. Video streaming embracing a characteristic of live streaming can be utilized to monitor a certain situation to discover something about it or to be as CCTV. By virtue of the aforementioned phenomenon, the research reported in this paper centered around the issue of the designed, implementation and the analysis of a service system of video streaming and CCTV. It particularly aims to analyze the quality of a service system of video streaming and CCTV based on Red5 in virtual server and to figure out whether this system is adequate to be implemented. Regarding the experiment and the analysis, the findings revealed that service system of video streaming based on Red5 in virtual server worked properly and this service system was adequate to be utilized. Besides, according to ITU-T G1010, level of delay, packet loss and throughput were in suitable category and of an acceptable standard.

Keywords : video streaming, CCTV, Red5, ITU-T G1010, virtual server, web.

1. Pendahuluan

Dilihat dari terus meningkatnya penggunaan internet sekarang ini, suatu aplikasi multimedia menjadi sangat dibutuhkan. Hal ini dapat dilihat dari semakin beragamnya penyampaian informasi yang tidak hanya berupa teks, tetapi juga meliputi suara, data, gambar maupun video. Hal ini juga yang merupakan salah satu sudut pandang utama konsep NGN (*Next Generation Networks*) yaitu layanan yang meliputi suara, data, video dan Internet. Hal terpenting dalam suatu konsep NGN adalah layanan yang berbasis IP (*Internet Protocol*).

Salah satu layanan atau aplikasi yang sangat menarik untuk diterapkan adalah video *streaming*, beberapa tahun belakangan ini *video streaming* menjadi salah satu layanan yang paling banyak digunakan oleh kebanyakan orang. Penerapan teknologi video *streaming* menjadi solusi baru dalam penyampaian multimedia berbasis jaringan.

Video *streaming* juga dapat digunakan untuk memantau suatu lokasi tertentu atau berperan sebagai CCTV. Video *streaming* sebagai CCTV ini merupakan salah satu aplikasi video *streaming* yang bersifat *live streaming*

yang memungkinkan suatu lokasi dapat terpantau dan didistribusikan melalui jaringan Internet atau lokal secara langsung.

Pada penelitian sebelumnya yang dilakukan oleh Aldy Prasetyo, telah dirancang penerapan teknologi video *streaming* berbasis Red5 server yang diimplementasikan dalam sebuah aplikasi Internet-TV^[10]. Penelitian tersebut melatar belakangi penelitian ini, pada penelitian ini akan dirancang dan diimplementasikan suatu sistem video *streaming* dan CCTV yang berbasis Red5 pada *virtual server* dengan tujuan untuk mengetahui apakah layanan video *streaming* dan CCTV ini sudah layak diterapkan. Penelitian juga dilakukan untuk menganalisis kualitas dari layanan tersebut berdasarkan parameter-parameter QoS (*Quality of Service*).

Pembahasan pada penelitian ini dibatasi pada hal-hal berikut :

1. *Virtual server* menggunakan perangkat lunak Proxmox.
2. *Server* menggunakan Red5 server yang merupakan *server open source*.
3. Pengguna menggunakan *web browser* untuk mengakses layanan.
4. Sistem menggunakan IPv4 yang terhubung dengan jaringan lokal .
5. Penelitian hanya membahas kualitas dan performansi pada layanan video *streaming* dan CCTV yang dilewatkan pada jaringan lokal dari Fakultas Teknik Universitas Diponegoro.
6. Percobaan dan pengambilan data hanya dilakukan di wilayah sekitar kampus Fakultas Teknik Universitas Diponegoro.
7. Pembahasan bahasa pemrograman hanya dilakukan pada antarmuka untuk pengguna yaitu *web*.
8. Tidak membahas keamanan pada jaringan yang dipakai pada perangkat keras baik di sisi *server* serta infrastruktur jaringan dimana sistem ini diimplementasikan.
9. Tidak membahas mengenai jaringan kampus Universitas Diponegoro.

2. Metode

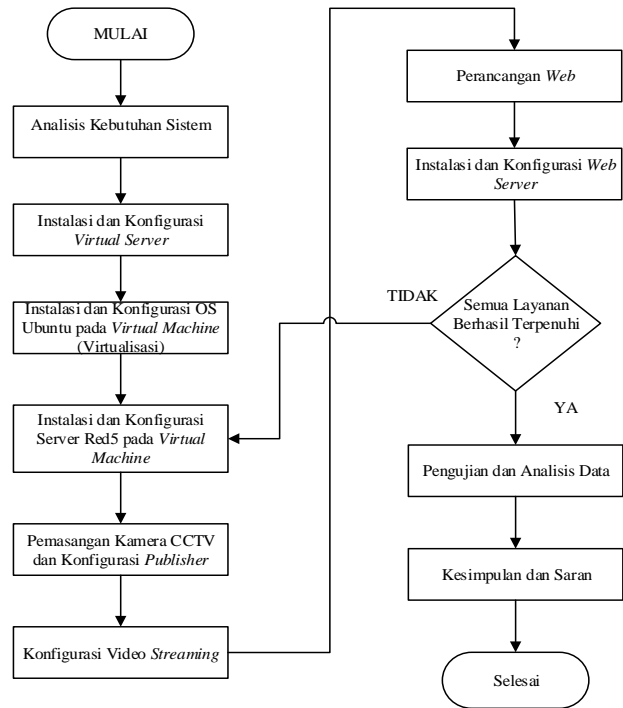
2.1 Perancangan Sistem

Tahapan-tahapan dari perancangan sampai implementasi terlihat dalam bagan-alir untuk memudahkan proses perancangan sistem. Bagan-alir perancangan dan implementasi seperti yang terlihat pada **Gambar 1**.

penelitian ini dibangun dengan beberapa komponen seperti pada **Gambar 2** dibawah ini.



Gambar 2 Komponen Sistem



Gambar 1Bagan-alir Perancangan dan Implementasi

- *Publisher* berperan sebagai *encoder*, untuk mengubah data-data yang ditangkap dari alat penangkap (video/audio).*Publisher* juga dapat melakukan perekaman video hasil tangkapan dari alat penangkap (video/audio).
- *Server Red5* memiliki 2 fungsi utama, yang pertama berfungsi untuk menerima *stream* dari *publisher*, sedangkan yang kedua berfungsi untuk mengirimkan *stream* video (video *streaming*) yang terdapat pada *server*, sehingga video tersebut dapat diakses *client*.
- *Client* merupakan sisi pengguna, yang menampilkan *Flash movie* yang merupakan hasil *stream* dari *server Red5*.

3. Hasil dan Analisa

3.1 Implementasi Sistem

Perangkat-perangkat yang digunakan dalam perancangan sistem dibagi ke dalam dua jenis, yaitu perangkat keras dan perangkat lunak. Perangkat lunak *Server Red5* berada dalam satu komputer *server* dengan menggunakan sistem operasi Ubuntu 12.04 yang dipasang pada perangkat lunak Proxmox yang merupakan sebuah perangkat virtualisasi *server*. Berikut ini komponen perangkat keras yang digunakan seperti pada **Gambar 3**.

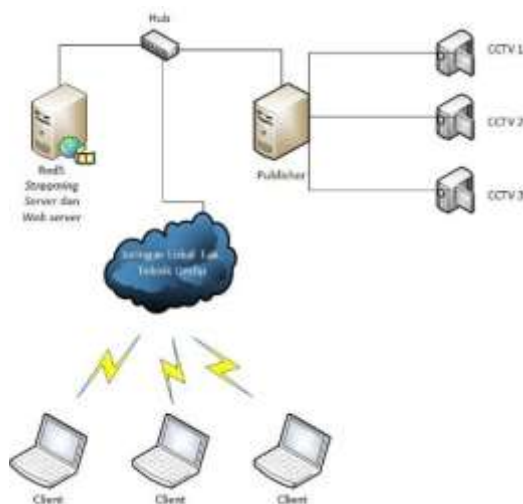
3.1.1 Instalasi dan Konfigurasi Proxmox

Proxmox pada penelitian ini bersifat *open source* berfungsi untuk menjalankan mesin virtual berbasis KVM berupa sistem operasi Linux yang nantinya digunakan

sebagai *server* Red5 dengan layanan *video streaming* dan CCTV.

Beberapa tahap yang penting dan perlu diperhatikan pada proses instalasi adalah proxmox akan menggunakan seluruh ruang penyimpanan yang ada, pembuatan partisi akan ditangani secara otomatis dan ruang penyimpanan yang digunakan diformat total, maka perlu diperhatikan untuk mem-backup dahulu data sebelum memulai instalasi, tahap yang kedua yang perlu diperhatikan adalah mengatur *password* untuk *user root* yang digunakan untuk mengakses *web administrator*, tahap ketiga yang juga penting adalah pengaturan jaringan (*IP address, netmask, gateway*).

Proses virtualisasi di Proxmox dilakukan melalui *web browser* dengan mengakses alamat IP dari Proxmox yang sudah dikonfigurasi sebelumnya yaitu 10.31.17.163, dan akan dibuat virtualisasi berupa sistem operasi Linux yang nantinya digunakan sebagai *server* Red5 dengan layanan *video streaming* dan CCTV.



Gambar 3 Komponen Perangkat Keras

3.1.2 Instalasi dan Konfigurasi *Server* Red5

Instalasi *server* Red5 dilakukan pada sistem operasi Linux 12.04 yang terpasang pada mesin *virtual server* Proxmox yang telah di instal dan dikonfigurasi sebelumnya. Sebelum proses instalasi dan konfigurasi, terlebih dahulu lakukan konfigurasi IP di Linux dan tentukan IP yang digunakan untuk koneksi ke jaringan lokal atau internet, dalam penelitian ini IP yang digunakan untuk *server* Red5 adalah 10.31.17.164.

Beberapa komponen perangkat lunak penting yang dibutuhkan sebagai pendukung dari *server* Red5 yaitu paket-paket Java, Ant yang digunakan untuk proses kompilasi dan membangun kelas-kelas Java, dan juga *subversion*. *Server* Red5 menggunakan *framework* Java

sehingga komponen-komponen pendukung tersebut sangat dibutuhkan.

Server Red5 menggunakan Flash oleh karena itu diperlukan juga *plugin* Flash yang terpasang baik pada sisi *server* maupun pada sisi *client*. Tahap selanjutnya adalah proses instalasi *server* Red5, langkah-langkah nya adalah sebagai berikut :

1. Mempersiapkan paket *server* Red5 yang bisa didapat dengan mengetikkan perintah berikut di *terminal* :
`#wgethttp://www.red5.org/downloads/red5/1_0_1/red5-1.0.1.tar.gz`
2. Mengekstrak *filered5-1.0.1.tar.gz* yang telah diunduh dengan mengetikkan perintah berikut di *terminal* :
`#sudo tar xvFz red5-1.0.1.tar.gz`
3. Memindahkan seluruh isi *folderred5-server-1.0* yang merupakan hasil ekstrak dari *file red5-1.0.1.tar.gz* ke *folder* baru yaitu *folder red5*. Perintah yang dilakukan adalah :
`#sudo mv red5-server-1.0 red5`
4. Memindahkan *folder red5* tadi ke direktori */usr/share/* dengan mengetikkan perintah :
`#sudo mv red5 /usr/share/`
5. Mengakses direktori */usr/share/red5/* dan menjalankan *server* Red5 dengan mengetikkan perintah berikut di *terminal* :

```
#sudo cd /usr/share/red5/
#sh red5.sh
```

Akan keluar pada *terminal* :

```
Running on Linux
Starting Red5
Red5 root: /usr/share/red5
.....
[INFO][Launcher:/installer]org.red5.serv
er.service.Installer -Installer service
created
```

Server Red5 sudah berjalan namun tidak berjalan secara otomatis pada saat *booting* atau saat *reboot*. *Server* Red5 dapat berjalan secara otomatis pada saat *booting* atau saat *reboot*, jika sudah dilakukan pembuatan *file* skrip *init* untuk Red5 di direktori */etc/init.d*.

Server Red5 dapat dijalankan, di-stop atau di-restart dengan menggunakan perintah :

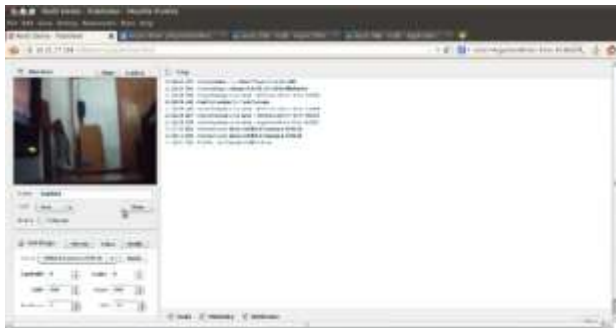
```
#!/etc/init.d/red5 {start|stop|restart} atau
#service red5 {start|stop|restart}
```

3.1.3 Konfigurasi *Publisher*

Publisher berperan sebagai *encoder* yaitu untuk mengubah data-data yang ditangkap dari alat penangkap (*video/audio device*) menjadi format file lain yang dapat diteruskan ke *server* Red5, dan *server* Red5 yang akan mempublikasikan data-data *stream* tersebut. penelitian ini menggunakan kamera USB *Webcam* sebagai alat penangkap video dan data-data video diubah menjadi Flash video. USB *Webcam* diposisikan sebagai kamera CCTV dan ditempatkan di lokasi-lokasi yang

akan dipantau. Setelah kamera terpasang, konfigurasi *publisher* dapat dilakukan dengan mengakses alamat URL <http://10.31.17.164:5080/demos/publisher.html> pada *web browser PC publisher*.

Setelah konfigurasi selesai dan *publisher* sudah terhubung dengan *server Red5* maka layanan CCTV sudah bisa di-*publish*. Terdapat 3 tipe *publish* yang dapat dipilih yaitu *live* apabila *publisher* hanya menampilkan video secara langsung tanpa melakukan perekaman, *record* dan *append* apabila *publisher* menampilkan video secara langsung sekaligus melakukan perekaman. Hasil perekaman video dari *publisher* tersimpan dalam folder `/usr/share/red5/webapps/oflademo/streams` yang terdapat pada *PC server*. Berikut ini tampilan *publisher* seperti yang terlihat pada Gambar 4 di bawah ini.



Gambar 4 *Publisher*

3.1.4 Konfigurasi Video Streaming

Untuk layanan video *streaming* konfigurasi dilakukan dengan menambahkan *file-file* video yang ingin di-*streaming*-kan dan disimpan dalam folder `/usr/share/red5/webapps/oflademo/streams` yang terdapat pada *PC server*. Penambahan *file-file* video pada penelitian ini dilakukan dengan menggunakan program WinSCP yang merupakan SFTP, SCP, dan FTP klien untuk Windows. WinSCP memungkinkan transfer file secara aman antara komputer lokal dengan *virtual server*. Format *file* video yang dapat dimainkan pada layanan video *streaming* yaitu *flv*, *mp4*, dan *mov*.

3.1.5 Perancangan Web

Web digunakan pada sisi *client* untuk dapat menjalankan dan menampilkan layanan video *streaming* dan CCTV. *Client* dapat menggunakan layanan video *streaming* dan CCTV dengan mengakses *web browser* dari PC atau laptop nya masing-masing yang terhubung dengan jaringan lokal, dan *web browser* tersebut harus sudah terpasang *plugin* Flash. *Client* membuka alamat URL dari *web* yang digunakan untuk menampilkan layanan video *streaming* dan CCTV.

Perancangan *web* dilakukan dengan terlebih dahulu membuat *layout web* dan selanjutnya adalah membuat

halaman *web* berdasarkan *layout web* yang sudah dibuat ke dalam skrip-skrip dengan menggunakan HTML (*HyperText Markup Language*) dan CSS (*Cascading Style Sheet*).

Halaman *web* yang dibuat dalam perancangan *web* ini ada 2 yaitu halaman untuk layanan CCTV dan video *streaming*. Yang pertama adalah halaman *web* layanan CCTV, untuk membuat halaman *web* CCTV dibuat skrip HTML yang disimpan dan diberi nama *index.html*, agar halaman *web* yang ditulis dalam dokumen HTML tersebut lebih teratur dan menarik tampilannya maka diperlukan sentuhan CSS, maka dibuat juga skrip CSS yang disimpan dan diberi nama *styles.css*. Selanjutnya adalah halaman *web* layanan video *streaming*, untuk membuat halaman *web* video *streaming* dibuat skrip HTML yang disimpan dan diberi nama *webvideo.html* dan dengan skrip CSS yang disimpan dan diberi nama *styles2.css*. Skrip HTML yang telah dibuat akan memanggil skrip CSS nya masing-masing pada saat skrip HTML tersebut dijalankan di *web browser* sehingga tampilan *web* dari skrip HTML lebih teratur dan menarik.

Web harus dapat terhubung dengan *server Red5* agar *web* dapat menampilkan dan menjalankan layanan video *streaming* dan CCTV, untuk itu diperlukan *Flash media player* yang sudah mendukung *streaming RTMP*. *Web* ini menggunakan *Flash media player* JWPlayer yang dipasang bersama dengan skrip-skrip HTML dan CSS di *web server* untuk dapat menampilkan *streaming RTMP* dari *server Red5*.

Skrip-skrip HTML disisipkan skrip dari *media player* yang nantinya akan menghubungkan *web* dengan *server Red5*. Skrip *media player* disisipkan di dalam tag `<head>` seperti berikut ini :

```
<script type="text/javascript" src="jwplayer.js"></script>
```

Skrip diatas fungsinya adalah menyatakan bahwa skrip melibatkan file *JavaScript* yang bernama *jwplayer.js* yang merupakan *source JavaScript* dari JWPlayer. Selanjutnya skrip *media player* disisipkan di dalam tag `<body>` pada lokasi dimana *player* akan muncul.

1) Skrip *index.html*

```
<video
src="cctv1"
height="300"
id="container"
width="520">
</video>
<script type="text/javascript">
jwplayer("container").setup({
flashplayer: "player.swf",

streamer:"rtmp://10.31.17.164/oflaDemo/",
volume: 50,
autostart: true
});
</script>
```

Skrip diatas merupakan skrip untuk memanggil pengaturan *player* dari JWPlayer pada tag <video>, dalam tag <video> terdapat nilai dari ukuran layar, nama *id* dan *source* dari video yang akan dimainkan, *source* video tersebut merupakan nama *file* video yang ditangkap oleh komputer *publisher* dan dikirimkan ke *server* Red5 yang kemudian hasil *stream* tersebut dikirim untuk ditampilkan oleh *player*. Skrip `jwplayer ("container") .setup ()` JWPlayer memeriksa tag <video> dengan *id* "container" dan memuat atributnya sebagai pilihan konfigurasi. Di dalam `setup ()` skrip menyediakan daftar pilihan untuk mengkonfigurasi *player*. Beberapa pilihan konfigurasi dalam skrip tersebut adalah seperti berikut ini :

1. `flashplayer: "player.swf"` menyatakan lokasi dari *flashplayer* "player.swf" yang akan dimuat.
2. `streamer:"rtmp://10.31.17.164/oflaDemo/"` menyatakan lokasi alamat IP dari *server* Red5 yang akan mengirimkan *streaming* video.
3. `volume: 50` menyatakan nilai dari volume saat video dimainkan.
4. `autostart: true` menyatakan bahwa *player* akan otomatis mulai memutar video saat halaman dimuat.

2) Skrip *webvideo.html*

```
<video
height="590"
id="container"
width="680">

<script type="text/javascript">
jwplayer("container") .setup({
flashplayer: "player.swf",
autostart: true,
volume: 50,
playlist: [
{
file:"hobbit_vp6.flv",
streamer:"rtmp://10.31.17.164/oflaDemo/"
, title: "Trailer Hobbit (flv)",
duration: 151 }
],
"playlist.position": "bottom",
"playlist.size": 150
});
</script>
```

Sama seperti pada skrip *index.html*, skrip diatas merupakan skrip untuk memanggil pengaturan *player* dari JWPlayer pada tag <video>. Yang membedakan adalah di dalam `setup ()` pada skrip *webvideo.html* ini terdapat tambahan `playlist: []` yang berisi kumpulan video yang dapat dimainkan oleh *player* dan pilihan-pilihan pengaturan yang dapat disesuaikan, tiap satu *item* video berisi :

1. `file:` menyatakan nama *file* video pada *server* Red5 yang akan di-*streaming*-kan.
2. `streamer:` menyatakan lokasi alamat IP dari *server* Red5 yang akan mengirimkan *streaming* video tersebut.

3. `title:` menyatakan judul *file* yang akan muncul di *Playlist*.
4. `duration:` menyatakan durasi dari *file* video dalam detik.

Tambahan yang lain adalah `"playlist.position":` yang menyatakan posisi dari *playlist* yang akan muncul, dan `"playlist.size":` yang menyatakan ukuran dari *playlist* yang akan muncul. Setelah tahapan-tahapan membuat *web* menggunakan HTML dan CSS selesai, selanjutnya adalah memasukkan seluruh komponen-komponen *web* tersebut kedalam satu *folder* yang diberi namawebcctv.

3.1.6 Instalasi Web Server dan Upload Komponen Web ke Dalam Web Server

Perancangan *web* selanjutnya adalah proses instalasi *web* server, agar *web* bisa berjalan dan ditampilkan pada *web* browser PC *client* maka diperlukan sebuah *web* server. penelitian ini menggunakan *web* server open source Apache yang dibundel dalam *software bundles* LAMPP (Linux Apache MySQL PHP Perl). *Web* server dipasang pada PC *server* sistem operasi Linux bersamaan dengan *server* Red5 yang sudah terhubung ke jaringan.

3.2 Hasil Pengujian

3.2.1 Pengujian Web layanan video streaming dan CCTV berbasis Red5

Pengujian dilakukan pada *web* sebagai aplikasi *client* layanan video *streaming* dan CCTV berbasis Red5 yang fungsinya untuk menjalankan dan menampilkan layanan video *streaming* dan CCTV. Pengujian ditujukan untuk menguji setiap komponen layanan yaitu video *streaming* dan CCTV untuk mengetahui apakah sistem telah bekerja sesuai dengan perancangan dan konfigurasi yang telah diberikan. Pengujian dilakukan dengan mengakses alamat URL dari *web* layanan video *streaming* dan CCTV yaitu `10.31.17.164/webcctv/`. Pengujian yang dilakukan adalah sebagai berikut :

1. Pengujian halaman *web* layanan CCTV

Halaman layanan CCTV dengan skrip *index.html* merupakan halaman awal yang pertama kali tampil ketika *client* mengakses *web* layanan video *streaming* dan CCTV dengan alamat URL `10.31.17.164/webcctv/`. Halaman ini digunakan untuk melihat hasil pantauan video secara langsung yang ditangkap dari kamera CCTV yang terhubung dengan komputer *publisher*, yang kemudian hasil tangkapan tersebut diteruskan ke *server* Red5. *Server* Red5 yang tugasnya mengirim hasil *stream* tersebut ke *client* dan ditampilkan melalui Flash *player* yang ada pada halaman *web* ini. Terdapat tiga layar video yang ditampilkan pada halaman ini yang merupakan tiga pantauan video langsung dari tiga lokasi yang berbeda.



Gambar 5 Tampilan halaman web layanan CCTV.

Gambar 5 menampilkan halaman web layanan CCTV dengan skrip *index.html* yang menjadi halaman awal yang pertama kali tampil ketika *client* mengakses web layanan video streaming dan CCTV, didalam halaman ini terdapat dua menu navigasi yaitu CCTV dan Video Streaming, untuk menu CCTV merupakan link menuju halaman ini yaitu halaman layanan CCTV sedangkan untuk menu Video Streaming merupakan link menuju halaman layanan video streaming. Terlihat bahwa tiga layar menampilkan video hasil tangkapan kamera CCTV di tiga lokasi, kamera CCTV tersebut terhubung dengan komputer *publisher* yang kemudian hasil tangkapan tersebut diteruskan ke *server* Red5. Video yang ditampilkan pada halaman ini merupakan hasil stream dari *server* Red5 yang terhubung dengan halaman web layanan CCTV ini. Hal ini membuktikan bahwa komputer *publisher* berhasil menangkap gambar dari kamera CCTV dan meneruskannya ke *server*, halaman web juga berhasil melakukan sambungan dengan *server* dan menampilkan video streaming dari *server*, dapat terlihat ketiga video CCTV yang ditampilkan terlihat jelas. Mode perekaman yang dilakukan lewat komputer *publisher* juga berhasil merekam gambar yang ditangkap dari kamera CCTV, dan bisa dimainkan pada *playlist* di halaman web layanan video streaming.

2. Pengujian halaman web layanan video streaming

Halaman layanan video streaming dengan skrip *webvideo.html* merupakan halaman yang digunakan untuk menampilkan video yang tersimpan pada *server* Red5. Video-video tersebut tersimpan dalam folder */usr/share/red5/webapps/oflademo/streams* yang terdapat pada *server* Red5. *Server* Red5 akan mengirimkan streaming video tersebut sesuai dengan permintaan dari *client* sehingga *client* dapat memainkan dan menonton video streaming tersebut melalui *Flash Player* yang ada pada halaman web ini. Pada halaman ini juga terdapat *Playlist* yang berisi kumpulan video yang dapat dimainkan dan dipilih oleh *client*. Gambar 8 menunjukkan tampilan dari halaman layanan video streaming.



Gambar 6 Tampilan halaman web layanan video streaming.

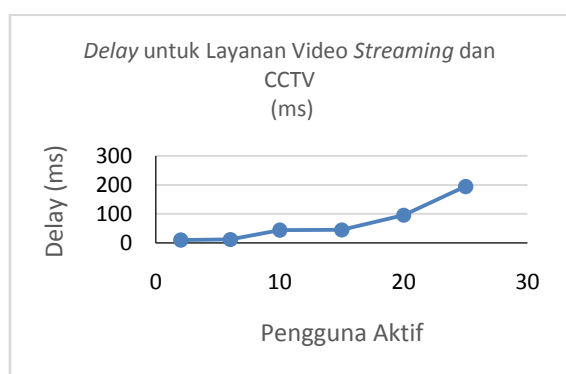
Gambar 6 menampilkan halaman web layanan video streaming dengan skrip *webvideo.html*. Halaman ini diakses melalui menu navigasi Video Streaming yang ada pada halaman web layanan CCTV. Terlihat bahwa *Flash Player* yang ada pada halaman web ini berhasil menampilkan video yang tersimpan pada *server* Red5. Format file video yang dapat dimainkan pada layanan video streaming yaitu flv, mp4, dan mov. *Client* dapat memainkan dan menonton video streaming yang ada pada *Playlist* dibawah layar, *server* Red5 akan mengirimkan streaming video sesuai dengan permintaan dari *client*. Hal ini membuktikan bahwa halaman web berhasil melakukan sambungan dengan *server* dan menampilkan video streaming dari *server*, dapat terlihat video yang ditampilkan terlihat jelas.

3.2.2 Pengujian Kualitas Layanan

Pengujian kualitas layanan video streaming dan CCTV berbasis Red5 pada *virtual server* dilakukan untuk mengukur aspek-aspek *Quality of Service (QoS)* yang terdiri dari pengukuran *delay* (waktu tunda), *packet loss* (kerugian paket), dan nilai *throughput*. Hal ini dilakukan untuk mengetahui kualitas dari layanan video streaming dan CCTV berdasarkan aspek-aspek tersebut sesuai dengan standar ITU-T G.1010. Untuk *jitter* tidak dilakukan pengukuran karena pada standar ITU-T G.1010 nilai *jitter* untuk aplikasi video tidak diketahui.

Pengujian layanan dilakukan dengan menerapkan beberapa rangkaian skenario penggunaan layanan. Pengujian dilakukan pada sisi *server*, *wireshark* mencatat arus pertukaran data yang terjadi antara *server* dengan *clients* setiap kali penggunaan layanan dilakukan kemudian mengetahui nilai dari parameter *Quality of Service (QoS)* yang diuji yaitu nilai *delay*, *packet loss* dan *throughput*. Beberapa rangkaian skenario penggunaan layanan sebagai berikut :

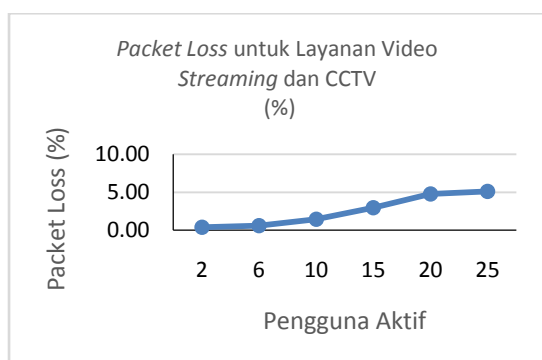
1. Penggunaan layanan dengan beberapa pengguna aktif
Gambar 7 menunjukkan grafik hasil pengukuran *delay* pada penggunaan layanan dengan beberapa pengguna aktif.



Gambar 7 Grafik hasil pengukuran *delay* pada penggunaan layanan dengan beberapa pengguna aktif

Standar nilai *delay* untuk aplikasi video satu arah berdasarkan ITU-T G.1010 adalah tidak lebih besar dari 10 s. Berdasarkan hasil pengukuran, terlihat dari grafik pada Gambar 4.14 bahwa nilai *delay* pada penggunaan layanan dengan beberapa pengguna aktif masih dalam kategori baik dan memenuhi standar *delay* berdasarkan ITU-T G.1010. Dapat terlihat pada grafik, semakin banyak jumlah pengguna aktif maka nilai *delay* semakin tinggi, nilai *delay* tertinggi yaitu pada saat penggunaan layanan dengan 25 pengguna aktif sebesar 195,09 ms, sementara penggunaan terbaik dengan nilai *delay* terendah adalah pada saat penggunaan layanan dengan 2 pengguna aktif sebesar 10,37 ms.

Gambar 8 menunjukkan grafik hasil pengukuran *packet loss* pada penggunaan layanan dengan beberapa pengguna aktif.

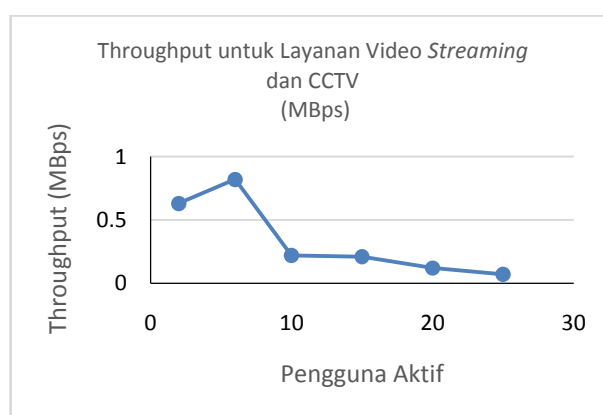


Gambar 8 Grafik hasil pengukuran *packet loss* pada penggunaan layanan dengan beberapa pengguna aktif

Standar nilai *packet loss* untuk aplikasi video satu arah berdasarkan ITU-T G.1010 adalah tidak lebih besar dari 1 %. Berdasarkan hasil pengukuran, terlihat dari grafik pada Gambar 4.18 bahwa nilai *packet loss* pada penggunaan layanan dengan beberapa pengguna aktif yang masih dalam kategori baik dan memenuhi standar *packet loss*

berdasarkan ITU-T G.1010 adalah ketika dilakukan pengukuran pada 2 sampai 6 pengguna aktif. Sedangkan nilai *packet loss* untuk 10 sampai 25 pengguna aktif melebihi standar nilai *packet loss* berdasarkan ITU-T G.1010. Dapat terlihat pada grafik, semakin banyak jumlah pengguna aktif maka nilai *packet loss* semakin tinggi, nilai *packet loss* tertinggi yaitu pada saat penggunaan layanan dengan 25 pengguna aktif sebesar 5,10 %, sementara penggunaan terbaik dengan nilai *packet loss* terendah adalah pada saat penggunaan layanan dengan 2 pengguna aktif sebesar 0,39 %.

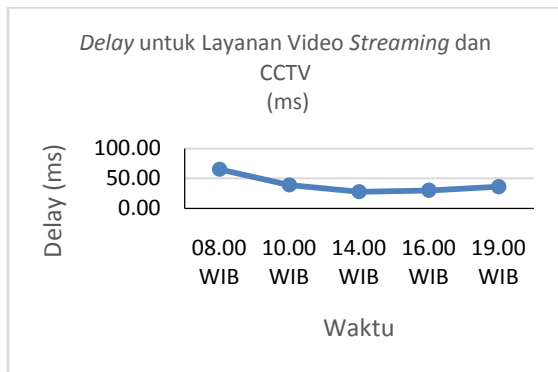
Gambar 9 menunjukkan grafik hasil pengukuran *throughput* pada penggunaan layanan dengan beberapa pengguna aktif.



Gambar 9 Grafik hasil pengukuran *throughput* pada penggunaan layanan dengan beberapa pengguna aktif

Berdasarkan hasil pengukuran, terlihat dari grafik pada Gambar 9 nilai *throughput* pada penggunaan layanan dengan beberapa pengguna aktif. Grafik menampilkan adanya penurunan nilai *throughput* ketika jumlah pengguna aktif bertambah dari 6 sampai 25 pengguna aktif. Namun pada penggunaan layanan dengan 2 sampai 5 pengguna aktif terjadi ketidaksesuaian dimana seharusnya nilai *throughput* nya lebih besar, dikarenakan dengan jumlah pengguna yang meningkat maka nilai *throughput* seharusnya akan semakin menurun. Hal ini disebabkan oleh sejumlah faktor yang terjadi pada saat pengukuran seperti penurunan *signal* jaringan dalam media jaringan lokal, dan juga terjadi trafik tinggi sehingga beban jaringan lebih besar pada saat pengukuran. Nilai *throughput* tertinggi yaitu pada saat penggunaan layanan dengan 6 pengguna aktif sebesar 0,82 Mbps, sementara penggunaan dengan nilai *throughput* terendah adalah pada saat penggunaan layanan dengan 25 pengguna aktif sebesar 0,07 Mbps.

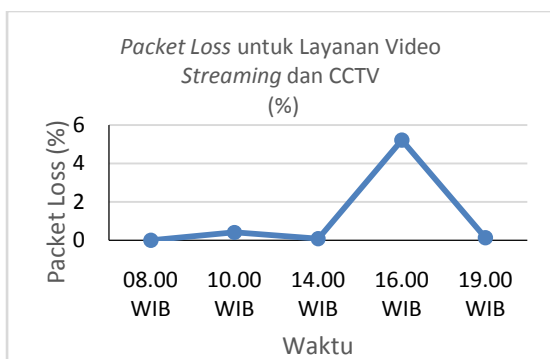
2. Penggunaan layanan pada waktu-waktu tertentu
 Gambar 10 menunjukkan grafik hasil pengukuran *delay* pada penggunaan layanan pada waktu-waktu tertentu.



Gambar 10 Grafik hasil pengukuran *delay* pada penggunaan layanan pada waktu-waktu tertentu

Standar nilai *delay* untuk aplikasi video satu arah berdasarkan ITU-T G.1010 adalah tidak lebih besar dari 10 s. Berdasarkan hasil pengukuran, terlihat dari grafik pada Gambar 4.15 bahwa nilai *delay* pada penggunaan layanan pada waktu-waktu tertentu masih dalam kategori baik dan memenuhi standar *delay* berdasarkan ITU-T G.1010. Nilai *delay* tertinggi yaitu pada saat pengukuran pada jam 08.00 WIB sebesar 65,43 ms, hal ini terjadi karena pada jam tersebut merupakan jam dimulainya kegiatan dikampus sehingga memungkinkan trafik tinggi pada jam tersebut. Semakin siang nilai *delay* menurun, penggunaan terbaik dengan nilai *delay* terendah adalah pada saat penggunaan layanan pada jam 14.00 WIB sebesar 27,94 ms dan kembali naik pada saat sore dan malam hari karena pada saat sore dan malam hari banyak terjadi aktifitas unduh data.

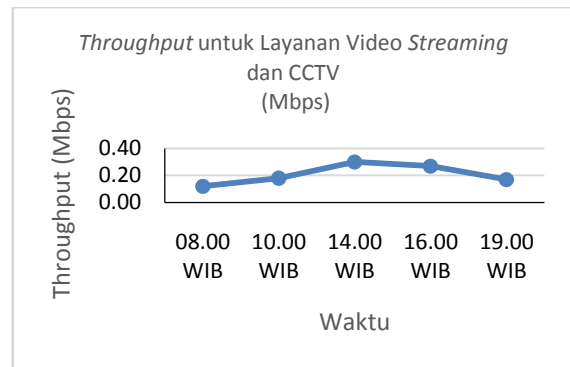
Gambar 11 menunjukkan grafik hasil pengukuran *packet loss* pada penggunaan layanan pada waktu-waktu tertentu.



Gambar 11 Grafik hasil pengukuran *packet loss* pada penggunaan layanan pada waktu-waktu tertentu

Standar nilai *packet loss* untuk aplikasi video satu arah berdasarkan ITU-T G.1010 adalah tidak lebih besar dari 1%. Berdasarkan hasil pengukuran, terlihat dari grafik pada Gambar 11 bahwa nilai *packet loss* pada penggunaan

layanan pada waktu-waktu tertentu yang masih dalam kategori baik dan memenuhi standar *packet loss* berdasarkan ITU-T G.1010 adalah ketika dilakukan pengukuran pada jam 08.00 WIB, 10.00 WIB, 14.00 WIB dan 19.00 WIB. Sedangkan nilai *packet loss* pada jam 16.00 WIB melebihi standar nilai *packet loss* berdasarkan ITU-T G.1010. Hal ini disebabkan oleh sejumlah faktor yang terjadi pada saat pengukuran seperti penurunan *signal* jaringan dalam media jaringan lokal, dan juga terjadi trafik tinggi sehingga beban jaringan lebih besar. Dari hasil pengukuran nilai *packet loss* didapat hasil yang tidak sesuai dengan hasil pengukuran *delay*, nilai *packet loss* tertinggi yaitu pada saat penggunaan layanan pada jam 16.00 WIB sebesar 5,22 %, sementara penggunaan terbaik dengan nilai *packet loss* terendah adalah pada saat penggunaan layanan pada jam 14.00 WIB sebesar 0,09 %. Gambar 12 menunjukkan grafik hasil pengukuran *throughput* pada penggunaan layanan pada waktu-waktu tertentu.



Gambar 12 Grafik hasil pengukuran *throughput* pada penggunaan layanan pada waktu-waktu tertentu

Berdasarkan hasil pengukuran, terlihat dari grafik pada Gambar 12 nilai *throughput* pada penggunaan layanan pada waktu-waktu tertentu. Nilai *throughput* tertinggi yaitu pada saat penggunaan layanan pada jam 14.00 WIB sebesar 0,30 Mbps, sementara penggunaan dengan nilai *throughput* terendah adalah pada saat penggunaan layanan pada jam 08.00 WIB sebesar 0,12 Mbps. Grafik menampilkan adanya kesesuaian antara hasil pengukuran nilai *throughput* dengan hasil pengukuran *delay*, dimana ketika *delay* paling tinggi yaitu pada jam 08.00 WIB maka nilai *throughput* terendahpun terdapat pada jam 08.00 WIB. Terlihat adanya kesinambungan antara *delay* dan *throughput*, dimana ketika *delay* tinggi maka *throughput* akan turun.

4. Kesimpulan

Dari hasil perancangan dan implementasi penelitian sistem layanan video *streaming* dan CCTV berbasis Red5 pada *virtual server* dapat diambil beberapa kesimpulan yaitu sistem layanan video *streaming* dan CCTV berbasis

Red5 pada *virtual server* yang telah dirancang dan diimplementasikan berhasil menampilkan video-video hasil *streaming* dari *server*, baik pada layanan video *streaming* maupun CCTV sesuai dengan perancangan dan konfigurasi yang telah diberikan, dan memenuhi standar berdasarkan ITU-T G.1010. Pada pengukuran *delay* dari keempat skenario penggunaan layanan didapat nilai *delay* tertinggi pada saat pengujian adalah sebesar 195,09 ms sementara penggunaan terbaik dengan nilai *delay* terendah adalah sebesar 10,37 ms. Pada pengukuran *delay* pada penggunaan layanan dengan beberapa pengguna aktif didapat bahwa semakin banyak jumlah pengguna aktif maka nilai *delay* semakin tinggi. Pada pengukuran *packet loss* dari keempat skenario penggunaan layanan didapat nilai *packet loss* tertinggi pada saat pengujian adalah sebesar 5,22 % sementara penggunaan terbaik dengan nilai *packet loss* terendah adalah sebesar 0 %. Pada pengukuran *packet loss* pada penggunaan layanan dengan beberapa pengguna aktif didapat bahwa semakin banyak jumlah pengguna aktif maka nilai *packet loss* semakin tinggi. Pada pengukuran *throughput* dari keempat skenario penggunaan layanan didapat nilai *throughput* tertinggi pada saat pengujian adalah sebesar 0,82 Mbps sementara penggunaan terbaik dengan nilai *throughput* terendah adalah sebesar 0,06 Mbps. Pada pengukuran *throughput* terlihat adanya kesinambungan antara *delay* dan *throughput*, dimana ketika *delay* tinggi maka *throughput* akan turun. Hasil pengukuran kualitas layanan berdasarkan parameter *delay*, *packet loss*, dan *throughput* untuk layanan video *streaming* dan CCTV masih dalam kategori baik dan layak digunakan karena hasil pengukuran memenuhi standar berdasarkan ITU-T G.1010.

Referensi

- [1]. Abdullah Fauzi, Aab, *Apa itu CCTV*, http://www.ensiklopedia1.com/apa-itu-cctv/_2013.
- [2]. Austerberry, David, *The Technology Of Video & Audio Streaming*, Focal Press, USA, 2005.
- [3]. Dewi, Asri, *Rancang Bangun dan Unjuk Kerja QoS Objektiv VoIP Berbasis SIP Menggunakan Codec G-711 M-Law dan GSM 06.10*, Skripsi S-1, Universitas Indonesia, Depok, 2009.
- [4]. Follansbee, Joe, *Get Streaming! Quick Steps to Delivering Audio and Video Online*, Focal Press, USA, 2004.
- [5]. Hidayat, Rachmat, *Mengenal Proxmox VE Lebih Dalam*, <http://www.cloudindonesia.or.id/mengenal-proxmox-ve-lebih-dalam.html>, 2010.
- [6]. Kurniawan, Agus, *Network Forensics Panduan Analisis dan Investigasi Paket Data Jaringan menggunakan Wireshark*, ANDI, Yogyakarta, 2012.
- [7]. Kristanto, Andri, *Jaringan Komputer*, Graha Ilmu, Yogyakarta, 2003.
- [8]. Monfort, Jean-Yves, *Basic Requirement to Quality of Service (IP centric)*, Workshop on Standardization E-Health, Swiss, 2003.
- [9]. Nanda, S. and T. Chiueh, *A Survey On Virtualization Technologies*, Stony Brook University, Research Preliminary Exam Report, Brookhaven, New York, 2005.
- [10]. Prasetyo, Aldy, *Analisis dan Implementasi Beewatch (Internet-TV BINUS) Berbasis Red5 Server*, Skripsi S-1, Binus, Jakarta, 2009.
- [11]. Proxmox Server Solution, *Proxmox Virtual Enviroment Datasheet*, <http://www.proxmox.com/downloads/proxmox-ve/misc/116-proxmox-ve-datasheet/download>, Proxmox Server Solution GmbH, Vienna, 2013.
- [12]. Sofana, Iwan, *CISCO CCNA & Jaringan Komputer*, Informatika, Bandung, 2012.
- [13]. Szigeti, Tim. Christina Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs*, Cisco Press, Amerika Serikat, 2004.
- [14]. ---, *Adobe Flash*, http://id.wikipedia.org/wiki/Adobe_Flash, 2013.
- [15]. ---, *Apache Ant*, http://en.wikipedia.org/wiki/Apache_Ant, 2013.
- [16]. ---, *Bandwidth*, http://id.wikipedia.org/wiki/Lebar_pita, 2013.
- [17]. ---, *Linux*, <http://opensource.telkomspeedy.com/wiki/index.php/Linux>, 2013.
- [18]. ---, *Red5 Introduction*, <http://red5.googlecode.com/svn/doc/trunk/reference/html/introduction.html>, 2013.
- [19]. ---, *Red5 system requirements*, http://red5.googlecode.com/svn/doc/trunk/reference/html/system_requirements.html, 2013.
- [20]. ---, *TCP/IP*, http://opensource.telkomspeedy.com/wiki/index.php/Networking_Fundamental#TCP.2FIP_Model, 2013.