

PEMANFAATAN APLIKASI RED5 SEBAGAI *STREAMING SERVER* DI UNIVERSITAS SUMATERA UTARA

Toni Andrian, T. Ahri Bahriun

Konsentrasi Teknik Komputer, Departemen Teknik Elektro
Fakultas teknik Universitas Sumatera Utara (USU)
Jl. Almamater, Kampus USU Medan 20155 INDONESIA
e-mail: toni.andrian@students.usu.ac.id

Abstrak

Pemanfaatan *streaming* video menjadi sarana yang efektif untuk berkomunikasi dan menyampaikan informasi. Aplikasi Red5 dapat berperan sebagai *flash media server* dan dimanfaatkan untuk melakukan *live streaming video*. Paper ini membahas tentang pengujian keandalan kinerja Red5 *server* yang di-*install* di Debian. Keandalan kinerja aplikasi *streaming server* diuji dengan melibatkan 30 klien dan dilakukan dengan pergantian parameter *quality* dan *keyframe*. Pada sisi *server* dihitung CPU *usage*, RAM, penggunaan *bandwidth*, serta tundaan waktu pada sisi klien.

Kata kunci: Red5, *streaming*, *server*

1 Pendahuluan

Perkembangan internet pada saat ini sudah sangat pesat dengan pemanfaatan sangat luas dan hampir mencakup di segala bidang. Pemanfaatan umum dalam bidang komunikasi dan informasi adalah *streaming* video sehingga setiap *event* yang terjadi di suatu tempat bisa disaksikan oleh orang lain yang berada di tempat lain hanya dengan memanfaatkan koneksi internet dan PC/laptop. Sistem *streaming* video membutuhkan sebuah *server* agar proses tersebut bisa berjalan.

Ada berbagai *event* penting, khususnya di Universitas Sumatera Utara (USU), yang hanya dapat disaksikan oleh sebagian orang saja misalnya seperti acara wisuda atau seminar besar. Agar orang lain yang tidak berada di tempat tersebut bisa menyaksikan acara tersebut maka salah satu solusinya adalah dengan melakukan *live streaming* sehingga bisa disaksikan melalui internet secara *live*. Kendala yang dihadapi saat ini adalah belum tersedianya *server* khusus *streaming* sehingga tidak bisa melakukan *streaming* selain harus memanfaatkan *server* luar yang saat ini tidak ada yang menyediakannya secara gratis.

Dengan kecepatan akses intranet di USU saat ini diperkirakan tidak akan mengalami kesulitan mengakses layanan video *streaming*. Perkembangan *open source* membuat para pengembang aplikasi dan sistem operasi semakin berusaha membuat aplikasi yang

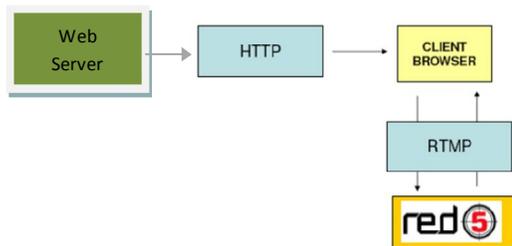
mampu memenuhi kebutuhan. Red5 pada dasarnya adalah sebuah *flash media server* yang dapat melakukan *streaming* audio maupun video, *shared object*, *recorded client streams*.

2 Konsep Dasar Video Streaming Server

Komputer yang dijadikan *server* harus didukung dengan spesifikasi *hardware* yang lebih tinggi daripada komputer personal yang biasa digunakan sehari-hari oleh orang pada umumnya. *File* yang terletak pada sebuah *server* dapat secara langsung dijalankan pada komputer klien sesaat setelah ada permintaan dari pengguna. Proses ini disebut juga *on-demand streaming*. Jika *file* tersebut langsung di-*capture* dan kemudian ditampilkan langsung (tanpa proses penyimpanan di *server*) maka proses ini disebut *live streaming*.

Ketika terjadi proses komunikasi di dalam jaringan komputer dibutuhkan protokol untuk mengatur proses komunikasi tersebut. Dalam komunikasi yang melibatkan Red5 sebagai *streaming server* terdapat 2 protokol yang berpengaruh terhadap berlangsungnya komunikasi tersebut, yaitu *Real Time Messaging Protocol* (RTMP) dan *Hypertext Transfer Protocol* (HTTP). Protokol HTTP digunakan untuk komunikasi antara klien dengan *web server* dan protokol RTMP digunakan untuk komunikasi antara klien

dengan Red5 server seperti ditunjukkan pada Gambar 1.



Gambar 1. Proses komunikasi klien dengan server dan web server.

Pada saat melakukan *streaming* media terdapat beberapa hal yang harus diperhatikan agar proses *streaming* dapat berjalan dengan lancar dan baik sehingga klien dapat menyaksikan *streaming* dengan nyaman dan server dapat bekerja pada keadaan optimal. Adapun indikator-indikator tersebut antara lain adalah *delay* waktu, pemakaian *bandwidth*, kinerja CPU dan RAM.

Delay (tundaan) waktu dihitung berdasarkan perbedaan selang waktu antara video sumber (*source*) dengan video yang ditampilkan pada klien. Pada proses *live streaming* diharapkan agar *delay* yang terjadi seminimal mungkin.

Bandwidth sangat berpengaruh dalam proses *streaming*. Semakin besar *bandwidth* yang tersedia pada suatu jaringan maka semakin banyak klien yang dapat menyaksikan *streaming* tersebut.

CPU *usage* dan RAM yang ditinjau adalah pada sisi server. Hal ini untuk mengantisipasi beban maksimum yang dapat diterima oleh server ketika diakses oleh klien dalam jumlah yang besar. Jika terjadi kelebihan beban (*overloading*) maka akan terjadi *error* pada server.

Agar komunikasi antara klien dengan Red5 server terjalin terdapat dua file yang berperan yaitu yang pertama adalah *file server* (.asc) yang dirancang untuk mendukung tujuan dan komunikasi yang ingin dicapai sehingga mampu merespon interaksi yang diberikan dari aplikasi klien dan yang kedua adalah *file* (.swf) yang merupakan aplikasi *flash* yang memberikan serta menyediakan tampilan layar untuk *end-user*. Komunikasi yang terjadi melalui aplikasi Red5 server yang dikirimkan kepada aplikasi klien (*flash player*) ketika aplikasi klien menggunakan komponen Red5 server dan hubungan terjalin dengan Red5 server untuk menyediakan aliran informasi yang diinginkan, sedangkan *file*

HTML digunakan sebagai penampung dari aplikasi klien (*flash player*) untuk dijalankan.

Gambar 2 menunjukkan bentuk tampilan *publisher* yang tersedia pada Red5. Halaman ini diatur agar *source* (video) yang akan di-*streaming*-kan bisa terhubung ke Red5 server.



Gambar 2. Tampilan halaman *publisher*

Ketika akan melakukan *streaming* video menggunakan Red5 server, pada bagian enkoder tersedia beberapa parameter yang bisa diatur agar data yang akan di-*streaming*-kan sesuai dengan yang diinginkan sesuai kondisi jaringan di tempat tersebut. Parameter yang diatur pada penelitian ini adalah *quality* dan *keyframe*. Gambar 3 menunjukkan halaman pengaturan video pada Red5.



Gambar 3. Halaman pengaturan video

3 Pengujian

Pengujian aplikasi Red5 sebagai *streaming server* merupakan hal penting yang bertujuan untuk mengetahui keandalan aplikasi tersebut sehingga didapatkan kelebihan maupun kekurangannya. Parameter yang diatur dalam penelitian ini adalah *quality* dan *keyframe*. Nilai masing-masing parameter yang diatur adalah pada nilai *default*, 50, dan 100.

Pengujian dilakukan dengan melakukan *streaming* data dengan mengatur nilai kedua parameter secara bergantian. Data tersebut diakses oleh klien dalam 3 tahapan. Tahapan pertama dengan 10 klien, kedua dengan 20 klien, dan ketiga dengan 30 klien. Pada tiap tahapan tersebut diukur besar *delay* (tundaan) waktu

yang terjadi antara video sumber (*source*) dengan video yang di-*streaming*-kan, besarnya pemakaian *bandwidth* selama proses *streaming* berlangsung, penggunaan RAM, dan penggunaan CPU.

4 Hasil Pengujian dan Analisa Hasil

Pengujian pertama dilakukan menggunakan beban sebanyak 10 klien. Besarnya *delay* waktu yang terjadi pada tiap pergantian parameter ditunjukkan pada Tabel 1.

Tabel 1. Besar *delay* waktu dengan 10 PC klien

No	Quality	Keyframe	Delay (det)
1	Default (0)	Default (5)	3,03
2	Default (0)	50	3,06
3	Default (0)	100	3,14
4	50	Default (5)	3,08
5	50	50	3,10
6	50	100	3,13
7	100	Default (5)	4,05
8	100	50	4,54
9	100	100	4,78

Dari Tabel 1 terlihat bahwa tidak terdapat perbedaan yang besar ketika nilai parameter *quality* diatur pada *default* (0) dan 50 terhadap besar *delay* waktu yang terjadi. *Delay* waktu terbesar pada kondisi ini adalah 3,14 detik dan yang terkecil adalah 3,03 detik. Pada saat parameter *quality* diatur pada 100 terjadi penambahan *delay* waktu. *Delay* waktu terbesar pada kondisi ini adalah 4,78 detik dan yang terkecil adalah 4,05 detik.

Pengaruh pergantian parameter terhadap pemakaian *bandwidth* selama melakukan proses *streaming* data ditunjukkan pada Tabel 2.

Tabel 2. Besar penggunaan *bandwidth* dengan 10 PC klien

No	Quality	Keyframe	Bandwidth	
			Received	Sent
1	Default (0)	Default (5)	33 kB	263 kB
2	Default (0)	50	20 kB	176 kB
3	Default (0)	100	20 kB	174 kB
4	50	Default (5)	30 kB	268 kB
5	50	50	18 kB	132 kB
6	50	100	20 kB	170 kB
7	100	Default (5)	80 kB	680 kB
8	100	50	78 kB	670 kB
9	100	100	80 kB	700 kB

Tabel 2 menunjukkan besarnya pemakaian *bandwidth* selama proses *streaming*

berlangsung. Jumlah paket tersebut digunakan untuk menjawab permintaan (*request*) 10 klien secara serentak. Dari tabel tersebut diperoleh informasi bahwa setiap perubahan parameter *quality* maka akan mempengaruhi jumlah data yang harus dikirim ke klien.

pengukuran penggunaan RAM dan penggunaan CPU (*CPU usage*) ketika parameter *quality* dan *keyframe* diubah dengan beban 10 PC klien yang mengakses data yang di-*streaming*-kan. Hasil tersebut ditunjukkan pada Tabel 3.

Tabel 3. Besar penggunaan CPU dan RAM dengan 10 PC klien

No	Quality	Keyframe	CPU (%)	RAM (MB)
1	Default (0)	Default (5)	3,5	200
2	Default (0)	50	3,5	200
3	Default (0)	100	2,5	200
4	50	Default (5)	2,4	200
5	50	50	2,5	200
6	50	100	3,6	200
7	100	Default (5)	3,5	200
8	100	50	4,4	200
9	100	100	3	200

Berdasarkan Tabel 3 tampak bahwa pergantian nilai kedua parameter tidak banyak berpengaruh terhadap kinerja CPU dan RAM dengan jumlah klien 10 PC.

Pengujian kedua dilakukan menggunakan beban sebanyak 20 klien. Besarnya *delay* waktu yang diukur pada tiap pergantian parameter ditunjukkan pada Tabel 4.

Tabel 4. Besar *delay* waktu dengan jumlah 20 PC klien

No	Quality	Keyframe	Delay (det)
1	Default (0)	Default (5)	3,14
2	Default (0)	50	3,15
3	Default (0)	100	3,18
4	50	Default (5)	3,17
5	50	50	3,24
6	50	100	3,22
7	100	Default (5)	5,63
8	100	50	5,50
9	100	100	5,83

Dari Tabel 4 terlihat bahwa tidak terdapat perbedaan yang besar ketika nilai parameter *quality* diatur pada *default* (0) dan 50 terhadap besar *delay* waktu yang terjadi. *Delay* waktu terbesar pada kondisi ini adalah 3,24 detik dan yang terkecil adalah 3,14 detik. Pada saat parameter *quality* diatur pada 100 terjadi penambahan *delay* waktu. *Delay* waktu terbesar

pada kondisi ini adalah 5,83 detik dan yang terkecil adalah 5,50 detik.

Pemakaian *bandwidth* selama pengujian menggunakan 20 PC klien ditunjukkan pada Tabel 5.

Tabel 5. Besar penggunaan *bandwidth* dengan 20 PC klien

No	<i>Quality</i>	<i>Keyframe</i>	<i>Bandwidth</i>	
			<i>Received</i>	<i>Sent</i>
1	<i>Default</i> (0)	<i>Default</i> (5)	70 kB	870 kB
2	<i>Default</i> (0)	50	55 kB	710 kB
3	<i>Default</i> (0)	100	50 kB	560 kB
4	50	<i>Default</i> (5)	50 kB	770 kB
5	50	50	20 kB	270 kB
6	50	100	38 kB	320 kB
7	100	<i>Default</i> (5)	80 kB	1,15 MB
8	100	50	100 kB	1,26 MB
9	100	100	80 kB	0,93 MB

Tabel 5 menunjukkan besarnya pemakaian *bandwidth* selama proses *streaming* berlangsung. Besar paket tersebut digunakan untuk menerima dan menjawab permintaan 20 klien secara serentak. Dari tabel tersebut diperoleh informasi bahwa setiap perubahan parameter *quality* maka jumlah data yang harus dikirim ke klien berubah secara signifikan.

Besar penggunaan RAM dan penggunaan CPU yang diukur pada pengujian ini ditunjukkan pada Tabel 6.

Tabel 6. Besar penggunaan CPU dan RAM dengan 20 PC klien

No	<i>Quality</i>	<i>Keyframe</i>	CPU (%)	RAM (MB)
1	<i>Default</i> (0)	<i>Default</i> (5)	3	200
2	<i>Default</i> (0)	50	2,5	200
3	<i>Default</i> (0)	100	3	200
4	50	<i>Default</i> (5)	2	200
5	50	50	3,5	200
6	50	100	4	200
7	100	<i>Default</i> (5)	5	210
8	100	50	3	210
9	100	100	3	200

Pada percobaan menggunakan 20 PC klien terjadi kenaikan penggunaan RAM dari 200 MB menjadi 210 MB pada parameter *quality* 100 dengan *keyframe default* (5) dan parameter *quality* 100 dengan *keyframe* 50.

Pengujian ketiga dilakukan menggunakan beban sebanyak 30 klien. Besarnya *delay* waktu yang diukur pada tiap pergantian parameter ditunjukkan pada Tabel 7.

Tabel 7. Besar *delay* waktu dengan jumlah 30 PC klien

No	<i>Quality</i>	<i>Keyframe</i>	<i>Delay</i> (det)
1	<i>Default</i> (0)	<i>Default</i> (5)	4,03
2	<i>Default</i> (0)	50	4,32
3	<i>Default</i> (0)	100	4,25
4	50	<i>Default</i> (5)	4,34
5	50	50	4,92
6	50	100	4,20
7	100	<i>Default</i> (5)	6,80
8	100	50	6,02
9	100	100	6,28

Dari Tabel 7 terlihat bahwa tidak terdapat perbedaan yang besar ketika nilai parameter *quality* diatur pada *default* (0) dan 50 terhadap besar *delay* waktu yang terjadi. *Delay* waktu terbesar pada kondisi ini adalah 4,92 detik dan yang terkecil adalah 4,03 detik. Pada saat parameter *quality* diatur pada 100 terjadi penambahan *delay* waktu. *Delay* waktu terbesar pada kondisi ini adalah 6,80 detik dan yang terkecil adalah 6,02 detik. Hal ini juga sama terjadi percobaan dengan jumlah 10 PC dan 20 PC klien. Dari ketiga percobaan ini dengan jumlah klien semakin bertambah tidak terjadi penambahan *delay* waktu yang linear dengan penambahan jumlah klien.

Pemakaian *bandwidth* selama pengujian menggunakan 30 PC klien ditunjukkan pada Tabel 8.

Tabel 8. Besar penggunaan *bandwidth* dengan 30 PC klien

No	<i>Quality</i>	<i>Keyframe</i>	<i>Bandwidth</i>	
			<i>Received</i>	<i>Sent</i>
1	<i>Default</i> (0)	<i>Default</i> (5)	120 kB	1,95 MB
2	<i>Default</i> (0)	50	110 kB	1,96 MB
3	<i>Default</i> (0)	100	110 kB	1,97 MB
4	50	<i>Default</i> (5)	40 kB	690 kB
5	50	50	20 kB	360 kB
6	50	100	30 kB	460 kB
7	100	<i>Default</i> (5)	95 kB	1,59 MB
8	100	50	110 kB	2 MB
9	100	100	100 kB	1,98 MB

Tabel 8 menunjukkan besarnya pemakaian *bandwidth* selama proses *streaming* berlangsung. Besar paket tersebut digunakan untuk menjawab permintaan 30 klien secara serentak. Dari tabel tersebut diperoleh informasi bahwa setiap perubahan parameter *quality* maka jumlah data yang harus dikirim ke klien juga terjadi perubahan. Hal ini juga terjadi pada percobaan dengan jumlah 10 PC klien dan 20 PC klien. Dari ketiga percobaan ini didapat kesimpulan bahwa yang sangat mempengaruhi

pemakaian *bandwidth* adalah dengan perubahan parameter *quality*.

Besar penggunaan RAM dan penggunaan CPU yang diukur pada pengujian ketiga ini ditunjukkan pada Tabel 9.

Tabel 9. Besar penggunaan CPU dan RAM dengan 30 PC klien

No	Quality	Keyframe	CPU (%)	RAM (MB)
1	Default (0)	Default (5)	3	210
2	Default (0)	50	4,5	200
3	Default (0)	100	4	200
4	50	Default (5)	4	200
5	50	50	2,5	200
6	50	100	4,5	200
7	100	Default (5)	3,5	200
8	100	50	4,5	200
9	100	100	3	200

Pada percobaan menggunakan 30 PC klien terjadi kenaikan RAM 210 MB pada parameter *quality default* (0) dengan *keyframe default* (5). Dari ketiga percobaan ini dengan jumlah klien semakin bertambah tidak terjadi penambahan yang linear pada penggunaan RAM dan CPU. Dengan kata lain tidak terjadi penambahan kerja pada RAM dan CPU ketika jumlah klien bertambah. Sehingga bisa diambil kesimpulan bahwa dengan jumlah klien berapapun yang mengakses halaman *streaming* yang sama tidak akan terjadi kelebihan pengguna (*overload*).

Dari ketiga pengujian tersebut pemakaian CPU dan RAM relatif stabil meskipun terjadi penambahan beban PC klien. Dengan kata lain tidak terjadi penambahan kerja pada RAM dan CPU ketika jumlah klien bertambah. Sehingga bisa diambil kesimpulan bahwa dengan jumlah klien berapapun yang mengakses halaman *streaming* yang sama tidak akan terjadi kelebihan pengguna (*overload*). Sedangkan penggunaan *bandwidth* mengalami kenaikan yang signifikan ketika terjadi penambahan klien dan perubahan parameter *Quality*. Penggunaan *bandwidth* yang minimal ketika terjadi penambahan klien adalah pada nilai parameter *Quality* adalah 50.

5 Kesimpulan

Berdasarkan uji coba dan analisa hasil di atas maka dapat diambil beberapa kesimpulan:

1. Penelitian yang dilakukan terhadap *delay* (tundaan) waktu yang terjadi selama proses pendistribusian dengan mengatur parameter *quality* dan *keyframe* pada bagian *encoder* yang disediakan Red5 diperoleh kesimpulan bahwa besarnya

delay sangat ditentukan oleh pengaturan parameter *quality* dan tidak dipengaruhi oleh *keyframe*.

2. Penggunaan *bandwidth* selama proses penerimaan *request* (*received*) dari klien ke *server* dan *server* menjawab dengan mengirimkan *file* yang diminta oleh klien (*sent*) dipengaruhi lebih dominan oleh pergantian nilai parameter *quality*. Penggunaan *bandwidth* paling minimal adalah pada nilai parameter *quality* 50 pada setiap tahapan pengujian dengan jumlah klien yang berbeda.
3. Kenaikan penggunaan *bandwidth* pada pengujian ini tidak linear dengan penambahan jumlah klien.
4. CPU *usage* selama berlangsung proses *live streaming* tidak dipengaruhi dengan setiap kenaikan jumlah klien dan pergantian nilai parameter yang ditetapkan.
5. RAM yang terpakai selama pengujian hampir konstan dengan nilai 200 MB pada tiap tahapan pengujian.
6. Untuk proses *live streaming*, *source file* (audio atau video) harus berasal dari dalam lingkup intranet.

6 Ucapan Terimakasih

Ucapan terima kasih kepada ayahanda Ubaidillah dan ibunda Elinar selaku orangtua, Ir. T. Ahri Bahriun, M.Sc, Ir. Pernantin Tarigan, M.Sc, F. Rizal Batubara, ST, MT, dan Ir. Kasmir Tanjung yang sudah membantu dalam menyelesaikan paper ini, serta teman-teman yang sudah memberikan dukungan selama pembuatan paper ini.

7 Daftar Pustaka

1. Allen, Chris. *Open Source Flash Development*. USA: Friends of, 2008.
2. Firestone, Scott., Ramalingan, Thiya., dan Fry, Steve. *Voice and Video Conferencing Fundamentals*. Indianapolis: Cisco System, 2007.
3. Follansbee, Joe. *Get Streaming! Quick Steps to Delivering Audio and Video Online*. USA: Focal Press, 2004.
4. Richter, Stefan dan Ozer, Jan. *Hands-on Guide to Flash Video*. USA: Focal Press, 2007.
5. <http://www.red5.org/> (diakses tanggal 11 Oktober 2012).