

PENGARUH *LOAD BALANCING* PADA PEMROSESAN PARALEL UNTUK KOMPRESI VIDEO

Sudaryanto¹, Teguh Bharata Adji, Hanung Adi Nugroho

¹Program Studi S2 Teknik Elektro dan Teknologi Informasi, Fakultas Teknik, Universitas Gadjah Mada,
Jl Grafika No. 2, Kampus UGM Yogyakarta 55281
Email: ryan_stta@yahoo.com

Abstract

Communication and multimedia, especially video data processing require very high resource both computing resources and communication traffic. This requires high-end machines such as servers with high specifications are of course very expensive. This results builds a web based application that implements the concept of parallel processing with load balancing process based CPU Usage to compress video files with FFmpeg software. The results are conditioned compression has half the resolution of the original video data. Based on the test results indicate with load balancing process parallel concepts used, the compression process showed an average speed up value of 8.07% faster than parallel Non load balancing process with 2 compressors, 37.57% with 3 compressors, and 41.24% with 4 compressors. The level of processor efficiency by 28.76% more efficient than parallel Non load balancing process with 2 compressors, 37.57% with 3 compressor, and 41.24% with 4 compressors.

Keywords: pemrosesan paralel, kompresi video, Load Balancing, CPU Usage

1. Pendahuluan

Data multimedia memiliki ukuran yang relatif besar dibanding data teks. Hal ini jelas menjadi kendala yang harus dihadapi untuk dapat mentransmisikan *file* tersebut di internet.

Ber macam cara dilakukan untuk manipulasi data multimedia agar data tersebut dapat dicecilkan salah satunya dengan mengompres. Permasalahan baru muncul yaitu beban komputasi yang diperlukan untuk mengompres *file* tersebut tidak kecil. *File* video harus diparsing menjadi sejumlah *frame* gambar kemudian mengurangi ukuran gambar tersebut dan menyatukan kembali (*merge*) menjadi *file* video utuh. Hal ini memerlukan mesin *high-end* seperti server dengan spesifikasi yang tinggi yang tentu saja sangat mahal. Maka pada penelitian ini dibangun sebuah aplikasi berbasis web yang mengimplementasikan konsep pemrosesan paralel *load balancing* untuk mengompres *file* video dengan software FFmpeg.

Teknologi pemrosesan *load balancing* dipilih mengingat pada [1] sudah diterapkan teknologi

pemrosesan paralel *non load balancing* yang dirasa masih mengalami kendala dari sisi waktu

Beberapa pendekatan telah diajukan oleh para peneliti seperti pada [1][2][3] untuk mengurangi waktu pemrosesan paralel. Tujuan dari penelitian ini adalah mencari teknik pemrosesan paralel yang lebih efisien, mencari efektifitas pengelolaan data multimedia dengan ukuran yang besar agar dapat lebih mudah ditransmisikan pada jaringan internet dan dengan pemrosesan paralel yang diimplementasikan pada aplikasi ini, diharapkan waktu kompresi video akan semakin cepat.

2. Metode Penelitian

Kompresi video digital adalah isu utama dalam *coding* video yang memungkinkan sebuah informasi video dapat dipertukarkan secara efisien dan terdistribusi seperti dalam komunikasi multimedia maupun penyiaran televisi digital [4].

Konsep pemrosesan paralel adalah memanfaatkan lebih dari satu *Central Processing Unit* (CPU) untuk mengerjakan sebuah tugas (*task*)

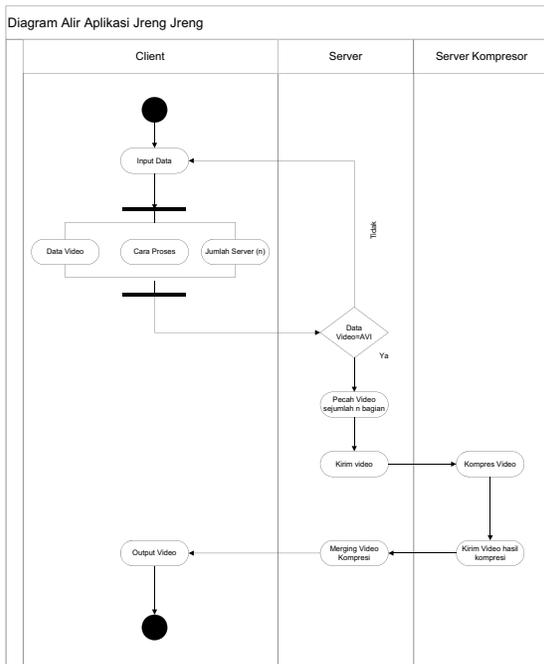
secara bersama-sama dan simultan sehingga dapat mempercepat terselesainya tugas tersebut.

Load balancing adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. Pada penelitian ini algoritma *load balancing* yang digunakan adalah algoritma *ratio*, dimana beban komputasi dibagi berdasarkan CPU Usage (Grafik yang menunjukkan waktu yang dibutuhkan prosesor dan indikator dari kegiatan prosesor dalam bekerja yang ditampilkan dalam persen) pada masing masing mesin. Jika CPU Usage tinggi maka beban yang diberikan pada mesin kompresor sedikit dan jika CPU Usage rendah beban yang diberikan untuk dikompresi besar.

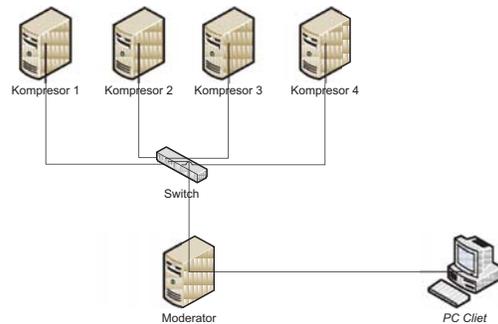
Gambar 1 menunjukkan cara kerja dari Aplikasi yang dimulai ketika *client* mengakses halaman dan memasukkan *file* video berformat AVI / MP4 yang ingin dikompres. Apabila *file input* tidak sesuai dengan kondisi yang telah ditetapkan, maka *client* akan diminta mengulangi proses *input* video. Setelah *file* video berformat AVI / MP4 berhasil dimasukan *client* memilih

proses parallel dengan *load balancing* atau *non load balancing*. Selain memilih proses, *client* juga harus memasukan jumlah kompresor yang akan digunakan untuk mengerjakan proses kompresi. Saat kondisi terpenuhi, maka *file* video akan dikompresi oleh sistem. Setelah proses kompresi telah berhasil dilakukan, *client* akan dibawa ke halaman *output* yang menyajikan informasi *file* video hasil kompresi dan *link* untuk mengunduh video hasil kompresi .

Aplikasi terdiri dari dua bagian, yaitu server moderator dan server kompresor. Server moderator berfungsi untuk melayani permintaan user, memecah video (task) menjadi sejumlah “n” bagian sesuai dengan pilihan client, mengirimkan masing-masing pecahan data video ke server kompresor, dan menggabungkan kembali menjadi file video utuh untuk dikembalikan kepada client. Sedangkan server kompresor berfungsi untuk mengompres tiap pecahan data video tersebut dan mengirimkan kembali data tersebut setelah proses kompresi berhasil dilakukan. Skema dasar jaringan komputer yang akan dibangun dalam perancangan aplikasi dapat dilihat pada Gambar 2.



Gambar 1 Diagram alir Aplikasi



Gambar 2 Skema Jaringan Komputer

Untuk mengukur kinerja pemrosesan paralel yang diajukan pada penelitian ini, diperlukan perhitungan atas kedua parameter [5] berikut ini:

a. *Speed Up* (SU)

Speed up adalah peningkatan kecepatan suatu proses yang diperoleh dalam menggunakan teknik paralel. Semakin besar nilai *speed up* juga dapat diartikan kinerja yang diperoleh dari pemrosesan paralel lebih cepat daripada pemrosesan secara serial. Penghitungan nilai *speed up* tersebut menggunakan Persamaan 2.1.

$$S(p) = \frac{t_s}{t_p} \tag{2.1}$$

Keterangan:

$S(p)$ = nilai *speed up*

t_s = waktu pemrosesan sekuensial (s)

t_p = waktu pemrosesan paralel dengan p prosesor (s)

b. Nilai Efisiensi (NE)

Dalam mengukur kinerja suatu sistem paralel efisiensi tidak dapat dipisahkan dari *speed up*. Dari definisi *speed up* tersebut, lahir nilai efisiensi, $E(p)$, untuk sistem dengan sejumlah p prosesor. Perhitungan nilai efisiensi didefinisikan sebagai berikut :

$$E(p) = \frac{t_s}{t_p \times p} \times 100\% \tag{2.2}$$

Keterangan:

$E(p)$ = nilai efisiensi (%)

t_s = waktu pemrosesan sekuensial (s)

t_p = waktu pemrosesan paralel dengan p prosesor (s)

p = jumlah prosesor yang digunakan

Pada perhitungan nilai *speed up* dan nilai efisiensi terdapat variabel waktu komputasi pemrosesan paralel *non load balancing* dan *load balancing*, di mana data tersebut diambil dari rata-rata waktu lima (5) kali uji coba proses kompresi. Waktu komputasi yang dihasilkan adalah lamanya proses komputasi yang dilakukan hingga menghasilkan suatu video yang terkompresi.

Peneliti juga akan menganalisis proses parallel *non load balancing* dan *load balancing* terhadap komunikasi data antar server yang dapat dilihat dari catatan waktu proses kompresi. Catatan waktu komputasi tersebut akan berpengaruh pada nilai *speed up* dan nilai efisiensi per jumlah prosesor yang digunakan. Peneliti akan membandingkan nilai rata-rata *speed up* dan rata-rata nilai efisiensi dengan menggunakan Persamaan 2.3 dan Persamaan 2.4.

$$\overline{SU} = \frac{\sum SU}{n} \tag{2.3}$$

Keterangan :

\overline{SU} : Rata-rata nilai *speed up*

$\sum SU$: Jumlah semua nilai *speed up*

n : Banyaknya data

$$\overline{NE} = \frac{\sum NE}{n} \tag{2.4}$$

Keterangan :

\overline{NE} : Rata-rata nilai efisiensi (%)

$\sum NE$: Jumlah semua nilai efisiensi (%)

n : Banyaknya data

Apabila terjadi perbedaan, maka akan dianalisis dari segi persentase kenaikan maupun persentase penurunan dengan menggunakan Persamaan 2.5 dan Persamaan 2.6, sehingga bisa diketahui pengaruh *load balancing* pada pemrosesan paralel apakah bisa mempercepat proses kompresi video.

$$\%Kenaikan = \frac{\text{Nilai Kenaikan}}{\text{Nilai Sebelum Kenaikan}} \times 100\% \tag{2.5}$$

$$\%Penurunan = \frac{\text{Nilai Penurunan}}{\text{Nilai Sebelum Penurunan}} \times 100\% \tag{2.6}$$

3. Hasil dan Pembahasan

Tampilan antar muka *client* dengan sistem kompresi video aplikasi dibangun menggunakan bahasa pemrograman PHP ditunjukkan pada Gambar 3. Pengujian sistem kompresi video ini dilakukan pada *Local Area Network* (LAN) dengan menyediakan 1 komputer sebagai server moderator dan 4 buah komputer sebagai server kompresor. Pengujian dilakukan terhadap delapan buah data video dengan format *.MP4 masing-masing berukuran antara 5,1 *Mega Bytes* (MB) hingga 291,8 MB. Waktu proses kompresi merupakan peubah yang diamati dalam penelitian ini.



Gambar 3 Tampilan Antar Muka *Client*

Pengujian terhadap data uji dilakukan sebanyak dua tahap. Tahap pertama adalah pengujian prosedur kompresi terhadap data video secara *non load balancing* dan tahap kedua dilakukan pengujian kompresi terhadap data video secara *load balancing* yang keduanya menggunakan satu server kompresor,

dua server kompresor, tiga server kompresor dan dengan empat server kompresor. Pada pengujian tahap pertama dan kedua kondisi mesin *server* kompresor 2 CPU usage dalam keadaan 90%-100%. Pengujian ini dilakukan terhadap delapan buah data video yang bervariasi dari segi ukuran *file* dan ukuran *frame*. Pada tiap tahap pengujian akan dilakukan pendataan terhadap ukuran data hasil kompresi dan waktu proses kompresi.

Setelah dilakukan uji coba tahap pertama, aplikasi berhasil melakukan proses kompresi pada kedelapan data uji sehingga menghasilkan output video dengan ukuran *file* yang cukup signifikan dibanding dengan ukuran *file* asli. Dan berdasarkan uji coba, pengukuran waktu proses kompresi aplikasi terhadap kedelapan data uji, dapat dihitung nilai *Speed up* (SU) dan Nilai Efisiensi (NE) setiap kompresor. Data parameter kinerja pemrosesan paralel aplikasi dari data hasil uji coba pada pemrosesan paralel *non load balancing* dapat dilihat pada Tabel 1.

Data hasil uji tersebut diperoleh dari perhitungan berdasarkan persamaan 2.1 dan persamaan 2.2 menggunakan software Microsoft Excel. Perhitungan manual dari Tabel 1 sebagai berikut:

File 11 dengan 4 kompresor

$$1. \quad SU = \frac{441,10}{202,02}$$

$$SU = 2,18$$

$$2. \quad NE = \frac{441,10}{202,02 \times 4} \times 100\%$$

$$NE = 54,59\%$$

Dari data hasil uji coba pemrosesan paralel *non load balancing* dapat diketahui bahwa nilai SU yang dapat dicapai oleh kompresi paralel dibandingkan dengan kompresi sekuensial berkisar antara 0,81 sampai dengan 2,18. Nilai SU maksimal dicapai oleh kompresi paralel dengan 4 server kompresor. Sedangkan nilai efisiensi masing-masing kompresor berkisar antara 22,70% sampai dengan 55,53%. Analisis yang dapat ditarik dari paparan data diatas adalah kecepatan waktu komputasi berbanding lurus dengan jumlah server kompresor yang digunakan dalam proses kompresi. Semakin banyak server kompresor yang digunakan maka waktu komputasi yang dibutuhkan untuk mengompres data video semakin cepat. Hal ini terlihat pada Grafik 2 bahwa kompresi video dengan menggunakan empat kompresor hampir mencapai

Tabel 1 Data hasil uji coba aplikasi dengan proses *Non Load Balancing*

Nama File	Kompresor	UF Asli	UF Kompres	ts	SU	NE
1.MP4	1	5,1	1,67	12,43	1,00	100,00
	2	5,1	1,67	15,25	0,81	40,74
	3	5,1	1,67	13,12	0,95	31,57
	4	5,1	1,67	13,69	0,91	22,70
3.MP4	1	16,3	6,46	35,35	1,00	100,00
	2	16,3	6,46	38,09	0,93	46,41
	3	16,3	6,46	29,41	1,20	40,07
	4	16,3	6,46	23,49	1,51	37,63
2.MP4	1	24,1	7,38	33,38	1,00	100,00
	2	24,1	7,38	35,03	0,95	47,64
	3	24,1	7,38	26,78	1,25	41,55
	4	24,1	7,38	24,26	1,38	34,40
5.MP4	1	45,1	15,66	65,32	1,00	100,00
	2	45,1	15,64	68,48	0,95	47,69
	3	45,1	15,55	50,20	1,30	43,37
	4	45,1	15,53	38,80	1,68	42,08
7.MP4	1	63,6	12,68	75,08	1,00	100,00
	2	63,6	12,64	78,44	0,96	47,86
	3	63,6	12,63	65,30	1,15	38,33
	4	63,6	12,36	56,51	1,33	33,21
13.MP4	1	82,6	17,21	117,85	1,00	100,00
	2	82,6	17,16	121,49	0,97	48,51
	3	82,6	17,14	78,55	1,50	50,01
	4	82,6	17,01	64,43	1,83	45,73
12.MP4	1	145,7	30,80	224,75	1,00	100,00
	2	145,7	30,79	217,99	1,03	51,55
	3	145,7	30,71	146,51	1,53	51,14
	4	145,7	30,68	115,19	1,95	48,78
11.MP4	1	291,8	64,42	441,10	1,00	100,00
	2	291,8	64,40	433,71	1,02	50,85
	3	291,8	64,31	264,80	1,67	55,53
	4	291,8	64,15	202,02	2,18	54,59

dua kali lebih cepat dari kompresi video sekuensial. Maka dapat disimpulkan bahwa pemrosesan paralel mempercepat proses kompresi video dibandingkan dengan mesin tunggal atau sekuensial.



Gambar 1 Grafik Perbandingan Waktu Komputasi dengan Jumlah Kompresor



Gambar 2 Grafik Hubungan Jumlah Kompresor dan Rata-Rata Nilai Speed Up

Berdasarkan data statistik rata-rata nilai efisiensi yang dicapai pada uji coba *non load balancing* terhadap jumlah kompresor pada Grafik 3, dapat dianalisis bahwa semakin banyak server kompresor yang digunakan maka beban komputasi pada masing-masing kompresor menurun. Nilai efisiensi maksimal terjadi pada percobaan tahap pertama, karena proses kompresi dikerjakan satu server kompresor. Sedangkan nilai efisiensi minimal sebesar 22,70% terjadi pada percobaan dengan menggunakan empat buah mesin server kompresor, karena beban komputasi untuk mengompres video dibagi ke empat server kompresor yang lain. Hal ini sesuai dengan tujuan awal dari penelitian ini yaitu dengan pemrosesan paralel beban komputasi pada mesin kompresor menurun sehingga mesin-mesin middle end atau low end dapat dimanfaatkan untuk mengolah data multimedia



Gambar 3 Grafik Hubungan Jumlah Kompresor dan Rata-Rata Efisiensi

Dan berdasarkan hasil uji coba tahap kedua, yaitu pengukuran waktu proses kompresi video dengan menggunakan pemrosesan paralel *Load Balancing* terhadap kedelapan data uji, maka dapat dihitung rata-rata nilai *Speed up* (SU) dan rata-rata Nilai Efisiensi (NE) pada setiap proses percobaan baik dengan 2 server kompresor, 3 server kompresor dan 4 server kompresor. Kemudian hasil perhitungan rata-rata nilai *Speed Up* dan rata-rata Nilai Efisiensi dibandingkan dengan hasil perhitungan pada pemrosesan paralel *Non Load Balancing*. Data parameter kinerja pemrosesan paralel dari data hasil perbandingan uji coba pada pemrosesan paralel dapat dilihat pada Tabel 2 dan Tabel 3.

Tabel 2. Data perbandingan Rata-Rata Speed Up (SU) pada proses *Non Load Balancing* dan *Load Balancing*

Proses	Jumlah Kompresor			
	1	2	3	4
Non Load Balancing	1,00	0,95	1,32	1,60
Load Balancing	1,00	1,03	1,81	2,25

Data tersebut diperoleh dari perhitungan berdasarkan Persamaan 2.3 menggunakan software Microsoft Excel. Perhitungan manual dari tabel 2 sebagai berikut:

- a. Rata-rata *speed up* dengan 3 kompresor pada *Non Load Balancing*

$$\overline{SU} = \frac{0,95 + 1,20 + 1,25 + 1,30 + 1,15 + 1,50 + 1,53 + 1,67}{8}$$

$$\overline{SU} = 1,32$$

- b. Rata-rata *speed up* dengan 3 kompresor pada *Load Balancing*

$$\overline{SU} = \frac{1,30 + 1,74 + 1,61 + 2,09 + 1,70 + 1,72 + 2,26 + 2,09}{8}$$

$$\overline{SU} = 1,81$$

Dilihat dari data hasil uji coba pemrosesan paralel *Non Load Balancing* dan *Load Balancing* Tabel 2 dan Grafik 4 dapat diketahui bahwa nilai rata-rata SU yang dapat dicapai oleh kompresi dengan pemrosesan paralel *Load Balancing* lebih cepat dibandingkan dengan Pemrosesan paralel *Non Load Balancing* dengan nilai 8,07% lebih cepat dengan dua kompresor, 37,57% dengan tiga kompresor dan 41.24% dengan empat kompresor. Prosentase tersebut didapat dari hasil perbandingan Proses paralel dengan *Non Load Balancing* dengan Proses paralel *Load Balancing* dengan melakukan perhitungan berdasarkan Persamaan 2.5.

- a. Porsentase kenaikan dengan 2 kompresor

$$\%Kenaikan = \frac{(1,03 - 0,95)}{0,95} \times 100\%$$

$$\%Kenaikan = 8,07$$

- b. Porsentase kenaikan dengan 3 kompresor

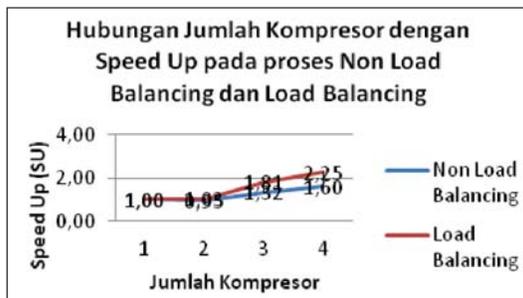
$$\%Kenaikan = \frac{(1,81 - 1,32)}{1,32} \times 100\%$$

$$\%Kenaikan = 37,57$$

- c. Porsentase kenaikan dengan 4 kompresor

$$\%Kenaikan = \frac{(2,25 - 1,60)}{1,60} \times 100\%$$

$$\%Kenaikan = 41,24$$



Gambar 4 Grafik Perbandingan Jumlah Kompresor dengan Speed Up pada proses *Non Load Balancing* dan *Load Balancing*

Sedangkan dilihat dari rata-rata Nilai Efisiensinya pada Tabel 3, data hasil uji coba menunjukkan bahwa rata-rata nilai efisiensi pada pemrosesan paralel *Load Balancing* lebih tinggi dibandingkan dengan pemrosesan paralel *Non Load Balancing*.

Tabel 3 Data perbandingan Rata-Rata Nilai Efisiensi (NE) pada proses *Non Load Balancing* dan *Load Balancing*

Proses	Jumlah Kompresor			
	1	2	3	4
Non Load Balancing	100,00	47,66	43,94	39,89
Load Balancing	100,00	61,36	60,46	56,34

Data tersebut diperoleh dari perhitungan berdasarkan Persamaan 2.4 menggunakan software Microsoft Excel. Perhitungan manual dari tabel 3 sebagai berikut:

- a. Rata-rata Nilai Efisiensi dengan 2 kompresor pada *Non Load Balancing*

$$\overline{NE} = \frac{40,47 + 46,41 + 47,46 + 47,69 + 47,86 + 48,51 + 51,55 + 50,85}{8}$$

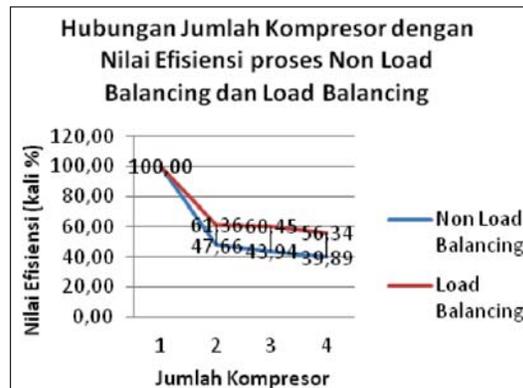
$$\overline{NE} = 47,66$$

- b. Rata-rata Nilai Efisiensi dengan 2 kompresor pada *Load Balancing*

$$\overline{NE} = \frac{43,55 + 50,40 + 48,08 + 54,60 + 53,69 + 53,41 + 54,31 + 52,87}{8}$$

$$\overline{NE} = 61,36$$

Dari data di atas dapat dianalisis bahwa semakin banyak server kompresor yang digunakan maka beban komputasi pada masing-masing kompresor menurun. Nilai efisiensi maksimal terjadi pada percobaan tahap pertama dapat dilihat pada Grafik 5, karena proses kompresi dikerjakan satu server kompresor.



Gambar 5 Grafik Perbandingan Jumlah Kompresor dengan Nilai Efisiensi pada proses *Non Load Balancing* dan *Load Balancing*

Analisis yang dapat ditarik dari data di atas adalah penggunaan load balancing pada pemrosesan paralel sangat berpengaruh dilihat dari proses kecepatan waktu komputasi, bahwa kecepatan komputasi pada load balancing lebih cepat dibandingkan dengan pemrosesan paralel Non load balancing. Dan dilihat dari nilai efisiensi pembagian beban komputasi penggunaan load balancing benar-benar membuat pembagian beban komputasi menjadi lebih efektif dan efisien jika dibandingkan dengan pemrosesan Non load balancing dengan nilai 28,76% lebih efisien dengan 2 kompresor, 37,57% dengan 3 kompresor dan 41,24% dengan 4 kompresor. Karena pembagian beban komputasi yang sesuai dengan aktifitas CPU Usage pada sebuah server kompresor (jika CPU Usage dalam keadaan sibuk dengan adanya aplikasi Load Balancing ini server kompresor akan diberikan beban yang ringan begitu pun sebaliknya) sangat efisien.

4. Kesimpulan

Berdasarkan hasil pengujian dari bagian sebelumnya, peneliti dapat menjabarkan beberapa kesimpulan, yaitu:

1. Penggunaan komputasi paralel hanya akan efektif untuk data dengan ukuran lebih dari 60 MB, karena pemrosesan paralel ditujukan untuk pemrosesan data dengan proses komputasi yang rumit maupun besar.
2. Penggunaan kompresi paralel baik dengan menggunakan *Load Balancing* maupun *Non Load Balancing*, paling optimal dicapai dengan menggunakan empat buah server kompresor, sehingga semakin banyak jumlah mesin kompresor yang digunakan waktu kompresi akan semakin cepat.
3. Dengan adanya *load balancing* pada pemrosesan paralel pembagian beban komputasi pada mesin server kompresor menjadi lebih bijaksana/adil sesuai dengan kesibukan (CPU Usage).
4. Konsep pemrosesan paralel load balancing yang diimplementasikan ke sistem kompresi video berhasil mempersingkat waktu kompresi bila dibandingkan dengan pemrosesan paralel non load balancing, dengan rata-rata nilai speed up sebesar 8,07% lebih cepat dari pemrosesan paralel *non load balancing* dengan dua kompresor; 37,57% dengan tiga kompresor

dan 41,24% kali dengan empat kompresor. Adapun tingkat efisiensi prosesor sebesar 61,36% dengan dua kompresor, 60,46% dengan tiga kompresor, dan 56,34% dengan empat kompresor.

5. Saran

Adapun saran untuk penelitian selanjutnya adalah:

1. Proses *Load Balancing* hanya fokus pada *CPU Usage* tanpa memperhatikan spesifikasi dari ke 4 server kompresor yang digunakan. Diharapkan pada penelitian berikutnya dalam penentuan rasio pembagian beban perlu memperhatikan atau ditambahkan kriteria Load Balancing berdasarkan *CPU Usage* dan Spesifikasi server kompresornya.
2. Pada penelitian ini untuk pembagian beban dimasing-masing mesin server kompresor berdasarkan durasi video. Dalam teknik kompresi beban kompresi terletak pada ukuran bit frame video yang dikompresi bukan pada durasi video. Sehingga diharapkan pada penelitian selanjutnya untuk pembagian beban komputasi pada masing-masing server kompresor berdasarkan pada ukuran bit frame video.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Bapak Teguh Bharata Adji selaku pembimbing 1, Bapak Hanung Adi Nugroho selaku pembimbing 2 dan Haruno Sajati teman seperjuangan serta Rita wulandari yang telah memberi dukungan financial terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Yuliani Indrianingsih, Hero Wintolo, Ika Kartika Sari, 2013, Penerapan *Grid Computing* untuk Mengkompilasi Program Berbahasa C/C++, Proceeding Senatik STTA, ISSN:2337-3881 Vol 1, 2013.
- [2] Ardhytia, S. N. dan Hiryanto, L. 2010. Algoritma Kompresi Fraktal Sequential dan Paralel untuk Kompresi Citra. Jurnal Ilmu Komputer dan Informasi, Vol 3, No 2 (2010). <http://jiki.cs.ui.ac.id/index.php/jiki/article/viewArticle/148>. Diunduh pada tanggal 2 April 2014.

- [3] Haruno Sajati, Yenni Astuti dan Chatrine Hernanda Octaviana, 2014, Analisis Pemrosesan Paralel untuk Kompresi Video Pada Jaringan Komputer Berbasis IPV6, Jurnal Angkasa: Volume 6, Nomer 6, November 2014 Teknik Informatika, STTA Yogyakarta.
- [4] Richardson, I. E. G., 2003, H.264 and MPEG-4 Video Compression Video Coding for Next Generation Multimedia, John Wiley & Sons. <http://lib.mdp.ac.id/ebook/Karya%20Umum/VideoCompression-Video-Coding-for-Next-generation-Multimedia.pdf>, diunduh pada tanggal 17 Februari 2014.
- [5] Parhami, B., 1999, Introduction to Parallel Computing. Kluwer Academic Publisher. University of California at Santa Barbara. Santa Barbara, California. <http://www.pnu.ac.ir/Portal/File/ShowFile.aspx?ID=7920fed-9312-4a70-b714c76893ec0327>, diunduh pada tanggal 20 Juni 2011.