

ANALISIS PEMROSESAN PARALEL UNTUK KOMPRESI VIDEO PADA JARINGAN KOMPUTER BERBASIS IPV6

Haruno Sajati, Yenni Astuti, Chatrine Hernanda Octaviana

Teknik Informatika
Sekolah Tinggi Teknologi Adisutjipto Yogyakarta
informatika@stta.ac.id

Abstrak

Komunikasi dan multimedia, khususnya pemroses data video memerlukan sumber daya yang sangat tinggi, baik dalam hal sumber daya komputasi maupun trafik komunikasinya. Hal ini terjadi salah satunya karena kurangnya alokasi alamat dari IPv4 Internet. Kekurangan alamat di IPv4 ini diatasi dengan adanya *Network Address Translation* (NAT), yang memaksa terjadinya penurunan kualitas layanan atau *Quality of Services* (QoS). Penelitian ini mengatasi permasalahan tersebut dengan membangun suatu aplikasi berbasis web yang menerapkan konsep pemrosesan paralel untuk mengkompresi suatu video dengan perangkat lunak FFmpeg dalam jaringan berbasis *Internet Protocol* versi 6 (IPv6). Pada penelitian ini, kompresi dikondisikan memiliki setengah resolusi dari data video asli. Berdasar hasil percobaan, diperoleh rasio rerata kompresi 4,06 kali dan rerata persentase penyimpanan sekitar 73,47%. Dengan penggunaan konsep paralel, proses kompresi menunjukkan suatu rerata nilai *speed up* 1,21 kali dengan menggunakan 2 kompresor; 1,39 kali dengan menggunakan 3 kompresor; dan 1,81 kali dengan 4 kompresor. Tingkat efisiensi prosesor bernilai 60% menggunakan 2 kompresor, 46% menggunakan 3 kompresor, dan 45% menggunakan 4 kompresor.

Keywords: pemrosesan paralel, kompresi video, IPv6

Abstract

Communication and multimedia, especially video data processing require very high resource both computing resources and communication traffic. This is compounded by the depletion of IPv4 address allocation in the Internet which forced administration degrade network quality of service (QoS) of communication in many ways one of them with Network Address Translation (NAT). Both try circumvented this problem by building a web-based application that implements the concept of parallel processing to compress a video file with FFmpeg software on the network based on Internet Protocol version 6 (IPv6). The results are conditioned compression has half the resolution of the original video data. Based on the test results indicate an average compression ratio of 4.06 times and the average percentage savings of 73.47%. With parallel concepts used, the compression process showed an average speed up value of 1.21 times with 2 compressors, 1.39 times with 3 compressors, and 1.81 times with 4 compressors. The level of processor efficiency by 60% with 2 compressors, 46% with 3 compressor, and 45% with 4 compressors.

Keywords: parallel processing, video compression, IPv6

1. Pendahuluan

Data multimedia memiliki ukuran yang relatif besar dibanding data teks. Sebuah *file* dengan format video yang memiliki durasi 183 detik dan ukuran *frame* hanya 400x320 *pixel* saja dapat memiliki ukuran 8 *Mega Byte* (MB). 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 dkecilkan salah satunya dengan mengompres. Permasalahan baru muncul yaitu beban komputasi yang diperlukan untuk mengompres *file* tersebut tidak kecil. *File* video harus di-*parsing* 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 untuk mengompres *file* video dengan *software* Ffmpeg pada jaringan berbasis *Internet Protocol version 6* (IPv6). Teknologi IPv6 dipilih mengingat teknologi ini sedang berkembang dan diprediksikan akan menggantikan *Internet Protocol version 4* (IPv4) yang alamatnya semakin menipis dan sarat kekurangan dalam hal kualitas.

Beberapa pendekatan telah diajukan oleh para peneliti seperti pada [1][2] untuk mengurangi waktu pemrosesan paralel. Tujuan dari penelitian ini adalah menghasilkan video dengan ukuran yang lebih kecil sehingga lebih efisien dalam penyimpanan maupun proses transfer. Dengan proses 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 [3].

Sebuah sinyal yang membawa data dapat dikompres dengan membuang *redundancy* atau kelebihan data dari sinyal tersebut. Pada teknik kompresi *lossless compression*, kelebihan data dibuang sehingga data asli bisa disusun kembali dengan sempurna di sisi penerima. Sayangnya, metode kompresi ini hanya mampu mengurangi sedikit dari ukuran *file* gambar maupun video. Sehingga banyak dipraktikkan teknik kompresi *lossy compression* dalam kompresi *file* multimedia yang mampu mengompres *file* lebih besar meskipun *file* hasil kompresi menjadi tidak sama dengan *file* asli. Kemampuan *visual* manusia sangat sensitif hingga ke frekuensi terkecil sehingga masih bisa mengenali gambar meskipun banyak informasi yang telah dihilangkan dari *file* tersebut.

Metode kompresi video pada penelitian ini memanfaatkan *codec standard* H.264. Standar H.264 merupakan salah satu standar kompresi video yang dalam hal fleksibilitas kompresi data visual, lebih menitikberatkan pada efisiensi dan reliabilitas kompresi *frame* video. Standar H.264 di publikasikan oleh *International Telecommunications Union* (ITU-T) untuk mendukung aplikasi komunikasi video dua arah (konferensi video maupun video *telephony*), *coding* untuk *broadcast* video berkualitas tinggi dan video *streaming* melalui jaringan internet [3].

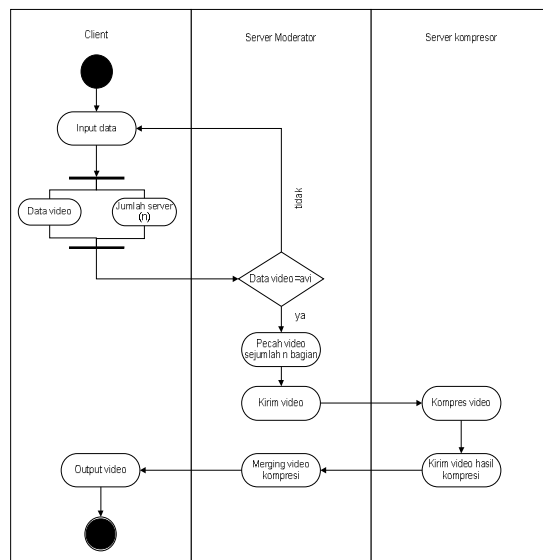
Gambar 1 menunjukkan cara kerja dari proses kompresi aplikasi “Shrink it”. Dari gambar tersebut dapat diketahui bahwa kompresi ini menggunakan *file* video berformat *Audio Video Interleave* (AVI) sebagai data input. Video input ini kemudian akan dipecah menjadi beberapa potongan video yang sama, sejumlah server kompresornya. Langkah selanjutnya adalah server moderator mengirimkan tiap bagian video tersebut ke server kompresor untuk dikompres. Setelah proses kompresi telah berhasil dilakukan, maka masing-masing kompresor tersebut akan mengirimkan kembali *file* hasil kompresi ke server moderator.

Server moderator akan menyatukan kembali *file* hasil kompresi tersebut menjadi satu *file* video utuh ke *client*.

Untuk menghitung kinerja kompresi video suatu algoritma kompresi perlu mempertimbangkan dua parameter [4] berikut yaitu:

a. Rasio Kompresi

Perbandingan antara ukuran *file* asli terhadap ukuran *file* hasil proses kompresi. Nilai rasio kompresi menunjukkan berapa kali kehandalan hasil kompresi yang dicapai dalam suatu proses kompresi. Semakin besar nilai faktor kompresi maka hasil kompresi semakin memuaskan. Namun nilai rasio kompresi ini berbanding terbalik dengan kualitas hasil kompresi. Sehingga dapat diartikan bahwa semakin tinggi nilai rasio kompresi maka kualitas hasil kompresi tersebut akan semakin menurun. Perhitungan nilai rasio kompresi dapat dirumuskan pada persamaan 2.1.



Gambar 1. Activity Diagram Aplikasi “Shrink it”

$$rasio\ kompresi = \frac{ukuran\ file\ asli}{ukuran\ file\ hasil\ kompresi} \quad (2.1)$$

b. Persentase Penghematan

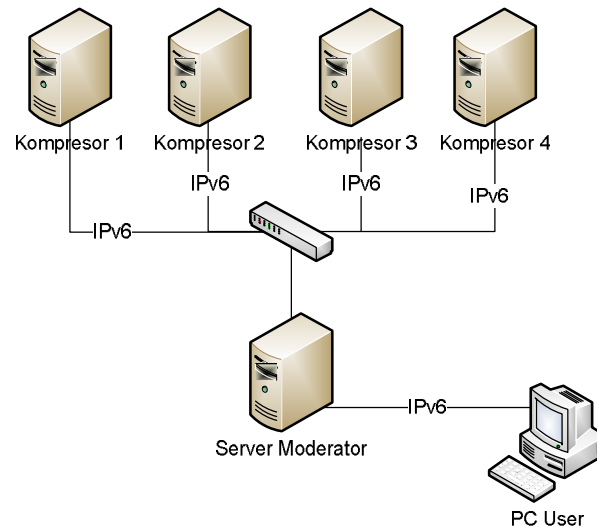
Persentase dari perbandingan antara selisih ukuran *file* dari sebelum proses kompresi dan sesudah proses kompresi dengan ukuran *file* sebelum proses kompresi. Angka persentase penghematan menunjukkan seberapa besar penghematan yang dicapai dalam suatu proses kompresi, semakin besar persentase maka hasil kompresi semakin memuaskan. Penghitungan nilai persentase penghematan video tersebut menggunakan persamaan 2.2.

$$PP = 100\% - \left(\frac{A}{O} \times 100\% \right) \quad (2.2)$$

Keterangan:

- PP = Persentase penghematan (%)
- A = ukuran *file* video hasil kompresi (MB)
- O = ukuran *file* video asli (MB)

Dilihat dari gambar 1, aplikasi “Shrink it” 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 “Shrink it” dapat dilihat pada gambar 2.



Gambar 2. Skema Jaringan Komputer yang Dibangun

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 2.3.

$$S(p) = \frac{t_s}{t_p} \quad (2.3)$$

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 pada persamaan 2.4.

$$E(p) = \frac{t_s}{t_p \times p} \times 100\% \quad (2.4)$$

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

Dapat dilihat bahwa efisiensi merupakan indikator atas tingkat kinerja *speed up* yang dicapai dibandingkan dengan nilai maksimum yang dapat dicapai. Dari persamaan-persamaan di atas dapat dilihat bahwa $1 \leq S(p) \leq p$, dan $1/p \leq E(p) \leq 1$. Efisiensi maksimum terjadi saat seluruh kode program dijalankan secara sekuensial pada satu prosesor, dan efisiensi terendah dicapai saat seluruh kode program dijalankan pada sejumlah p -prosesor.

Pada perhitungan nilai *speed up* dan nilai efisiensi terdapat variabel waktu komputasi pemrosesan serial maupun paralel, dimana data tersebut diambil dari rata-rata waktu tiga kali uji coba proses kompresi. Waktu komputasi yang dihasilkan adalah lamanya proses komputasi yang dilakukan hingga menghasilkan suatu citra yang terkompresi.

3. Hasil dan Pembahasan

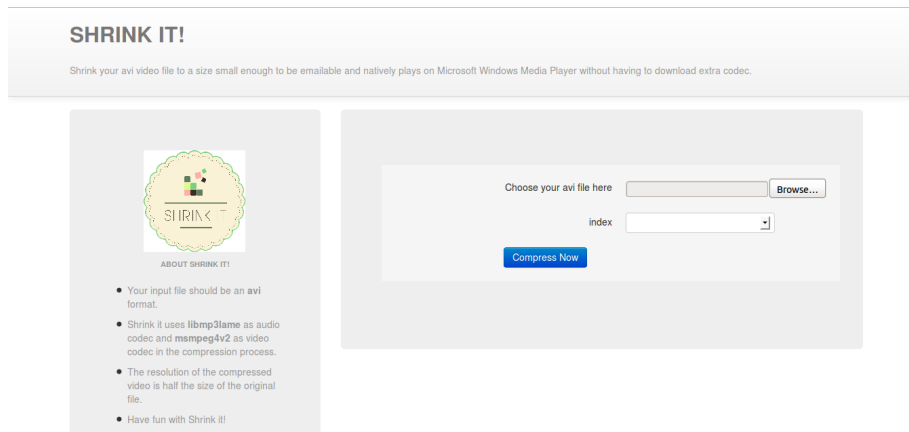
Tampilan antar muka *client* dengan sistem kompresi video aplikasi "Shrink it" dibangun menggunakan bahasa pemrograman PHP. Pengujian sistem kompresi video ini dilakukan pada *Local Area Network* (LAN) dengan menyediakan 1 komputer sebagai server moderator dan 4 buah komputer sebagai server kompresor. Pengujian dilakukan terhadap enam buah data, yaitu lima buah video dengan format *.AVI dan 1 (satu) video dengan format *.MP4 untuk menampilkan kotak dialog validasi data. Masing-masing video berformat *.AVI berukuran antara 23,5 *Mega Bytes* (MB) hingga 233 MB. Ukuran video hasil kompresi dan waktu proses kompresi merupakan peubah yang diamati dalam penelitian ini.

Aplikasi "Shrink it" terdiri dari dua bagian, yaitu pada Halaman Index dan Halaman Unduh. Halaman Index merupakan halaman yang pertama kali muncul saat "Shrink it" diakses. Pada halaman ini terdapat modul *form browse file* untuk *client* memilih file video yang akan dikompresi. *Combo-box* untuk memilih jumlah server kompresor yang akan digunakan.

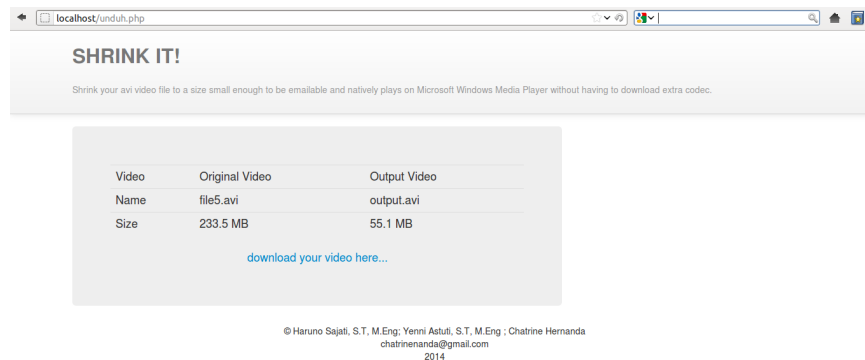
Nantinya nilai dari pilihan *client* ini akan disimpan pada sebuah *file* berekstensi *.TXT agar memudahkan pengecekan jumlah video yang dikirim ke server moderator. Tombol Compress Now untuk memulai proses kompresi. Halaman Index dapat dilihat pada gambar 3.

Hasil dari proses kompresi *file* video yang telah dipilih oleh *client* pada Halaman Index akan ditampilkan pada halaman unduh. Jika format video *input client* benar, maka Halaman Unduh akan muncul dengan menampilkan perbandingan ukuran video asli dengan video hasil kompresi dalam bentuk tabel dan *link* untuk mengunduh video hasil kompresi seperti yang ditunjukkan pada gambar 4.

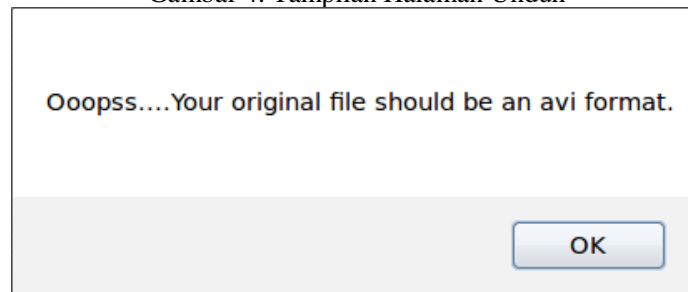
Namun, halaman ini tidak akan muncul apabila *file* video *input* tidak berformat AVI seperti yang sudah dikondisikan oleh peneliti. Sebagai gantinya akan muncul kotak dialog peringatan yang berisi informasi bahwa video *input* berformat selain AVI. Peringatan proses unggah video yang gagal setelah peneliti mengunggah *file* berekstensi MP4 ditunjukkan pada gambar 5.



Gambar 3. Tampilan Halaman Index



Gambar 4. Tampilan Halaman Unduh



Gambar 5. Message Box Proses Unggah Video

Pengujian terhadap data uji dilakukan sebanyak empat tahap. Tahap pertama adalah pengujian prosedur kompresi terhadap data video secara sekuensial atau menggunakan satu server kompresor. Tahap kedua dilakukan dengan dua server kompresor. Tahap ketiga dilakukan dengan tiga server kompresor. Tahap keempat dengan empat server kompresor. Pengujian ini dilakukan terhadap lima 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.

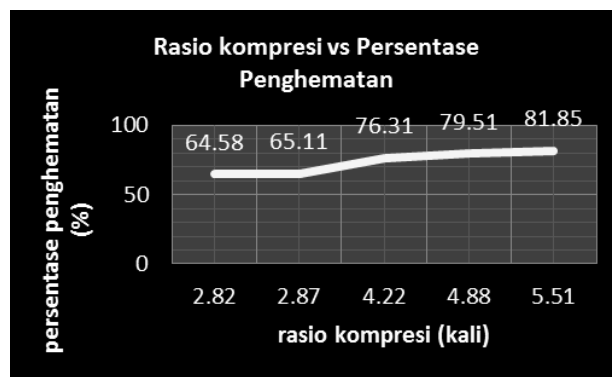
Setelah dilakukan uji coba tahap pertama, didapatkan hasil bahwa, aplikasi “Shrink it” berhasil melakukan proses kompresi pada kelima data uji dan menghasilkan output video

dengan ukuran *file* yang cukup signifikan dibanding dengan ukuran *file* asli. Data perbandingan ukuran *file* asli terhadap ukuran *file* kompresan digunakan dalam menghitung kedua parameter kinerja teknik kompresi yaitu rasio kompresi dan persentase penghematan pada tabel 1.

Tabel 1. Data hasil uji coba aplikasi *shrink it* dengan 1 server kompresor (sekuensial)

File	Ukuran Awal (MB)	Ukuran Akhir (MB)	Rasio kompresi	Persentase Penghematan
1	23,5	8,2	2,87	65,11%
2	61	12,5	4,88	79,51%
3	92	16,7	5,51	81,85%
4	124,8	44,2	2,82	64,58%
5	233	55,2	4,22	76,31%
Rata-rata			4,06	73,47%

Dilihat dari statistik perhitungan parameter kompresi terhadap kelima data input, terlihat bahwa nilai rasio kompresi berkisar antara 2,87 sampai dengan 5,51, serta persentase penghematan berkisar antara 64,58% sampai dengan 81,85%. Setelah menganalisis kedua parameter kinerja kompresi ini, peneliti membandingkan antara nilai rasio kompresi dan persentase penghematan pada gambar 6.



Gambar 6 Grafik Perbandingan Rasio Kompresi dan Persentase Penghematan

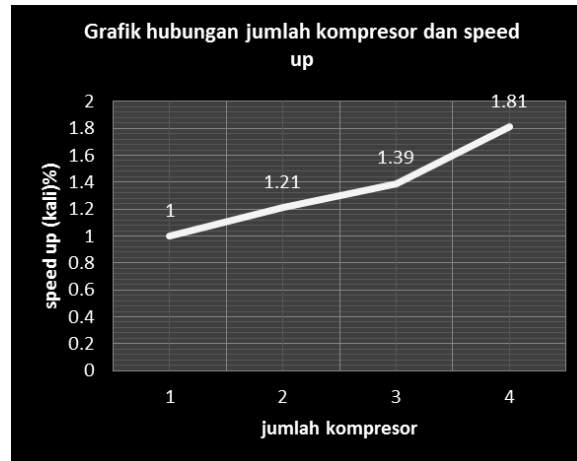
Berdasarkan data perbandingan diatas dapat disimpulkan bahwa besarnya nilai rasio kompresi berbanding lurus dengan persentase penghematan yang dicapai sistem kompresi. Maka terbukti pernyataan sebelumnya bahwa semakin besar rasio kompresi maka semakin besar pula data terkompres dari data asli.

Setelah analisis kinerja kompresi dari data hasil uji coba kompresi video secara sekuensial, selanjutnya adalah uji coba kompresi video secara paralel. Berdasarkan pengukuran waktu proses kompresi aplikasi “Shrink it” terhadap kelima data uji diatas, dapat dihitung nilai *Speed up* (SU) dan Nilai Efisiensi (NE) setiap kompresor. Data parameter kinerja pemrosesan paralel aplikasi *Shrink it* dari data hasil uji coba pada jaringan IPv6 dapat dilihat pada tabel 2.

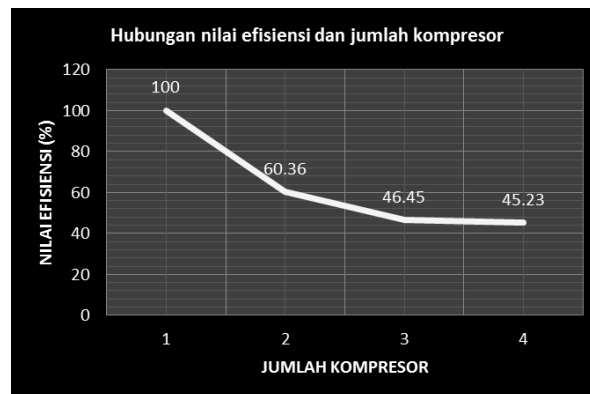
Dari data hasil uji coba dapat diketahui bahwa nilai SU yang dapat dicapai oleh kompresi paralel dibandingkan dengan kompresi sekuensial berkisar antara 1,1 sampai dengan 2. Nilai SU maksimal dicapai oleh kompresi paralel dengan 4 server kompresor.

Sedangkan nilai efisiensi masing-masing kompresor berkisar antara 39.64% sampai dengan 74,54%.

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 7 bahwa kompresi video dengan menggunakan empat kompresor hampir mencapai 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 7. Grafik Hubungan Jumlah Kompresor dan Rata-Rata Nilai *Speed Up*

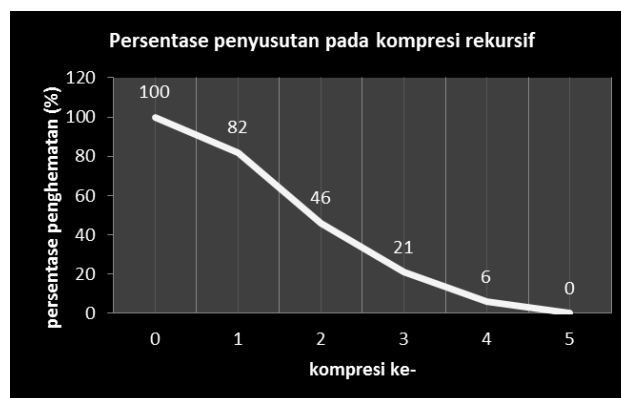


Gambar 8. Grafik Hubungan Rata-Rata Nilai Efisiensi dan Jumlah Kompresor

Berdasarkan data statistik rata-rata nilai efisiensi yang dicapai pada uji coba di jaringan IPv6 terhadap jumlah kompresor pada grafik diatas, 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 45,23% terjadi pada percobaan tahap keempat dengan empat server kompresor, karena beban komputasi untuk mengompres video dibagi ke tiga 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.

Tabel 2. Data hasil uji coba aplikasi *shrink it* pada jaringan IPv6

File	Kompresor	Ukuran File Output (MB)	\bar{t} (s)	SU	NE
1	1	8,2	44,470	1.0	100.00%
	2	8,2	41,196	1.1	53.97%
	3	8,2	36,136	1.2	41.02%
	4	8,2	23,526	1.9	47.26%
2	1	12,5	108,236	1.0	100.00%
	2	12,5	95,874	1.1	56.45%
	3	12,5	89,079	1.2	40.50%
	4	12,4	68,261	1.6	39.64%
3	1	16,7	154,262	1.0	100.00%
	2	16,7	126,999	1.2	60.73%
	3	16,5	99,110	1.6	51.88%
	4	16,5	80,242	1.9	48.06%
4	1	44,2	260,845	1.0	100.00%
	2	44,2	232,414	1.1	56.12%
	3	44,2	195,075	1.3	44.57%
	4	44,1	156,377	1.7	41.70%
5	1	55,2	581,861	1.0	100.00%
	2	55,2	390,316	1.5	74.54%
	3	55,2	357,403	1.6	54.27%
	4	55,1	293,971	2.0	49.48%



Gambar 9. Grafik Kompresi Rekursif pada *file 3*

Beban komputasi juga dapat diamati dari nilai efisiensi tiap prosesor saat proses kompresi baik pada jaringan IPv6. Grafik perbandingan rata-rata nilai efisiensi tiap prosesor akan dipaparkan pada gambar 8.

Pada uji coba penelitian juga dilakukan kompresi rekursif terhadap data uji 3 dengan ukuran awal data sebesar 92 MB. Pengujian ini dilakukan untuk menguji teknik kompresi yang digunakan. Setelah proses kompresi pertama dilakukan, dan *user* mendapatkan *output* video kompresi. Video keluaran tersebut menjadi data masukan untuk proses kompresi yang kedua. Peneliti menyebut proses ini dengan kompresi rekursif. Kompresi akan terus

dilakukan sampai proses kompresi tidak memberikan efek kompresi terhadap data *input*. Pengujian kompresi rekursi ditunjukkan pada gambar 9.

Berdasarkan pada gambar 9, setelah proses kompresi keempat, sistem tidak mampu menyusutkan ukuran *file* input. Hal ini dikarenakan kompresi video merupakan teknik kompresi *lossy*. Dimana teknik kompresi *lossy* mengalami *generation loss* jika melakukan berulang kali proses kompresi, yang akan mengakibatkan *file* kehilangan kualitas secara progresif. Maka dari itu, *file* hasil kompresi keempat sudah tidak dapat dikompres lagi.

4. Penutup

4.1 Kesimpulan

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

1. Konsep paralel yang diimplementasikan ke sistem kompresi video pada jaringan IPv6 berhasil mempersingkat waktu kompresi dengan rata-rata nilai *speed up* sebesar 1,21 kali dengan dua kompresor; 1,39 dengan tiga kompresor dan 1,81 kali dengan empat kompresor. Adapun tingkat efisiensi prosesor sebesar 60% dengan dua kompresor, 46% dengan tiga kompresor, dan 45% dengan empat kompresor.

2. Kinerja teknik kompresi pada penelitian ini berhasil dengan rata-rata rasio kompresi sebesar 4,06 kali dan rata-rata persentase penghematan 73,47%.

3. Sebuah file video akan mengalami *generation loss* atau kehilangan kualitas apabila dilakukan proses kompresi sebanyak empat kali.

4. Kualitas video hasil kompresi paralel sama dengan video hasil kompresi tunggal karena perintah *codec* yang digunakan sama. Namun ukuran video hasil kompresi paralel bisa berbeda dengan video hasil kompresi tunggal.

5. Penggunaan komputasi paralel hanya akan efektif untuk data dengan ukuran lebih dari 60 MB, karena pemrosesan paralel ditujukan untuk pemrosesan data rumit maupun besar.

6. Semakin banyak jumlah mesin kompresor yang digunakan, maka waktu kompresi semakin cepat. Hal ini dilihat dari penggunaan kompresi paralel paling optimal dicapai dengan menggunakan empat buah server kompresor.

4.2 Saran

Adapun saran untuk penelitian selanjutnya adalah:

1. Kompresi *multifile*. Diharapkan pada penelitian selanjutnya, aplikasi “Shrink it” mampu menangani kompresi beberapa data video dalam waktu bersamaan.

2. Penerapan *load balancing* pada sistem kompresi video. Sehingga beban atau *task* yang diberikan oleh server moderator sesuai dengan spesifikasi dan *traffic* yang terjadi pada masing-masing server kompresor tersebut.

3. Sebaiknya sistem kompresi video tidak hanya berfokus ke pemampatan ukuran saja, namun juga mempertimbangkan sisi kualitas video yang dihasilkan.

Daftar Pustaka

- [1] Kartika, I. , 2013, *Grid Computing* dalam Proses Kompilasi Kode Program C/C++, Proceeding Senat STTA, ISSN: 2337-2881 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] 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/Video-Compression-Video-Coding-for-Next-generation-Multimedia.pdf> , diunduh pada tanggal 17 Februari 2014.
- [4] Napitupulu, H S., 2012, *Analisa Perbandingan Kinerja Teknik Kompresi Citra Menggunakan Metode JPEG dan WAVELET Multi Variabel*.
<http://lontar.ui.ac.id/file?file=pdf/metadata-20308365.pdf> , diunduh pada tanggal 20 Maret 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=7920fefd-9312-4a70-b714-c76893ec0327> , diunduh pada tanggal 20 Juni 2011.