

KOMPUTASI PARALEL RENDER OBYEK 3D BERBASIS CLUSTER LAN

Anritsu S.Ch. Polii

(Staf Pengajar Teknik Elektro Politeknik Negeri Manado)

Abstract; The background of this research is the length of time required to perform runtime rendering 3D objects. Identification problems show that each hardware resource has a computing speed threshold where despite the high specification hardware and software support multitasking, but only until the execution of a single computing at a certain threshold. The research aimed to accelerate the rendering process of a 3D object using parallel computing techniques substance animation rendering 3D objects. Research results obtained in the form of accelerated runtime of 264.64%, the minimization of the number of samples per second up to 156.94%, reducing the number of update frames by 40%, and speed up the execution of the sample per second at 68.02%. So in general it can be concluded that the research could accelerate the execution time rendering of 3D object animation file reaches two to three times the normal speed of execution by single computing.

Keywords; Parallel computing, Cluster LAN, farm render, render

Abstrak; Penelitian ini dilatarbelakangi oleh lamanya waktu *runtime* yang dibutuhkan dalam melakukan render obyek 3D. Identifikasi permasalahan menunjukkan bahwa setiap sumber daya perangkat keras memiliki ambang batas kecepatan komputasi dimana meskipun dengan spesifikasi perangkat keras yang tinggi dan dukungan perangkat lunak yang *multitasking*, namun eksekusi *single computing* hanya sampai pada ambang batas tertentu. Penelitian ini bertujuan mempercepat proses render obyek 3D menggunakan teknik *parallel computing* substansi *rendering* animasi obyek 3D. Hasil penelitian yang didapatkan berupa percepatan waktu *runtime* sebesar 264.64%, minimalisasi jumlah sampel per detik sebesar 156.94%, mengurangi jumlah *update frame* sebesar 40%, dan mempercepat eksekusi sampel per detik sebesar 68.02%. Sehingga secara umum dapat disimpulkan bahwa penelitian ini mampu mempercepat waktu eksekusi *render file* animasi obyek 3D mencapai dua sampai tiga kali kecepatan normal eksekusi *single computing*.

Kata-kata Kunci; Komputasi Paralel, Cluster LAN, farm render, render

PENDAHULUAN

Kegiatan render animasi obyek 3D memerlukan sumber daya perangkat keras komputer dengan spesifikasi tinggi dan perangkat lunak yang memadai serta bersesuaian dengan perangkat kerasnya. Untuk mengolah gambar yang banyak dan kompleks memerlukan *PC* dengan spesifikasi tinggi (Alwi dkk., 2012) dan untuk menangani proses *rendering* suatu gambar baik 2D atau 3D yang kompleks membutuhkan waktu yang cukup lama (Kusuma dkk., 2012). Semakin tinggi spesifikasi perangkat keras yang tersedia maka semakin cepat proses rendering animasi obyek 3D.

Permasalahan yang muncul adalah setiap perangkat keras komputer yang tersedia mempunyai ambang batas kecepatan proses komputasi, artinya meskipun dengan spesifikasi yang tinggi dan dilakukan optimasi terhadap perangkat lunak dan perangkat

kerasnya, kenaikan proses *rendering* tidak begitu signifikan apabila hanya menggunakan satu komputer saja. Pembagian *resource* komputer guna melakukan proses *distributed rendering* (Prawira dkk., 2011) sangat dibutuhkan untuk mempercepat *render* obyek 3D.

Dalam rangka percepatan proses *rendering* obyek 3D maka penelitian ini akan membahas tentang komputasi perangkat keras multi komputer (*parallel computing*) secara simultan untuk *render* obyek 3D memanfaatkan *network render*.

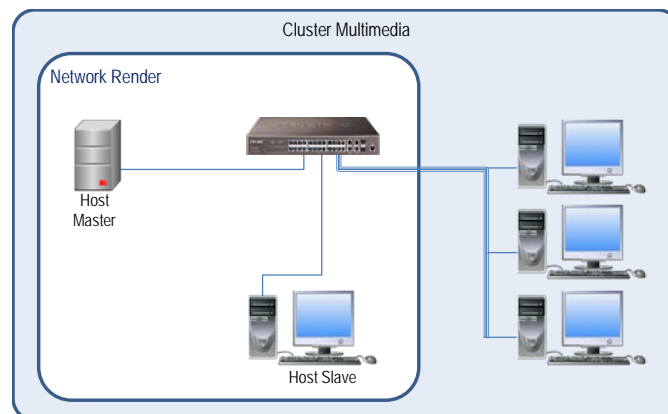
Perumusan masalah untuk diteliti, yaitu bagaimana merancang dan mengimplementasi *farm render* pada *cluster* jaringan, bagaimana mengukur kecepatan proses *render*. Parameter batasan masalah yang ditetapkan, yaitu *clustering* jaringan model *client-server* pada jaringan lokal, pengukuran kecepatan *render* berdasarkan waktu, sampel per pixel, sampel per detik, dan jumlah *update frame*.

Penelitian ini bertujuan, yaitu mempercepat proses *render* obyek 3D dan membandingkan kinerja *single computing* dengan *parallel computing* substansi proses *rendering*. Sehingga manfaat penelitian berupa efisiensi waktu proses kegiatan *render* animasi obyek 3D.

METODOLOGI PENELITIAN

Lokasi penelitian bertempat di Laboratorium Multimedia Politeknik Negeri Manado yang mempunyai ciri khas lokasi terdapat fasilitas komputer *cluster* spesialisasi multimedia yang saling terkoneksi melalui jaringan *LAN* dengan dukungan teknologi *gigabit ethernet* (1000 Mbps).

Rancangan penelitian topologi konseptual *farm render* dalam lingkungan *cluster multimedia* dapat dilihat pada Gambar 1.



Gambar 1. Rancangan Topologi konseptual *farm render*

Rancangan topologi *farm render* terdiri atas *host* yang saling terkoneksi melalui *intermediary device (switch)* dalam kelompok *network render* pada satu jaringan *cluster multimedia*.

Rancangan variabel *farm render* terdiri atas *host master* yang bertugas sebagai komputer pengelola untuk melakukan *render* obyek 3D, sedangkan *host slave* difungsikan untuk membantu proses *render* yang dilakukan oleh *host master*. Rancangan parameter-parameter *host master* terdiri atas penggunaan *network*, *hostname*, dan menjalankan *host* sebagai *master*, sedangkan parameter-parameter *host slave* meliputi penggunaan *network*, *hostname* dan menjalankan *host* sebagai *slave*.

Rancangan pengujian berupa *render file* obyek 3D dengan menggunakan satu *host* atau *standalone (single computing)*, selanjutnya untuk mengukur perbandingan kinerja dilakukan *render file* obyek 3D secara *parallel* menggunakan dua *host (parallel computing)*, yaitu *host master* dan *host slave* yang dikelompokkan pada *network render* (Gambar 1).

Populasi obyek yang diteliti mempunyai ciri khas berupa *file video* animasi multimedia obyek 3D dengan teknik pemilihan sampel berdasarkan nilai parameter waktu *runtime*, sampel per *pixel*, sampel per detik, dan jumlah *update frame* yang diproses.

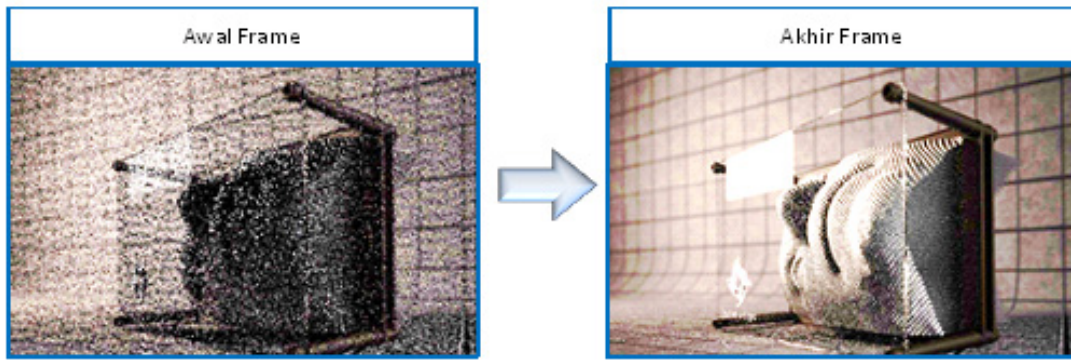
Pengumpulan data dilakukan menggunakan metode *blackbox* dengan mengambil hasil keluaran nilai terbaik proses *render* pada *single computing* dan *parallel computing* sesuai parameter yang ditetapkan.

Data yang dikumpulkan selanjutnya akan dianalisis menggunakan teknik perbandingan kinerja *single computing* dan *parallel computing* dalam bentuk grafik dan tabel perbandingan.

PEMBAHASAN

Hasil Penelitian

Hasil penelitian dari rancangan pengujian *render file* obyek 3D ditunjukkan pada Gambar 2.

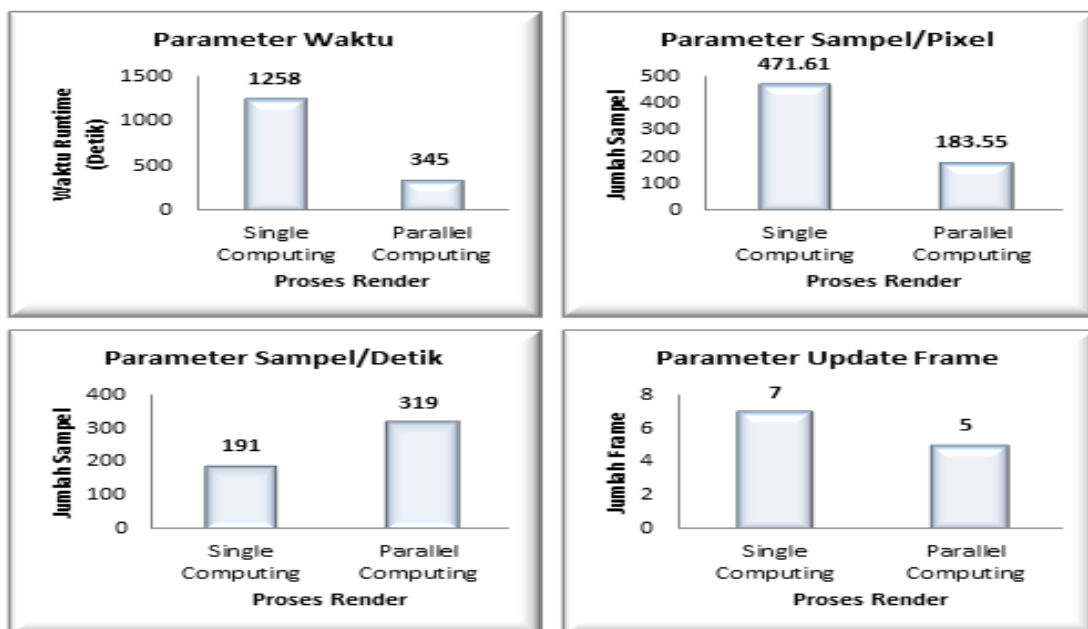


Sumber : Hasil pengujian

Gambar 2. Hasil Render Frame

Gambar 2 menunjukkan proses render dari awal *frame* secara *real-time* hingga mendapatkan hasil akhir *render frame*.

Berdasarkan proses render awal *frame* dan akhir *frame* kemudian dilakukan pengukuran terhadap proses komputasi menggunakan *single computing* dari awal *frame* hingga akhir *frame*. Selanjutnya dengan situasi pengujian yang sama dilakukan pengukuran dengan menggunakan *parallel computing*. Hasil keluaran pengukuran langsung terhadap proses *render* untuk individual parameter pengujian dapat dilihat pada grafik berikut.



Gambar 3. Perbandingan individual parameter kinerja proses *render*

Gambar 3 di atas merupakan perbandingan kinerja pengukuran langsung terhadap hasil

pengujian (Gambar 2) yang dipetakan berdasarkan parameter-parameter, yaitu waktu, sampel per pixel, sampel per detik, dan *update frame* sebagaimana di uraikan pada Tabel 1.

Tabel 1. Perbandingan Proses Render

Proses Render	Waktu (detik)	Sampel pe Pixel	Sampel per Detik (Max(k))	Jumlah Update Frame
Single Computing	1258	471.61	191	7
Parallel Computing	345	183.55	319	5

Sumber : Hasil pengujian

Analisa Hasil Penelitian

Data hasil penelitian pengukuran kinerja *render parallel computing* secara keseluruhan menunjukkan peningkatan kecepatan *render* dibandingkan *render single computing* yang mempunyai ambang batas komputasi yang dapat dihitung dalam MIPS (Pamungkas dkk., 2013). Dalam *parallel computing* semakin sedikit atau singkat proses *render* setiap parameter, maka semakin efektif dan efisien hasil *render* yang didapatkan.

Untuk menghitung selisih kenaikan persentase dari masing-masing parameter yang diujikan menggunakan formula:

$$\text{Persentase} = \frac{(\text{Single Computing} - \text{Parallel Computing})}{\text{Parallel Computing \%}}$$

Kinerja parameter waktu *runtime* pengujian secara *parallel computing* untuk menyelesaikan proses *render file* terjadi kenaikan yang signifikan yaitu sebesar 264.64 % dari waktu yang diperlukan oleh *single computing*. Waktu yang diperlukan untuk *render* secara *parallel* menggunakan dua *host* pada penelitian ini selama 5 menit 45 detik, sedangkan waktu *render* untuk 100 *frame* hanya selama 2 menit 50 detik untuk 20-40 *nodes* (Chong dkk., 2006). Meskipun waktu yang diperlukan pada penelitian ini masih lebih besar namun dapat dikatakan lebih baik karena jumlah *host* yang digunakan untuk *parallel render* jauh lebih sedikit, yaitu 2 berbanding 20.

Untuk menghasilkan sebuah *frame* gambar dengan kualitas bagus dibutuhkan jumlah sampel yang lebih banyak (Shirley dkk., 2011). Parameter sampel per *pixel* merupakan total jumlah sampel yang dibutuhkan untuk menyelesaikan *render file*. Jumlah sampel yang dibutuhkan untuk proses *render* secara *parallel* pada *host master* lebih sedikit

dibandingkan *standalone*, dampaknya proses *render* secara *parallel* terjadi kenaikan sebesar 156.94% dari jumlah sampel *single computing*.

Update frame mempunyai waktu ideal (Kopta dkk., 2012). Parameter jumlah *update frame* mengindikasikan jumlah *frame* yang dieksekusi untuk menghasilkan gambar akhir. Data hasil menunjukkan bahwa jumlah *update frame* untuk *parallel computing* lebih sedikit dari eksekusi *single computing*, sehingga terjadi kenaikan sebesar 40% dari jumlah *update frame* *single computing*.

Jumlah *frame* untuk satu host pada penelitian ini sebanyak 7 *frame* dengan waktu eksekusi selama 20 menit 58 detik. Dengan melakukan pembagian waktu terhadap jumlah *frame* maka dibutuhkan waktu 2 menit 59 detik untuk satu *frame*. Hasil penelitian tentang *framework* untuk *distribusi rendering* (Sheharyar dkk., 2013) menunjukkan dibutuhkan akumulasi waktu 260 menit untuk 80 *frame* pada satu *host*. Setelah dilakukan pembagian waktu terhadap jumlah *frame* maka dibutuhkan waktu 3 menit 15 detik untuk satu *frame*. Mengacu pada perbandingan waktu yang dibutuhkan untuk eksekusi satu *frame* maka dapat dikatakan bahwa penelitian ini masih lebih baik dengan agregasi waktu 16 detik lebih cepat.

Kinerja parameter sampel per detik diukur berdasarkan kecepatan eksekusi sampel per detik, sehingga semakin besar jumlah sampel yang dieksekusi per detik maka semakin cepat proses *render* (Underdahl, 2006). Oleh karena itu kenaikan persentase sampel per detik menggunakan formula berikut.

$$\text{Persentase} = \frac{(\text{Parallel Computing} - \text{Single Computing})}{\text{Single Computing \%}}$$

Parameter sampel per detik merupakan kecepatan eksekusi pemrosesan sampel dalam satu detik. Data menunjukkan bahwa kecepatan maksimum *parallel computing* lebih cepat 68.02% dalam memproses sampel dibandingkan kecepatan maksimum *single computing*.

Keuntungan yang ditawarkan pada penelitian ini yaitu peningkatan kecepatan *render* obyek 3D sehingga mampu mempersingkat waktu *render file* video animasi. Sedangkan kekurangan dari sistem yang ditawarkan adalah ketergantungan terhadap model *client-server* sehingga apabila terjadi kegagalan pada *host master* ketika proses *render* sedang berjalan maka harus diulang dari awal.

Kesimpulan

Mengacu pada permasalahan yang dikemukakan, untuk menjawab tujuan yang ditetapkan dan berdasarkan hasil yang didapatkan maka dapat ditarik kesimpulan, yaitu:

- 1). Waktu yang dibutuhkan *render* obyek 3D dapat dipercepat dengan menggunakan *parallel computing* memanfaatkan *farm render* sehingga eksekusi waktu lebih singkat karena dikerjakan oleh multi komputer secara simultan dengan capaian dua sampai tiga kali kecepatan *single computing*.
- 2). Dibandingkan dengan *single computing* maka *parallel computing* mampu mempercepat waktu *runtime render file* hingga 264.64%, mampu menekan total jumlah sampel yang dieksekusi hingga 156.94%, mampu mengurangi jumlah *update frame* yang dieksekusi mencapai 40%, dan mampu mempercepat eksekusi sampel per detik sebesar 68.02%.

Saran

Mengacu pada kekurangan yang teridentifikasi, maka sebagai *future work* untuk pengembangan sistem disarankan untuk melakukan penjadwalan kembali (*re-schedulling*) terhadap proses *render* sehingga jika terjadi kegagalan pada *host master*, maka dapat dilakukan *resume* hasil *render*.

DAFTAR PUSTAKA

- Alwi E. & Adji T. (2012). *Implementasi dan Analisis Kinerja Cluster MPI pada Proses Distributed Rendering Menggunakan POV-RAY*. Tesis. Yogyakarta: UGM.
- Chong A., Sourin A. & Levinski K. (2006). *Grid-based Computer Animation Rendering*. GRAPHITE '06 Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, hal.39-47. New York, USA: ACM.
- Kopta D., Ize T., Spjut J., Brunvand E., Davis A. & Kensler A. (2012). *Fast, Effective BVH Updates for Animated Scenes*. I3D '12 Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, hal.197-204. New York, USA: ACM.
- Kusuma A. & Adji T. (2012). *Analisis Perbandingan Kinerja Sistem Cluster PVM dengan MPI Terhadap Aplikasi POV-RAY*. Tesis. Yogyakarta: UGM.

- Pamungkas F., Muttaqin A. & Adams E. (2013). *Analisis dan Implementasi Render Farm*. Doro Jurnal. Vol. 2(No. 1): PTIIK, UB.
- Prawira D. & Adji T. (2011). *Analisis Kinerja Sistem Cluster pada Proses Distributed Rendering Menggunakan DrQueue*. Tesis. Yogyakarta: UGM.
- Sheharyar A. & Bouhali O. (2013). *A Framework for Creating a Distributed Rendering Environment on the Compute Clusters*. International Journal of Advanced Computer Science and Applications. Vol. 4(No. 6), hal.117-123: IJACSA.
- Shirley P. dkk., (2011). *A Local Image Reconstruction Algorithm for Stochastic Rendering*. Proceeding I3D '11 Symposium on Interactive 3D Graphics and Games, hal.9-14. New York, USA: ACM.
- Underdahl K. (2006). *Digital Video for Dummies 4th Edition*. Indiana: John Wiley & Sons.