

Implementasi *Rendering Farm* dengan Teknologi *Cluster Computing* Menggunakan *Back Burner* di Laboratorium Multimedia

Sutarti¹, Ngatono², Noverla Seticaya³

^{1,2,3} Program Studi Sistem Komputer, Fakultas Teknologi Informasi,
Universitas Serang Raya, Serang - Banten

e-mail: ¹sutarti86@gmail.com, ²ngatono77@gmail.com,
³verlacaya@gmail.com

Intisari

Ide untuk menggunakan komputasi paralel berawal dari permasalahan waktu proses *render*, jika menggunakan satu komputer bisa memakan waktu yang cukup lama dan menghasilkan hanya sebuah *frame* gambar dari proses, jika sebuah *file* animasi *render* yang diproses menggunakan proses komputer paralel atau dengan konsep jaringan *cluster* dapat menggunakan waktu seoptimal mungkin dan menghasilkan bagian-bagian dari proses *rendering*. Artinya setiap bagian-bagian dari proses *rendering* animasi tersebut diproses di komputer *slave* dan dicatat dari proses tersebut dan disimpan di komputer *master*.

PC cluster yang diimplementasikan dalam kasus ini menggunakan *middleware backburner*. Dalam hal ini memungkinkan sebagai media pembelajaran ilmu baru dan menunjang kegiatan praktikum di Laboratorium Multimedia karena sifat *rendering farm* lebih efisien terhadap waktu kerjanya dan memungkinkan Laboratorium Multimedia Universitas Serang Raya dapat dijadikan tempat atau fasilitas *rendering* untuk berbagai kalangan, seperti pelajar, mahasiswa, dan yang lainnya. Setelah dilakukan pengujian maka didapatkan bahwa *rendering* dengan menggunakan *PC cluster* tiga *slave* lebih efisien dalam pemakaian *processor* masing-masing *PC* dan pemanfaatan waktu lebih baik karena dapat melakukan *rendering* secara bersamaan, dibandingkan dengan *single computer*.

Kata kunci— *Backburner, Cluster Computer, Efisien, Master, Renderfarm*

Abstract

The idea of using parallel computing starts from the problem of the rendering process time, if using a computer can take a long time and produce only an image frame from the process, if a rendering animation file that is processed using a parallel computer process or with a cluster network concept can use time as optimal as possible and produce parts of the rendering process. This means that every part of the animation rendering process is processed on the slave computer and recorded from the process and stored on the master computer.

The PC cluster implemented in this case uses the backburner middleware. In this case it is possible as a new learning media and support practical activities in the Multimedia Laboratory because the rendering farm is more efficient in its working time and allows the Multimedia Laboratory of Universitas Serang Raya to be used as a rendering facility for various groups, such as students and others. After testing, it was found that the rendering using three slave cluster PCs was more efficient in using each PC processor and better time utilization because it could rendering simultaneously, compared to single computers.

Keywords— Backburner, Cluster Computer, Efficient, Master, Renderfarm

PENDAHULUAN

Fakultas Teknologi Informasi di Universitas Serang Raya memiliki tiga laboratorium, meliputi: Laboratorium Pemrograman, Laboratorium Multimedia, Laboratorium *Hardware* dan *Software*. Laboratorium tersebut biasa digunakan untuk praktikum dan perkuliahan, serta untuk kegiatan pengembangan keahlian dan ketrampilan bagi civitas akademik yang biasa dilaksanakan dengan kegiatan-kegiatan pendidikan dan pelatihan. Pada Laboratorium Multimedia diperlukan pembaruan sistem jaringan yang berbasis *Rendering Farm*.

Renderfarm adalah teknik komputer paralel yang menerapkan sistem penggabungan beberapa unit komputer sekaligus untuk mengerjakan proses yang telah dibagi-bagi secara bersamaan untuk *me-render* animasi 3D. Ide untuk menggunakan komputasi paralel berawal dari permasalahan waktu proses *render* animasi 3D, jika menggunakan satu komputer bisa memakan waktu yang cukup lama dan hanya menghasilkan sebuah hasil *frame* gambar dari proses *render* animasi 3D, jika sebuah *file* animasi *render* yang diproses menggunakan proses komputer paralel atau dengan konsep jaringan *cluster* dapat menggunakan waktu seoptimal mungkin dan menghasilkan bagian-bagian dari proses *rendering*. Artinya setiap bagian dari proses *rendering* animasi tersebut diproses di komputer *client* dan dicatat dan disimpan di komputer *master*.

Pada faktanya proses *render* biasa memerlukan waktu yang lebih lama karena diperlukan sebuah *core processor* untuk melakukan *rendering*, sedangkan dengan *rendering farm* akan terdiri dari ribuan *core processor* yang digabung untuk melakukan *rendering* dengan teknologi *clustering* atau komputer paralel, sehingga waktu yang dibutuhkan untuk *me-render* akan lebih singkat dengan membagi pekerjaan ke beberapa komputer.

Pemanfaatan *cluster computing* diharapkan bisa menjadi media pembelajaran ilmu baru dan menunjang kegiatan praktikum di Laboratorium Multimedia karena sifat *rendering farm* lebih efisien terhadap waktu kerjanya dan memungkinkan Laboratorium Multimedia Universitas Serang Raya dapat menjadi tempat atau fasilitas *rendering* bagi mahasiswa dan memungkinkan dapat disewakan untuk berbagai kalangan, seperti pelajar, mahasiswa luar kampus Universitas Serang Raya, maupun pekerja seni di bidang multimedia. Dalam implementasiannya penulis menggunakan teknologi *cluster computer* yang berbasis *rendering farm* menggunakan aplikasi *3DS MAX – Back Burner*.

Kajian Pustaka

Penelitian tentang pembangunan *Render Farm* untuk *Blender* di sistem operasi *Microsoft Windows* telah dilakukan dengan hasil penelitiannya adalah kinerja *render farm* yang telah dibangun dengan perbedaan *hardware* memberikan perbandingan kinerja yang signifikan dalam sistem *loki render farm*, dan penerapan *loki render* terbukti dapat mempercepat proses *rendering* animasi 3D [1].

Penentuan harga komputasi *grid* komersial dalam *rendering farm* berbasis waktu dan beban prosesor dengan hasil penelitiannya adalah dengan memberikan tugas pada *render farm* sistem dapat menghitung biaya yang dikenakan kepada *user* [2]. Dalam pengujian harga Rp 1.000,-/Ghz per jam menunjukkan bahwa semua

proses (100%) dapat dihitung biaya yang harus dikenakan kepada pengguna dengan *error* perhitungan rata-rata 1,17%. Apabila proses *render* terjadi *error*, maka sistem perhitungan tersebut tetap akan menunggu sehingga perlu dilakukan penghentian secara paksa, dan proses perhitungan tersebut dilakukan pada lapisan data sehingga dapat berdiri sendiri ketika menghitung biaya.

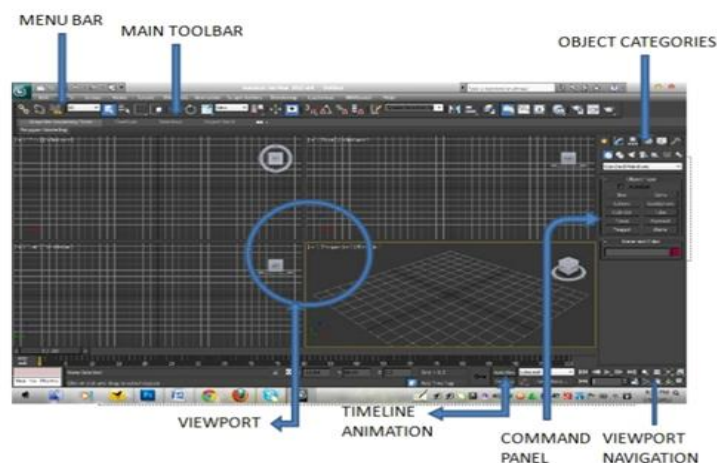
Penelitian dengan tema perancangan *PC cluster* untuk *render* animasi 3D dengan hasil penelitiannya adalah melakukan proses *rendering* dengan *PC cluster* lebih cepat dibandingkan dengan *single computer* [3]. Dalam hal ini terjadi peningkatan *performance* pada *PC cluster* seiring dengan bertambahnya jumlah *node*. *Rendering* pada *file* yang berbeda dapat menyebabkan perbedaan waktu *rendering* yang disebabkan oleh perbedaan karakteristik dari *file* itu sendiri.

Rendering 3D

Teknologi *rendering* 3D telah berhasil berkembang dalam kehidupan jutaan orang di seluruh dunia. Baik untuk *game* konsol yang terhubung ke televisi, animasi perangkat lunak pada sebuah *workstation* atau *blockbuster* terbaru efek khusus untuk film. Pada tahun 1960, desainer William Fetter merancang sebuah sistem 3D di dalam *cockpits* pesawat Boeing untuk memaksimalkan efisiensi tata letak *cockpits*. *Rendering* adalah suatu proses untuk mengubah model geometri menjadi suatu gambar. Untuk melakukan proses *render* biasanya membutuhkan waktu yang lama, terutama untuk *rendering* gambar yang *photo-realistic*.

3D Studio Max

3D Studio Max atau biasa dikenal dengan *3DS Max* adalah suatu *software* untuk membuat sebuah grafik vektor 3 dimensi dan animasi. Para desain grafis banyak menggunakan *software* ini, digunakan untuk membuat film animasi, arsitektur rumah, ataupun membuat logo suatu perusahaan. Untuk menjalankan aplikasi ini membutuhkan spesifikasi komputer yang cukup tinggi agar mendapat kenyamanan saat menggunakannya. Karena banyak proses yang membutuhkan grafik yang tinggi untuk melakukan *rendering* dan sebagainya.



Sumber: <http://dwimuri.files.wordpress.com/2014/11/3ds.png>

Gambar 1. Tampilan Awal Autodesk 3D Studio Max

Autodesk Backburner

Autodesk Backburner adalah aplikasi perangkat lunak manajemen *render* dikemas dengan *3DS MAX*, *Maya*, *Flame* dan berbagai produk lainnya dari *Autodesk*. *Backburner* terinstal secara *default* ketika menginstal *Autodesk 3DS MAX* atau *Autodesk 3DS MAX design*. *Backburner* merupakan perangkat lunak yang kompatibel, tetapi pengaturannya tidak bermigrasi dari versi sebelumnya.

Backburner bertindak sebagai manajer antrian untuk latar belakang dan didistribusikan tugas pengolahan jaringan, yang memungkinkan tugas-tugas ini harus dilakukan pada satu atau lebih komputer dengan menjaga fungsi program utama yang telah tersedia untuk digunakan. Dalam *setup* dengan beberapa mesin jaringan, *backburner* dapat membuat gambar lebih cepat dan efisien dengan membagi pekerjaan menjadi bagian-bagian kecil. *Backburner* dikenal karena memiliki *built in render que* di mana dapat mengirimkan pekerjaan dan sistem menunggu dalam antrian, serta secara otomatis dikirimkan ke *slave* atau *node* (komputer yang sedang tidak bekerja) yang telah tersedia dan menyelesaikan pekerjaannya.

METODE PENELITIAN

Alat dan Bahan

Dalam penelitian ini memerlukan *hardware* yang terdiri dari:

1. Komputer *Server*
2. *Switch*
3. Lima unit komputer *slave*

Sedangkan sistem operasi yang digunakan adalah *Windows* dan *software 3DS Max*.

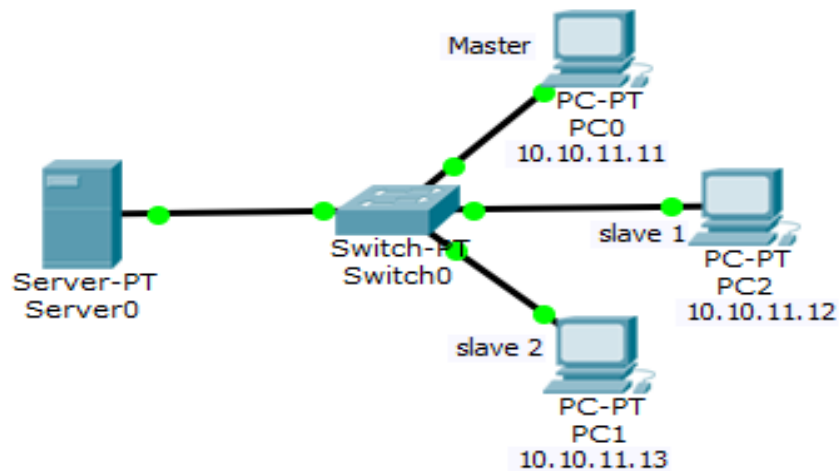
Diagram Alir Penelitian

Penelitian ini dilakukan dengan beberapa tahapan sebagai berikut:

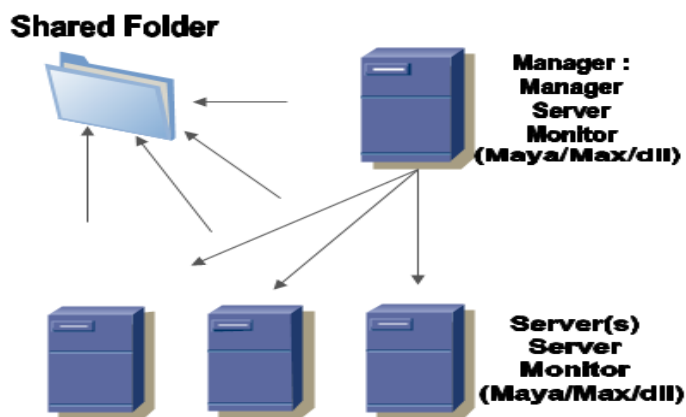
1. Merancang jaringan LAN sesuai dengan topologi jaringan usulan
2. Membangun rancangan sistem *rendering farm*
3. Instalasi *Autodesk 3DS Max 2013*
4. Instalasi *Backburner*

1. Topologi Jaringan Usulan

Adapun sistem jaringan usulan untuk Laboratorium Multimedia Fakultas Teknologi Informatika adalah dengan menggunakan teknologi *cluster computing* untuk *rendering farm* menggunakan sistem operasi *Windows* dan *software 3DS Max*. *Software* tersebut sebelumnya sudah digunakan pada Laboratorium Multimedia Universitas Serang Raya, namun akan dikembangkan menjadi suatu sistem *rendering* yang mempermudah *user* dalam *me-render* dan mempersingkat waktu *render* dengan menggunakan *software Backburner* sebagai sistem *manager render*-nya.



Gambar 2. Topologi Jaringan Usulan Lab Multimedia

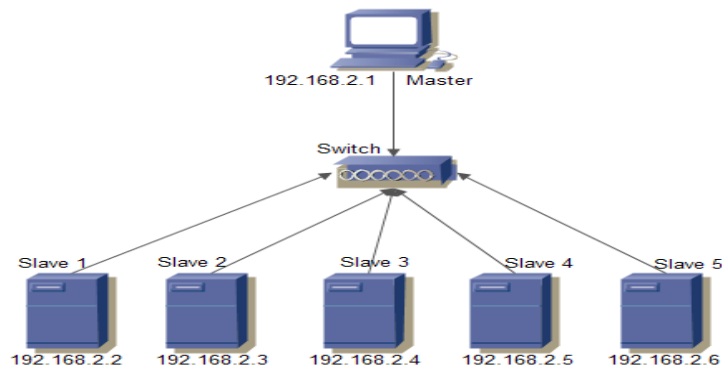


Gambar 3. Skema Rendering Farm

Skema *Rendering Farm* tersebut menjelaskan sistem kerja *Renderfarm* dalam jaringan komputer paralel atau *cluster computing*. Di mana terdapat komputer sebagai *manager (master computer)* yang dapat berperan sebagai *manager, server* dan *monitor* dalam satu komputer sekaligus, dan terdapat beberapa *server (slave)* yang dapat ikut serta me-*monitoring* pekerjaan dari komputer *server* tersebut, lalu terdapat suatu *folder* yang harus bisa di *sharing* untuk seluruh komputer yang masuk ke dalam jaringan yang telah terbentuk, yang fungsinya sebagai penyimpanan hasil dari *rendering* yang telah dikerjakan oleh seluruh *server (slave)*.

2. Rancangan Sistem *Rendering Farm*

Perancangan jaringan yang dibuat menggunakan topologi *star* di mana semua *nodes* terhubung pada *switch* yang menjadi titik pemersatu antara satu sama lain. *IP* yang digunakan adalah 192.168.2.0/24, alokasi *IP* pada tiap *node* dapat dilihat pada gambar 4.



Gambar 4. Perancangan Jaringan

3. Instalasi *Autodesk 3DS Max 2013*

Autodesk 3DS Max berperan sebagai mesin yang melakukan proses *render* (*render engine*). *Autodesk 3DS Max* di-*install* pada mesin pekerja (*slave node*) namun tidak ada salahnya jika di-*install* pada *node master*, karena komputer *master* sebagai *manager* yang mampu berperan sebagai *server* atau *slave*. Pada saat penelitian ini aplikasi 3D yang digunakan adalah *Autodesk 3DS Max 2013*.

4. Instalasi *Backburner*

Backburner merupakan *middleware* dari *cluster computing* yang dirancang pada *Autodesk 3DS MAX 2013* yang nantinya akan menjadi *render engine*-nya dan peran *backburner* sebagai jembatan antara aplikasi 3D *render* dengan *networking* (jaringan) yang dapat mengatur distribusi *render* di dalam jaringan. Disini penulis menggunakan *backburner* versi 2013, karena menggunakan aplikasi *Autodesk 3DS MAX* versi 2013, dalam hal ini antara aplikasi dengan *middleware*-nya harus disamakan versinya agar tidak ada kesalahan sistem nantinya.

Prosesnya *file* yang diperlukan untuk *render* jaringan akan disalin secara otomatis ke direktori *root backburner* ketika telah menginstal *3DS Max 2013*. *Manager* dan *server* di sini memerlukan *setting IP Address* sebelum dapat dijalankan. Pertama mengatur program manajer, kemudian *server*. Setelah selesai dapat mulai *render* jaringan. Setelah *setup*, dapat menjalankan keduanya, manajer dan *server* yang akan dipantau melalui *network render monitor (Queue)*.

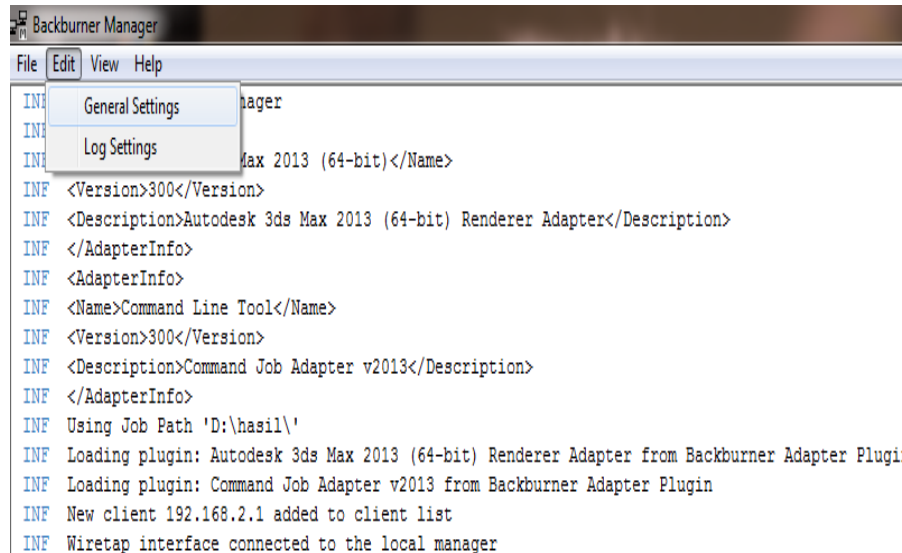
Pengujian Sistem *Rendering Farm*

Dalam tahap pengujian digunakan satu parameter, yaitu dengan membandingkan kinerja *rendering* secara *single computer*, secara *clustering* (komputer paralel) menggunakan aplikasi 3D menggunakan *middleware backburner* dalam proses *rendering*.

Berikut langkah yang dilakukan dalam pengujian:

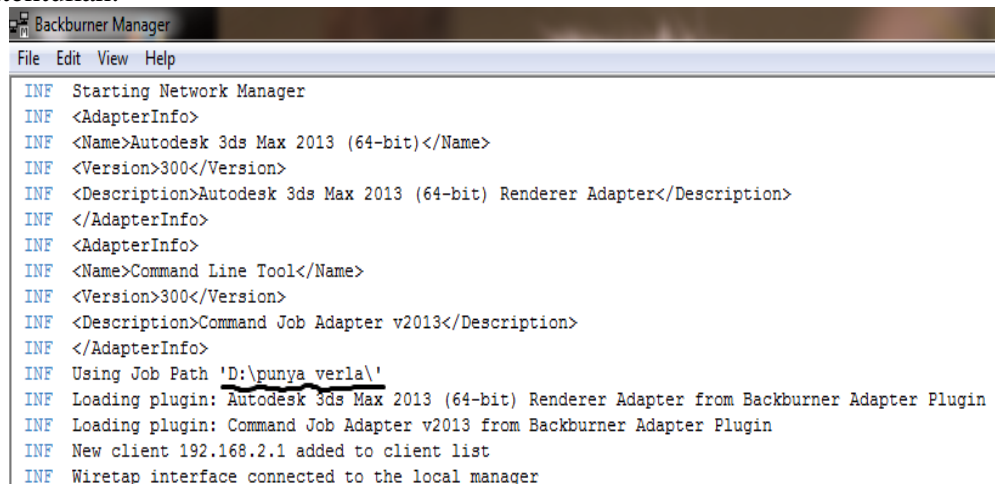
1. Konfigurasi jaringan LAN dengan menggunakan *IP Address* menjadi *static*.
Tahap penyettingan selanjutnya adalah dengan memilih *IPv4* untuk membuat alamat *IP manager* secara statis, agar mudah dibaca atau di temukan oleh klien. Pada komputer *server*-pun yang akan menjadi *slave* tahapan dalam me-*setting*

- alamat *IP* sama dengan tahapan yang dilakukan pada komputer *manager* ini, komputer *manager* akan menjadi komputer *master* dalam sistem *renderfarm*.
2. Konfigurasi sebuah *computer master* (sebagai *manager*)



Gambar 5. Tampilan Tahapan Konfigurasi *Computer Master*

Pada Gambar 6 di bawah ini menunjukkan lokasi penyimpanan yang telah ditentukan.

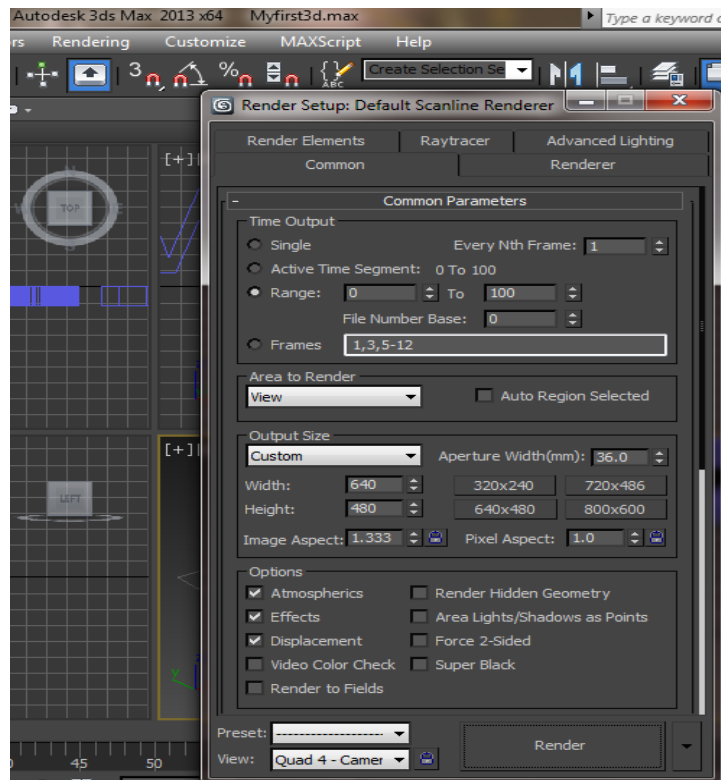


Gambar 6. Tampilan *Manager* dengan Perubahan Lokasi

3. Konfigurasi beberapa komputer sebagai *slave* (sebagai *servers*)

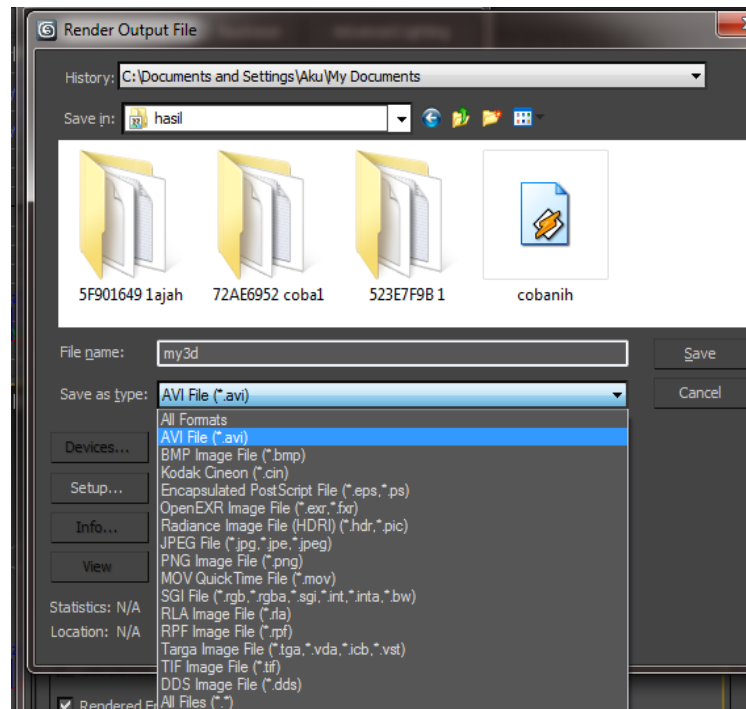
Pengujian Sistem *Render Single Computer*

Tampilan gambar 7 menunjukkan tahapan-tahapan *rendering* secara *single computer*, dimana proses *rendering* dijalankan secara langsung pada aplikasi *3DS Max 2013*.



Gambar 7. Tampilan *Rendering Setup* 3DS MAX 2013

Pada *rendering setup* dipilih tempat penyimpanan dan menyimpan *file* dengan format sesuai kebutuhan.

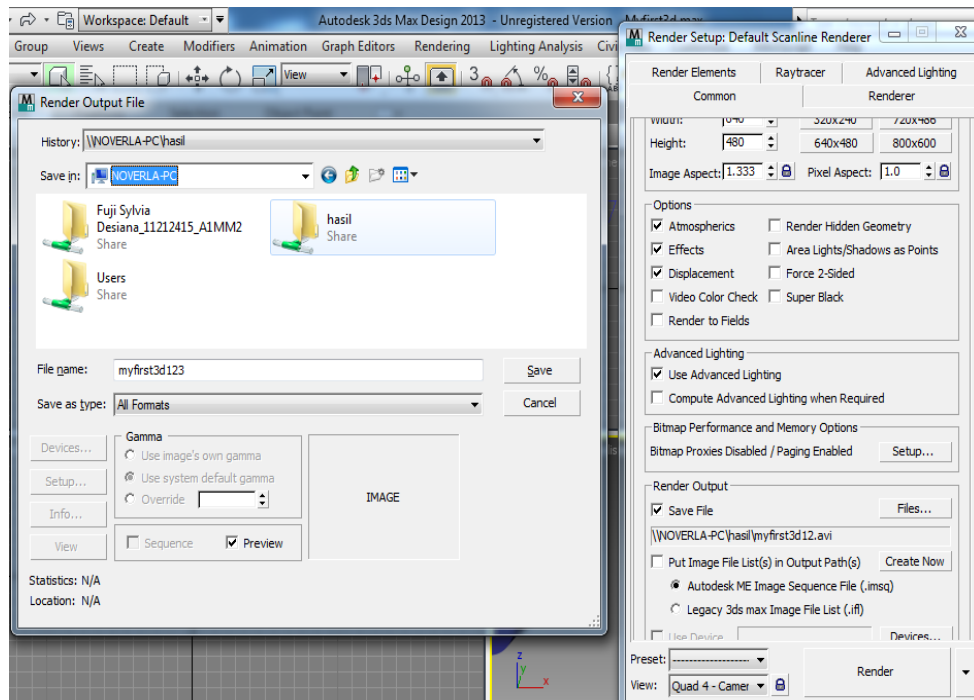


Gambar 8. Tampilan Tempat Penyimpanan Hasil Render

Pengujian Sistem *Render Clustering*

Pada pengujian *render* secara *clustering* proses *render* akan diatur oleh *manager* atau komputer *master*, dan pada *slave* yang sudah diatur pengalamatan *IP* nya sudah dapat digunakan untuk *rendering*, dan proses *render* maupun daftar *job* yang sedang di-*render* dapat dilihat di bagian *network render backburner monitor (Queue)*. Di bawah ini merupakan tahapan *rendering* secara komputer paralel:

1. Jaringan harus saling terkoneksi antara komputer *master* dengan komputer *server* (penyetingannya sudah diterangkan pada sub bab sebelumnya pada penyetingan *backburner*).
2. Buka aplikasi *3D* pada komputer *server* (karena pekerjaan ini dilakukan oleh *servers / slaves*), di sini penulis menggunakan *Autodesk 3DS MAX 2013*, *open file* yang sudah siap untuk di *render* lalu pilih *Render Setup* pada *Menu Render*.



Gambar 9. Tampilan Penyimpanan Hasil Render

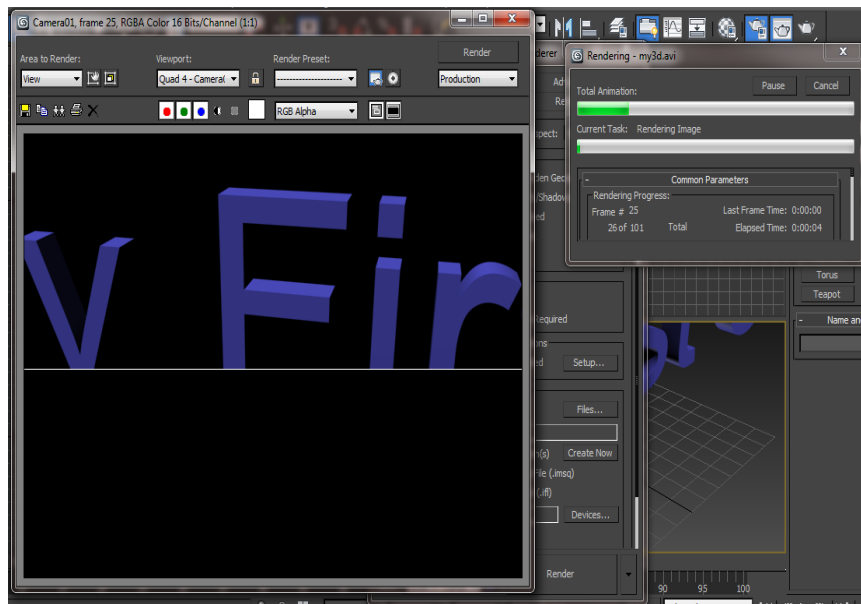
Tampilan Gambar 9 di atas menunjukkan untuk memilih tempat untuk penyimpanan hasil *rendering*, yaitu hasil *rendering* akan disimpan pada *folder* “hasil” yang telah di-*share* secara *localhost* dari *folder* milik komputer *master*, di sini NOVERLA-PC sebagai komputer *master*-nya. Lalu simpan dengan memberi nama dan pilih *type file* yang akan menjadi hasil *render*-nya, di sini penulis menamai *file* hasil *render* dengan nama “myfirst3d123” dengan *type file* (*avi).

3. Selanjutnya adalah tahap *monitoring* dengan membuka mesin *network render monitor*.
Sebelumnya *server* maupun *manager* yang ingin *monitoring* terlebih dahulu masuk dengan mengkonfigurasi alamat *manager* yang dituju, jika sudah melakukan konfigurasi maka komputer yang ingin *monitoring* dapat memonitor pekerjaannya.

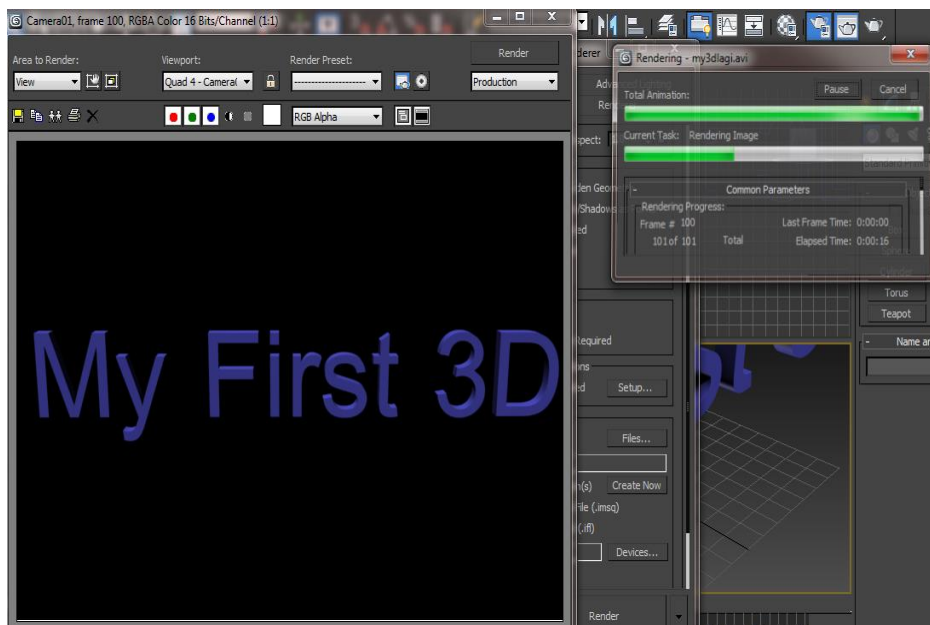
HASIL DAN PEMBAHASAN

Hasil Pengujian Sistem *Render Single Computer*

Dalam hal ini penulis, melakukan *render* dengan memberi nama *file* “my3d” dan menyimpan *file* dengan *format* (*avi) yang akan menghasilkan hasil *render* berupa animasi bergerak.



Gambar 10. Tampilan Proses *Render* Tahap Awal

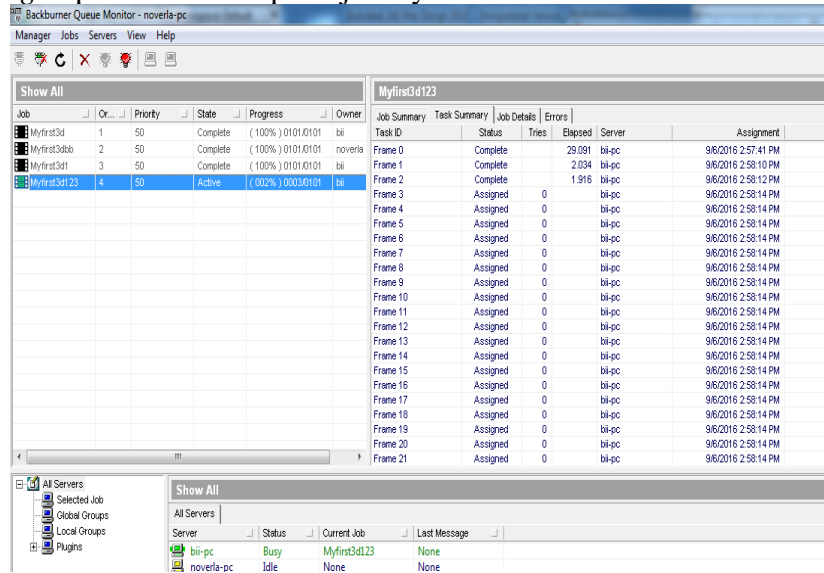


Gambar 11. Tampilan Proses *Render* Tahap Akhir

Dari tampilan gambar-gambar di atas menunjukkan tahapan *render* secara *single computer* dan proses *render* hanya terfokus pada satu *file* yang sedang di-*render* saja, dan jika penulis ingin me-*render file* yang lainnya harus menunggu *file* yang sedang di-*render* selesai.

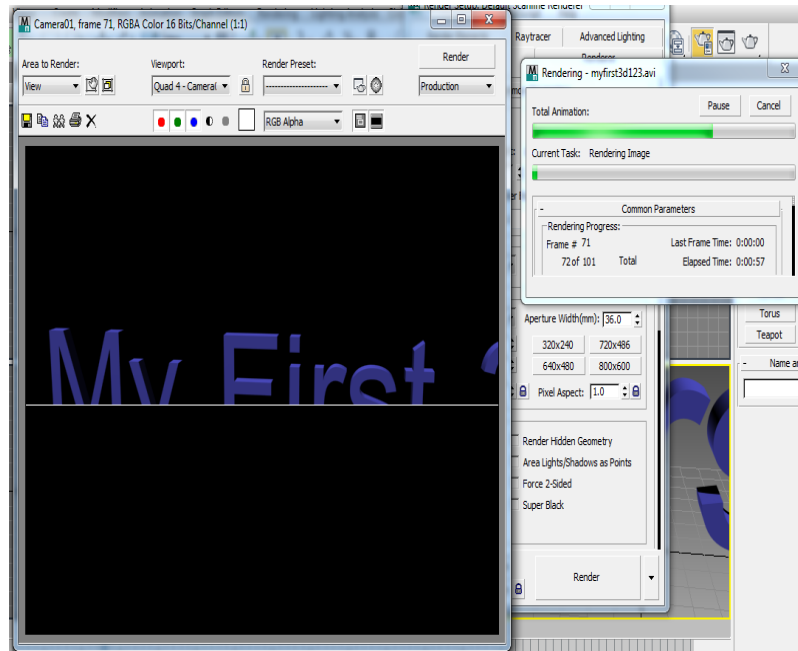
Pengujian Sistem *Render Clustering*

Pada pengujian sistem *Render Clustering*, dilakukan *monitoring* dengan membuka mesin *network render monitor*. Sebelumnya *server* maupun *manager* yang ingin *monitoring* terlebih dahulu masuk dengan mengkonfigurasi alamat *manager* yang dituju, jika sudah melakukan konfigurasi maka komputer yang ingin *monitoring* dapat memonitor pekerjaannya.



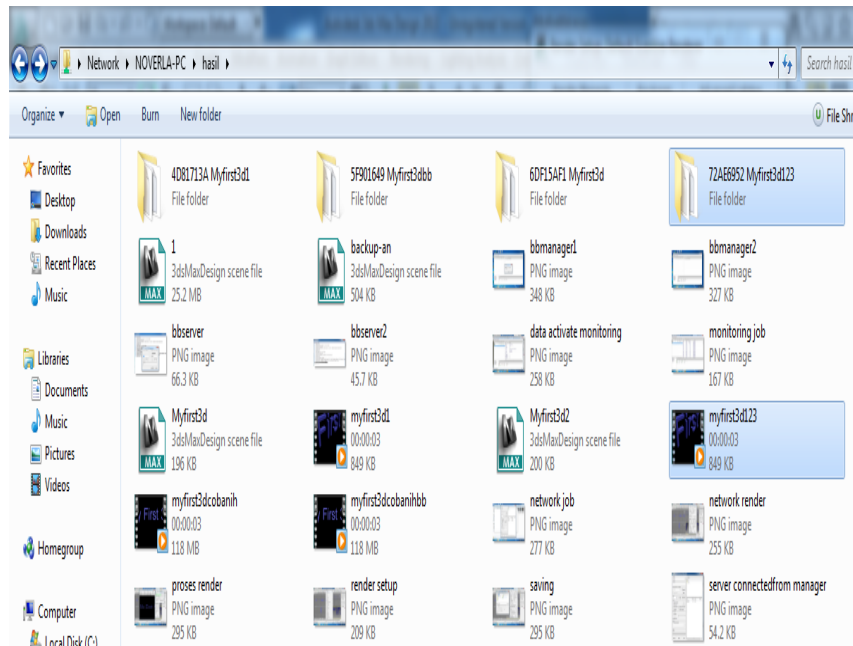
Gambar 12. Tampilan Monitoring

Pada gambar 12 menunjukkan proses *render* yang dilakukan oleh *server* dengan nama komputer “bii-pc” dengan status yang sedang aktif me-render file “myfirst3d123”. Pada info *server* yang terkoneksi menerangkan terdapat dua komputer yang siap melakukan pekerjaan, namun disini yang sedang melaksanakan *render* adalah komputer “bii-pc” namun komputer “noverla-pc” siap untuk diberi pekerjaan.



Gambar 13. Tampilan Proses Rendering

Tampilan di bawah ini merupakan *file* dan *folder* hasil dari *rendering* yang dikerjakan oleh komputer *server* "bii-pc"



Gambar 14. Tampilan Hasil dari Rendering

Dari hasil pengujian menunjukkan bahwa pemanfaatan teknologi komputer paralel dapat membentuk sistem *rendering farm* yang mana tahapan awalnya komputer yang saling terkoneksi dapat mengenali dengan mengatur *IP Address*

secara statis. Komputer *master* berperan sebagai *manager* yang dapat mengatur pekerjaan yang dilakukan oleh komputer *server* atau *slaves* yang saling terkoneksi, dan komputer *manager* pun dapat menjadi komputer *server* atau *slave* untuk melakukan *rendering*, serta dapat juga *monitoring* pekerjaan yang sedang berjalan.

Komputer *server* atau *slave* hanya bisa menerima pekerjaan untuk *me-render* dan juga bisa *monitoring* pekerjaan yang sedang berjalan, karena jika proses *render* terjadi kesalahan atau ingin menghentikan sementara waktu dan dilanjutkan pada hari berikutnya, komputer *server* dapat mengatur pekerjaannya sendiri di bagian *monitoring* tersebut atau pada sistem *network render monitor*. Pada *network render monitor*, mesin tersebut dapat dilihat oleh komputer *master* ataupun komputer *server*. Komputer *server* dapat mengatur *file rendering*-nya di mesin *network render monitor* tersebut, seperti halnya mematikan proses *rendering* dan akan dilanjutkan pada keesokan harinya, atau menghapus *file* yang telah sukses di *render* atau mengalami kesalahan dalam *rendering*-nya.

Penyimpanan hasil dari *rendering* disimpan pada komputer *master* atau komputer *manager*, maka *folder* yang di *sharing* adalah milik komputer *master*. Dengan menggunakan *middleware backburner* melakukan *rendering* dapat lebih efisien waktu karena hanya dengan satu mesin *backburner* tersebut pekerjaan dari beberapa *file* dapat dikerjakan dalam satu waktu bersamaan tanpa harus menunggu *file* antrian pertama selesai. Selain itu, dengan menggunakan *middleware backburner* pun akan lebih hemat penggunaan *hardware software* karena proses *render* terpisah dengan *file* yang *render* maupun hasil *rendering*-nya.

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, ada beberapa hal yang penulis simpulkan, yaitu:

1. Pemanfaatan teknologi komputer paralel dapat membentuk sistem *rendering farm* yang mana tahapan awalnya komputer yang saling terkoneksi dapat mengenali dengan mengatur *IP Address* secara statis.
2. Komputer *master* berperan sebagai *manager* yang dapat mengatur pekerjaan yang dilakukan oleh komputer *server* atau *slaves* yang saling terkoneksi, dan komputer *manager* pun dapat menjadi komputer *server* atau *slave* untuk melakukan *rendering*, serta dapat juga *monitoring* pekerjaan yang sedang berjalan.
3. Dengan menggunakan *middleware backburner* melakukan *rendering* dapat lebih efisien waktu karena hanya dengan satu mesin *backburner* tersebut pekerjaan dari beberapa *file* dapat dikerjakan dalam satu waktu bersamaan tanpa harus menunggu *file* antrian pertama selesai.

4. Dengan menggunakan *middleware backburner* pun akan lebih hemat penggunaan *hardware software* karena proses *render* terpisah dengan *file* yang *render* maupun hasil *rendering*-nya.

SARAN

Dari keterbatasan yang ditemui selama penelitian, disarankan untuk dilakukan penelitian yang lebih lanjut sehingga dapat menangani *file* yang rusak ketika di-*render*. Sistem ini ditujukan untuk menghindari adanya proses *render* yang *error* dan sistem akan secara otomatis memaksa proses *render* terhenti.

DAFTAR PUSTAKA

- [1] Alma'Arif, T. dkk. (2013). "Penentuan Harga Komputasi Grid Komersial dalam *Rendering Farm* Berbasis Waktu dan Beban Prosesor." *E-journal Teknik dan Komputer*, Surabaya.
- [2] Gaus, M.A. dkk. (2011). *Pembangunan Render Farm untuk Blender di Sistem Operasi Microsoft Windows*, Riau.
- [3] Malubaya. dkk. (2014). "Perancangan PC Cluster untuk Render Animasi 3D." *E-journal Teknik dan Komputer*, Manado.
- [4] Pandu, A. (2011). *Blender Template*. [Online], Tersedia: <http://panduaji.net/2011/03/blender-template.html>. [21 Mei 2016].
- [5] Qamar, S. (2014). "Sistem Manajemen Rendering Film Animasi Pada Server Cluster Menggunakan Software Royal Render." *Seminar Nasional Teknologi Informasi dan Multimedia*, Vol. 05. No (1). 57 – 62, Yogyakarta.
- [6] Sandra, W. S. (2013). *Rendering Farm*. [Online], Tersedia: http://sandrawulansari.blogspot.co.id/2013/05/rendering-farm_3.html. [12 Mei 2016]
- [7] Sofana, I. (2008). *Membangun Jaringan Komputer*, Penerbit Informatika, Bandung.
- [8] Suhendra. (2010). *Implementasi Sistem PC Cluster pada Operasi Perkalian Matriks*, Manado.