Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id

E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari

OPEN ACCESS

https://doi.org/10.35870/siskom.v5i1.1318

Analisis Renderfarm Menggunakan Teknologi Cloud Computing dengan Virtual Container Berbasis Docker: Studi Kasus InkubatorX

Muhammad Fajar [1*], NM Faizah [2], Panser Karo - Karo [3]

[1*,2,3] Program Studi Ilmu Komputer, Universitas Tama Jagakarsa, Kota Jakarta Selatan, Daerah Khusus Ibu kota Jakarta, Indonesia.

Corresponding Email: mrfajar232@gmail.com [1*], novianti@jagakarsa.ac.id [2], panserkarokaro@jagakarsa.ac.id [3]

Histori Artikel:
Dikirim 10 Januari 2025; *Diterima dalam bentuk revisi* 15 Januari 2025; *Diterima* 25 Januari 2025; *Diterbitkan* 28 Februari 2025. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIK Indonesia Banda Aceh.

Abstrak

Perkembangan industri animasi 3D terus berkembang pesat, namun sering terkendala oleh keterbatasan sumber daya komputasi yang memadai. Proses rendering, yang mengubah model 3D menjadi gambar nyata, memerlukan daya komputasi tinggi dan sering kali memakan waktu lama. Penelitian ini menggunakan metode eksperimen untuk mengukur waktu eksekusi rendering dengan variasi nilai frame. Variasi nilai frame diterapkan pada setiap komputer slave, kemudian dibandingkan untuk mengetahui percepatan waktu eksekusi. Dalam eksperimen ini, observasi dilakukan untuk memahami dampak perubahan nilai frame terhadap waktu rendering. Dengan penerapan sistem render farm, kesalahan dalam proses rendering dapat otomatis disubmit ulang hingga selesai, dan penggunaan beberapa sumber daya komputasi secara paralel dapat mempercepat proses rendering. Hal ini membuat proses rendering lebih efisien dan membantu menyelesaikan proyek animasi dengan lebih cepat. Proses perancangan sistem dimulai dengan observasi umum, dilanjutkan dengan perancangan perangkat lunak, dan diakhiri dengan penerapan render farm pada server. Pengujian komputer paralel dilakukan dengan menyelaraskan waktu antara Master, Slave, dan Client menggunakan protokol waktu jaringan. Hasil pengujian menunjukkan bahwa penggunaan render farm dapat mempercepat waktu render secara signifikan, memberikan solusi yang lebih efisien dalam mengatasi keterbatasan perangkat keras dalam proses rendering animasi 3D.

Kata Kunci: Render farm; Animasi 3D; Rendering; Eksperimen; Efisiensi.

Abstract

The 3D animation industry continues to grow rapidly, but it is often hindered by the limitations of adequate computational resources. The rendering process, which converts 3D models into realistic images, requires significant computational power and often takes a long time. This study uses an experimental method to measure execution time during rendering with varying frame values. These variations are applied to each slave computer and compared to determine the speedup in execution time. Observations were made to understand the impact of changing frame values on rendering time. With the implementation of a render farm system, errors in the rendering process can automatically be re-submitted until completion, and utilizing multiple computational resources in parallel can speed up the rendering process. This makes the rendering process more efficient and helps complete animation projects more quickly. The system design process begins with a general observation, followed by software design, and concludes with the deployment of the render farm on the server. Parallel computer testing was carried out by synchronizing the time between the Master, Slave, and Client using network time protocols. The results of the tests indicate that the use of a render farm can significantly reduce rendering time, providing an efficient solution to overcome hardware limitations in 3D animation rendering.

Keyword: Render farm; 3D animation; Rendering; Experiment; Efficiency.

Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318

1. Pendahuluan

Perkembangan animasi 3D mengalami pertumbuhan pesat, namun sering terhambat oleh kebutuhan komputasi yang sangat besar, khususnya dalam proses rendering. Rendering adalah proses yang mengubah model tiga dimensi menjadi gambar yang terlihat nyata, dengan pencahayaan dan tekstur yang sesuai. Proses ini memerlukan daya komputasi tinggi karena setiap titik koordinat dalam model harus dihitung, dan kompleksitas data yang digunakan dapat memperpanjang waktu rendering secara signifikan (Felani *et al.*, 2020). InkubatorX menghadapi tantangan serupa, di mana keterbatasan perangkat keras sering kali menghambat kemampuan untuk melakukan rendering secara efisien. Salah satu solusi yang banyak diterapkan untuk mengatasi masalah ini adalah pembangunan sistem render farm. Render farm memungkinkan beberapa komputer bekerja secara bersamaan dalam jaringan untuk mempercepat proses rendering melalui komputasi paralel (Kim *et al.*, 2022). Dengan membagi beban kerja antar komputer, proyek yang memerlukan waktu render lama dapat diselesaikan lebih cepat. Namun, untuk mengimplementasikan render farm secara efektif, dibutuhkan infrastruktur yang tepat untuk mendukung pembagian tugas dan distribusi beban secara efisien ("Security Implications In Docker Based Virtual Environment", 2021). Teknologi cloud computing menjadi salah satu solusi yang menjanjikan, karena memungkinkan penggunaan sumber daya komputasi secara fleksibel dan skalabel, tanpa terikat oleh perangkat keras fisik tertentu (Verma & Dhawan, 2017). Docker, sebagai platform yang memungkinkan pengemasan aplikasi beserta semua dependensinya dalam container, dapat menjadi alat yang sangat berguna dalam implementasi sistem render farm berbasis cloud. Dengan Docker, aplikasi rendering dapat dijalankan dalam lingkungan terisolasi, yang memungkinkan distribusi tugas secara efisien antar komputer tanpa khawatir tentang masalah kompatibilitas (Zerouali *et al.*, 2019). Selain itu, Docker juga mempermudah manajemen dan pemeliharaan sistem, sehingga memaksimalkan performa render farm (Sergeev *et al.*, 2022). Penelitian menunjukkan bahwa penggunaan Docker dalam cloud computing dapat meningkatkan kecepatan dan efisiensi proses rendering 3D dengan memanfaatkan sumber daya yang ada secara optimal (Saha *et al.*, 2018). Penelitian ini bertujuan untuk menganalisis penggunaan cloud computing dan Docker dalam membangun sistem render farm di InkubatorX. Fokus utama penelitian ini adalah pada perancangan dan implementasi sistem yang dapat meningkatkan kecepatan dan efisiensi proses rendering 3D. Dengan menerapkan pendekatan ini, diharapkan InkubatorX dapat mengatasi masalah keterbatasan perangkat keras dan memanfaatkan teknologi modern untuk meningkatkan produktivitas dalam proyek animasi.

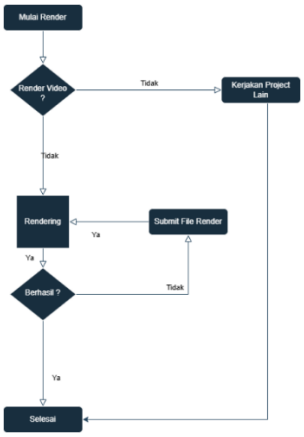
Melalui penelitian ini, diharapkan dapat ditemukan solusi yang tepat untuk mendukung proses rendering animasi 3D yang lebih cepat dan efisien, serta memberikan gambaran mengenai keuntungan dan tantangan penerapan teknologi cloud computing dan Docker di dunia animasi. Renderfarm merupakan sistem komputasi yang digunakan untuk mempercepat proses rendering dalam produksi grafis visual, animasi, dan pengolahan data besar. Proses rendering sering kali memerlukan daya komputasi yang besar, sehingga membutuhkan perangkat keras dan sumber daya yang dapat diakses secara efisien (Cho & Bahn, 2021). Salah satu solusi untuk mengatasi tantangan ini adalah dengan memanfaatkan teknologi cloud computing yang memungkinkan alokasi sumber daya komputasi secara fleksibel dan skalabel (Annette *et al.*, 2018). Dalam beberapa tahun terakhir, Docker sebagai platform virtualisasi berbasis container telah mendapatkan perhatian signifikan karena kemampuannya dalam menyediakan lingkungan yang konsisten dan terisolasi untuk aplikasi. Dengan Docker, proses deploy dan manajemen aplikasi menjadi lebih efisien, yang sangat menguntungkan bagi sistem seperti renderfarm yang memerlukan pengelolaan banyak node komputasi secara bersamaan (Setyaningfebry *et al.*, 2023). Teknologi ini juga memungkinkan penggunaan sumber daya yang lebih optimal, dengan memanfaatkan infrastruktur cloud yang terdistribusi (Wang *et al.*, 2019). Penelitian menunjukkan bahwa penerapan Docker dalam konteks cloud rendering dapat meningkatkan efisiensi dan mengurangi biaya operasional (Annette & Banu, 2018). Penelitian ini bertujuan untuk menganalisis penerapan renderfarm dengan teknologi cloud computing yang menggunakan virtual container berbasis Docker, dengan fokus pada studi kasus InkubatorX. InkubatorX dipilih sebagai studi kasus karena memiliki kebutuhan yang tinggi terhadap

Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E - ISSN: 3047-7565 | P - ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318

komputasi grafis dan merupakan contoh perusahaan yang mengintegrasikan teknologi cloud computing dalam operasionalnya (Varghese & Buyya, 2018). Melalui analisis ini, diharapkan dapat ditemukan wawasan baru mengenai efisiensi, skalabilitas, dan potensi penerapan teknologi ini dalam industri kreatif dan pengolahan data besar (Santiko & Rosidi, 2018). Dengan memanfaatkan cloud render farm, InkubatorX dapat mengatasi keterbatasan perangkat keras dan meningkatkan produktivitas dalam proyek animasi (Fuchs *et al.*, 2019).

2. Metode Penelitian

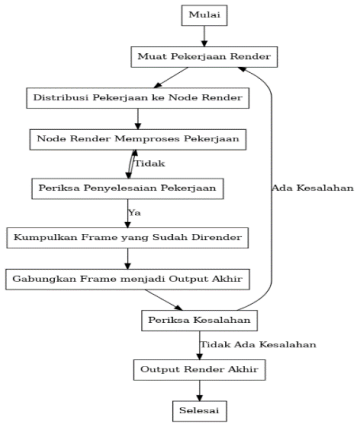
Penelitian ini dilakukan di InkubatorX, sebuah perusahaan yang bergerak di bidang animasi 3D dan teknologi. Pemilihan InkubatorX sebagai lokasi penelitian didasarkan pada relevansi permasalahan yang dihadapi perusahaan terkait dengan keterbatasan perangkat keras dalam proses rendering animasi. InkubatorX memiliki proyek-proyek yang melibatkan rendering animasi 3D yang kompleks, yang membutuhkan daya komputasi tinggi. Dengan keterbatasan perangkat keras yang ada, perusahaan ini mengalami kesulitan dalam menjalankan proses rendering secara efisien. Oleh karena itu, penelitian ini bertujuan untuk mencari solusi yang dapat mengatasi masalah tersebut. Penelitian ini berlangsung selama 7 bulan, dimulai dari Januari 2024 hingga Juli 2024. Metode yang digunakan dalam penelitian ini adalah eksperimen untuk menganalisis efisiensi sistem render farm berbasis teknologi cloud computing dengan virtual container berbasis Docker. Dalam eksperimen ini, peneliti menguji berbagai konfigurasi render farm dengan variasi nilai frame rendering pada beberapa komputer slave yang terhubung dalam jaringan. Setiap komputer slave diatur untuk menjalankan tugas render tertentu, dan waktu eksekusi untuk setiap tugas diukur untuk mengevaluasi kinerja sistem. Eksperimen dilakukan dengan menggunakan perangkat lunak animasi 3D Blender pada setiap komputer slave, yang menjalankan proses rendering menggunakan file animasi yang sama namun dengan nilai frame yang bervariasi. Variasi nilai frame tersebut akan memberikan gambaran tentang seberapa besar pengaruh jumlah frame terhadap waktu rendering, serta bagaimana beban kerja dapat dibagi antara berbagai komputer slave dalam jaringan render farm. Hasil dari eksperimen ini kemudian dianalisis untuk melihat seberapa efektif penerapan teknologi cloud computing dan Docker dalam mempercepat proses rendering animasi 3D. Selain itu, observasi dilakukan untuk memahami pengaruh parameter teknis lainnya, seperti jumlah komputer yang terlibat dalam sistem dan konfigurasi jaringan, terhadap waktu eksekusi render. Penelitian ini diharapkan dapat memberikan pemahaman yang lebih baik mengenai penerapan teknologi cloud dan Docker dalam meningkatkan efisiensi proses rendering animasi 3D, serta memberikan solusi yang praktis bagi perusahaan seperti InkubatorX untuk mengatasi keterbatasan perangkat keras yang mereka hadapi.



Gambar 1. Rancangan Sistem Terdahulu

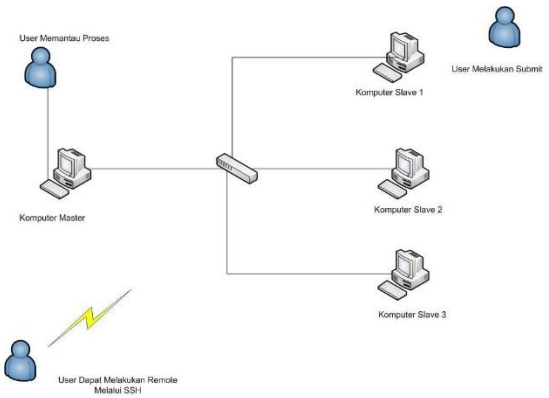
Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318

Dalam flowchart diatas terlihat proses rendering sebelum adanya sistem renderfarm dilakukan secara manual dan rentan error saat proses rendering dan memakan banyak waktu karena jika rendering gagal atau tidak berhasil file yg direnderakan disubmit manual dan akan memakan banyak waktu sedangkan project di InkubatorX, dibutuhkan keandalan dan efesiensi waktu untuk menyelesaikan project dengan cepat dan efesien.



Gambar 2. Rancangan Sistem Terbaru

Dengan adanya sistem renderfarm maka pekerjaan render video akan lebih mudah karena jika terdapat kesalahan dalam rendering akan otomatis disubmit ulang hingga selesai di kerjakan terlebih lagi akan menghemat waktu karena rendering video akan di bantu beberapa resource (tergantung jumlah komputer) yang akan membantu proses ini lebih mudah cepat serta efisen,tentu nya membantu menyelesaikan project. Sistem direncanakan untuk mencerminkan struktur keseluruhan dari sistem yang dibuat. Arsitektur sistem yang telah direncanakan dapat dilihat melalui gambar berikut ini.



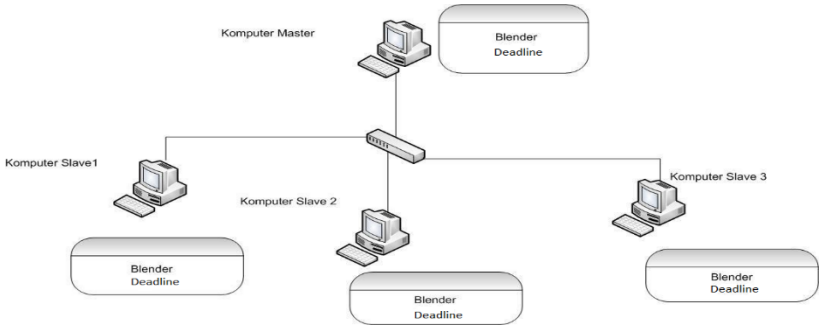
Gambar 3. Rancangan Sistem Render Farm

Render farm dapat dipahami sebagai penggunaan beberapa komputer yang bekerja bersama-sama untuk melakukan proses rendering. Komputer-komputer tersebut terhubung dalam jaringan di mana salah satunya bertindak sebagai komputer master yang digunakan untuk memantau proses kerja render farm ketika sedang berjalan. Komputer lainnya beroperasi sebagai slave untuk menjalankan proses rendering secara bersamaan.

Blender, sebuah perangkat lunak untuk membuat animasi 3D, digunakan sebagai alat tambahan dalam sistem render farm untuk melakukan proses rendering terhadap file .blend sebelum dikirimkan ke dalam render farm. Sebagai perangkat lunak animasi 3D, fungsi render Blender akan digunakan oleh sistem render farm untuk menyelesaikan setiap pekerjaan rendering yang dikirimkan oleh pengguna. Selain itu, kita memiliki Deadline, yang merupakan alat yang

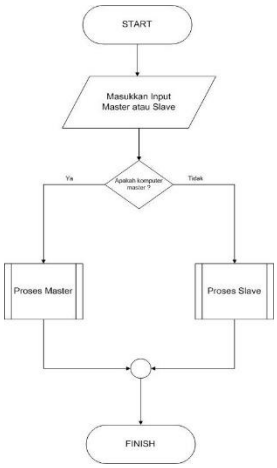
Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318

digunakan untuk membangun sistem render farm. Tugas-tugas dalam sistem render farm akan dilaksanakan sesuai dengan tenggat waktu, seperti distribusi frame untuk proses rendering, pengaturan koneksi antar server, serta pekerjaan-pekerjaan lain yang dibutuhkan.



Gambar 4. Blender dan Deadline dalam Sistem Renderfarm

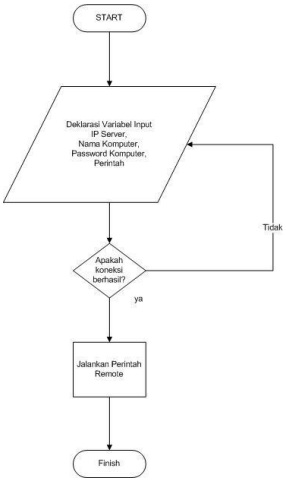
Program rancangan dibagi menjadi tiga proses rancangan algoritma, yaitu algoritma untuk instalasi sistem, aktivasi sistem, submit job, dan remote.



Gambar 5. Flowchart Instalasi Sistem Render Farm

Pada Gambar 5. diuraikan diagram alir langkah-langkah pemasangan sistem secara rinci. Karena terdapat dua jenis komputer yang digunakan, setiap jenis memiliki fungsi yang berbeda. Pengguna akan memiliki dua opsi proses instalasi yang dapat dipilih, yaitu sebagai master atau slave. Setiap komputer, baik master maupun slave, memiliki proses dan input yang berbeda.

Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E - ISSN: 3047-7565 | P - ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318



Gambar 6. Flowchart Rancangan Remote Sistem

Flowchart berikutnya menunjukkan proses remote terhadap server komputer. Pada tahap ini, pengguna diminta untuk memasukkan informasi berupa alamat IP tujuan, nama komputer, dan perintah yang akan dijalankan. Sistem akan mencoba terhubung dengan komputer yang dituju. Jika koneksi gagal, sistem akan meminta pengguna untuk memasukkan kembali informasi tersebut.

3. Hasil dan Pembahasan

3.1. Hasil

3.1 Implementasi Rancangan

Pembangunan komputer paralel yang sedang diuji dimulai dengan menyelaraskan waktu masing-masing komputer Master, Slave, dan Client menggunakan protokol waktu jaringan. Semua komputer dilengkapi dengan penunjuk waktu untuk memastikan pengujian dapat dilakukan secara seragam pada semua komputer. Perlunya waktu yang konsisten untuk mengetahui durasi eksekusi yang dibutuhkan saat proses rendering. Dua unit komputer digunakan sebagai Slave dalam penelitian ini, dengan perbedaan jumlah gpu yaitu satu gpu dipasang pada setiap komputer di kedua perangkat tersebut. Komputer Slave digunakan untuk menjalankan proses rendering. Semakin banyak komputer Slave yang digunakan, maka semakin singkat waktu yang dibutuhkan untuk melakukan proses rendering.

Tabel 1. Hardware yang di gunakan untuk pengujian

Hardware	
CPUs	Intel Core i3-10100 @3.60GHz (8CPUs)
RAM	32GB
SSD	512GB
GPU	NVIDIA GeForce RTX 3070

Tabel 2. Software yang di gunakan untuk pengujian

Software	
Windows 10	*Windows 10 Profesional (64 bit) and later
NVIDIA Driver	*Game ready driver
Thinkbox Deadline	v10.1
MongooDB	*include in thinkbox deadline
Blender	v24.0

Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318

Docker	*New version

Tabel 3. Render Farm Data

No	Title	Duration	Frame	Software	Source
1	BMW27	00:00	1	Blender	https://www.blender.org/download/demo-files/
2	InkubatorX	00:16	480	Blender	Kalunk (tigadimensi)
3	Hi, my name is Amy'	00:06	180	Blender	https://www.blender.org/download/demo-files/

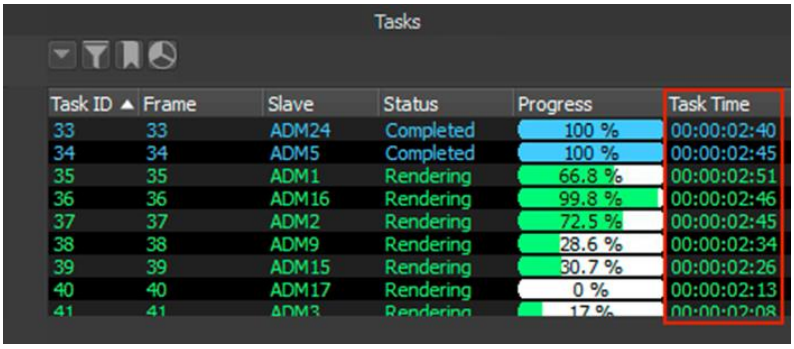
Gambar 6. Asset File BMW27

Gambar 7. Asset File Inkubator X

3.1.2 Skenario Pengujian

Setelah rancangan dan implementasi sistem render farm dibahas, kini saatnya penulis melakukan pengujian terhadap sistem yang telah dibangun untuk mengukur efisiensi waktu serta fleksibilitas pengerjaan. Proses pengujian ini bertujuan untuk memastikan bahwa sistem render farm dapat bekerja dengan baik dan memenuhi tolok ukur yang telah ditetapkan. Langkah pertama adalah membuka Deadline Monitor. Setelah itu, pengguna dapat membuka tab Submit, lalu pilih 3D dan Blender untuk mengirimkan proyek Blender ke sistem render farm. Pada langkah selanjutnya, pengguna diminta untuk mengatur Job name dan memilih Group, yaitu memilih worker yang digunakan untuk rendering, seperti rig-001, rig-002, atau semua GPU yang terhubung. Selanjutnya, pengguna harus menentukan Blender file, dengan menyimpan file Blender di path renderwork/tigadimensi/submit untuk file yang akan dirender, dan renderwork/tigadimensi/result untuk hasil rendering. Setelah itu, pengguna dapat mengatur frame list sesuai kebutuhan, dan jika pengaturan sudah selesai, proyek dapat langsung disubmit ke server render farm. Hasil pengujian menunjukkan bahwa file yang dirender berhasil tanpa mengalami error. Untuk memastikan hasil rendering, pengguna dapat memeriksa file explorer untuk melihat hasil rendering yang telah selesai. Selain menggunakan Deadline Monitor untuk merender, pengguna juga dapat merender secara langsung melalui aplikasi atau software dengan integrasi plugin yang disediakan oleh Deadline. Caranya adalah dengan membuka aplikasi Blender, kemudian pergi ke menu Edit dan pilih Preferences. Setelah itu, pengguna harus mencentang plugin Submit Blender To Deadline. Setelah plugin aktif, pengguna dapat melakukan submit proyek dengan cara yang sama seperti yang telah dijelaskan sebelumnya.

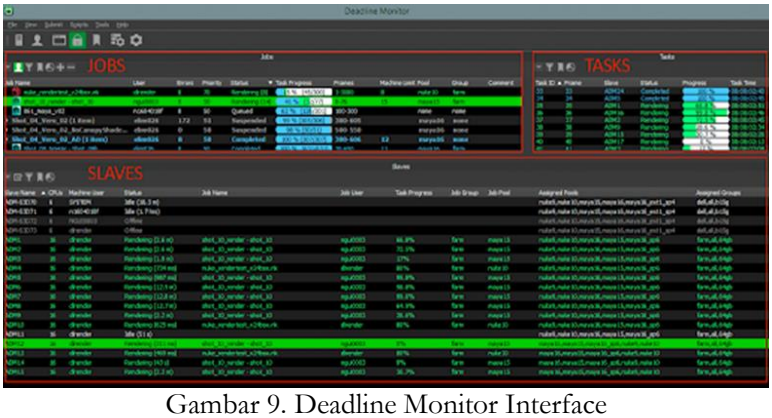
Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318



Gambar 8. Deadline Monitor Rendering Result

Penulis akan menjelaskan secara lebih rinci hasil pengujian yang telah dilakukan sebelumnya, guna menganalisis kinerja sistem render farm agar berjalan lebih optimal dan efisien. Langkah pertama dalam pengujian adalah membuka Deadline Monitor yang terletak di sistem tray di pojok kanan bawah layar. Deadline Monitor ini berfungsi untuk memantau proses pengerjaan video, memonitor frames, estimasi waktu pengerjaan, serta jumlah worker (komputer) yang sedang digunakan atau tidak. Dengan menggunakan Deadline Monitor, pengguna dapat dengan mudah memonitor dan mengelola pengerjaan render farm secara efisien.

1) Tabel Hasil Test Case Render 1
   Pada tabel di atas, terlihat bahwa pengujian test case pertama menunjukkan bahwa dengan jumlah frame sebanyak 30 dan ukuran file 3,56 MB, proses rendering menggunakan render farm lebih cepat 39 detik.
2) Tabel Hasil Test Case Render 2
   Pada tabel di atas, terlihat bahwa pengujian test case kedua menunjukkan bahwa dengan jumlah frame sebanyak 500 dan ukuran file 7,77 MB, proses rendering menggunakan render farm lebih cepat 37 menit.
3) Tabel Hasil Test Case Render 3
   Pada tabel di atas, terlihat bahwa pengujian test case ketiga menunjukkan bahwa dengan jumlah frame sebanyak 181 dan ukuran file 1,97 MB, proses rendering menggunakan render farm lebih cepat 29 menit dan 77 detik.
4) Tabel Temperature GPU Sebelum dan Setelah Render
   Pengujian ketiga test case menunjukkan terjadinya peningkatan suhu pada GPU (Graphic Processing Unit) saat sirkulasi udara kurang optimal dalam perakitan hardware. Oleh karena itu, diperlukan kipas tambahan untuk menurunkan suhu pada GPU.



Gambar 9. Deadline Monitor Interface

Penulis akan jelaskan mengenai Deadline Monitor. Terdapat banyak informasi yang terdapat pada gambar di atas, yaitu:

Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318

1) Job
   a) ID Job: Identifikasi unik untuk setiap job.
   b) Nama Job: Nama yang diberikan pada job.
   c) Status: Status saat ini dari job (misalnya, Antri, Rendering).
   d) Selesai: Persentase penyelesaian job.
   e) Progress Tugas: Indikator visual kemajuan tugas.
   f) Pengguna: Pengguna yang mengirimkan job.
   g) Prioritas: Tingkat prioritas job.
   h) Ukuran Job: Jumlah frame atau tugas dalam job.
   i) Komentar: Catatan tambahan yang dapat diberikan oleh pengguna.
   j) Gambar 38: Taskbar Deadline Monitor
2. Slave
   a) Nama Slave: Nama node render yang terhubung.
   b) Status: Status node (misalnya, Idle, Rendering).
   c) Nama Job: Job yang sedang dikerjakan pada node tersebut.
   d) E. Progress Tugas: Kemajuan tugas yang sedang dikerjakan.
   e) ID Tugas: ID tugas yang sedang dikerjakan.
   f) Pool yang Ditugaskan: Pool render tempat node tersebut beroperasi.
   g) Grup yang Ditugaskan: Grup render tempat node tersebut terhubung.
   h) Penggunaan RAM: Penggunaan memori oleh node tersebut.
   i) Penggunaan CPU: Pemakaian prosesor oleh node tersebut.
   j) Komentar: Catatan tambahan terkait status node.
3. Fitur Tambahan:
   a. Tombol Toolbar: Untuk mengelola job dan slave, seperti memulai, menghentikan, atau menjeda.
   b. Filter dan Pencarian: Opsi untuk memfilter dan mencari job atau slave tertentu.
   c. Statistik dan Log: Akses statistik kinerja dan log sistem.
   d. Manajemen Prioritas dan Pool: Mengelola prioritas job dan alokasi sumber daya ke pool tertentu.
   e. Fitur-fitur tersebut membantu pengguna dalam mengelola dan memantau tugas rendering di berbagai node dalam render farm dengan lebih efisien.

3.2 Pembahasan

Penerapan sistem render farm berbasis Docker yang diuji dalam penelitian ini berfokus pada optimasi penggunaan sumber daya komputasi dalam proses rendering animasi 3D. Hasil eksperimen menunjukkan bahwa penggunaan render farm dapat mempercepat waktu rendering dengan signifikan, sesuai dengan temuan yang dijelaskan oleh Felani *et al.* (2020) yang menyatakan bahwa penggunaan Docker dalam manajemen sumber daya virtual dapat meningkatkan efisiensi dalam pengelolaan aplikasi berbasis cloud. Dengan memanfaatkan beberapa komputer slave dalam jaringan, beban rendering dibagi secara paralel, yang pada gilirannya mengurangi waktu rendering, seperti yang ditemukan dalam eksperimen ini (Kim *et al.*, 2022).

Sistem render farm yang diterapkan di InkubatorX menunjukkan kemudahan dalam pengelolaan job dan distribusi tugas rendering, terutama dengan penggunaan Deadline Monitor. Dengan adanya sistem ini, pengguna dapat memantau dan mengelola proses rendering secara efisien, sebagaimana dijelaskan oleh Sergeev *et al.* (2022) yang menekankan pentingnya kemudahan dalam mengelola kontainer Docker untuk aplikasi komputasi berat. Melalui integrasi sistem ini, kesalahan dalam proses rendering otomatis disubmit ulang hingga selesai, yang tidak hanya mempercepat tetapi juga meminimalkan kemungkinan kesalahan yang mempengaruhi proyek animasi. Penting untuk dicatat, seperti yang disarankan oleh Zerouali *et al.* (2019), bahwa penggunaan kontainer Docker yang terbarukan dan terkelola dengan baik memiliki dampak signifikan terhadap kinerja sistem. Oleh karena itu, pemeliharaan dan pembaruan kontainer secara teratur sangat penting untuk mencegah masalah yang terkait dengan kerentanannya. Selain itu,

Jurnal Sistem Komputer (SISKOM)
https://journal.stmiki.ac.id
E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318

penggunaan sistem render farm berbasis cloud computing, seperti yang dijelaskan oleh Wang *et al.* (2019), memungkinkan distribusi sumber daya yang lebih optimal dan meminimalkan bottleneck dalam proses rendering.

Hasil pengujian ini menunjukkan bahwa render farm berbasis Docker dapat menjadi solusi yang efektif untuk mengatasi keterbatasan perangkat keras, mengoptimalkan penggunaan sumber daya komputasi, dan meningkatkan efisiensi dalam proyek animasi 3D. Hal ini sejalan dengan rekomendasi dari Annette *et al.* (2018) yang menyatakan bahwa menggunakan layanan render farm berbasis cloud dapat meningkatkan performa rendering, mengurangi biaya operasional, dan memberikan solusi yang lebih fleksibel dalam pengelolaan sumber daya komputasi. Penerapan render farm berbasis Docker di InkubatorX terbukti efektif dalam meningkatkan kecepatan dan efisiensi rendering, serta memberikan solusi praktis untuk mengatasi kendala keterbatasan perangkat keras dalam produksi animasi 3D. Teknologi ini membuka jalan bagi perusahaan seperti InkubatorX untuk meningkatkan produktivitas dan memenuhi kebutuhan rendering animasi yang semakin kompleks dan membutuhkan daya komputasi tinggi.

4. Kesimpulan

Berdasarkan rumusan masalah yang telah penulis susun dan melalui tahapan analisis sistem, perancangan sistem, implementasi berupa pembuatan, serta pengujian, dapat disimpulkan beberapa poin penting. Sistem render farm yang telah dirancang dan diimplementasikan menggunakan teknologi cloud computing terbukti menjadi solusi efektif untuk mengatasi keterbatasan sumber daya komputasi dalam proses rendering animasi 2D dan 3D. Melalui analisis yang telah dilakukan, penambahan Graphical User Interface (GUI) pada sistem render farm telah terbukti sangat membantu dalam mempercepat dan mempermudah berbagai proses penting. Proses seperti instalasi, aktivasi server, serta pengiriman, pemantauan, dan peninjauan hasil pekerjaan pengguna dapat dilakukan dengan lebih efisien dan terstruktur. GUI yang ditambahkan ke dalam sistem render farm memungkinkan pengguna untuk mengelola seluruh proses render dengan lebih intuitif. Pengguna tidak perlu lagi melalui langkah-langkah yang rumit dan teknis karena antarmuka grafis mempermudah akses ke berbagai fungsi utama dalam sistem. Hal ini juga meningkatkan kenyamanan pengguna dalam melakukan pengaturan dan pemantauan proses rendering secara real-time. Selain itu, berdasarkan hasil pengujian yang telah dilakukan, penulis menyimpulkan bahwa render farm merupakan solusi yang efektif dalam menangani masalah rendering gambar dan video. Terdapat perbedaan waktu rendering yang signifikan, dengan render farm yang mampu menyelesaikan pekerjaan lebih cepat dan efisien dibandingkan dengan metode rendering tradisional. Dengan adanya render farm, proses produksi animasi dapat dilakukan dengan lebih optimal dan mengurangi ketergantungan pada perangkat keras yang terbatas.

5. Daftar Pustaka

Annette, J. and Banu, W. (2018). Ranking cloud render farm services for a multi criteria recommender system. Sadhana, 44(1). https://doi.org/10.1007/s12046-018-0981-0

Annette, R., W, A., & P, S. (2018). A multi criteria recommendation engine model for cloud renderfarm services. International Journal of Electrical and Computer Engineering (Ijece), 8(5), 3214. https://doi.org/10.11591/ijece.v8i5.pp3214-3220

Cho, K. and Bahn, H. (2021). Design and implementation of a distributed versioning file system for cloud rendering. Ieee Access, 9, 138716-138723. https://doi.org/10.1109/access.2021.3119077

E-ISSN: 3047-7565 | P-ISSN: 2775-3115
Vol. 5 No. 1 (2025) | Februari
OPEN ACCESS
https://doi.org/10.35870/siskom.v5i1.1318

Felani, R., Al-Azam, M., Adi, D., Widodo, A., & Gumelar, A. (2020). Optimizing virtual resources management using docker on cloud applications. Ijccs (Indonesian Journal of Computing and Cybernetics Systems), 14(3), 319. https://doi.org/10.22146/ijccs.57565

Fuchs, L., Orero, L., Namoi, N., & Neufeldt, H. (2019). How to effectively enhance sustainable livelihoods in smallholder systems: a comparative study from western kenya. Sustainability, 11(6), 1564. https://doi.org/10.3390/su11061564

Kim, B., Lee, S., Lee, Y., Park, Y., & Jeong, J. (2022). Design and implementation of cloud docker application architecture based on machine learning in container management for smart manufacturing. Applied Sciences, 12(13), 6737. https://doi.org/10.3390/app12136737

Santiko, I. and Rosidi, R. (2018). Pemanfaatan private cloud storage sebagai media penyimpanan data e-learning pada lembaga pendidikan. Jurnal Teknik Informatika, 10(2), 137-146. https://doi.org/10.15408/jti.v10i2.6992

Sergeev, A., Rezedinova, E., & Khakhina, A. (2022). Stress testing of docker containers running on a windows operating system. Journal of Physics Conference Series, 2339(1), 012010. https://doi.org/10.1088/1742-6596/2339/1/012010

Setyaningfebry, F., Felasari, S., & Michelle, B. (2023). Developing a website for rendering based on renderers' experience. Grafica, 11(21), 23-36. https://doi.org/10.5565/rev/grafica.235

Varghese, B. and Buyya, R. (2018). Next generation cloud computing: new trends and research directions. Future Generation Computer Systems, 79, 849-861. https://doi.org/10.1016/j.future.2017.09.020

Verma, M. and Dhawan, M. (2017). Towards a more reliable and available docker-based container cloud.. https://doi.org/10.48550/arxiv.1708.08399

Wang, R., Zhang, B., Wu, M., Zhang, J., Guo, X., Zhang, X., … & Ma, S. (2019). Performance bottleneck analysis and resource optimized distribution method for iot cloud rendering computing system in cyber-enabled applications. Eurasip Journal on Wireless Communications and Networking, 2019(1). https://doi.org/10.1186/s13638-019-1401-9

Zerouali, A., Mens, T., Robles, G., & González-Barahona, J. (2019). On the relation between outdated docker containers, severity vulnerabilities, and bugs., 491-501. https://doi.org/10.1109/saner.2019.8668013

(2021). Security implications in docker based virtual environment.. https://doi.org/10.29121/web/v18i4/141