

PERBANDINGAN BANYAKNYA *FRAME RENDER* DENGAN PENGGUNAAN SUMBERDAYA YANG DIGUNAKAN MENGGUNAKAN ANOVA ONE WAY DENGAN MEMANFAATKAN *PARALLEL COMPUTING RENDER ENGINE*

Ichwan Kurniawan, Much. Rifqi Maulana
STMIK Widya Pratama
Jl. Patriot 25 Pekalongan. Telp (0285)427816
email : ichwan.ana10@gmail.com, rifqi_13@yahoo.com

ABSTRAKS

Untuk memperoleh hasil render yang maksimal dengan waktu render yang singkat dapat memanfaatkan beberapa sumberdaya komputer secara bersama-sama dalam sebuah jaringan. Dengan memanfaatkan *Parallel computing render engine* untuk *rendering 3D model* pada *open-source software* agar hasil akhir dari *rendering 3D model* dapat lebih maksimal dan waktu *rendering* menjadi lebih cepat. Dalam penelitian ini *render engine* yang digunakan adalah *blender network render*, dengan menggunakan 12 unit perangkat komputer, yaitu 10 unit *slave mode*, 1 unit *master* dan 1 unit *client*. Uji coba *rendering* dilakukan 5 kali penambahan *frame render*, yaitu 12, 24, 38, 60, 120 *frame render*. Pada penelitian ini pengukuran komparasi menggunakan teknik uji komparasi statistik *Anova one way*. Hasil dari uji komparasi statistik *Anova one way* adalah $18.11 > 2.53$ untuk probabilitas 5% dan $18.11 > 3.77$ untuk probabilitas 1%. $F_{hitung} > F_{tabel}$ maka H_a diterima dan H_0 ditolak yaitu Terdapat Perbedaan Banyaknya Sumberdaya Komputer (*Slave/Node*) Yang Digunakan Sebelum Penambahan *Frame Render* Dari 12, 24, 36, 60, 120 *Frame* Dengan Memanfaatkan *Parallel Computing render Engine* Pada Platform *Open-Source Software*. Terdapat 2 korelasi yang Hipotesis Alternatif diterima, yaitu korelasi $X_3:X_4$ dengan harga T_{hitung} (2.98) lebih kecil dari harga T_{tabel} (2.979 < 2.101). Dengan demikian H_a diterima, dan H_0 ditolak yaitu Terdapat Perbedaan Penggunaan *Slave/Node Render* Sebelum Menambah 36 *Frame* Menjadi 60 *Frame*, dan korelasi $X_4:X_5$ dengan harga T_{hitung} (4.81) lebih kecil dari harga T_{tabel} (4.81 < 2.101). Dengan demikian H_a diterima, dan H_0 ditolak yaitu Terdapat Perbedaan Penggunaan *Slave/Node Render* Sebelum Menambah 60 *Frame* Menjadi 120 *Frame*.

Keywords: GPS, Aplikasi AR Perangkat Bergerak, AR Browser, Spherical Law of Cosines

1 PENDAHULUAN

1.1 Latar Belakang

Proses *rendering* dengan model 3D yang kompleks dan *photo-realistic*, membutuhkan waktu yang lebih lama dan membutuhkan sumber daya komputasi yang memadai, hal ini mengasumsikan bahwa waktu dan tenaga berbanding terbalik satu sama lain. artinya jika daya komputasi yang kurang akan membuat waktu *render* akan menjadi lebih lama, sebaliknya jika daya komputasi besar akan membuat waktu *render* menjadi lebih singkat (Patoli, Gkion and Al-Barakati 2009). *rendering* merupakan tahap akhir dalam pembuatan animasi 3D, dalam proses *rendering* data-data yang berhubungan dengan model, animasi, material, texture, lighting dan parameter tertentu akan dikonversi menjadi file image atau video. Tahap *rendering* ini menentukan baik-buruknya kualitas *model 3D* dihasilkan. Tahap *rendering* merupakan proses pengubahan model geometri menjadi sebuah image atau video. Biasanya pada proses

rendering ini membutuhkan waktu yang lama dan membutuhkan spesifikasi *hardware* yang besar. Dalam proses *rendering* perangkat *hardware* yang paling penting adalah CPU (Central Processing Unit), VGA (Video Graphic Adaptor) dan RAM (Random Access Memory), ke tiga perangkat *hardware* ini yang digunakan sebagai pemroses dan penyimpanan dalam proses *rendering*.

Selain membutuhkan perangkat *hardware* yang besar, pemilihan aplikasi *render engine* juga sangat penting. Pemilihan aplikasi *render engine* dilakukan pada *software* yang digunakan, dan mengatur parameter render dapat mempengaruhi hasil akhir dari *model 3D*. Selain itu dengan memanfaatkan beberapa sumberdaya komputer yang ada dapat meminimalkan waktu *rendering*. Hal ini nantinya akan mengurangi waktu render dengan membagi beban kerja pada tiap-tiap sumberdaya komputer.

Memanfaatkan internet untuk keperluan *rendering* grafis terdistribusi menjadi

kebutuhan bagi produsen gambar dan animasi, dengan model seperti ini, dapat mengoptimalkan *rendering* dengan memerlukan waktu *rendering* yang sedikit dan tidak mempengaruhi kualitas hasil *render* yang diperoleh (Zhurong, Wei and Yuhui 2009). Dengan memanfaatkan banyak sumberdaya komputer dapat meminimalkan waktu *render* dan tidak mempengaruhi kualitas hasil akhir *rendering*, dari banyak sumberdaya tersebut beban *render* akan dibagi dan didistribusikan ke dalam tiap-tiap sumberdaya yang ada. Hal ini dapat diasumsikan semakin banyak sumberdaya yang dimanfaatkan akan meminimalkan waktu *rendering*, dan dari hasil *rendering* akan di *compositng* kedalam komputer server tanpa mengurangi kualitas dari hasil *render*.

Dalam konsep jaringan distribusi data local *Local Area Network (LAN)*, semua komputer dapat menggunakan *network* sepanjang media *network* tidak digunakan komputer lain. Mungkin saja pada saat yang bersamaan ada komputer lain yang juga mengirimkan data, akibatnya dapat terjadi *data collision*. Untuk mengatasi permasalahan ini *Ethernet* telah dilengkapi dengan *collision detection*, jika terjadi *collision* komputer pengirim akan diberikesempatan untuk mengirim ulang data (Sofian 2010). Dengan menggunakan Jaringan Local *Ethernet* sumberdaya komputer dapat digunakan secara bersama-sama dalam satu kegiatan. Beberapa perangkat jaringan yang dapat dimanfaatkan pada local jaringan adalah *hub* dan *switch* yang mempunyai topology jaringan sendiri-sendiri.

Dari pemaparan di atas selain penentuan spesifikasi *hardware* dan aplikasi *render engine* merupakan hal yang penting dalam proses *rendering*, sedangkan dengan memanfaatkan beberapa sumberdaya komputasi dapat meminimalkan waktu *rendering* tanpa mengurangi kualitas hasil *render*. Oleh karena itu dengan memanfaatkan *parallel computing* dengan menggunakan banyak sumberdaya komputer dapat meminimalkan waktu *render* tanpa mengurangi kualitas *render*.

Dalam penelitian ini, akan memanfaatkan *parallel computing render engine* untuk *render 3d model* pada *open-space software* dengan membandingkan banyaknya *frame render* apakah akan mempengaruhi banyaknya sumberdaya komputer (*slave/node*) yang digunakan. Dalam penelitian ini, teknik statistik yang digunakan untuk menguji hipotesis

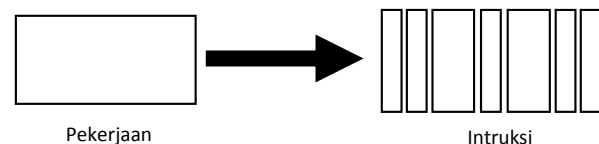
menggunakan teknik *Analisis of Variants (Anova) one way*.

Analisis of Variants (Anova) merupakan teknik uji statistik yang digunakan untuk membandingkan dua atau lebih variable penelitian (Verma and Patel 2017). Menurut G. Keller (2014) *Anova* umumnya dapat digunakan sebagai pengujian signifikansi dari pengamatan experimental atau perbedaannya (Bokoro and Malandala 2017). Dengan menggunakan teknik statistik hipotesis *Anova*, banyaknya *frame render* akan dibandingkan satu sama lain apakah akan mempengaruhi banyaknya sumberdaya komputer (*slave/node*) yang digunakan.

1.2 Landasan Teori

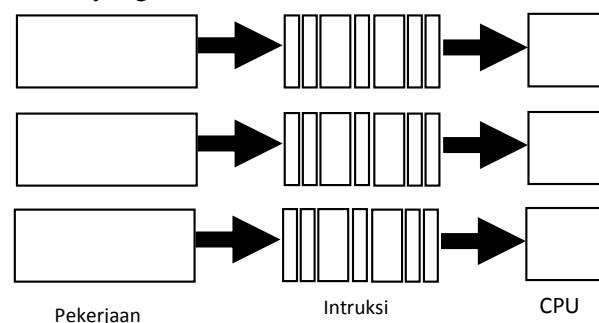
1.2.1 Parallel Computing

Parallel computing adalah penggunaan lebih dari satu sumber daya komputasi secara simultan untuk memecahkan persoalan komputasi (Rumagit 2013). Proses komputasi umumnya dikerjakan secara serial yang dikerjakan oleh prosesor tunggal, artinya intruksi yang diberikan kepada prosesor berurutan dan dalam saat yang sama intruksi tidak boleh dikerjakan sebelum intruksi yang masih dikerjakan selesai.



Gambar 1 Serial Computing

Dengan memanfaatkan *parallel Computing* proses komputasi akan dipecah menjadi beberapa bagian yang dapat dikerjakan secara bersama-sama dalam satu waktu (Rumagit 2013). Artinya proses komputasi tidak dikerjakan oleh satu prosesor saja, namun proses komputasi akan dibagi menjadi beberapa bagian komputasi yang akan dikerjakan oleh beberapa prosesor secara bersama-sama dalam waktu yang sama.



Gambar 1 Parallel Computing

Alasan utama penggunaan *parallel computing* adalah (Rumagit 2013):

1. Dapat menghemat waktu
2. Dapat memecah persoalan yang lebih besar
3. Dapat menghemat biaya
4. Dapat mengatasi keterbatasan fisik dari komputasi serial

Alasan lain yang cukup penting adalah *fault tolerant*, jika salah satu prosesor mengalami kegagalan prosesor lain dapat menggantikannya, meskipun dengan perform yang menurun.

1.2.2 *Rendering*

Rendering adalah suatu proses untuk mengubah model geometri menjadi suatu gambar. Proses untuk membangun sebuah gambar membutuhkan beberapa fase seperti *modelling*, pengaturan *material* dan *texture*, penempatan virtual *light*, dan proses render (Lysy C, Arthur and Brave 2013). Dalam proses *rendering* ini penggunaan *hardware* yang mencukupi akan berpengaruh pada waktu *rendering* model 3D

Rendering memainkan peran yang sangat vital pada proses penciptaan animasi dan gambar. Rendering dapat digunakan untuk meniru objek visual yang nyata (*photorealistic rendering*), atau *stylistic fashion (non-photorealistic renderings)* dengan baik. Dengan mengesampingkan *style*, *rendering* dapat diselesaikan dengan menggunakan algoritma yang memang dikhususkan untuk proses render. Algoritma ini bisa menjadi sangat kompleks. Semakin kompleks algoritma yang digunakan maka semakin lama waktu yang dibutuhkan untuk melakukan proses kalkulasi. Dengan menggunakan *hardware* yang moderen proses render dari film sederhana dapat menghabiskan waktu antara beberapa menit sampai dengan dua ratus menit untuk setiap frame-nya (Rumagit 2013). Dapat disimpulkan bahwa waktu yang dibutuhkan dalam proses render animasi dapat dilihat dari banyak frame yang dimiliki dalam sebuah adegan. Dapat gambarkan jika dalam satu frame perkiraan waktu render adalah dua menit maka untuk mengetahui waktu keseluruhan render, dapat dikalikan dengan banyak frame dalam satu adegan animasi.

Pada umumnya proses render yang dilakukan pada saat ini masih bekerja secara single dengan menggunakan sebuah mesin yang

memiliki sumber daya yang besar. Kerugian dari proses render yang bekerja secara single adalah Waktu yang diperlukan untuk melakukan proses ini masih dirasakan cukup lama walaupun mesin yang digunakan memiliki sumber daya yang cukup besar (Lysy C, Arthur and Brave 2013). Hal ini dikarenakan satu sumber daya komputer menangani semua proses render, hal ini mengakibatkan proses render membutuhkan waktu yang lama. Jika proses render dikerjakan menggunakan banyak sumber daya computer, proses render akan dibagi menjadi beberapa bagian dan akan didistribusikan kesemua sumber daya yang ada. Dengan cara seperti itu proses render akan menjadi lebih singkat dan lebih cepat, karena semua sumber daya computer bekerja bersama dalam satu waktu.

1.2.3 *Blender Network Render*

Blender adalah aplikasi *Opens Source pembentuk* grafis 3D, aplikasi ini mendukung 3D *Pipeline-modeling, rigging, animation, simulation, rendering, compositing, motion tracking, even video editing* dan pembuatan *game*. Pengguna tingkat lanjut dapat menggunakan *API Blender* dengan menggunakan *script python* (Blender, About n.d.). Dalam aplikasi *blender* karena bersifat *open source* dapat meminimalkan beban biaya dan kemudian dengan aplikasi *blender* fitur-fitur yang dimiliki lebih variatif.

Network Render tujuannya bukan untuk menggantikan *render farm*, melainkan dirancang untuk secara interaktif merender sebuah gambar melalui jaringan komputer dengan umpan balik langsung di *Blender*. Dalam *network render* memerlukan desain yang berbeda dengan pembuatan *render farm*, meskipun dalam kasus sederhana dapat menjadi pengganti (Blender, Network Render n.d.). Cara *network render* hampir sama dengan *render farm*, namun dengan menggunakan *network render* pekerjaan render secara paralel dapat dilakukan dalam aplikasi *blender* sendiri.

1.2.4 *Analisis of Variants (Anova) one way*

Analisis varians dapat digunakan untuk menguji hipotesis komparasi rata-rata k sample bila data yang diolah berbentuk interval atau rasio. Satu sampel dalam k kejadian atau pengukuran berarti sampel tersebut berpasangan, model *befor-after*. Satu sampel diberi perlakuan sampai 5 kali, hal ini artinya 5

sampel tersebut berpasangan. Sedangkan k sampel dalam satu kejadian berarti sampel *independen*. Terdapat dua jenis *Anova*, *Anova single classification* dan *Anova multiple classification* (Sugiyono 2017).

Analisis single classification yang sering disebut *Anova one way* dapat digunakan untuk menguji hipotesis komparasi rata k sampel, bila pada setiap sampel hanya terdiri atas satu kategori. Dalam teknik ini setiap sampel akan mempunyai Mean (rata-rata) dan Varians (simpangan baku kuadrat), dimana n = jumlah sample, M = mean/rata-rata sedangkan s^2 = varians (Sugiyono 2017).

Dalam *Anova* terdapat (Sugiyono 2017):

1. Deviasi Total, yaitu jarak antara nilai individual yang ada dalam seluruh sampel dengan Mean Total. Dalam hal ini misal $(X - M_{tot})$.
2. Deviasi antara kelompok (*between*), yaitu jarak antara Mean setiap kelompok dengan Mean Total. Dalam hal ini misal $(M_4 - M_{tot})$.
3. Deviasi dalam kelompok (*within*), yaitu jarak nilai seluruh individu dalam satu kelompok dengan Mean kelompok itu. Dalam hal ini misal $(X - M_4)$.

Kemudian jarak suatu nilai dalam kelompok terhadap rata-rata $(X_i - M)$ dikuadratkan menjadi $(X_i - M)^2$. Kuadrat ini selanjutnya disingkat dengan JK dan merupakan varian dari kelompok tersebut. Karena dalam pengujian hipotesis melibatkan lebih dari dua kelompok sampel, maka akan terdapat beberapa macam JK (Sugiyono 2017), yaitu:

1. Jumlah Kuadrat Total (JK_{tot}) merupakan perjumlahan kuadrat deviasi nilai individual dengan M_{tot}

$$JK_{tot} = \sum X_{tot}^2 - \frac{(\sum X_{tot})^2}{N}$$

Rumus 1 Jumlah Kuadrat Total

N = jumlah seluruh anggota sampel

2. Jumlah Kuadrat Antara (JK_{ant}) merupakan jumlah selisih kuadrat Mean Total (M_{tot}) dengan Mean setiap kelompok (M_1), dikalikan dengan jumlah sampel setiap kelompok.

$$JK_{ant} = \sum \frac{(\sum X_k)^2}{n_k} - \frac{(\sum X_{tot})^2}{N}$$

Rumus 2 Jumlah Kuadrat Antara

3. Jumlah Kuadrat dalam Kelompok (JK_{dal})

$$JK_{dal} = JK_{tot} - JK_{ant}$$

Rumus 3 Jumlah Kuadrat dalam Kelompok

Setiap sumber variasi didampingi dengan dk , dan dk untuk setiap sumber variasi tidak sama.

Untuk Antar Kelompok $dk = m - 1$

Untuk Dalam Kelompok $dk = N - m$

Total $dk = N - 1$

M = Jumlah kelompok sampel

N = Jumlah seluruh anggota sampel

Untuk dapat menghitung harga F hitung, maka beberapa sumber variasi harus dihitung mean kelompoknya, yang meliputi Mean Antara Kelompok dan Mean Dalam Kelompok.

Untuk Antar Kelompok:

$$MK_{ant} = \frac{JK_{ant}}{m - 1}$$

Rumus 4 Mean Antar Kelompok

Untuk Dalam Kelompok :

$$MK_{dal} = \frac{JK_{dal}}{N - m}$$

Rumus 5 Mean dalam Kelompok

$$F \text{ hitung} = \frac{MK_{ant}}{MK_{dal}}$$

Rumus 6 F Hitung

Jadi untuk pengujian hipotesis dalam *Anova one way* diperlukan langkah-langkah sebagai berikut (Sugiyono 2017):

1. Menghitung Jumlah Kuadrat Total (JK_{tot}).
2. Menghitung Jumlah Kuadrat Antar Kelompok (JK_{ant}).
3. Menghitung Jumlah Kuadrat dalam Kelompok (JK_{dal}).
4. Menghitung Mean Kuadrat Antar Kelompok (MK_{ant}).
5. Menghitung Mean Kuadrat Dalam Kelompok (MK_{dal}).
6. Menghitung F Hitung (F_{hit}).
7. Menbandingkan harga F hitung dengan F table dengan dk pembilang $(m - 1)$ dan dk penyebut $(N - 1)$. Harga F hasil perhitungan tersebut selanjutnya disebut F hitung (F_{hit}), yang berdistribusi F dengan dk pembilang $(m - 1)$ dan dk penyebut $(N - 1)$ tertentu. Ketentuan pengujian sipotesis: *Bila F hitung lebih kecil atau sama dengan harga F table ($F_h \leq F_t$) maka H_0 diterima dan H_a ditolak,*

sebaliknya bila $F_h > F_t$, maka H_a diterima dan H_o ditolak.

- Membuat kesimpulan pengujian hipotesis:
Ho diterima Ho ditolak.

Setelah mengetahui apakah hipotesis H_a diterima atau tidak, selanjutnya untuk membuktikan apakah perbedaan dua sampel yang diterima atau tidak dengan menggunakan t-test.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2} - 2r \left(\frac{S_1}{\sqrt{n_1}} \right) \left(\frac{S_2}{\sqrt{n_2}} \right)}}$$

Rumus 7 t-test related/berpasangan

r = Korelasi antara dua sampel

\bar{X} = Rata-rata sampel

2 METODE PENELITIAN

2.1 Metode Penelitian

- Membuat Model 3D dengan penambahan pengaturan modifier, lighting, camera, material dan texture, yang nantinya akan digunakan sebagai bahan penelitian.
- Analisis dan perancangan konfigurasi topologi jaringan komputer dengan menggunakan *switch*, dalam hal ini terdapat 12 perangkat komputer dan 1 perangkat *switch*.
- Mengatur konfigurasi perangkat komputer dalam *network render*, dalam hal ini terdapat 10 perangkat komputer yang berfungsi sebagai *slave mode*, 1 perangkat komputer yang berfungsi sebagai *master mode* dan 1 perangkat komputer yang berfungsi sebagai *client mode*.
- Implementasi *rendering 3D Model* dengan menggunakan *blender network render*, dalam hal ini experiment dilakukan dengan melakukan uji coba dengan *render, 12 frame render, 24 frame render, 36 frame render, 60 frame render* dan *120 frame render*.
- Evaluasi dengan melakukan komparasi banyaknya *frame render* apakah akan mempengaruhi banyaknya sumberdaya komputer (*slave/node*) yang digunakan. Dalam hal ini dalam tahap uji komparasi akan menggunakan teknik statistik *Anova one way* dengan menentukan hipotesis H_a dan H_o terlebih dahulu, kemudian akan mengukur tingkat korelasi antara antar sampel dengan sampel yang lain untuk mengetahui

perubahan penggunaan sumberdaya (*slave/node*) yang signifikan.

2.2 Pengumpulan Data

1. Model 3D

Dalam pembuatan model 3d terdapat 6 model 3d yaitu Monitor LCD, Meja, Keyboard, Mouse, Lampu Meja dan Mouse Path, detail dari properties model 3d sebagai berikut.

Table 1 *Properties Model 3D*

No	Model	Modifier	Material
1	Monitor LCD : Transform (loc_x : 0.0; loc_y : -1.78; loc_z : 0.0)	Edge Split (Split Angle:30°) Curve (object curve:Bezier)	Body (Glossy BSDF:GGX; color: 000000; Roughness:0.058) Layar (texture coordinate, image texture, emission)
2	Meja: Transform (loc_x : 0.0; loc_y : 0.0; loc_z : 1.78)	-	Meja (texture coordinat, image texture, bump, glossy bsdf)
3	Keyboard: Transform (loc_x : 0.0; loc_y : -2.1; loc_z : 0.1)	Subdivision surface (view : 2; render : 2)	Body (Glossy BSDF:GGX; color: 000000; Roughness:0.058) Key (texture coordinate, image texture, emission)
4	Mouse: Transform (loc_x : 3.1; loc_y : -2.2; loc_z : 0.09)	Subdivision surface (view : 2; render : 3) Mirror (exis : x; option: clipping)	Body (Glossy BSDF:GGX; color:
5	Lampu Meja: Transform (loc_x : - 3.4; loc_y : -0.9; loc_z : 0.04)	Subdivision surface (view : 2; render : 2)	Body (Glossy BSDF:GGX; color:
6	Path: Transform (loc_x : -3.1; loc_y : -2.2; loc_z : 0.00)	ParticleSyste m (Type : Hair; render : 100)	Path (image texture, Hair BSDF)

Selanjutnya setelah model 3d telah selesai, kemudian model 3d akan ditambahkan material dan texture.

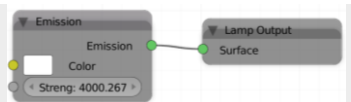
2. Material Node Editor

Pada kasus ini, karena *engine render* menggunakan *network render* dengan *engine cyle*, material dan texture memanfaatkan *node editor* untuk mengolahnya. Dengan memanfaatkan *node editor* penggunaan *node material* dapat dilakukan dengan mudah.

3. Lighting

Kemudian setelah model 3d telah terbentuk, selanjutnya adalah mengatur pencahayaan pada lingkungan render. Ada 3 type lighting yang dimanfaatkan yaitu: *Spot Light, Point Light* dan *Emission Light*.

Table 2 Properties Lighting

No	Type	Properties
1	Spot Light	Size: 2.837; Max Bounces: 1024; color: FFFFFFFF; Spot Shape (Size: 84°; blend: 0.033); Cast Shadow
2	Point Light	Size: 10.000; Max Bounces: 1024; color: FFFFFFFF; Cast Shadow
3	Emission Light	

Kemudian setelah mengatur posisi objek dan lighting, dan menentukan intensitas cahaya yang akan dihasilkan oleh tiap-tiap light, mengatur tampilan shadow atau bayangan model 3d yang dihasilkan oleh lighting. Selanjutnya adalah mengatur posisi kamera, yang nantinya dari tampilan kamera ini hasil render akan ditampilkan.

4. Camera

Dengan mengatur posisi, resolusi, preview render sebelum dan sesudah dirender sebagai berikut.

2.3 Analisis dan Perencanaan

2.3.1 Analisis Kebutuhan Hardware

Pemanfaatan *Parallel Computing Render Engine* pada *platform Open-Source Software*, membutuhkan banyak sumberdaya komputer yang nantinya akan digunakan sebagai mesin *rendering* yang berkerja secara bersama-sama atau *parallel*. Dalam penelitian ini terdapat 12 perangkat komputer dan 1 perangkat *switch*, detail analisis kebutuhan system sebagai berikut.

Table 3 Analisis Kebutuhan Hardware Perangkat Komputer

No	Processor	Ram Gb	Cardlan
1	Intel(R) Core(TM) i3 @3.20 GHz	4	Realtek PCIe GBE Family Controller
2	Intel(R) Core(TM) i3 @3.30 GHz	4	Realtek PCIe GBE Family Controller
3	Intel(R) Core(TM) i3 @3.30 GHz	4	Realtek PCIe GBE Family Controller
4	Intel(R) Core(TM) i3 @3.30 GHz	4	Realtek PCIe GBE Family Controller
5	Intel(R) Core(TM) i3 @3.30 GHz	4	Qualcomm Atheros AR8151 PCI-E Gigabit Ethernet Controller
6	Intel(R) Core(TM) i3 @3.30 GHz	4	Realtek PCIe GBE Family Controller

No	Processor	Ram Gb	Cardlan
7	Intel(R) Core(TM) i3 @3.30 GHz	4	Realtek PCIe GBE Family Controller
8	Intel(R) Core(TM) i3 @3.30 GHz	4	Realtek PCIe GBE Family Controller
9	Intel(R) Core(TM) i3 @3.30 GHz	4	Realtek PCIe GBE Family Controller
10	Intel(R) Core(TM) i3 @3.30 GHz	4	Qualcomm Atheros AR8151 PCI-E Gigabit Ethernet Controller
11	Intel(R) Core(TM) i3 @3.30 GHz	4	Realtek PCIe GBE Family Controller
12	Intel(R) Core(TM) i3 @3.30 GHz	4	Qualcomm Atheros AR8151 PCI-E Gigabit Ethernet Controller

Sedangka kebutuhan perangkat *switch*, sebagai berikut

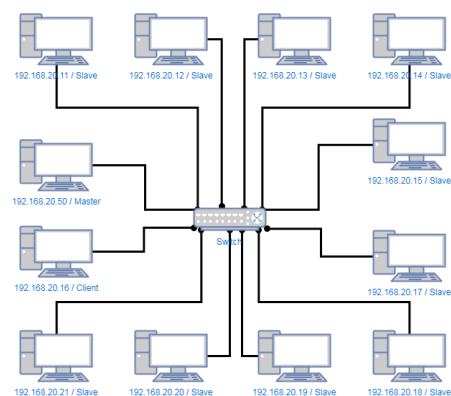
2.3.2 Perancangan Konfigurasi Jaringan

Setelah mempersiapkan kebutuhan *hardware*, kemudian 12 perangkat komputer dan 1 *switch* akan dikonfigurasi menjadi satu jaringan *Local Area Network (LAN)* dengan menggunakan topologi jaringan *Bus*.

Table 4 Konfigurasi Software

No	IP Address	Mode Operation	Name	OS
1	192.168.20.50	MASTER	STMIK-III-DOSEN	Win 7 32bit
2	192.168.20.11	SLAVE	STMIK-III-11	Win 7 32bit
3	192.168.20.12	SLAVE	STMIK-III-12	Win 7 32bit
4	192.168.20.14	SLAVE	STMIK-III-14	Win 7 32bit
5	192.168.20.15	SLAVE	STMIK-III-15	Win 7 32bit
6	192.168.20.16	CLIENT	STMIK-III-16	Win 7 32bit
7	192.168.20.17	SLAVE	STMIK-III-17	Win 7 32bit
8	192.168.20.18	SLAVE	STMIK-III-18	Win 7 32bit
9	192.168.20.19	SLAVE	STMIK-III-19	Win 7 32bit
10	192.168.20.20	SLAVE	STMIK-III-20	Win 7 32bit
11	192.168.20.21	SLAVE	STMIK-III-21	Win 7 32bit
12	192.168.20.22	SLAVE	STMIK-III-22	Win 7 32bit

Sedangkan skema konfigurasi topologi jaringan LAN sebagai berikut.



Gambar 2 Slema Topologi jaringan

Setelah mengatur dan melakukan konfigurasi LAN, selanjutnya akan dilakukan konfigurasi pada *blender network render*. Dalam hal ini setiap sumberdaya komputer akan diidentifikasi sesuai fungsinya masing.

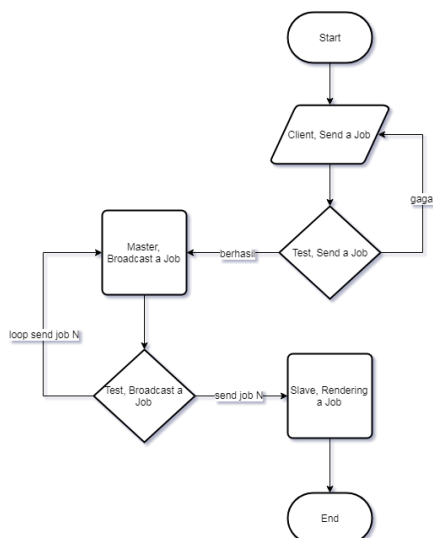
2.3.3 Konfigurasi Network Render

Seperti yang telah dijelaskan dalam pembahasan sebelumnya bahwa terdapat 1 komputer dengan mode *Master* yang berfungsi sebagai pengontrol dan mem-*broadcast* pekerjaan kesemua bagian yang lain, 1 komputer dengan mode *Client* yang berfungsi sebagai *client* yang mengirimkan pekerjaan ke komputer *master* selanjutnya mem-*broadcast* ke komputer *slave* dan 10 komputer dengan mode *Slave* yang berfungsi sebagai semua komputer yang menerima pekerjaan dari komputer *master*. Konfigurasi pada *properties render* tiap mode sebagai berikut.

Table 5 Konfigurasi blender Network Render

No	Mode	Properties
1	Master	Network setting (type: Master mode; address: default; port:8000; Start Service: active) Master setting (broadcast)
2	Client	Network setting (type: Client mode; address: 192.168.20.50; port:8000) Job setting (type: blender; name: default; engine: cycles; send job)
3	Slave	Network setting (type: Slave mode; address: 192.168.20.50; port:8000; Start Service: active) Slave setting (tag: non; render on slave: on; bake on slave: on; clear on exit: on; output rener on console: on; threads: auto-detect)

Gambaran skema alur kerja adam *blender network render* sebagai berikut.



Gambar 3 Slema Alur blender network render

Sedangkan Setelah melakukan konfigurasi pada *blender network render*, selanjutnya adalah tahap implementasi *rendering 3D Model*. Pada tahap berikutnya ini *Model 3D* yang telah dibuat akan di-*render* dengan *frame render* yang berbeda-beda.

2.4 Implementasi Rendering 3D Model

1. Rendering 12 frame render

Pada tahap *rendering 12 frame render* ditentukan terlebih dahulu *start* dan *end frame*, dalam hal ini *start frame: frame 1* dan *frame end: frame 12*. Pada saat proses *12 frame render* sumberdaya komputer (*slave/node*) yang dimanfaatkan hanya 3 yaitu komputer dengan *ip address 192.168.20.15, 192.168.20.19* dan *192.168.20.17* sedangkan 7 komputer yang lain tidak menerima *job*. Dari proses *rendering* tersebut dapat di lihat bahwa *start time: 15:35:43 – finished time: 15:36:32 = line time: 49second*.

2. Rendering 24 frame render

Pada tahap *rendering 24 frame render* ditentukan terlebih dahulu *start* dan *end frame*, dalam hal ini *start frame: frame 1* dan *frame end: frame 24*. Pada saat proses *24 frame render* sumberdaya komputer (*slave/node*) yang dimanfaatkan hanya 5 yaitu komputer dengan *ip address 192.168.20.15, 192.168.20.19, 192.168.20.21, 192.168.20.14* dan *192.168.20.17* sedangkan 5 komputer yang lain tidak menerima *job*. Dari proses *rendering* tersebut dapat di lihat bahwa *start time: 15:38:31 – finished time: 15:39:25 = line time: 54second*.

3. Rendering 36 frame render

Pada tahap *rendering 36 frame render* ditentukan terlebih dahulu *start* dan *end frame*, dalam hal ini *start frame: frame 1* dan *frame end: frame 36*. Pada saat proses *36 frame render* sumberdaya komputer (*slave/node*) yang dimanfaatkan hanya 8 yaitu komputer dengan *ip address 192.168.20.18, 192.168.20.15, 192.168.20.19, 192.168.20.21, 192.168.20.22, 192.168.20.14, 192.168.20.12* dan *192.168.20.17* sedangkan 2 komputer yang lain tidak menerima *job*. Dari proses *rendering* tersebut dapat di lihat bahwa *start time: 15:41:11 – finished time: 15:42:09 = line time: 58second*.

4. Rendering 60 frame render

Pada tahap *rendering 60 frame render* ditentukan terlebih dahulu *start* dan *end frame*, dalam hal ini *start frame: frame 1* dan *frame end: frame 60*. Pada saat proses *60 frame render*

sumberdaya komputer (*slave/node*) yang dimanfaatkan semua sumberdaya untuk menerima *job* yaitu komputer dengan *ip address* 192.168.20.18, 192.168.20.15, 192.168.20.19, 192.168.20.21, 192.168.20.22, 192.168.20.14, 192.168.20.12, 192.168.20.11, 192.168.20.20 dan 192.168.20. Dari proses *rendering* tersebut dapat di lihat bahwa *start time:15:45:19 – finished time: 15:46:56 = line time:37second*.

5. *Rendering 120 frame render*

Pada tahap *rendering 120 frame render* ditentukan terlebih dahulu *start* dan *end frame*, dalam hal ini *start frame: frame 1* dan *frame end: frame 120*. Pada saat proses *120 frame render* sumberdaya komputer (*slave/node*) yang dimanfaatkan semua sumberdaya untuk menerima *job* yaitu komputer dengan *ip address* 192.168.20.18, 192.168.20.15, 192.168.20.19, 192.168.20.21, 192.168.20.22, 192.168.20.14, 192.168.20.12, 192.168.20.11, 192.168.20.20 dan 192.168.20. Dari proses *rendering* tersebut dapat di lihat bahwa *start time:15:48:01 – finished time: 15:50:29 = line time:28second*.

Dari hasil 7 tahap implementasi *rendering 3d model*, dapat dilihat bahwa penggunaan sumberdaya komputer (*slave/node*) semakin banyak *frame render* yang digunakan semaik banyak pula sumberdaya yang digunakan. Detail penggunaan sumberdaya tiap *frame render* sebagai berikut.

Table 6 *Job Slave Render tiap frame render*

IP Address	Jumlah Frame									
	12		24		36		60		120	
	J	T	J	T	J	T	J	T	J	T
192.168.20.11	0		0		0		5	8.4	15	
192.168.20.12	0		0		1	9.3	5	8.3	10	
192.168.20.14	0		4	8.6	5	8.5	10	8.7	5	10
192.168.20.15	2	9.1	5	8.4	5	8.5	5	8.5	15	
192.168.20.17	5	8.4	5	8.4	5	8.3	5	8.4	15	
192.168.20.18	0		0		5	8.6	5	8.6	15	
192.168.20.19	5	8.4	5	8.4	5	8.4	5	8.3	10	
192.168.20.20	0		0		0		5	8.5	10	
192.168.20.21	0		5	8.5	5	8.4	10	8.4	5	10
192.168.20.22	0		0		5	8.5	5	8.6	10	
Jumlah	12	8.6	24	8.5	36	8.6	60	8.5	120	8.7

Dari data *job slave render* tiap *frame render* ini adakan diuji perbandingan tiap *frame* menggunakan teknik pengujian perbandingan *Anova one way*.

3 EVALUASI DAN HASIL

3.1 Perbandingan dengan *Anova One Way*

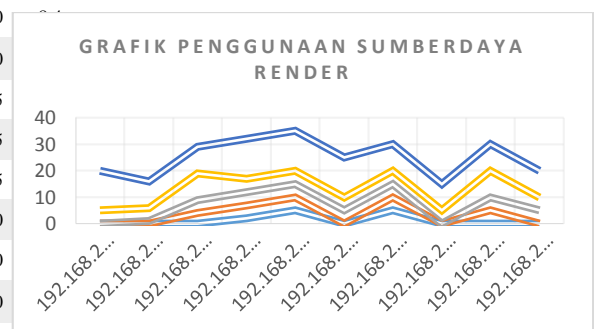
Penentuan hipotesis awal:

Ho : Tidak terdapat perbedaan penggunaan SLAVE/NODE RENDER dengan adanya penambahan frame (penambahan frame tidak berpengaruh terhadap penggunaan SLAVE/NODE RENDER).

Ha : Terdapat perbedaan penggunaan SLAVE/NODE RENDER dengan adanya penambahan frame (penambahan frame dapat mengurangi penggunaan SLAVE/NODE RENDER)

Table 7 *Job Slave Render tiap frame render (X₁,X₂,X₃,X₄,X₅)*

No	Frame Render					ΣX _{tot}	ΣX _{tot} ²
	12 X ₁	24 X ₂	36 X ₃	60 X ₄	120 X ₅		
192.168.20.11	0	0	0	5	15	20	275
192.168.20.12	0	0	1	5	10	16	152
192.168.20.14	0	4	5	10	10	29	266
192.168.20.15	2	5	5	5	15	32	329
192.168.20.17	5	5	5	5	15	35	350
192.168.20.18	0	0	5	5	15	25	300
192.168.20.19	5	5	5	5	10	30	225
192.168.20.20	0	0	0	5	10	15	150
192.168.20.21	0	5	5	10	10	30	275
192.168.20.22	0	0	5	5	10	20	159
Jml	12	24	36	60	120	252	2456
\bar{X}	1.2	2.4	3.6	6.0	12.0	n	10
$\sum X^2$	0	0	0	0	0	x	5
s	2.1	2.5	2.2	2.1	2.5	N	10x5 =50
s ²	4.4	6.4	5.1	4.4	6.6		
	0	9	6	4	7		



Gambar 4 *Grafik Penggunaan Sumberdaya Render*

Diketahui:

- \bar{X} : Rata-rata x
- $\sum X^2$: Mean X kuadrat
- s : Simpangan baku
- s² : Simpangan baku kuadrat

sebelum analisis varian dilakukan perlu homogenitas varians terlebih dahulu dengan menggunakan uji F.

$$F \text{ hitung} = \frac{\text{Varianterbesar}}{\text{Varianterkecil}}$$

Dapat dilihat bahwa varian terbesar = 6.67 dan varian terkecil = 4.40 sehingga F hitung yang diperoleh:

$$F \text{ hitung} = \frac{6.67}{4.40} = 1.52$$

Selanjutnya F hitung akan dibandingkan dengan F table dengan df (degr freedom) pembilang $n_7 - 1$ dan df penyebut $n_1 - 1$, dalam hal ini nilai n_7 dan n_1 bernilai sama yaitu 10 dan setelah dikurang 1 nilainya 9. Berdasarkan Tabel F, maka F tabel untuk 5% = 3.18 sedangkan F table untuk 1% = 5.35. Harga F hitung lebih kecil dari F table (**1.52 < 3.18**). Dengan demikian dapat disimpulkan bahwa varians data yang akan dianalisis homogen, sehingga perhitungan *anova* dapat dilanjutkan.

1. Jumlah Kuadrat Total (JK_{tot})

$$JK_{tot} = \sum X_{tot}^2 - \frac{(\sum X_{tot})^2}{N}$$

$$= 2456 - \frac{252^2}{50}$$

$$JK_{tot} = 1185.9$$

2. Jumlah Kuadrat Antara (JK_{ant})

$$JK_{ant} = \sum \frac{(\sum X_k)^2}{n_k} - \frac{(\sum X_{tot})^2}{N}$$

$$JK_{ant} = \frac{(\sum X_1)^2}{n_1} + \frac{(\sum X_2)^2}{n_2} + \dots + \frac{(\sum X_m)^2}{n_m} - \frac{(\sum X_{tot})^2}{N}$$

$$= \frac{12^2}{10} + \frac{24^2}{10} + \frac{36^2}{10} + \frac{60^2}{10} + \frac{120^2}{10} - \frac{252^2}{50}$$

$$= 14.4 + 57.6 + 129.6 + 360 + 1440 - 1270.1$$

$$= 731.52$$

3. Jumlah Kuadrat dalam Kelompok (JK_{dal})

$$JK_{dal} = JK_{tot} - JK_{ant}$$

$$JK_{dal} = 1185.9 - 731.52$$

$$JK_{dal} = 454.4$$

4. Menghitung Mean Kuadrat Antar Kelompok (MK_{ant}).

$$MK_{ant} = \frac{JK_{ant}}{m - 1}$$

$$= \frac{731.5}{5 - 1}$$

$$= 182.88$$

5. Menghitung Mean Kuadrat Dalam Kelompok (MK_{dal}).

$$MK_{dal} = \frac{JK_{dal}}{N - m}$$

$$= \frac{454.4}{50 - 5}$$

$$= 10.10$$

6. Menghitung F Hitung (F_{hit}).

$$F \text{ hitung} = \frac{MK_{ant}}{MK_{dal}}$$

$$= \frac{182.88}{10.10}$$

$$= 18.11$$

Table 8 Rencana Target Capaian Perhitungan

Sumber Validasi	df	Jumlah Kuadrat	MK	Fh	Ft	Kep
Total	50-1 = 49	1185.9			5 %	
Antar Kelompok k	5-1=4	731.52	182.88	18.11	2.5 %	Fh>Ft Jadi Ha diterima
Dalam Kelompok k	50-5=45	454.40	10.10		3.7 %	

7. Menbandingkan harga F hitung dengan F table dengan df pembilang $\{m(5) - 1 = 4\}$ dan df penyebut $\{N(50) - m(5)=45\}$, hasilnya adalah 253. Jadi $18.11 > 2.53$ untuk probability 5% dan $18.11 > 3.77$ untuk probality 1% , Harga F hitung (F_{hit}) lebih besar dari F table (F_{tab}) ($F_h > F_t$) tertentu. Jadi bila F hitung lebih besar harga F table ($F_h > F_t$) maka Ha diterima dan Ho ditolak.
8. Jadi Terdapat Perbedaan Banyaknya Sumberdaya Komputer (Slave/Node) Yang Digunakan Sebelum Penambahan Frame Render Dari 12, 24, 36, 60, 120 Frame Dengan Memanfaatkan Parallel Computingrender Engine Pada Platform Open-Source Software.

3.2 Hasil Pengujian Hipotesis

Dapat disimpulkan pada tahap ini, bahwa dengan menggunakan F uji Hipotesis Alternatif dapat diterima. Bahwa Terdapat Perbedaan Banyaknya Sumberdaya Komputer (Slave/Node) Yang Digunakan Sebelum Penambahan Frame Render Dari 12, 24, 36, 60, 120 Frame Dengan Memanfaatkan Parallel Computingrender Engine Pada Platform Open-Source Software.

DAFTAR PUSTAKA

- Blender. n.d. *About*. Blender.org. Accessed Mei 9, 2017.
<https://www.blender.org/about/>.
- . n.d. *Network Render*. Blender. Accessed Mei 9, 2017.
https://wiki.blender.org/index.php/Dev:Source/Render/Cycles/Network_Render.
- Bokoro, P, and K Malandala. 2017. "Condition assessment of SF6 circuit breakers using analysis of variance (ANOVA)." *AFRICON, 2017 IEEE*. Cape Town, South Africa: IEEE. 1374-1377.
- Lysy C, Moleong,, M. Rumagit Arthur, and A. Sugiarto Brave. 2013. "Implementasi Cluster Computing Untuk Render Animasi." *E-journal Teknik Elektro dan Komputer 2* (3): 1-8.
- Patoli, Muhammad Zeeshan , Michael Gkion, and Abdullah Al-Barakati. 2009. "An open source Grid based render farm for Blender 3D." *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*. Seattle, WA, USA: IEEE.
- Rumagit, Arthur. 2013. "IMPLEMENTASI GRID COMPUTING UNTUK HIGH THROUGHPUT COMPUTING." *JURNAL TEKNIK INFORMATIKA UNIVERSITAS SAM RATULANGI 2* (1).
- Sofian, Iwan. 2010. *CISCO CCNA & Jaringan Komputer*. Bandung: Informatika.
- Sugiyono, Prof. Dr. 2017. *Statistik untuk Penelitian*. Bandung: Penerbit Alfabeta.
- Verma, Shanti, and Kalyani Patel. 2017. "Association between shopping habit and demographics of m-commerce user's in India using two way ANOVA." *Convergence in Technology (I2CT), 2017 2nd International Conference for*. Mumbai, India: IEEE. 38-43.
- Zhurong, Zhou, Deng Wei, and Qiu Yuhui. 2009. "A Grid Based Graphics Rendering Design." *Information Technology and Applications, 2009. IFITA '09. International Forum on*. Chengdu, China: IEEE.