

Analisis dan Optimasi Rendering pada Autodesk Maya dengan Menggunakan UE4

Analysis and Optimization of Rendering in Autodesk Maya Using UE4

Winarja*¹, M. Suyanto², Asro Nasiri³

^{1,2,3} Magister Teknik Informatika, Universitas Amikom Yogyakarta

E-mail: *¹winarja@amikom.ac.id, ²yanto@amikom.ac.id, ³asro@amikom.ac.id

Abstrak

Proses rendering pada pembuatan film 3D animasi merupakan pekerjaan yang sangat memakan banyak waktu dikarenakan banyaknya frame yang diperlukan untuk membuat film 3D animasi. Pada proses rendering satu frame dalam film 3D animasi biasanya membutuhkan waktu hingga beberapa jam dikarenakan proses kalkulasi render yang terdiri dari data model, data shader, data texture, dan data lighting dalam sebuah shot pada adegan 3D animasi. Dengan menggunakan proses render secara frame per frame maka proses render akan memerlukan waktu yang lama sehingga penulis akan melakukan penelitian pada proses rendering dengan menggunakan software game engine pada Unreal Engine 4 sebagai kalkulasi render secara realtime. Dari data yang semula dilakukan proses render pada software maya dan selanjutnya akan dilakukan perubahan data dari file maya diubah menjadi FBX file sebagai data yang akan di proses pada software Unreal Engine 4 (UE4). Dengan penggunaan game engine sebagai hasil akhir dalam menampilkan hasil render secara realtime sangatlah membantu dalam mendapatkan kelancaran pada sebuah proses produksi pada film 3D animasi.

Kata Kunci—animasi 3D, rendering, real time render, game engine.

Abstract

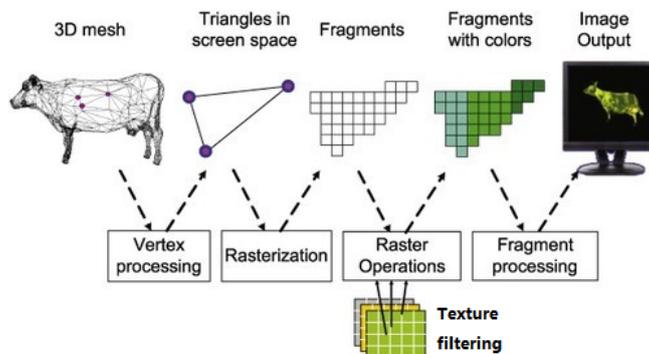
The rendering process in making animated 3D movies is a very time-consuming job due to the large number of frames needed to make animated 3D films. In the rendering process one frame in an animated 3D movie usually takes up to several hours because the rendering calculation process consists of model data, shader data, texture data, and lighting data in a shot on a 3D animated scene. By using the rendering process in a frame per frame, the rendering process will require a long time so the writer will conduct research on the rendering process by using the game engine software on Unreal Engine 4 as a realtime rendering calculation. From the data that was originally done in the virtual software rendering process and then the data will be changed from the virtual file to be converted into FBX file as data that will be processed in the Unreal Engine 4 software (UE4). Using the game engine as the final result in displaying rendering results in realtime is very helpful in getting fluency in a production process in 3D animated films.

Keywords—3D animation, rendering, real time rendering, game engine.

1. PENDAHULUAN

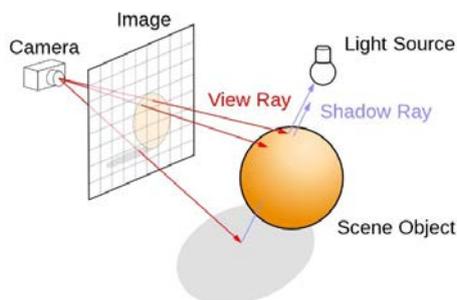
Rendering pada 3D animasi adalah pekerjaan yang sangat memakan waktu. Rendering merupakan proses mengubah model geometris 3D menjadi sebuah gambar grafik seperti pada Gambar 1. Rendering animasi menjadi tugas yang sangat memakan waktu dikarenakan banyaknya frame yang diperlukan untuk membuat animasi. Pada proses rendering satu frame dalam animasi biasanya membutuhkan waktu beberapa jam. Animasi komputer merupakan

sebuah proses pada serangkaian gambar dari hasil render per frame dan kemudian ditampilkan secara berurutan frame demi frame untuk biasa menghasilkan ilusi gambar berupa efek gerakan yang kontinu [1]. Image yang dihasilkan komputer untuk produksi film selalu memiliki persyaratan yang sangat khusus mengenai kualitas gambar dan set fitur sistem rendering yang digunakan. Karena cuplikan film sinematik sering diambil pada frame rate biasanya 24FPS [2].



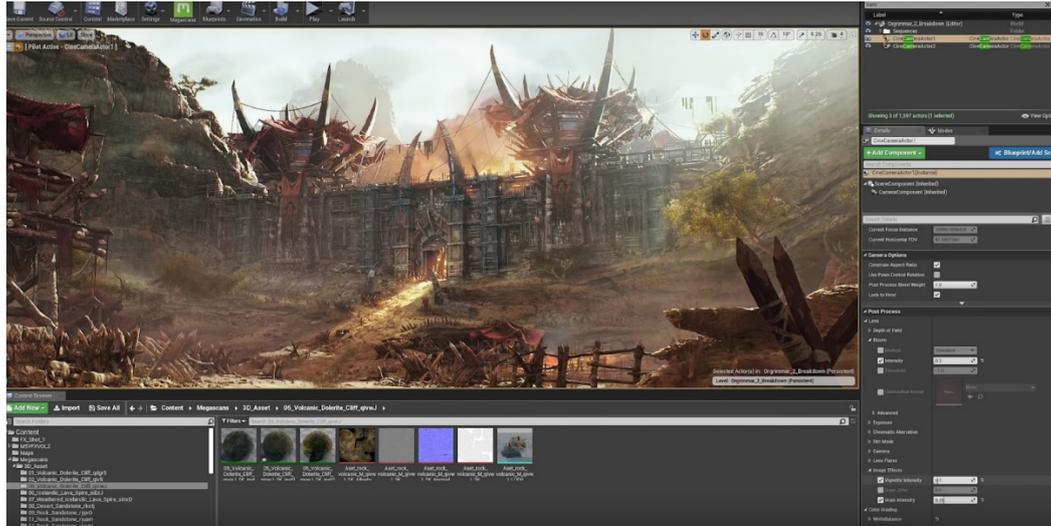
Gambar 1. Proses rendering

Ada dua jenis metode yang utama yang digunakan untuk rendering yaitu rasterisasi (juga dikenal sebagai scanline atau rendering poligon) [3] dan Ray Tracing [4]. Scanline rendering adalah algoritma untuk penentuan permukaan yang terlihat dari dalam grafik komputer 3D, yang bekerja secara baris per baris atau piksel per piksel. Semua poligon yang akan dirender diurutkan terlebih dahulu oleh koordinat y atas di mana pertama kali muncul. Kemudian tiap-tiap baris atau scanline dari image dikomputasikan dengan menggunakan perpotongan antara scanline dengan polygon. Proses pada scanline ini bergerak secara berurutan turun menuju kebawah. Sedangkan Ray Tracing adalah teknik rendering untuk menghasilkan gambar dengan menelusuri jalur cahaya sebagai piksel dalam bidang gambar dan mensimulasikan efek pertemuannya dengan objek virtual seperti pada Gambar 2. Teknik ini dapat menghasilkan tingkat realis visual yang sangat tinggi, dan biasanya hasil rendering ini lebih lama dari pada metode rendering scanline biasa.



Gambar 2. Ray Tracing

Frame pada film 3D animasi yang kita lihat ketika menonton film hari ini biasanya dibuat menggunakan render secara frame per frame, yang dapat membutuhkan banyak waktu beberapa jam untuk menghasilkan per frame. Proses yang memakan waktu ini menghasilkan konten yang sulit, sehingga untuk mengatasi masalah ini dengan memanfaatkan grafik real-time seperti pada Gambar 3 sebagai solusi untuk menyelesaikan pekerjaan secara efisien. Sementara grafik real-time [5] dapat digunakan sebagai pratinjau pada tingkat kualitas yang belum mencapai standar yang dibutuhkan pada film. Namun selama bertahun-tahun perkembangan yang dihasilkan oleh grafis real-time menjadi lebih halus dan bahkan sebagai pra-visualisasi dapat menggunakan dengan cara Virtual Reality [6]. Hal ini dapat memberikan lebih banyak konteks untuk menjadi lebih kreatif. Mereka juga akan menggambarkan beberapa tantangan mereka tentang bagaimana mereka meramalkan masa depan grafik real-time dalam dunia film [7].



Gambar 3. Realtime Render

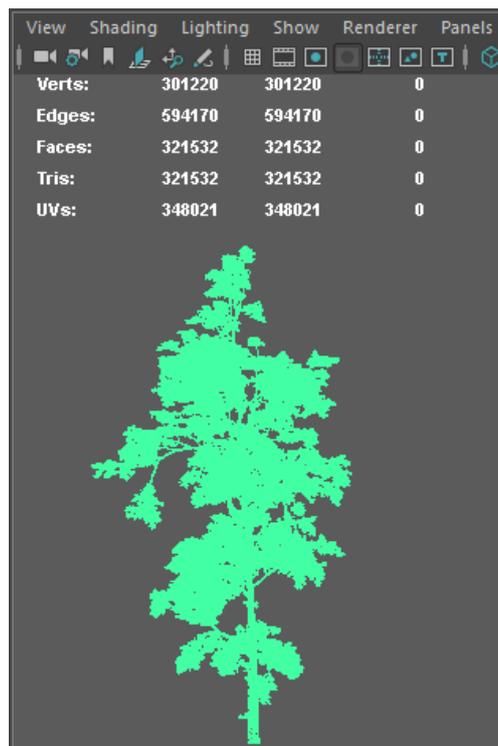
Penggunaan game engine sebagai hasil akhir dalam menampilkan hasil render secara realtime sangatlah membantu dalam mendapatkan kelancaran pada sebuah produksi film [8]. Meskipun UE4 dibuat untuk produksi game akan tetapi system pada UE4 ini memiliki system yang bagus untuk produksi film. Game engine yang semakin populer sampai saat ini dengan banyaknya jumlah pengembang yang saat ini menghasilkan banyak minat dari siswa dalam belajar dan menggunakan engine ini di universitas dengan cara yang berbeda. Karena mereka sangat fleksibel, mereka tidak hanya belajar desain game tetapi mereka dapat digunakan untuk memproduksi film pendek, serial TV, aplikasi Virtual Reality, Visualisasi Desain dan penelitian ilmiah [9].

2. METODE PENELITIAN

Dalam penelitian ini dilakukan pengamatan langsung pada objek yang sesuai dengan lingkup dari penelitian. Pada tahapan ini dilakukan pengumpulan data melalui study literatur dan pengambilan data dari objek penelitian yaitu PT. MSV Studio. Pada penelitian ini objek yang akan diuji untuk rendering adalah pada divisi environment dengan menggunakan jenis aset pohon. Peneliti mengidentifikasi dari objek pada penelitian yang meliputi 3d model pohon, tekstur dan pencahayaan. Selanjutnya dengan melihat hasil proses rendering dengan menggunakan maya dan proses render menggunakan UE4 maka proses analisis dilakukan untuk mendapatkan hasil pipeline rendering [10] secara *realtime* terhadap divisi render khususnya pada 3D aset environment.

2.1. Pengumpulan data

Tahapan yang dilakukan pada pengumpulan data dilakukan secara langsung dengan pengambilan sampel aset pada produksi film animasi 3D yang berjudul Ajisaka. Dari beberapa divisi yang terdapat pada produksi film animasi 3D penelitian hanya dilakukan pada divisi environment sebagai objek penelitian dan memenuhi kriteria dari penelitian yang mencakup shader atau material di dalam objek tersebut dengan beberapa jenis tekstur pada aset pohon. Gambar 4 merupakan 3D objek pohon mesh, texture color, texture specular dan texture normal map.

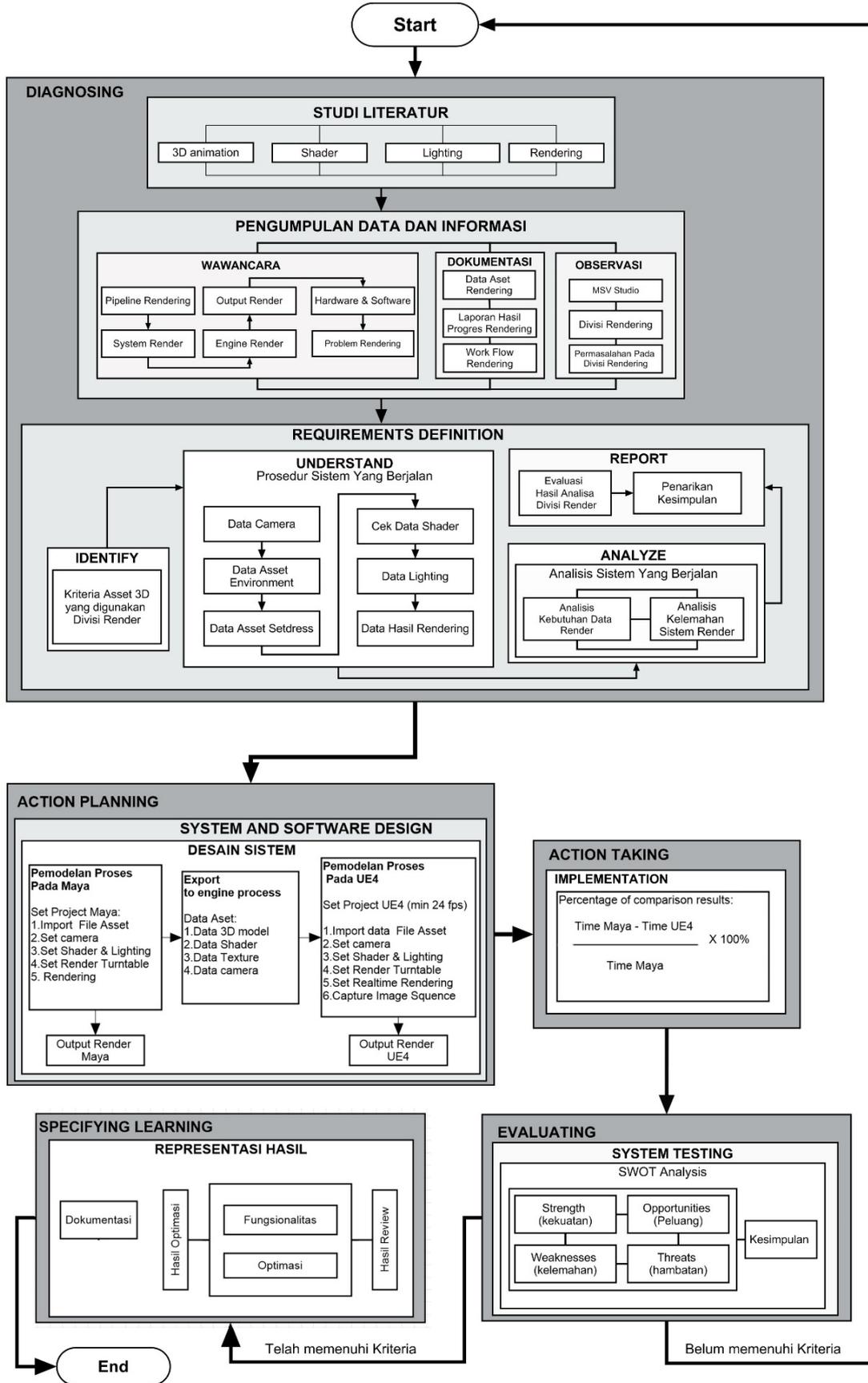


Gambar 4. 3D model pohon

2.1.1. Analisis data

Analisa data yang dilakukan yaitu pada Gambar 5 dengan pengambilan aset dari objek scene Setdress environment 3D yang merupakan bagian dalam pembuatan film animasi 3D. Analisis data pada rendering memerlukan proses export data asset dari maya menjadi FBX file, dikarenakan dalam pengujian proses rendering akan dilakukan dengan menggunakan dua software yaitu maya dan UE4.

Tahapan dalam menganalisa data dari objek Scene Setdress environment 3D asset pohon dengan proses pengujian render menggunakan camera turntable 360 [11] yang berputar mengelilingi objek pohon dengan jumlah waktu yang digunakan sebanyak 200 frame. Dari pengujian yang dilakukan pada software maya di perlukan proses export camera turntable sebagai proses pengujian render pada UE4. Output pada hasil render 200 frame akan diperoleh jumlah waktu yang diperlukan dalam menyelesaikan proses render pada maya dan UE4.



Gambar 5. Alur Penelitian

3. HASIL DAN PEMBAHASAN

Pengujian waktu render bertujuan untuk menguji kinerja rangkaian yang digunakan dan uji unjuk kerja pipeline pada divisi render. Dengan pengujian ini dapat diketahui apakah data aset yang telah dirancang dapat bekerja sesuai dengan yang diinginkan atau tidak. Berikut hasil dari pengujian yang telah dilakukan dengan beberapa indikator pengujian.

3.1 Analisis pada software maya

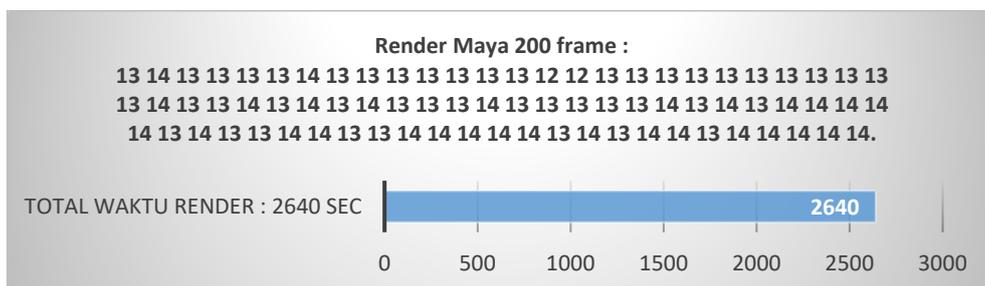
Proses pengujian dilakukan pada komputer dengan spesifikasi:

- Processor : Intel (R) Core (TM) i7-3770 CPU @ 3.40GHz
- RAM : 16 GB
- Kartu grafis (VGA) : NVIDIA GeForce GTX 1050 Ti

Pengujian yang pertama dilakukan pada pipeline software maya dengan menggunakan camera turntambe 360 sebanyak 200 frame dengan resolusi output 700x1000 pixel. Gambar 6 merupakan output image dan Gambar 7 sebagai total waktu render.



Gambar 6. Hasil render pada maya



Gambar 7. Hasil waktu render pada maya

3.2 Analisis pada software game engine UE4

Pengujian yang kedua dilakukan pada software game engine UE4 dengan menggunakan spesifikasi komputer yang sama dan pengujian hasil render juga dilakukan dengan metode yang sama dengan camera turntambe 360 sebanyak 200 frame dengan resolusi output 700x1000 pixel. Proses render pada UE4 ini tidak sama dengan proses render pada maya dikarenakan proses yang

dilakukan pada software maya secara frame per frame, sedangkan proses UE4 ini di render dengan cara realtime render sehingga proses yang dilakukan lebih efisien dari waktu proses render 200 frame tersebut. Hasil proses render pada pengujian ini membutuhkan waktu 25.9 detik seperti pada Gambar 8 sebagai output image dan Gambar 9 sebagai total waktu render.



Gambar 8. Hasil render pada UE4

```
Output Log
Filters Search Log
Cmd: MAP CHECK DONTDISPLAYDIALOG
MapCheck: New page: RND01 - Aug 1, 2020, 2:40:00 AM
MapCheck: Map check complete: 0 Error(s), 0 Warning(s), took 0.309ms to complete.
LogEditorBuildUtils: Build time 0:06
LogUObjectHash: Compacting FUObjectHashTables data took 0.41ms
Job is a success!
LogVolumetricLightmapImport: Imported Volumetric Lightmap in 0.019s
LogVolumetricLightmapImport: Indirection Texture 16x16x16 = 0.0Mb
LogVolumetricLightmapImport: BrickData (all levels) 0.0Mb
LogVolumetricLightmapImport: Bricks at depth
LogVolumetricLightmapImport: 0: 16.7% covering 100.0% of volume
LogVolumetricLightmapImport: 1: 16.7% covering 1.6% of volume
LogVolumetricLightmapImport: 2: 66.7% covering 0.1% of volume
LogVolumetricLightmapImport: Bricks in each level
LogVolumetricLightmapImport: RND01 6 bricks 100.0% (Persistent Level)
LogTexture: Display: Building textures: LQ_Lightmap_1_1 (DXT1, 256X256)
LogTexture: Display: Building textures: HQ_Lightmap1_1 (AutoDXT, 256X256)
LogStaticLightingSystem: RND01_BuiltData storing lightmap data for 3 meshes in 2
LightmapResourceClusters (1.5 Meshes per cluster).
Lightmass on DESKTOP-7V3TQ6M: 7.36 sec total, 643 ms importing, 153 ms setup, 2.82 sec
photons, 3.70 sec processing, 0 ms extra exporting [3/3 mappings]. Threads: 25.9 sec total,
4.16 sec processing.
LogUObjectHash: Compacting FUObjectHashTables data took 1.55ms
LogStaticLightingSystem: Illumination: 12.6 sec (12 ms encoding lightmaps, 0 ms encoding
shadowmaps)
LogStaticLightingSystem: Lightmap texture memory: 0.2 MB (0.2 MB streaming, 0.0 MB non-
streaming), 2 textures
LogStaticLightingSystem: Shadowmap texture memory: 0.0 MB (0.0 MB streaming, 0.0 MB non-
streaming), 0 textures
DumpUnbuiltLightInteractions
Lights with unbuilt interactions: 0
Primitives with unbuilt interactions: 0
Cmd: MAP CHECK NOTIFYRESULTS
MapCheck: New page: RND01 - Aug 1, 2020, 2:40:09 AM
MapCheck: Map check complete: 0 Error(s), 0 Warning(s), took 0.698ms to complete.
Cmd Enter Console Command View Options
```

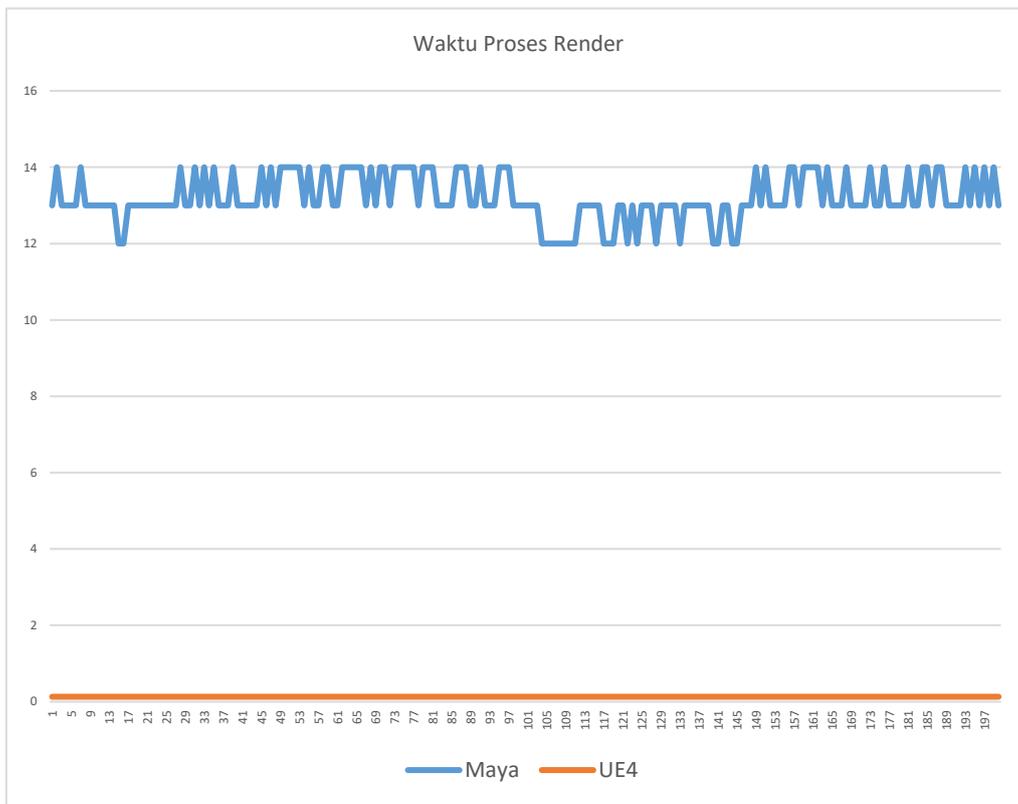
Gambar 9. Hasil waktu render pada UE4

Setelah dilakukan penelitian dari kedua proses render pada maya dan UE4 maka akan terlihat perbedaan waktu yang sangat jauh dari metode secara frame per frame dengan metode realtime render. Dari data yang di hasilkan maka proses render paa UE4 lebih cepat di bandingkan dengan maya dengan perbedaan hasil maya memakan waktu 2640 detik sedangkan UE4 memakan waktu 25.9 detik. Hasil perbandingan tersebut dapat dilihat pada Tabel 1.

Tabel 1. Hasil analisis perbedaan hasil waktu pada proses render maya dan UE4

Software	Jumlah frame	TotalWaktu Proses	Proses render
Maya	200 frames	2640 sec	Frame per frame
UE4	200 frames	25.9 sec	Realtime

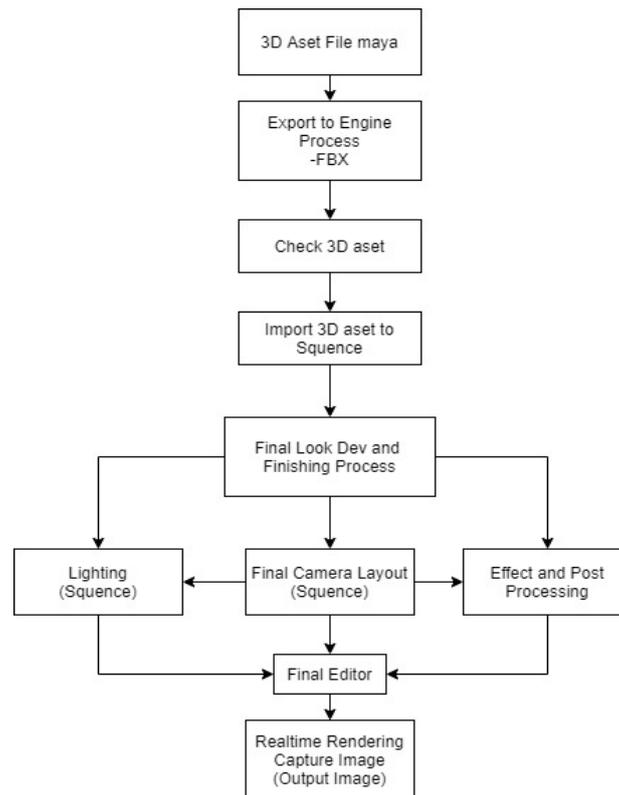
Dilihat dari hasil waktu proses render dapat diambil rata-rata waktu proses render pada software maya 13 detik per farama, sedangkan waktu proses render pada UE4 dapat di ambil rata-rata kurang dari 1 detik per frame. Grafik perbandingan dapat dilihat pada Gambar 10.



Gambar 10. Grafik perbandingan maya dan UE4

3.3 Pipeline Realtime Render

Dari hasil perbandingan render tersebut maka dapat di hasilkan pipeline pada Gambar 11 terkait dengan faktor-faktor yang dapat mempengaruhi kecepatan pada proses render dengan hasil kualitas image rander yang terdapat perbedaan pada proses render maya dengan proses kalkulasi reder pada setiap frame. Sedangkan proses pada UE4 dengan proses render secara realtime yang dapat menghasilkan proses waktu lebih cepat di bandingkan pada proses maya.



Gambar 11. Pipeline Realtime render

4. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian pada proses render secara realtime ini dapat diambil beberapa kesimpulan yaitu:

1. Proses render secara realtime dapat mengoptimasi waktu proses render menjadi lebih cepat sehingga deadline pekerjaan pada divisi render menjadi lebih cepat terselesaikan.
2. Realtime render dapat mengoptimasikan priview final render karena tidak perlu membutuhkan waktu lagi untuk kalkulasi pada setiap akan melakukan proses priview render.
3. Dengan proses secara realtime render maka dapat dilakukan proses render beberapa shot secara bersamaan berdasarkan jumlah camera pada masing-masing shot dalam satu scene environment.
4. Proses render secara realtime dapat membantu mengurangi revisi pada divisi shader karena setiap shader secara realtime dapat diketahui secara langsung benar dan salahnya shader pada scene environment tersebut.

5. SARAN

Agar penelitian dimasa mendatang bisa menjadi lebih baik, beberapa saran yang dapat diambil antara lain:

1. Untuk melengkapi pipeline pembuatan filem 3D animasi di perlukan penelitian pada divisi animasi yang terkait dengan data animasi supaya dapat di integrasikan pada setiap shot camera.
2. Pengembangan yang masih di perlukan pada divisi render diantaranya pada proses simulasi dari hair, cloth, dan vfx.

DAFTAR PUSTAKA

- [1] Chong, A., Sourin, A., Levinski. K., 2006, Grid-based computer animation rendering, GRAPHITE06: International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, Kuala Lumpur Malaysia, November
 - [2] Fascione, L., Hanika, J., Leone, M., Droske, M., Schwarzhaupt, J., Davidovič, T., Weidlich, A., Meng, J., 2018, Manuka: A Batch-Shading Architecture for Spectral Path Tracing in Movie Production, ACM Transaction of Graphics, No. 3, Vol. 37, Hal. 1–18.
 - [3] Francisco., R. P., 2011, Photorealistic Physically Based Render Engines A Comparative Study, Master's Degree in School of Computer Engineering, Polytechnic University of Valencia UPV, Spanish university.
 - [4] Liu, E., Llamas, I., Cañada, J., Kelly, P., 2019, Cinematic rendering in UE4 with real-time ray tracing and denoising, Ray Tracing Gems NVIDIA 2019, doi: https://doi.org/10.1007/978-1-4842-4427-2_19
 - [5] Larsson-stahl, Jennifer., 2020, A Study on the Perceived Realism Of Strand-based Hair Simulated By Style: Evaluating Real-time Hair-simulation in Unreal Engine 4, Thesis, Faculty of Computing, Blekinge Institute of Technology, Sweden.
 - [6] Yuan, J., Fan, A., Xing, R., 2015, Design and Implementation of Games Based on UE4, 2019 International Conference on Computationand Information Sciences (ICCIS 2019), Aljouf - Kingdom of Saudi Arabia, 10 - 11 April
 - [7] Jeremias-Vila, P., Libreri, K., Quaroni, G., Tatarchuk, N., Fagnou, D., 2018, SIGGRAPH '18: ACM SIGGRAPH 2018 Panels, Vancouver, 12 - 16 Agustus
 - [8] Stadler, V., Hlavacs, H., 2018, Blind adventure - A game engine for blind game designers, CHI PLAY 2018 - Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play, Melbourne, 28 - 31 Oktober
 - [9] Salomao, A., Andalo, F., Vieira, M. L. H., 2019, How Popular Game Engine Is Helping Improving Academic Research: The DesignLab Case, AHFE 2018: Advances in Human Factors in Wearable Technologies and Game Design, San Diego, 16 - 20 July
 - [10] Lowe, K. K., 2019, Analyzing and Developing Aspects of the Artist Pipeline for Clemson University Art, Thesis, Master of Fine Arts, Clemson University, South Carolina.
 - [11] Tamboli, R. R., Reddy, M. S., Kara, P. A., Martini, M. G., Channappayya, S. S., Jana, S., 2018, A High-angular-resolution Turntable Data-set for Experiments on Light Field Visualization Quality, 2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX), Sardinia - Italy, 29 - 31 Mei.
-