

# Analisis Perbandingan Render Engine *Cycles* & *Eevee* Pada Blender Melalui Metode *Quasi-Monte Carlo* Untuk Meningkatkan Rendering Di PT. Labtech Penta International

Aragani Timur Kanistren\*, Muhammad Liannardie\*

\* Multimedia Engineering Technology, Politeknik Negeri Batam, Indonesia

---

## Article Info

### Article history:

Received Nov 1st, 2024

Revised Dec 5th, 2024

Accepted Dec 23th, 2024

### Keyword:

Blender  
Cycles  
Eevee  
Rendering

---

## ABSTRACT

This research discusses about analysing the comparison of *Cycles* & *Eevee* rendering engines in Blender 3D opensource software. The test results will help PT Labtech Penta International and three-dimensional application users in choosing a suitable rendering engine for 3D model projects with still image output. Finding the configuration between the *Cycles* and *Eevee* rendering engines or researching it through the *Quasi-Monte Carlo* method, where the value or configuration in the settings of each rendering engine is carried out using several samples. In each sample, the value or configuration is set randomly.

The measurement parameters for this test use 3 variables, rendering speed, render file size, rendered image quality with respect to vertices and texture nodes on 3d objects. The results of this study were able to state that *Eevee* outperformed *Cycles* in the comparison of rendering speed, *Cycles* and *Eevee* in the output of the rendered results did not differ much in file size, and the quality of the rendered images stated that *Cycles* was superior to *Eevee*. Get a solution for rendering at PT Labtech Penta International with a stable and effective configuration.

---

### Corresponding Author:

Aragani Timur Kanistren

Multimedia Engineering Technology, Politeknik Negeri Batam, Indonesia

Email: aragani@polibatam.ac.id

---

## 1. INTRODUCTION

Dunia grafika komputer, render engine adalah inti dari proses rendering yang mengubah model tiga dimensi menjadi gambar dua dimensi dengan menggunakan teknik pencahayaan, *shading*, dan tekstur. Render engine berperan penting dalam menciptakan visualisasi yang memukau dan realistis. Ini menjadi kunci dalam pembuatan gambar komputer yang berkualitas tinggi, terutama dalam pengembangan film, permainan video dan simulasi. Perangkat lunak dalam pembuatan gambar dengan fitur yang lengkap untuk mengembangkan karya model visual realistis adalah blender (Flavell, 2010).

Blender adalah sebuah aplikasi yang berfungsi untuk membuat animasi tiga dimensi, *visual effect*, objek tiga dimensi dan *game*. *Software* ini mengumumkan dirinya sebagai *open source* tiga dimensi *creation suite* (Uma, 2022). Dalam meningkatnya industri visualisasi dan animasi, pemilihan render engine dalam

blender menjadi elemen penting untuk memperoleh kualitas visual yang ideal dalam mendapatkan keefisienan waktu dan hasil yang baik dalam rendering. Blender, sebagai media pemodelan dan rendering yang populer, menyediakan dua pilihan utama render *engine*, yaitu *Cycles* dan *Eevee*. Kedua *engine* ini mempunyai kelebihan dan kelemahan masing-masing, yang menimbulkan keraguan tentang bagaimana kita memperoleh kualitas hasil render berkualitas tinggi dengan memaksimalkan kelebihan dari masing-masing *engine*.

*Cycles*, dengan proses rendering yang lambat karena berbasis *ray tracing* tetapi memberikan hasil yang realistis. Di sisi lain *Eevee* menawarkan kecepatan render yang tinggi tetapi dengan beberapa keterbatasan dalam realisme visual. Pengoptimalan render *engine* pada penelitian ini menggunakan metode *Quasi-Monte Carlo*. Metode *Quasi-Monte Carlo* adalah metode pengambilan sampel nilai atau konfigurasi konfigurasi dari render *engine* secara acak. Sifat acak yang dimiliki metode *Quasi-Monte Carlo* dapat memastikan bahwa sampel yang diambil akan memiliki konfigurasi yang merata di setiap sampelnya, menurut *Website* resmi Blender yaitu *blender.org* versi 2.79. Metode *Quasi-Monte Carlo* merupakan metode yang paling realistis dan cocok untuk render *engine* pada Blender.

PT. Labtech Penta International yang bergerak dibidang *knowledge engineering*, menggunakan render *engine* Blender *Cycles* dan *Eevee* tanpa memaksimalkan kelebihan dan kekurangan yang ada pada *engine* ini. Oleh karena itu, penelitian ini bertujuan untuk memberikan wawasan yang mendalam tentang perbedaan fundamental antara *Cycles* dan *Eevee*, menyertakan aspek-aspek seperti waktu render, ukuran file outputnya, dan hasil visual yang dihasilkan dan memberikan panduan yang tajam bagi praktisi desain tiga dimensi di PT. Labtech Penta International. Dengan pemahaman yang lebih baik terpaut karakteristik keduanya, diinginkan penelitian ini dapat memberikan prinsipal yang stabil bagi para pengguna render *engine* Blender dalam memilih render *engine* yang paling sesuai dengan tujuan mereka, sekaligus berperan pada peningkatan lebih lanjut dalam bidang render *engine* pada Blender.

## 2. RESEARCH METHOD

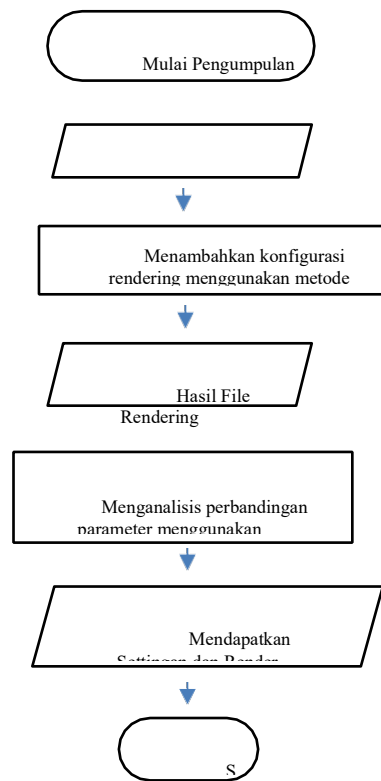
### 2.1 Metode *Quasi-Monte Carlo*

Penelitian pada render *engine* *Cycles* dan *Eevee* ini menggunakan metode *Quasi-Monte Carlo*. Metode *QMC* mengambil sampel dari nilai atau konfigurasi secara acak dan biasanya membutuhkan beberapa sampel lebih dan konfigurasi lebih untuk menentukan hasilnya. Metode ini dibuat seacak mungkin agar sampel memiliki nilai atau konfigurasi seluas luasnya. Namun, akan tersebar nilai atau konfigurasinya secara merata agar sampel yang diambil bermacam-macam dan mendapatkan hasil yang akurat dan menguntungkan (Roosendaal, 2017). Sampel sampel yang berisikan nilai atau konfigurasi acak untuk dijadikan patokan rendering dan akan di render satu per satu di setiap render *engine* *Cycles* dan *Eevee*.

Hasil dari renderingnya di kelompokkan sesuai sampel, render *engine* dan parameter pengukurannya dianalisis menggunakan metode kuantitatif. Metode kuantitatif disini berperan untuk menganalisa kualitas hasil render yang sebelumnya sudah dikumpulkan survei google form, menganalisa pengukuran kecepatan waktu render menggunakan data statistik yang berbentuk tabel statistik deskriptif dan hasil rendering tersebut dianalisis besar *output* ukuran file dari masing-masing *engine* satu per satu. Kemudian hasil dari statistik parameter pengukuran hasil render, kecepatan render dan ukuran file render dapat ditarik kesimpulan.

### 2.2 Tahapan Penelitian

Penelitian ini dijalankan menggunakan beberapa tahapan. Tahapan tersebut digunakan untuk membantu pelaksanaan dalam penelitian dan supaya pelaksanaannya lebih tertata. Tahapan penelitian tersebut dipaparkan pada gambar 3.1 berikut.



Gambar 3.1 Tahapan Penelitian

### 3.2 Tahapan Penelitian

Tahap pengumpulan data ialah langkah permulaan dalam melaksanakan penelitian ini. Dilakukan studi literatur pengumpulan data yang diraih dari beberapa literatur yang mengangkat permasalahan mengenai teknik rendering *Cycles* dan *Eevee*. Studi literatur ini dilakukan dengan menggali informasi dari jurnal, buku, dan artikel.

Dan juga pengumpulan data untuk mendapatkan informasi dan ilmu untuk meneliti penelitian ini, dari berbagai pihak serta menambahkan beberapa pertanyaan perenderan still image melalui *engine Cycles* dan *Eevee* kepada bidang ahli melalui pertanyaan pertanyaan yang akan dilakukan dengan cara observasi dan wawancara.

| No | List Pertanyaan Wawancara kepada Ahli Bidang   |
|----|--|
| 1. | Bagaimana mengoptimalkan pengaturan sampel untuk mencapai hasil render yang bebas <i>noise</i> ?                                 |
| 2. | Apa keuntungan dan kerugian menggunakan <i>raytracing</i> berbasis <i>CPU</i> dibandingkan dengan <i>GPU</i> ?                   |
| 3. | Apa teknik terbaik atau bagian sampel konfigurasi untuk mendapatkan tampilan realistis?  |
| 4. | Bagaimana mengoptimalkan pengaturan <i>Eevee</i> untuk mendapatkan hasil <i>real-time</i> yang halus?                            |
| 5. | Apa keuntungan dan batasan menggunakan <i>Eevee</i> dibandingkan dengan <i>Cycles</i> dalam hal waktu render dan kualitas akhir? |
| 6. | Apa teknik terbaik untuk menggunakan pencahayaan dalam <i>Eevee</i> ?  |

|     |  |
|-----|--|
| 7.  | Bagaimana cara terbaik untuk mengatur <i>Light path</i> agar terlihat realistis dalam <i>still image</i> ?                 |
| 8.  | Apa keuntungan dan kerugian merender <i>still image</i> menggunakan <i>engine Cycles</i> dan <i>Eevee</i> ?                |
| 9.  | Berikan konfigurasi yang harus diperhatikan dalam merender <i>still image</i> pada <i>engine Cycles</i> dan <i>Eevee</i> ? |
| 10. | Apa yang menjadi batasan dalam mengukur besar kecil <i>output file</i> dari merender <i>still image</i> ?                  |

Tabel 3.1 List Pertanyaan Wawancara kepada Ahli Bidang.

### 3.3 Tahapan Penelitian

Dalam hal ini alat penelitian atau perangkat yang digunakan untuk meneliti yaitu melibatkan penggunaan komputer (*device pc*). Perangkat komputer dapat mencakup komponen komponen seperti prosesor, memori, media penyimpanan, kartu grafis dan perangkat lainnya (Darmawan, 2012). Berikut spesifikasi komputer yang digunakan di PT. Labtech Penta International.

|    |                  |                        |
|----|------------------|------------------------|
| 1. | Sistem Informasi | Windows 10             |
| 2. | <i>Processor</i> | Intel® Core™ i5-7400   |
| 3. | Memory           | 8192MB RAM             |
| 4. | Grafis Card      | NVIDIA GeForce GTX 970 |

Tabel 3.2 Spesifikasi alat penelitian

### 3.5 Objek Penelitian

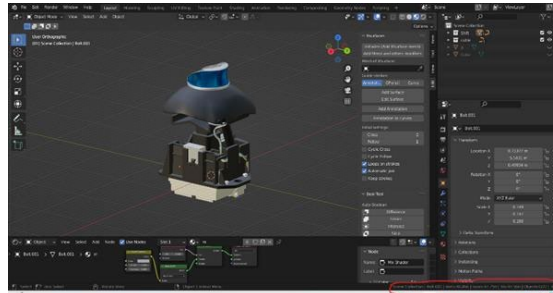
Objek penelitian yang dipakai pada teknik rendering *Cycles* dan *Eevee* adalah Material dan Model tiga dimensi yang proses pembuatannya memanfaatkan *software* Blender tiga dimensi. Daftar model tiga dimensi yang berisi banyaknya jumlah *vertices* dan *node textures* yang dibutuhkan ditampilkan pada tabel 3.3.

| No  | Model                    | Vertices  | Node Texture |
|-----|--------------------------|-----------|--------------|
| 1.  | Truck                    | 439.152   | 213          |
| 2.  | AC Indoor                | 59.880    | 29           |
| 3.  | Smoke Detector           | 61.190    | 18           |
| 4.  | High Beam Globe          | 3.429     | 42           |
| 5.  | Shell                    | 169.491   | 113          |
| 6.  | Core                     | 1.917.360 | 140          |
| 7.  | Damper Actuator          | 34.981    | 32           |
| 8.  | Fuse Box Car             | 76.323    | 30           |
| 9.  | Molded Plastic Manometer | 2.266     | 6            |
| 10. | Nissan Leaf Shift        | 49.325    | 63           |
| 11. | Parking                  | 28.652    | 62           |

|     |             |        |     |
|-----|-------------|--------|-----|
|     | Break Valve |        |     |
| 12. | Suspension  | 14.521 | 372 |

Tabel 3.3 Spesifikasi alat penelitian

Pembuatan beberapa model tiga dimensi yang digunakan untuk mengetahui pengaruh banyaknya *vertices* dalam rendering dibuat secara *poly modelling*, dengan menggunakan *subdivision surface* untuk menambah *polygon* dan mengaktifkan *shade smooth* agar pada bagian tepi terlihat lebih halus dan penggunaan *node texture* yang nantinya akan mempengaruhi besar ukuran file render.

Gambar 3.2 Model tiga dimensi Nissan Leaf Shifter, *vertices* dan *node texture* nya.

### 3.6 Parameter Pengukuran

Kualitas render, kecepatan render, dan ukuran file hasil render merupakan tiga aspek penting yang menentukan performa sebuah render *engine*. Untuk mengevaluasi aspek-aspek tersebut, beberapa parameter pengukuran dapat digunakan. Pada kualitas render, noise level dan *spectral accuracy* menilai tingkat kebisingan serta akurasi reproduksi warna pada gambar hasil render (Raghani et al., 2021). *Dynamic range* menunjukkan rentang kontras cahaya maksimal yang mampu direpresentasikan dengan baik oleh engine (Banterle et al., 2011). Semakin tinggi nilai parameter ini, maka semakin baik kualitas render yang dihasilkan.

Sedangkan pada kecepatan render, parameter utama yang digunakan adalah *time to complete single frame* yaitu lama waktu untuk me-render 1 *frame* gambar, serta rendering throughput yakni banyaknya *frame* yang dapat di-render per detik (Garanzha & Pantaleoni, 2017). Semakin singkat waktu render per *frame* dan semakin tinggi teroutput maka semakin cepat kemampuan rendering *engine* tersebut.

Adapun ukuran file hasil render diukur berdasarkan *frame size* yaitu total *bytes* yang dibutuhkan untuk menyimpan 1 *frame* render, serta *compression ratio* gambar yang didukung format render, misalnya *JPEG 2000* (Christensen & Jarosz, 2016). Kedua parameter ini menunjukkan efisiensi penyimpanan file render yang dihasilkan masing-masing *engine*. Tabel parameter pengukuran ditampilkan pada tabel 3.4.

| No | Pengukuran Variabel      | Indikator Variabel   | Skala Pengukuran  |
|----|--------------------------|--|-------------------|
| 1  | Kecepatan Rendering      | Membandingkan kecepatan saat proses rendering semua model tiga dimensi dari model nomor 1-12 | Minute dan Second |
| 2  | Ukuran Hasil File Render | Membandingkan ukuran file hasil render pada semua model tiga dimensi dari model nomor 1-12   | Megabyte (mb)     |
| 3  | Kualitas Gambar Hasil    | Membandingkan kualitas gambar hasil render   | Skala 1-5         |

|  |        |   |  |
|--|--------|---|--|
|  | Render | pada semua model tiga dimensi dari model 1-12 |  |
|--|--------|---|--|

Tabel 3.4 Parameter Pengukuran

### 3. RESULT AND ANALYSIS

#### 3.1 Konfigurasi Rendering

Menambahkan *environment node world texture* pada di setiap model untuk mendapatkan cahaya sekitar pada objek.

Gambar 4.1 *Environment node world texture*

Setelah itu membuat sampel konfigurasi render menggunakan metode *QMC*, yang berarti sampel acak dari render *engine Cycles* dengan hasil sebagai berikut:

|           |  |  |   |  |
|-----------|--|--|---|--|
| Sampel 1  | Menggunakan setingan awal dari Blender   |  |   |  |
| Sampel 2  | Max bounces light paths total 16, Diffuse 8, Glossy 8, Transmission 16, Volume 4 dan <u>Transparent 12</u>       | Clamping 10.00   |   |  |
| Sampel 3  | Max bounces light paths total 14, Diffuse 6, Glossy 10, Transmission 12, Volume 6, <u>Transparent 14</u>         | Clamping 10.00   | Fast GI dan Simplify <u>diaktifkan</u>  | Exposure 1.50                          |
| Sampel 4  | Sampling render noise threshold 0.0300 dan max samples 500   | Denoise <u>diaktifkan (OptiX)</u>                                | Max bounces light paths total 12, Diffuse 4, Glossy 4 dan yang <u>lainnya 0</u>                           | Clamping 1.00                          |
| Sampel 5  | Sampling render noise threshold 0.3000 dan max samples 128   | Denoise <u>diaktifkan (OptiX)</u>                                | Max bounces light paths total 12, Diffuse 8, Glossy 8, <u>Transmission 8</u> dan yang <u>lainnya 0</u>    | Filter Glossy 10.00                    |
| Sampel 6  | Sampling render noise threshold 1.0000 dan max samples 200   | Min light bounces 100, Min transparent 100, light threshold 1.00 | Max bounces light paths total 12, Diffuse 4, Glossy 4, Transmission 12, Volume 2 dan <u>Transparent 8</u> | Clamping 10.00 dan Filter Glossy 10.00 |
| Sampel 7  | Sampling render noise threshold 0.5000 dan max samples 500   | Denoise <u>OpenImageDenoise</u>                                  | Min light bounces 50, Min transparent 50, light threshold 1.00  | Fast GI dan Simplify <u>diaktifkan</u> |
| Sampel 8  | Max bounces light paths total 200, Diffuse 4, Glossy 100, Transmission 12, Volume 100 dan <u>Transparent 100</u> | Clamping 1000.00   |   |  |
| Sampel 9  | Filter Glossy 10.00  | Filter Glossy 10.00  |   |  |
| Sampel 10 | Max bounces light paths total 24, Diffuse 8, Glossy 8, Transmission 24, Volume 4 dan <u>Transparent 16</u>       | Filter Glossy 0  |   |  |

Tabel 4.1. 10 Sampel sampel konfigurasi render *engine Cycles* menggunakan metode *QMC*, Sampel 1-5 menggunakan *GPU* dan Sampel 6-10 menggunakan *CPU*.

Pada sampel sampel diatas, sudah dilakukannya pemilihan untuk hasil rendering mana yang paling bagus disetiap objek nya melalui survei *google form*, dari 36 responden yang menjawab kuesioner dinyatakan bahwa hasil nya konfigurasi sampel 2 yang terpilih dengan rata rata presentase 63,9%. Untuk hasil rendering nya dapat dilihat pada Gambar 4.2.

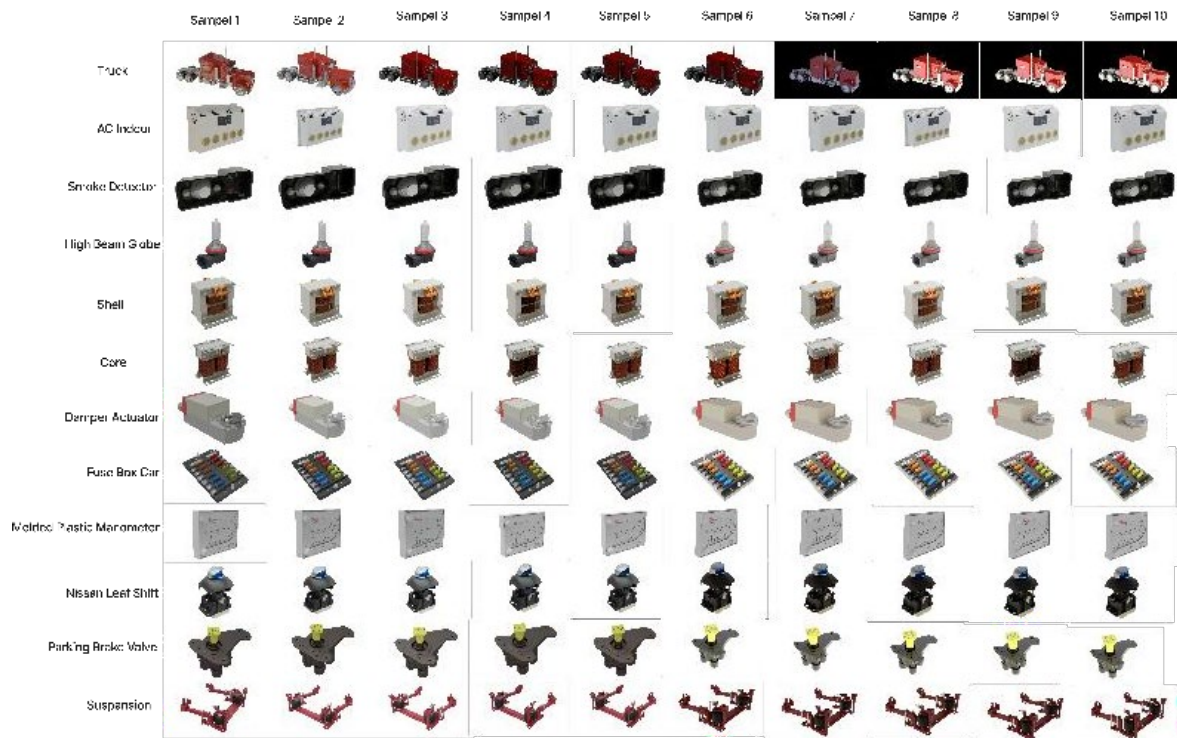
Setelah itu, membuat konfigurasi render bagian *engine Eevee* di setiap objek dengan hasil render yang sama dengan hasil render *engine Cycles* agar komparasi yang dilakukan nantinya akan lebih merata dan objektif, dengan hasil sebagai berikut:

| No | Objek                    | Konfigurasi   |  |  |  |
|----|--------------------------|---|--|--|--|
| 1  | Truck                    | Ambient Occlusion dan Bloom <b>diaktifkan</b>       | Screen Space Reflection <b>diaktifkan</b> (Refraction <b>diaktifkan</b> , thickness 20.200 dan clamp 19.600)   | Color management (sRGB, Filmic, Exposure 0.408 dan Gamma 1.900)  | Tambahkan Light jenis Sun (Strength 0.200)                                       |
| 2  | AC Indoor                | Ambient Occlusion dan Bloom <b>tidak diaktifkan</b> | Color management (sRGB, Standard, Exposure -1.000 dan Gamma 1.400)   | Tambahkan Light jenis Sun (Strength 8.000)   |  |
| 3  | Smoke Detector           | Ambient Occlusion dan Bloom <b>diaktifkan</b>       | Screen Space Reflection <b>diaktifkan</b> (Refraction <b>diaktifkan</b> , thickness 10.000 dan clamp 11.200) dan <b>tambahan node texture transparan</b> | Color management (sRGB, Standard, Exposure -1.200 dan Gamma 1.300)   | Tambahkan Light jenis Sun (Strength 10.000)                                      |
| 4  | High Beam Globe          | Ambient Occlusion dan Bloom <b>diaktifkan</b>       | Color management (sRGB, Standard, Exposure 1.000 dan Gamma 2.100)  | Tambahkan Light jenis Sun (Strength 5.400)   |  |
| 5  | Shell                    | Ambient Occlusion dan Bloom <b>diaktifkan</b>       | Color management (sRGB, Standard, Exposure 0.920 dan Gamma 1.000)  | Tambahkan Light jenis Sun (Strength 10.000)  |  |
| 6  | Core                     | Ambient Occlusion dan Bloom <b>diaktifkan</b>       | Color management (sRGB, Standard, Exposure 0.914 dan Gamma 0.976)  | Tambahkan Light jenis Sun (Strength 10.000)  |  |
| 7  | Damper Actuator          | Ambient Occlusion dan Bloom <b>diaktifkan</b>       | Color management (sRGB, Standard, Exposure 1.800 dan Gamma -1.050)   | Tambahkan Light jenis Sun (Strength 9.000)   |  |
| 8  | Fuse Box Car             | Hanya Ambient Occlusion yang <b>diaktifkan</b>      | Screen Space Reflection <b>diaktifkan</b> (Refraction <b>diaktifkan</b> , thickness 51.4 m dan clamp 10.000) dan <b>tambahan node texture transparan</b> | Color management (sRGB, Standard, Exposure -3.490 dan Gamma 1.772)   | Tambahkan Light jenis Sun (Strength 46.600)                                      |
| 9  | Molded Elastic Manometer | Ambient Occlusion dan Bloom <b>tidak diaktifkan</b> | Color management (sRGB, Filmic, Exposure -0.224 dan Gamma 0.900)   | Tidak pakai light tambahan   |  |
| 10 | Nissan Leaf Shift        | Ambient Occlusion dan Bloom <b>tidak diaktifkan</b> | Screen Space Reflection <b>diaktifkan</b> (Refraction <b>tidak diaktifkan</b> , thickness 1.7 m dan clamp 10.000)  | Motion blur <b>diaktifkan</b> dan soft shadow <b>juga</b> untuk color management (sRGB, Standard, Exposure -0.900 dan Gamma 1.500) | Tambahkan Light jenis Sun (Strength 2.200) dan contact shadow 0 m dan bias 0.001 |
| 11 | Parking Brake Valve      | Ambient Occlusion dan Bloom <b>tidak diaktifkan</b> | Screen Space Reflection <b>diaktifkan</b> (Refraction <b>tidak diaktifkan</b> , thickness 1.7 m dan clamp 10.000)  | Color management (sRGB, Standard, Exposure -1.128 dan Gamma 1.628)   | Tambahkan Light jenis Sun (Strength 10.000 dan Diffuse 0.17)                     |
| 12 | Suspension               | Ambient Occlusion dan Bloom <b>tidak diaktifkan</b> | Screen Space Reflection <b>diaktifkan</b> (Refraction <b>tidak diaktifkan</b> , thickness 1.7 m dan clamp 10.000)<br>Shadow 64px                         | Color management (sRGB, Standard, Exposure -0.900 dan Gamma 1.700)   | Tambahkan Light jenis Sun (Strength 8.000, Diffuse 0.80 dan Specular 0.22)       |

Tabel 4.2. Konfigurasi render engine Eevee.

### 3.2 Hasil Rendering

Proses rendering dilakukan setelah penyetingan render, kemudian hasil render engine Cycles yang menggunakan metode QMC & konfigurasi engine Eevee yang menyesuaikan objek render engine Cycles. Berikut hasil render Cycles menggunakan sampel sampel dari metode QMC yang ditampilkan pada Gambar 4.2.



Gambar 4.2 Hasil render engine Cycles dengan sampel metode QMC

Seperti yang kita sudah ketahui pada konfigurasi rendering dengan hasil rendering diatas dan hasil survei *google form* membuktikan bahwasannya sampel 2 yang terpilih menjadi yang terbaik dari segi hasil rendering pada *engine Cycles*.

Setelah mendapatkan sampel 2 render *engine Cycles* yang menjadi bahan komparasi render *engine Eevee*, oleh karena itu setelah dicari konfigurasi yang hasil rendering nya serupa dengan objek objek sampel 2 *engine Cycles*, dapat dilihat konfigurasi nya pada Tabel 4.2 dan untuk hasil rendering nya dapat dilihat pada Gambar 4.3.

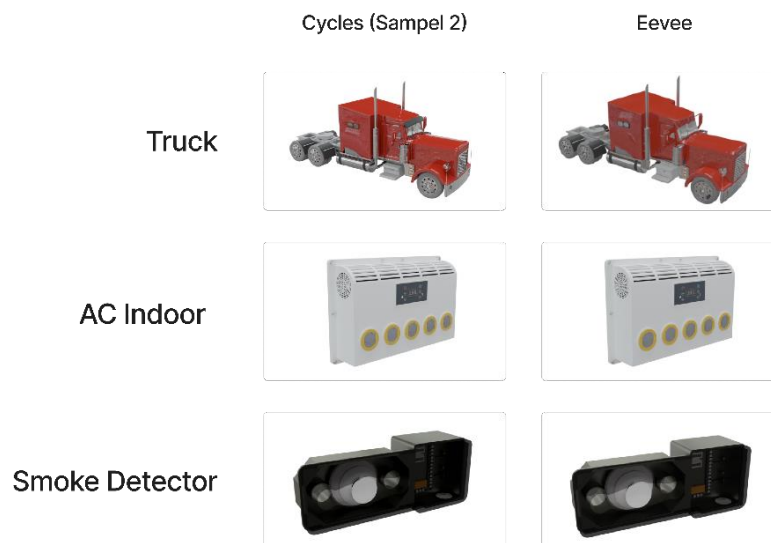


Gambar 4.3 Hasil render *engine Eevee*.

Hasil render *engine Eevee* yang sudah diserupakan dengan sampel render *engine Cycles* yang terpilih yaitu sampel 2 akan dibandingkan dengan parameter yang sudah ditentukan.

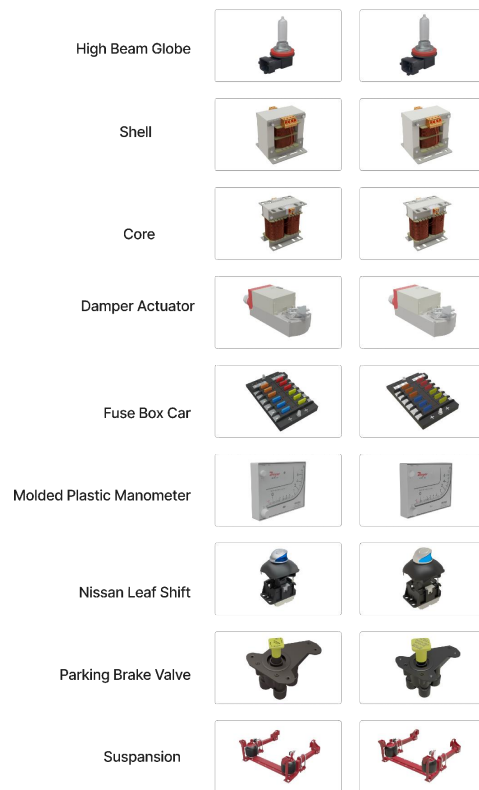
#### 4.2 Analisa Perbandingan Parameter

Menganalisa hasil render 12 objek tiga dimensi dan sampel yang sudah terpilih antara *engine Cycles & Eevee* pada hasil rendering menggunakan metode *QMC*. Banyaknya jumlah vertices dan node texture yang nantinya akan berpengaruh ke parameter pengukuran. Hasil perbandingan rendering nya sebagai berikut.



Gambar 4.4 Perbandingan hasil rendering.





Gambar 4.5 Perbandingan hasil rendering.

Perbandingan hasil rendering kedua engine dapat dipastikan tidak jauh berbeda, karena dalam konfigurasi render *engine Eevee* menyesuaikan render *engine Cycles* Sampel 2. Survei *google form* untuk perbandingan tersebut menyatakan bahwa rata rata presentase 55,3 % dari 36 responden melihat hasil render *engine Eevee* lebih baik dari hasil render *engine Cycles* pada objek yang menggunakan konfigurasi *Screen Space Reflection*.

Tetapi untuk keseluruhan objek, 36 responden menyatakan render *engine Cycles* lebih realistis dengan rata-rata presentase 58,1%. Namun, penting untuk diingat dalam mencapai hasil rendering Eevee untuk menyamai sampel 2 render *engine Cycles* membutuhkan akurasi dan konfigurasi yang kompleks dalam mencapai realistis *still image*, berbeda dengan render *engine Cycles* yang tidak banyak membutuhkan konfigurasi, dapat dilihat pada Tabel 4.1 dan Tabel 4.2. Tetapi pastinya *Cycles* ada yang dikorbankan seperti kecepatan rendering, dapat dilihat pada Tabel 4.3.

| No | Objek                    | Cycles (Sampel 2) | Eevee       |
|----|--------------------------|-------------------|-------------|
| 1  | Truck                    | 00:01:06.98       | 00:00:02.30 |
| 2  | AC Indoor                | 00:01:02.28       | 00:00:01.27 |
| 3  | Smoke Detector           | 00:00:28.52       | 00:00:01.44 |
| 4  | High Beam Globe          | 00:00:17.77       | 00:00:00.43 |
| 5  | Shell                    | 00:01:44.00       | 00:00:03.83 |
| 6  | Core                     | 00:00:59.46       | 00:00:03.90 |
| 7  | Dumper Actuator          | 00:00:29.52       | 00:00:01.50 |
| 8  | Fuse Box Car             | 00:00:30.76       | 00:00:01.42 |
| 9  | Molded Plastic Manometer | 00:00:41.21       | 00:00:00.25 |
| 10 | Nissan Leaf Shift        | 00:01:41.39       | 00:00:01.71 |
| 11 | Parking Break Valve      | 00:00:22.73       | 00:00:10.77 |
| 12 | Suspension               | 00:00:30.54       | 00:00:03.96 |

Tabel 4.3 Perbandingan kecepatan rendering

Dengan adanya hasil kecepatan rendering tabel 4.3 diatas, bahwasannya render *engine Cycles* lebih lama waktu perenderingannya dari pada render *engine Eevee*. Render *engine Cycles* lebih lama dikarenakan sebuah metode *engine* itu sendiri, yang memiliki proses yang sangat kompleks, dapat diambil contoh pada konfigurasi *Cycles* sampel 2 dalam *light path* nya memiliki efek tingkat lanjut yang semuanya berkontribusi pada realistis tinggi yang membuat waktu komputasi meningkat.

Di sisi lain konfigurasi pada *Eevee* seperti *Space Screen* yang metode komputasi nya jauh lebih cepat dihitung, karena ditangani dengan *Screen Space Reflection* sedangkan *Cycles* dihitung secara alami melalui *path tracing*. Faktor lainnya, sampling konfigurasi bawaan dari masing masing render *engine* berbeda, *Cycles* lebih besar dari pada *Eevee* seperti yang dapat kita lihat pada Gambar 4.6.

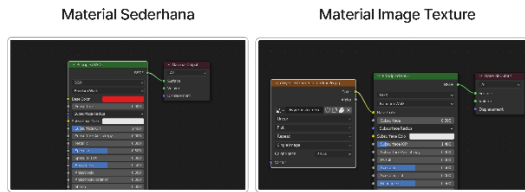


Gambar 4.6 Sampling Cycles (4096) dan Eevee (64)

Meskipun demikian, dalam ranah *still image*, *Cycles* & *Eevee* dapat menghasilkan hasil rendering yang sebanding tetapi tidak dengan waktu rendering. Pilihan antara keduanya sering bergantung pada prioritas proyek.

Jika membahas kecepatan rendering pada kedua *engine*, terdapat objek yang memiliki kecepatan rendering yang hampir waktu render nya mendekati waktu render *engine Eevee* dapat dilihat pada Tabel 4.3, yaitu pada objek *Smoke Detector* yang memiliki *vertices* terbanyak ke 5 dan *Node Texture* terdikit ke 2, dapat dilihat pada Tabel 3.2. *Node Texture* yang ada pada objek *Smoke Detector* sedikit ke 2 dari objek yang lainnya. Karena *Cycles* sangat memahami material, jadi meskipun memiliki banyak *vertices*, render bisa cepat menurut *Blender Foundation*.

Dikutip juga dalam *Blender Foundation* “*Material Settings*”, kecepatan rendering tidak hanya bergantung pada *vertices*, tetapi juga bergantung pada kompleksitas material. Oleh karena itu, objek *Molded Plastic Manometer* yang memiliki *Vertices* dan *Node Texture* paling sedikit dari yang lainnya dapat dilihat pada Tabel 3.2, bahwasannya kecepatan render *engine Cycles* nya lebih lama daripada objek *Smoke Detector* dikarenakan objek *Molded Plastic Manometer* memiliki material *Image Texture* sedangkan objek *Smoke Detector* hanya material biasa, yang membuat *Cycles* sangat efisien dalam menangani material yang relatif sederhana dan render bisa tetap cepat.



Gambar 4.7 Material dalam Node Texture

| No | Objek                    | Cycles (Sampel 2) | Eevee   |
|----|--------------------------|-------------------|---------|
| 1  | Truck                    | 2.01 MB           | 1.79 MB |
| 2  | AC Indoor                | 1.49 MB           | 1.67 MB |
| 3  | Smoke Detector           | 1.53 MB           | 1.01 MB |
| 4  | High Beam Globe          | 1.53 MB           | 1.92 MB |
| 5  | Shell                    | 1.64 MB           | 1.92 MB |
| 6  | Core                     | 1.65 MB           | 1.83 MB |
| 7  | Damper Actuator          | 1.39 MB           | 1.59 MB |
| 8  | Fuse Box Car             | 1.58 MB           | 1.65 MB |
| 9  | Molded Plastic Manometer | 1.65 MB           | 1.54 MB |
| 10 | Nissan Leaf Shift        | 1.37 MB           | 1.36 MB |
| 11 | Parking Break Valve      | 1.47 MB           | 1.49 MB |
| 12 | Suspension               | 1.52 MB           | 1.42 MB |

Tabel 4.4 Ukuran hasil file rendering kedua *engine*.

Dengan adanya Tabel 4.4 bahwasannya render *engine Cycles* & *Eevee* perbandingan hasil file rendering nya tidak jauh beda dikarenakan ukuran file rendering ditentukan oleh resolusi, kedalaman bit dan format file, bukan oleh komputasi (Akenine-Möller, 2018). Contohnya yang dapat diambil setiap objek yang memiliki warna, piksel dan bit jika dirender 2 dimensi dengan format yang sama, skenario nya tidak akan jauh beda karena kedua *engine* ini mengkompresi dengan informasi yang sama.

Namun, dalam hasil pada Tabel 4.4, dapat dilihat render *engine Cycles* rata rata objek semua nya lebih sedikit besar dari render *engine Eevee*, dikarenakan efektivitas kompresi dari konfigurasi yang berbeda membuat perbedaan ukuran hasil file rendering tidak jauh beda (Sayood, 2017).

#### 4. CONCLUSION

Solusi terbaik untuk bisa mendapatkan hasil render still image di PT. Labtech Penta International adalah dengan bergantung pada prioritas proyek, jika fotorealisme adalah yang utama, *Cycles* jawabannya. Tetapi jika kemampuan literasi cepat lebih penting, *Eevee* menjadi pilihan yang menarik.

Kecepatan rendering menggunakan render *engine Cycles* lebih lama daripada menggunakan render *engine Eevee*, tetapi kita juga bisa menyesuaikan kecepatan render *engine Eevee* dengan memakai render *engine Cycles* dengan memperhatikan penggunaan material *Node Texture* yang sederhana tanpa harus memperhatikan Jumlah *Vertices*.

Ukuran file hasil render pada teknik rendering *Cycles* dan *Eevee* tidak begitu jauh berbeda, karena kedua *engine* tersebut memiliki resolusi, kedalaman bit dan format file pada objek yang dikompresi dengan informasi yang sama, akan berbeda sedikit jika konfigurasi render rumit, karena akan mempengaruhi efektivitas kompresi.

Konfigurasi render *engine Cycles* akan lebih efektif jika material yang digunakan tidak terlalu rumit dan itu yang akan mempengaruhi kecepatan rendering dan tidak mempengaruhi kualitas dari render *engine Cycles* itu sendiri, jadi bisa menyeimbangi pemilihan konfigurasi dengan efek realistis tingkat lanjut yang membuat waktu komputasi meningkat.

Untuk konfigurasi render *engine Eevee* itu sendiri yang membuat nya dapat menyaingi realistis nya render *engine Cycles* adalah dengan memakai *Screen Space Reflection*, mengatur cahaya yang masuk dengan manajemen warna dan yang paling penting menambahkan *Add Light > Sun*, tambahan konfigurasi ini yang tidak ada pada render *engine Cycles* dan konfigurasi *Add Light > Sun* ini yang

membantu membandingkan dengan render *engine Cycles* secara *apple to apple* dengan memperhatikan dan mengatur kekuatan cahayanya. Maka dari itu, untuk para akademik yang lagi mendapatkan proyek *hard surface* dengan *output* render *still image*, bisa menggunakan konfigurasi yang telah dijabarkan.

## REFERENCES

- Andersson, R. (2016). *Comparative Study Of Real Time Render Engines In 3D Animation Production*.
- Banterle, F., Artusi, A., Debattista, K., & Chalmers, A. (2011). *Advanced high dynamic range imaging: theory and practice*. CRC Press.
- Budi Arifitama, Ade Syahputra, Ketut Bayu Yogha Bintoro (2022). Analisis Perbandingan Efektifitas Metode Marker dan Markerless Tracking pada Objek Augmented Reality. *Jurnal Integrasi*, 14(1)
- Chaitanya, C. R. A., Kaplanyan, A. S., Schied, C., Salvi, M., Lefohn, A., Nowrouzezahrai, D., & Aila, T. (2017). *Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder*. *ACM Transactions on Graphics (TOG)*, 36(4), 1-12.
- Christensen, P. H., & Jarosz, W. (2016). *The path to path-traced movies*. *Foundations and Trends® in Computer Graphics and Vision*, 10(2), 103-175.
- Dudek, S., & Dziedzic, K. (2022). *Comparative analysis of the Cycles and Eevee graphics engines on the example of rendering 3D models of archaeological artifacts*. *Journal of Computer Sciences Institute*, 24, 218–223.
- Fakenine-Möller, T., Haines, E., & Hoffman, N. (2018). *Real-time rendering*. AK Peters/CRC Press.
- Garanzha, V., Pantaleoni, J., & McAllister, D. (2011, December). *Simpler and faster HLBVH with work queues*. In ACM SIGGRAPH Asia 2011 papers.
- Goralczyk, A. (2019). Spring-2D Animated Short Made in Blender. In ACM SIGGRAPH 2019 Studio.
- Hendri Syahputra, Mahmuda Saputra, Buge Cipta Wijaya (2022). Implementasi Perbandingan dan Optimalisasi Teknik 3D Rendering pada Objek Animasi Profil Fakultas Teknik Universitas Gajah Putih Takengon. *Jurnal Informasi dan Teknologi*, 4(4)
- Raghani, A., Novák, J., Dixon, P., Bashford-Rogers, T., Hatchett, J., & Debattista, K. (2021). *Objective evaluation metrics for perceptual realism in rendering algorithms*. *Computers & Graphics*, 99, 1-13.
- Sahputra, E., Sucahyo, M., H. (2022). Analisis Teknik Rendering Eevee Engine pada Pembuatan Video Animasi 3D Rumah Sakit Mukomuko. *Jurnal Komitek* 2(2).
- Tengku Hisyam Muhammad Umar, Doni Winarso (2020). Analisis Perbandingan Teknik 3D Rendering *Cycles* Dan *Eevee* Pada *Software* Blender. *Jurnal Fasilkom*, 10(1).
- Ton Roosendaal (2017). Blender, *Blender Render, Quasi-Monte Carlo*. Blender.org 2.79.
- Sayood, K (2017). *Introduction to Data Compression*. Morgan Kuafmann.