

ANALISIS PERFORMA PROTOTYPE GAME PADA PLATFORM ANDROID

Riwinoto*, William Tan**

* Informatics Engineering, Batam State Polytechnic

** Multimedia and Network Engineering, Batam State Polytechnic

Article Info

Article history:

Received Des 12th, 2021

Revised Des 20th, 2021

Accepted Des 26th, 2021

Keyword:

Performansi game

Mobile game

Unity profiler

unity 3d

komputasi

ABSTRACT

Saat ini, Perkembangan teknologi sudah sangat maju. Sebagian masyarakat sudah menggunakan dan memiliki *Smart Phones* ataupun *Mobile Phones*. Salah satu aspek didalam *Smart Phones* adalah *Game Mobile* yang dapat didownload di *App Store*, maupun *Play Store*. *Game Mobile* memiliki kesesuaian dengan pengguna saat ini. Dalam pengembangan game mobile, salah satu yang penting adalah performansi yang optimum game bisa berjalan di perangkat. Penelitian merupakan penelitian awal untuk mengetahui performansi prototype game yang sedang dikerjakan. Analisa performansi mencakup komputasi rendering, scripting, physics, animasi, garbage collection, global illumination, UI dan others. Performansi dilakukan dengan menggunakan tools *unity profiler* yang tersedia di *unity 3D IDE*. Hasil observasi menunjukkan bahwa prototype game yang sudah dikembangkan masih mempunyai performansi komputasi yang buruk karena FPS yang dihasilkan hanya sampai level 15-20 FPS di mana idealnya mendapatkan 60 FPS. Diperlukan riset lanjutan untuk menentukan langkah optimasi pada prototype game sebelum melangkah ke pengembangan game yang full fitur.

Corresponding Author:

Riwinoto,

Jurusan Teknik Informatika

Politeknik Negeri Batam

Jalan Ahmad Yani No 1, Batam Kota, Batam, Kepulauan Riau, Indonesia

Email: riwi@polibatam.ac.id

1. INTRODUCTION

Pada perkembangan teknologi sekarang, *game* merupakan salah satu bagian yang mengalami perkembangan pesat. Salah satu target platform yang dinikmati oleh user adalah dengan *Mobile Planning*. *Mobile games* sudah berkembang dan sangat mudah untuk didapatkan [1]. Terlebih lagi, aplikasi game kebanyakan bisa didapat dengan mendownload secara gratis di *Google Play* maupun *App Store* [2].

Game untuk *Mobile Phones* memiliki kesesuaian dengan pengguna sekarang ini [3]. Dalam pembuatan game, terdapat berbagai macam *Game Engine* yang untuk mengembangkan game baik PC maupun *Mobile Phones*, *Game Engine* yang sering digunakan sekarang ini adalah *Unity Engine*, dan *Realtime Engine*, dan masih banyak lagi. Dalam pembahasan kali ini, penulis akan berfokus kepada *Unity Engine*. *Unity Engine* merupakan *game engine* yang dikembangkan oleh Unity Technologies. Software ini diluncurkan tahun 2005 dan sudah digunakan oleh banyak pengembang game di dunia [4].

Dalam pengembangan game, performa merupakan salah satu aspek yang dibahas. Performa sendiri merupakan evaluasi pada semua produk software yang menghitung serta membantu merumuskan seberapa besar sebuah perangkat bekerja, seperti seberapa besar penggunaan sebuah aplikasi [5], seberapa besar pemakaian CPU. Proses tersebut memerlukan percobaan dengan cara mengumpulkan data dan dilakukan analisa terhadap titik puncak sebuah perangkat dapat bekerja [6], dan dilakukan perbaikan penyebab dari masalah sebuah performa [7].

Optimasi yang buruk dapat menimbulkan rendahnya *frame rate*, *freezes*, *crashes*, *input lag*, *loading time*, bahkan menggunakan daya baterai yang sangat besar. Untuk mengetahui performa dalam Unity, kita dimudahkan dengan adanya *Unity Profiler* [7] yang dibuat untuk memudahkan optimisasi performa dengan cara mengecilkan laporan statistik tentang pemaikaina performa dalam Unity3D ketika sedang berlangsung.

Tahapan *profiler* dilakukan melaluiobservasi dan mengambil sampel *fps (frame per second)* . Pengambilan data dilakukan dengan cara *independent*, yaitu mengambil 2 sampel data kemudian dibandingkan dengan sebelum melakukan optimisasi sebuah performa dan setelah dilakukan optimisasi. Sebagai bahan kajian, penulis akan menganalis performansi sebuah prototype game yang sedang dikembangkan, nama game tersebut tidak disebutkan karena masih dalam tahap pengembangan.

2. RESEARCH METHOD

Metode analisis performa pada penelitian ini menggunakan metode kualitatif dan kuantitatif dengan mengukur standard optimisasi. Untuk mengetahui nilai hasil yang optimal dalam pengembangan game tersebut, diperlukan cara untuk melakukan penelitian pada optimisasi tersebut. Oleh karena itu, penulis mengambil cara tahapan *Optimization Cycle* dari *makaka.org* [8].

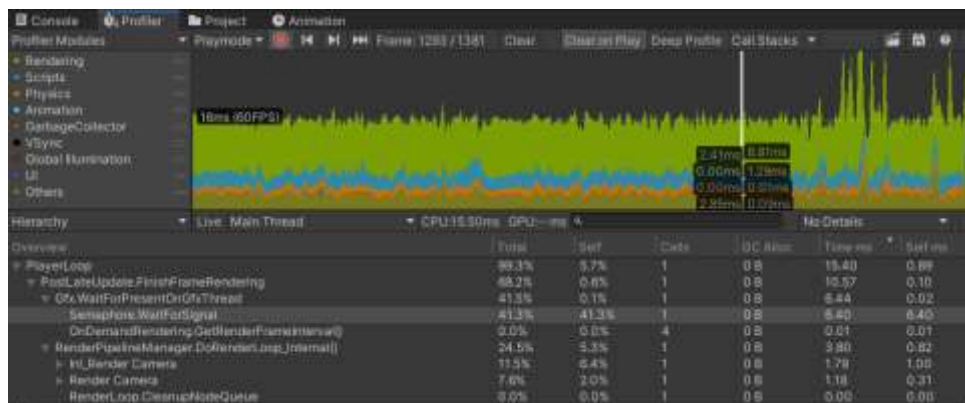
OPTIMIZATION CYCLE



Gambar 1. Unity Optimization Cycle

Terdapat 4 langkah dalam melakukan optimisasi dari Gambar 1. Langkah pertama , merupakan hasil dari game itu sendiri, dapat dilihat dalam profiler, menurut expert dari Dreams Studio [9] “*fps* dari *mobile device* minimal harus memiliki 60 fps”, dikarenakan layar *mobile device* jaman sekarang biasanya memiliki layar kecepatan 60 fps.

Kemudian dilakukan pengecekan *profiling*, menentukan bagian secara spesifik dengan membaca setiap proses yang dijalankan oleh aplikasi. *Profiling* sangat membantu dalam proses optimisasi.



Gambar 2. Profiler

Terdapat beberapa bagian data dalam sebuah *profiler* yang dapat membantu menentukan performa pada sebuah device. Yang pertama merupakan *CPU Usage*. *CPU Usage* merupakan bar yang berwarna warni yang menentukan kinerja pada setiap proses dalam *device*. Terdapat *rendering*, *scripts*, *physics*, *animation*, *GC*, *VSync*, *Global Illumination*, *UI*, serta yang lainnya. Dari situ kita dapat mengambil sampel dari 1 *frame* pada grafik tersebut yang kemudian diperjelas didalam opsi *hierarchy*.

Hierarchy merupakan performa yang dihasilkan oleh dan kemudian diperjelas dengan memberikan sampel bagian secara spesifik mengenai kinerja dari CPU, ataupun GPU. Dari situ akan didapat data penyebab masalah performa.

Setelah melakukan *Profiling*, kita melakukan *List to Do*, yaitu menentukan bagian apa yang ingin dioptimasi. Terdapat berbagai cara untuk melakukan optimasi, yaitu *Batching*, *Object Pooling*, *GPU Instancing*, *Lightmapping*, *Texture*, *Occlusion Culling*, *Scripting*, serta *Tris and Vertexes*. Masing-masing pada proses diatas dapat mengoptimisasikan performa dari game [8].

Object Pooling merupakan metode untuk meminimalisir objek yang sifat nya banyak, tetapi dalam waktu yang singkat, yang berfungsi untuk mengurangi kinerja CPU membuat objek berulang kali dengan menetapkan jumlah objek yang dihasilkan.

Batching merupakan metode penggabungan objek yang sama menjadi 1, sehingga CPU akan lebih mudah mengenali objek tersebut tanpa melakukan analisis pembacaan objek yang baru. GPU Instancing merupakan proses yang membuat GPU membaca objek sehingga tidak membebankan CPU untuk membaca *texture* dari objek tersebut.

Texture, merupakan kualitas dari gambar dari sebuah objek, semakin banyak *texture* dan semakin kompleks *texture* tersebut, maka akan mempengaruhi kinerja dari perangkat yang bekerja. Occlusion Culling merupakan metode untuk menghilangkan objek yang tidak tampak ketika kamera sedang merender, berfungsi untuk mengurangi memory yang digunakan untuk membaca seluruh map.

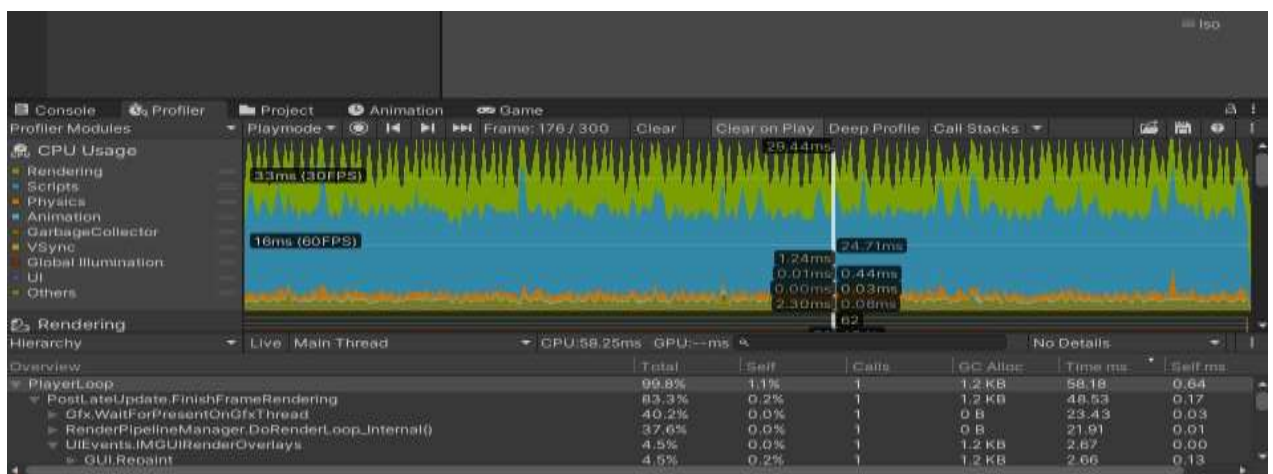
Scripting merupakan pembuatan perintah, yang akan berpengaruh pada performa ketika beberapa perintah tersebut melakukan kinerja yang cukup berat dan dihasilkan pengulangan tingkah laku sebuah objek, yang pada akhirnya dapat menurunkan performa dari perangkat. *Tris and Vertexes*, merupakan segitiga dari setiap *mesh* yang dihasilkan dan dibaca oleh kamera didalam sebuah fram. Untuk itu, semakin besar Tris dan Vertex, dapat mempengaruhi penurunan performa.

Sehingga dari sekian cara diatas, kita dapat membandingkan antara sebelum dan sesudah melakukan optimisasi pada game tersebut. Sehingga tercipta berbagai macam sampel yang akan mempengaruhi dan dijadikan sebagai acuan dalam analisa performa, jika belum teroptimasi maka, kembali lagi menuju tahapan awal dari pengecekan *profiler*.

3. RESULTS AND ANALYSIS

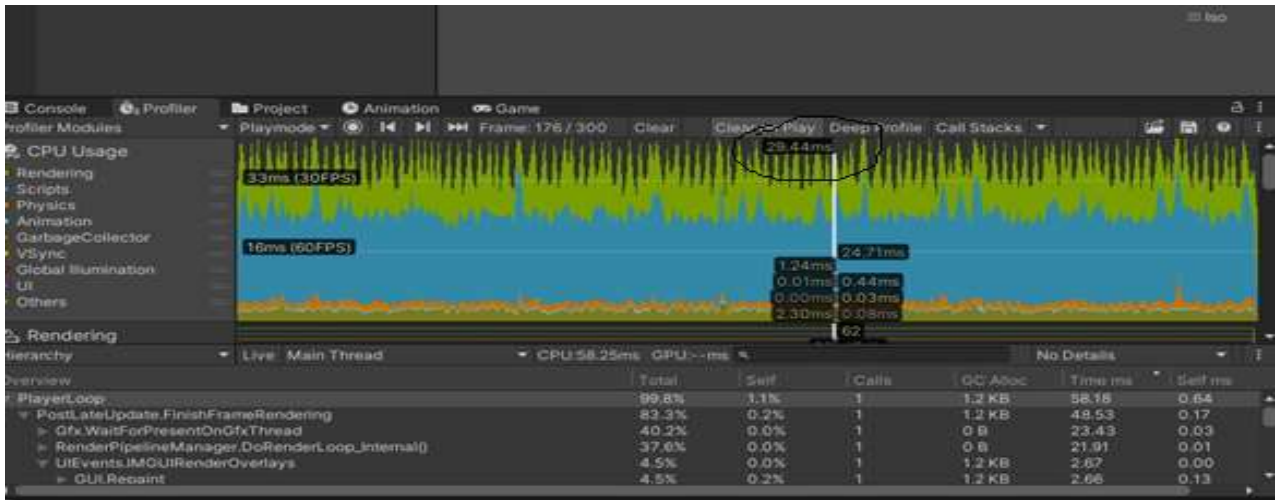
Game dianalisis adalah game 2.5D platformer, menggunakan unity 3D dan beroperasi di platform mobile, masih satu level dan dimainkan selama 2.000 frame atau 30 detik dengan setingan 1 detik adalah 60 frame dalam render game.

Berikut adalah hasil dari profiling dari game tersebut di unity 3D.



Gambar 3. Hasil profiling prototype game

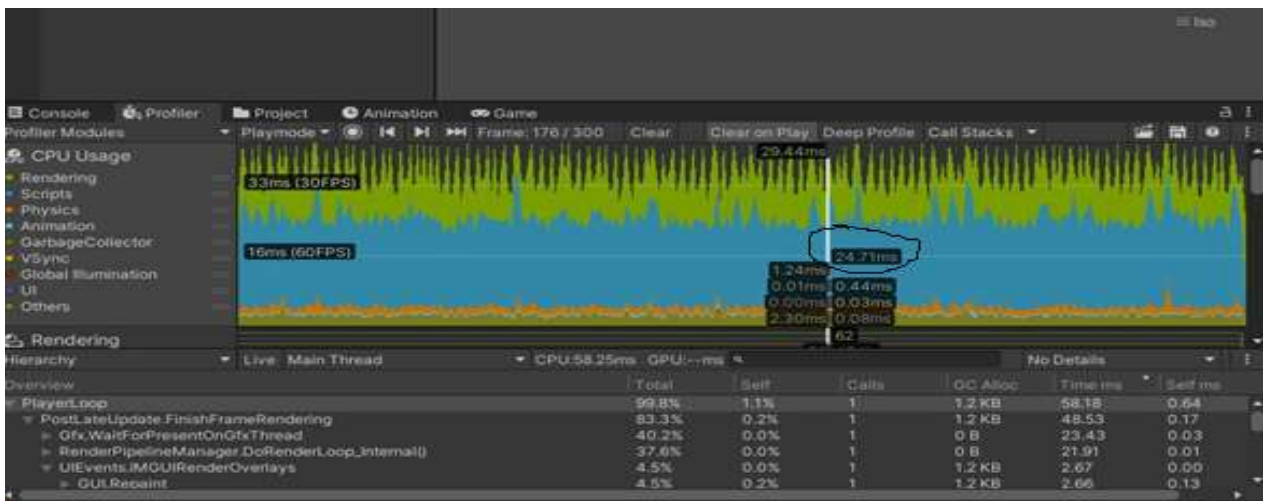
3.1 Analisa Rendering



Gambar 4. Hasil profiling rendering prototype game

Warna hijau pada gambar 4 menunjukkan hasil komputasi rendering pada game 25D pada frame yang dipilih menghasilkan 29.44 ms. Hasil rendering dengan waktu tersebut menyebabkan frame turun ke 30 FPS. Sedangkan syarat game yang baik diperlukan minimal 60 FPS untuk seluruh proses dengan total komputasi maksimal 16 ms. Dari hasil tersebut menunjukkan game tidak optimal hanya melihat dari rendering time saja.

3.2 Analisa Scripting

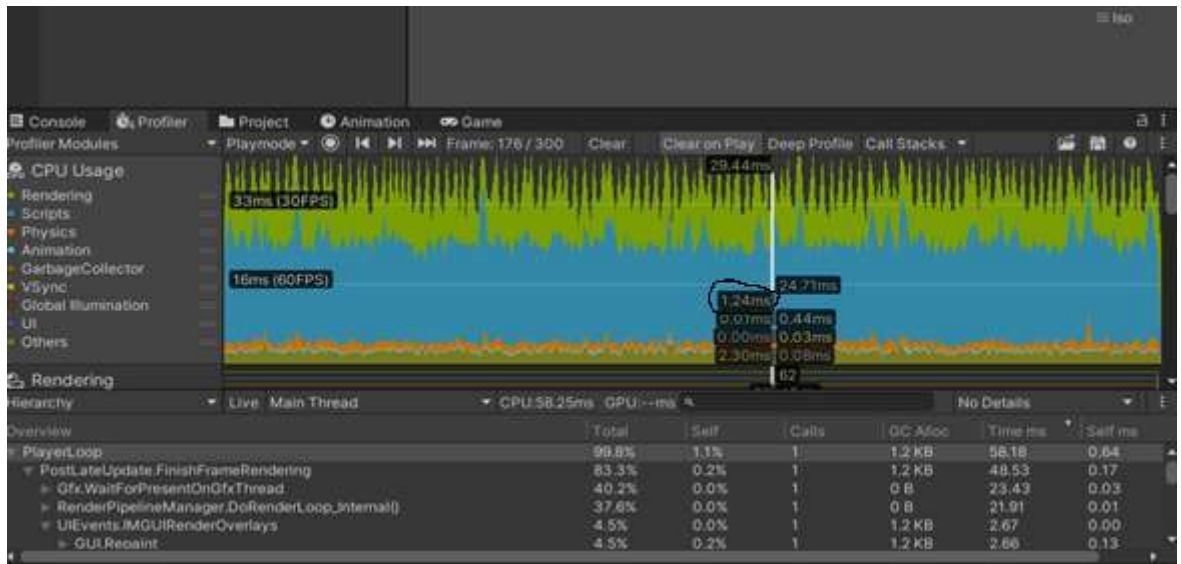


Gambar 5. Hasil profiling scripting prototype game

Warna biru pada gambar 5 menunjukkan hasil komputasi scripting pada game 25D pada frame yang dipilih menghasilkan 24.77 ms. Hasil komputasi scripting dengan waktu tersebut menyebabkan frame turun ke sekitar 45 FPS. Sedangkan syarat game yang baik diperlukan minimal 60 FPS untuk seluruh proses dengan total komputasi maksimal 16 ms. Dari hasil tersebut menunjukkan game tidak optimal hanya melihat dari rendering time saja. Dibandingkan dengan hasil point rendering, komputasi scripting lebih kecil efeknya dibanding rendering di mana rendering menyebabkan perfomansi turun ke 30 FPS sedangkan scripting bertahan di 45 FPS.

Meskipun demikian untuk standar game minimal 60 FPS dan komputasi maksimal 16 ms, komputasi scripting yang ideal adalah di bawah 3 ms. Jadi hasil komputasi scripting pada game dengan 24.77 ms sangat tidak optimal.

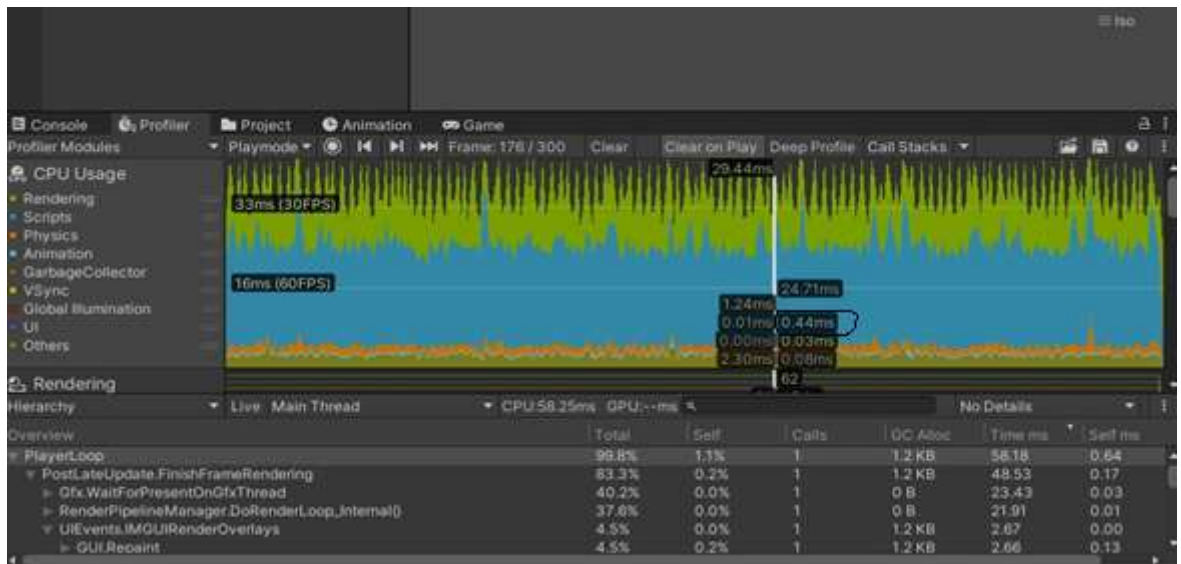
3.3 Analisa Physics



Gambar 6. Hasil profiling physics prototype game

Warna orange pada gambar 6 menunjukkan hasil komputasi physics pada game 25D pada frame yang dipilih menghasilkan 1.24 ms. Hasil ini menunjukkan sudah cukup baik mengingat yang dibutuhkan total adalah 16 ms. Jadi ada spread cukup lebar antara 1.24 ms-16 ms untuk komputasi aspek lain.

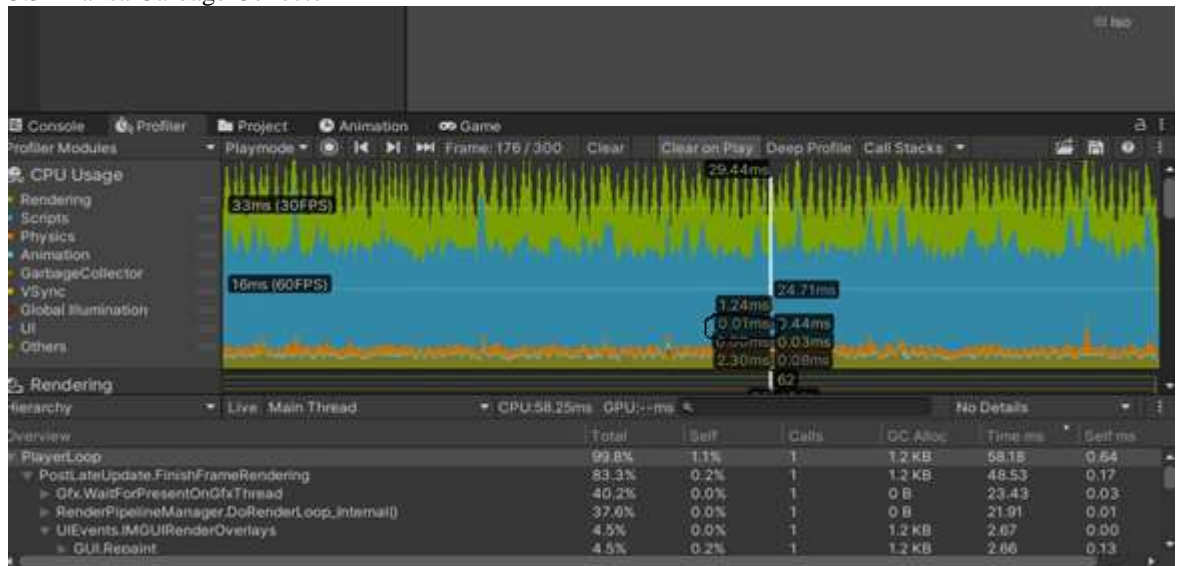
3.4 Analisa Animation



Gambar 7. Hasil profiling animation prototype game

Warna biru pucat pada gambar 7 menunjukkan hasil komputasi animation pada game 25D pada frame yang dipilih menghasilkan 0.44 ms. Hasil ini menunjukkan sudah cukup baik mengingat yang dibutuhkan total adalah 16 ms. Jadi ada spread cukup lebar antara 0.44 ms-16 ms untuk komputasi aspek lain.

3.5 Analisa Garbage Collector



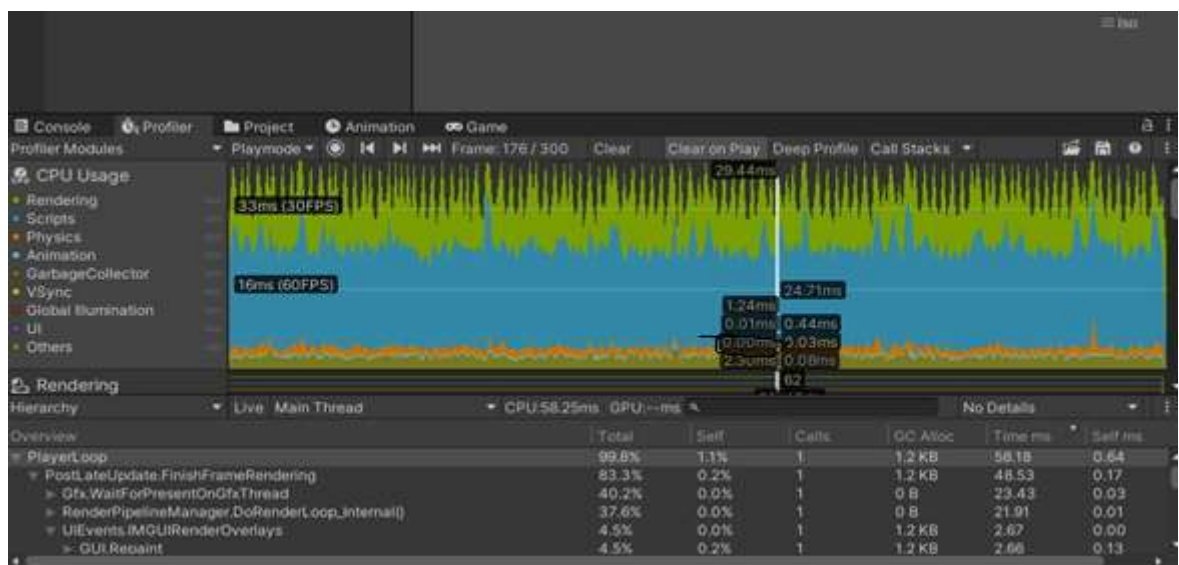
Gambar 8. Hasil profiling garbage collector prototype game

Warna hijau kekuningan pada gambar 8 menunjukkan hasil komputasi garbage collector pada game 25D pada frame yang dipilih menghasilkan 0.01 ms. Hasil ini sebenarnya menunjukkan sudah cukup baik mengingat yang dibutuhkan total adalah 16 ms. Jadi ada spread cukup lebar antara 0.01 ms-16 ms untuk komputasi aspek lain. Namun perlu dicek untuk seluruh frame apakah terdapat juga hasil memori pada komputasi yang sama dengan frame lain karena jika durasi game cukup panjang mengakibatkan kinerja memori menjadi lebih berat. Sebagai contoh dalam 1000 frame terdapat 1.2 kb per frame, maka memori garbage collection yang dihasilkan menjadi 1000 Kb. Jadi hasil tersebut perlu diwaspadai.

3.6 Analisa vsync

Hasil vsync tidak perlu dihitung karena untuk platform mobile. vsync dimatikan karena platform mobile mempunyai vsync tersendiri.

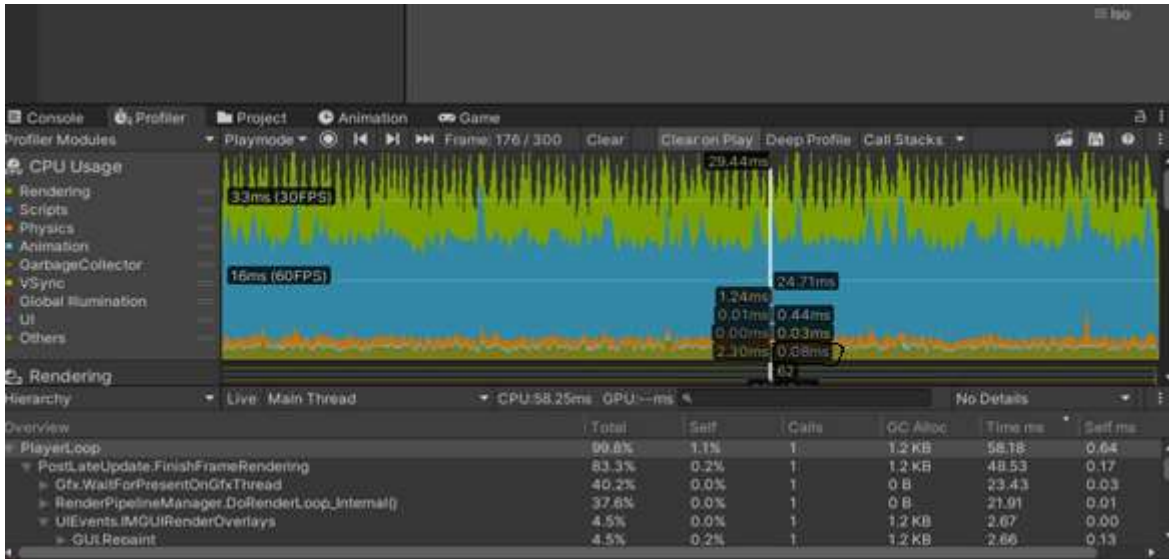
3.7 Analisa Global Illumination



Gambar 9. Hasil profiling global illumination prototype game

Warna merah pada gambar 9 menunjukkan hasil komputasi global illumination pada game 25D pada frame yang dipilih menghasilkan 0.00 ms. Hasil ini menunjukkan sudah baik mengingat yang dibutuhkan total adalah 16 ms. Jadi ada spread cukup lebar antara 0.00 ms-16 ms untuk komputasi aspek lain.

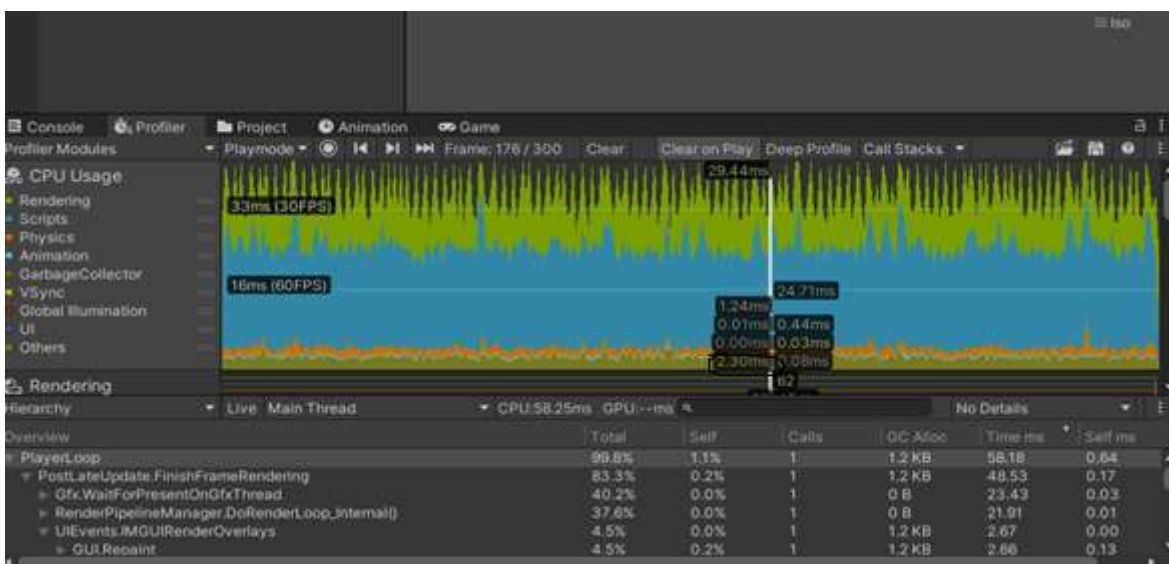
3.8 Analisa UI



Gambar 10. Hasil profiling UI prototype game

Warna ungu pada gambar 10 menunjukkan hasil komputasi UI pada game 25D pada frame yang dipilih menghasilkan 0.08 ms. Hasil ini menunjukkan sudah baik mengingat yang dibutuhkan total adalah 16 ms. Jadi ada spread cukup lebar antara 0.08 ms-16 ms untuk komputasi aspek lain. Memang pada game, tampilan UI belum banyak jadi wajar masih bagus komputasi.

3.9 Analisa Others



Gambar 11. Hasil profiling Others prototype game

Warna krem hijau pada gambar 11 menunjukkan hasil komputasi Others pada game 25D pada frame yang dipilih menghasilkan 2.30 ms. Hasil ini bisa dari berbagai kemungkinan misal hardware dan software lain. Hasil ini menunjukkan sudah baik mengingat yang dibutuhkan total adalah 16 ms. Jadi ada spread cukup lebar antara 2.38 ms-16 ms untuk komputasi aspek lain.

4. KESIMPULAN

Perfomansi komputasi prototipe game belum baik karena komputasi rendering dan scripting menyebabkan FPS pada game turun sekitar 15-20 FPS. Jika dilakukan play game maka hasilnya akan patah patah karena rate FPS normal minimal 24 FPS di mana mata manusia melihat gerakan dengan halus tanpa patah patah. Diperlukan riset lanjutan untuk menurunkan komputasi rendering dan scripting.

ACKNOWLEDGEMENTS

Terima kasih kepada PT ZETTA NUSANTARA yang telah membantu baik pendanaan dan fasilitas dalam penelitian ini.

REFERENCES

- [1] Penttinen, E. R. (2010). Mobile Games: Analyzing the Needs and Values of the Consumers. *Journal of Information Technology Theory and Application*, 5-22.
 - [2] Baskara Arya Pranata, A. K. (2015). *Mudah Membuat Game dan Potensi Finansialnya dengan Unity 3D*. Elex Media Computindo
 - [3] Rogers, R. (2011). In R. Rogers, *Learning Android Game Programming*. Addison-Wesley.
 - [4] Optimize Your Mobile Game Performance , *Unity3d.com*,2020. Diakses september 2021
 - [5] Praveen Kundurthy, ‘Software Performance Optimizations for Games: Best Practices), Intel Corporation, 2016
 - [6] Dickinson, C. (2015). *Optimization Unity*. Packt Publishing Ltd.
 - [7] Dickinson, C. (2017). *Unity 2017 Game Optimization*. Packt Publishing Ltd.
 - [8] Sirota, A. (2021). *Makaka Games. Unity Optimization : Mobile & Desktop*.
 - [9] Adryanto, J. (2021). *game optimazation*. unpublished.
-