

Optimasi Occlusion Culling dan Render Scale dalam Game Labirin Matematika Mazetic

Khindy Zharifany Fayyadhana Fattah^{a,1,*}, Bambang Robiin, S.T., M.T.^{b,2}

^a Program Studi S1 Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan, Indonesia

¹ khindy2100018062@webmail.uad.ac.id; ² bambang.robiiin@tif.uad.ac.id

* Penulis Korespondensi

ABSTRAK

Penelitian ini bertujuan untuk meningkatkan performa game edukasi matematika Mazetics pada perangkat mobile yang memiliki keterbatasan daya dan kemampuan rendering. Versi sebelumnya mengalami masalah performa, khususnya akibat penggunaan aset 3D yang belum optimal dan minimnya teknik optimasi. Optimasi dilakukan untuk meningkatkan FPS dan menurunkan beban GPU dengan menerapkan tiga teknik utama, yaitu Occlusion Culling (teknik optimasi dengan menyembunyikan objek yang tidak terlihat oleh kamera), Render Scale (teknik optimasi dengan menurunkan resolusi render internal lalu resolusinya akan ditingkatkan menjadi lebih tinggi), dan Vertex Decimation (mengurangi jumlah poligon pada model 3D tanpa mengubah bentuk secara signifikan). Metode penelitian yang digunakan adalah Research and Development (R&D) dengan pendekatan kuantitatif untuk membandingkan performa sebelum dan sesudah optimasi. Pengujian dilakukan menggunakan Unity Stat Profiler dengan parameter Draw Calls, jumlah Vertices, dan FPS. Hasil penelitian menunjukkan peningkatan performa yang signifikan, dengan kenaikan FPS berkisar antara 43 FPS (24,86%) hingga 174 FPS (167,05%) serta penurunan beban pada sebagian besar parameter rendering. Hasil ini menunjukkan bahwa penerapan teknik optimasi rendering efektif dalam meningkatkan performa game edukasi pada perangkat mobile.

Riwayat Artikel

Diterima 8 Januari 2026

Diperbaiki 14 Januari 2026

Diterbitkan 10 Februari 2026

Kata Kunci

Unity Optimization Graphic

Occlusion Culling

Game

Android

Vertex Decimation



This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

1. Pendahuluan

Dalam beberapa tahun terakhir, game mobile telah menjadi salah satu bentuk permainan yang paling banyak dimainkan, sehingga kebutuhan akan game yang menarik secara visual namun tetap ramah performa semakin menjadi salah satu aspek yang penting untuk diperhatikan. Meskipun perangkat mobile terus berkembang, pengembang seringkali masih menghadapi keterbatasan dalam hal kapasitas daya dan kinerja rendering[1], terlepas dari teknologi game yang saat ini berkembang pesat, dengan banyak pengembang yang kini menggunakan grafis 3D, optimasi tetap diperlukan seiring dengan kemajuan teknologi visualisasi [2]

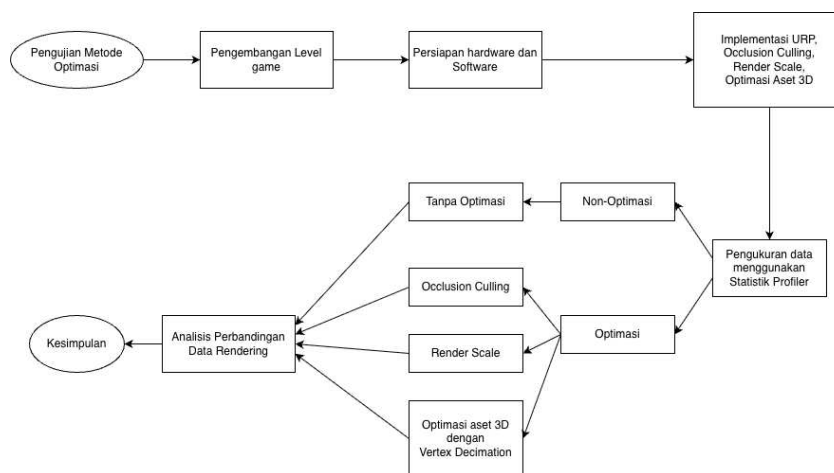
Salah satu game yang pernah dikembangkan sebelumnya yaitu Mazetics, game ini terdiri dari 15 level, namun setelah dilakukan observasi saat pengembangan, game ini masih terdapat kekurangan pada sisi performa seperti aset 3D yang belum optimal, serta kurangnya teknik optimasi yang diterapkan sehingga memberikan ruang untuk penambahan optimasi, maka penelitian ini berfokus untuk meningkatkan performa pada game Mazetics, khususnya untuk peningkatan Frame Per Second (FPS) dan mengurangi beban kerja GPU, penelitian ini melakukan optimasi menggunakan teknik Occlusion Culling[2], [3], Render Scale[4], dan Vertex Decimation[5] pada aset 3D untuk melakukan optimasi game secara keseluruhan. Teknik Occlusion Culling bekerja dengan hanya merender objek yang terlihat oleh kamera, sementara objek yang tidak terlihat atau tertutup oleh objek lain tidak dirender, sehingga mengurangi beban proses rendering pada CPU dan GPU. Teknik ini bekerja dengan optimal pada game Mazetics dikarenakan lokasi Scene yang kecil dan konsep game labirin yang membuat objek banyak terpisah oleh game objek lainnya, seperti contoh game Mazetic merupakan game indoor dengan banyak objek yang dipisahkan oleh objek tembok [6]. Render Scale

merupakan proses dimana rendering dilakukan pada resolusi yang lebih rendah untuk mengurangi beban komputasi, untuk nantinya kualitas gambar ditingkatkan kembali melalui metode machine learning, pada optimasi game Mazetic, teknologi FSR dipilih sebagai render scale yang digunakan[4]. Vertex Decimation merupakan proses optimasi aset 3D pada game yaitu dengan proses penyederhanaan model 3D yang telah banyak digunakan karena kesederhanaan dan kemampuannya untuk melakukan optimasi pada objek 3D tanpa harus menghilangkan banyak detail pada objek tersebut dimana pada metode optimasi vertex decimation tersebut berjalan dengan mencari 3 titik vertex yang membentuk segitiga, lalu menghapus vertex, lalu dimunculkan face baru dari garis yang berdekatan dengan vertex sebelumnya, sehingga mengurangi jumlah vertex pada objek tanpa harus mengurangi detail pada objek karakter tersebut[5].

Dengan penerapan tiga teknik optimasi tersebut, penelitian ini tidak hanya berfokus pada peningkatan performa game Mazetics, tetapi juga menawarkan kontribusi berupa penerapan kombinasi Occlusion Culling, Render Scale (FSR), dan Vertex Decimation secara bersamaan pada game edukasi labirin berbasis mobile, yang masih jarang dibahas dalam penelitian sebelumnya. Evaluasi dilakukan secara kuantitatif menggunakan parameter rendering sehingga hasil penelitian dapat menjadi acuan praktis bagi pengembang game edukasi berbasis Unity dalam melakukan optimasi performa pada perangkat berspesifikasi rendah.

2. Metode

Penelitian ini menggunakan metode R&D dengan pendekatan kuantitatif eksperimental melalui dua skenario pengujian, yaitu versi non-optimasi dan versi setelah penerapan optimasi menggunakan Occlusion Culling, Render Scale, dan Vertex Decimation. Data pengujian diperoleh menggunakan Unity Stat Profiler. Berdasarkan alur pada Gambar 1, kedua versi diuji menggunakan parameter rendering yang sama, kemudian hasilnya dibandingkan untuk mengetahui peningkatan performa yang dihasilkan.



Gambar 1 Kerangka Penelitian

2.1. Variabel Stat Profiler

Profiler adalah alat pada Unity untuk menampilkan data kinerja game, seperti penggunaan data rendering, jumlah Draw Calls, waktu render, dan lainnya[7]. Informasi ini membantu mengidentifikasi bottleneck dan melakukan optimasi yang diperlukan. Berikut aspek yang dibandingkan pada Profiler:

- *Draw Calls*: Jumlah permintaan kepada GPU untuk merender objek.
- *Batches*: Berapa banyak draw calls yang berhasil digabungkan menjadi satu batch untuk efisiensi rendering.

- *SetPass Calls*: Jumlah panggilan yang dibuat untuk mengatur keadaan render seperti material, shader, dan pengaturan lainnya.
- *Triangles*: Jumlah segitiga yang diterima oleh GPU untuk dirender.
- *Vertices*: Jumlah titik dalam ruang tiga dimensi yang diterima oleh GPU untuk dirender.
- *Shadow Casters*: Objek-objek yang melemparkan bayangan di dalam scene.
- *Used Texture* : jumlah dan jenis tekstur yang digunakan dalam proses rendering selama sesi profiling. Ini termasuk tekstur-tekstur yang diterapkan ke objek
- *Render Texture*: tekstur yang digunakan sebagai target render untuk operasi-render tertentu dalam aplikasi
- *Used Buffer* : penggunaan buffer dalam proses rendering. Buffer dalam konteks ini bisa merujuk pada berbagai jenis buffer, termasuk buffer geometri, buffer warna, dan buffer lainnya yang diperlukan untuk menyimpan data dan state selama proses rendering.
- *Vertex Buffer* : tipe buffer khusus yang menyimpan data titik-titik dalam ruang tiga dimensi yang akan dirender
- *Frame Per Second (FPS)* : Banyaknya output frame per detik yang didapatkan pada saat game dijalankan

2.2. Spesifikasi Perangkat

Tabel 1 Spesifikasi Perangkat Keras

| Komponen | Level 16 |
|----------|---------------------|
| CPU | Apple M2 8-core CPU |
| GPU | Apple M2 8-core GPU |
| Memory | 8 GB Unified Memory |

Tabel 2 Spesifikasi Perangkat Lunak

| Aplikasi | Level 16 |
|----------|-----------------|
| Unity | 2022.3.62f1 LTS |
| Blender | Blender 4 LTS |

3. Hasil dan Pembahasan

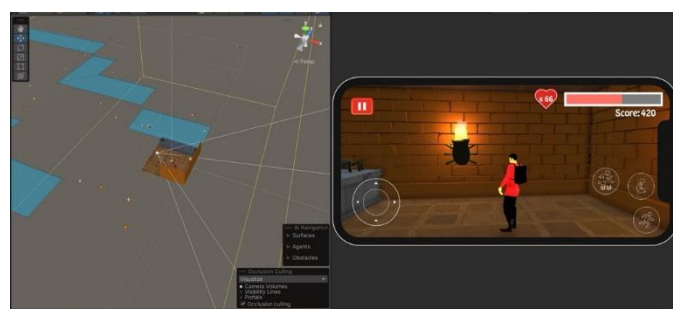
3.1. Penerapan Optimasi

3.1.1. Occlusion Culling

Occlusion Culling adalah teknik yang digunakan dalam melakukan optimasi pada game yang berpotensi untuk mengurangi beban kerja saat proses rendering game yang berpotensi meningkatkan jumlah FPS dengan cara kerja menyembunyikan gameobject yang tidak terlihat oleh kamera utama atau tertutup oleh gameobject lainnya sehingga gameobject tersebut tidak dirender oleh game (culling) [4], [6], sehingga GPU akan melakukan render objek yang lebih sedikit terbatas hanya oleh objek yang terlihat oleh kamera saja yang berpengaruh pada penurunan beban kerja GPU dan CPU[8] dengan menurunkan jumlah Draw Calls, Batches, SetPass Calls, Triangles, dan Vertices, contoh penerapan dapat dilihat pada gambar 2 dan gambar 3.



Gambar 1 Contoh Occlusion Culling



Gambar 2 Contoh Occlusion Culling

3.1.2. Render Scale

Render Scale atau Image Upscaling yaitu teknik optimasi dengan meningkatkan kualitas gambar dari resolusi rendah ke resolusi tinggi. Teknik ini memungkinkan proses rendering dilakukan pada resolusi yang lebih rendah untuk mengurangi beban komputasi[9], untuk nantinya kualitas gambar ditingkatkan kembali melalui metode machine learning. Beberapa teknologi render scale super resolution yaitu Nvidia DLSS, Intel Xess, TAA (Temporal AA Upscaling), dan AMD FidelityFX Super Resolution (FSR)[10] yang banyak dimanfaatkan untuk mencapai kualitas visual mendekati resolusi asli dengan penggunaan komputasi yang lebih rendah[4], pada optimasi kali ini pilihan render scale yang digunakan yaitu FSR 1.0 karena tersedia di game engine Unity dan bersifat open source [11], penerapan Render Scale dapat membantu mengurangi beban kerja GPU[12] karena melakukan rendering game pada resolusi yang relatif lebih rendah dari resolusi asli layar, contoh penerapan dapat dilihat pada gambar 4, 5, dan 6.



Gambar 3 Render Scale pada Skala 0.1



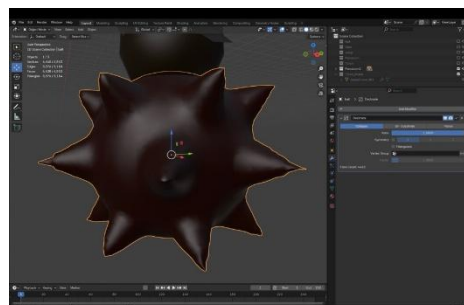
Gambar 4 Render Scale pada Skala 0.7



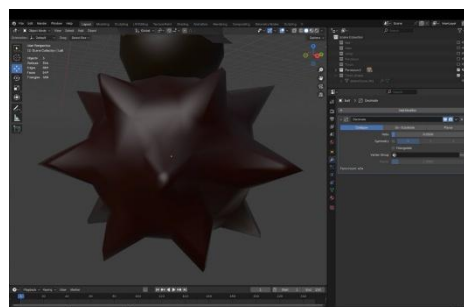
Gambar 5 Render Scale pada Skala 1

3.1.3. Vertex Decimation

Vertex Decimation adalah metode optimasi untuk melakukan optimasi secara cepat dan mudah[13] pada objek 3D tanpa harus menghilangkan banyak detail pada objek tersebut[14][15] dimana pada metode optimasi vertex decimation berjalan dengan mencari 3 titik vertex yang membentuk segitiga, lalu menghapus vertex, lalu dimunculkan face baru dari garis-garis yang berdekatan dengan vertex sebelumnya untuk menghasilkan jumlah vertex yang lebih sedikit namun hanya sedikit mengurangi detail dari objek 3D [5], Penerapan Vertex Decimation membantu dalam mengurangi jumlah Triangles dan Vertices pada objek 3D dengan resolusi tinggi sehingga mengurangi beban pada GPU, Vertex Decimation juga dapat membantu Occlusion Culling dengan mengurangi jumlah Triangles dan Vertices secara signifikan dengan membantu mengurangi jumlah Triangles dan Vertices.



Gambar 6 Aset 3D tanpa Vertex Decimation



Gambar 7 Aset 3D dengan Vertex Decimation

3.2. Pengambilan Sampel

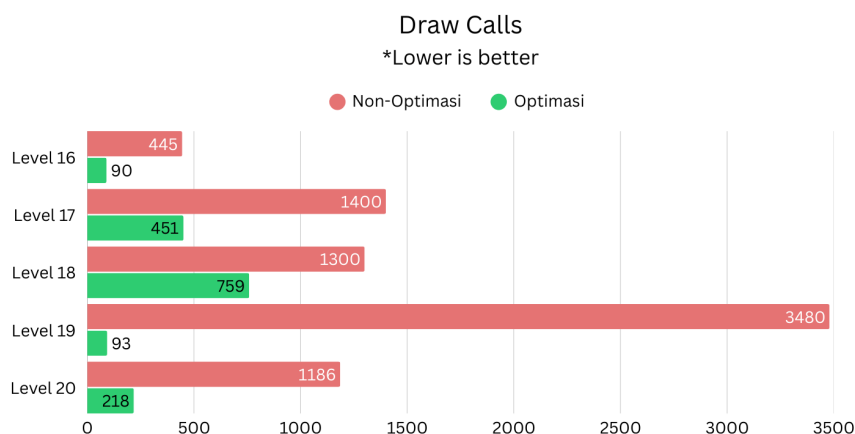
Pengambilan sampel dilakukan dengan mengambil data profiler pada level 16 sampai level 20 untuk masing-masing versi yaitu versi non-optimasi dan versi optimasi, data diambil pada saat awal scene level dijalankan dengan resolusi 2000x1149, opsi render scale default pada nilai 0.7, tanpa opsi grafis SSAO, data pada profiler akan diambil tiga kali untuk kemudian diambil rata-ratanya dengan hasil dapat dilihat pada tabel 3 dan tabel 4 dengan data penting yang berpengaruh pada performa game dapat dilihat pada gambar 9 sampai gambar 13.

Tabel 3 Data Profiler Rendering *Non Optimasi*

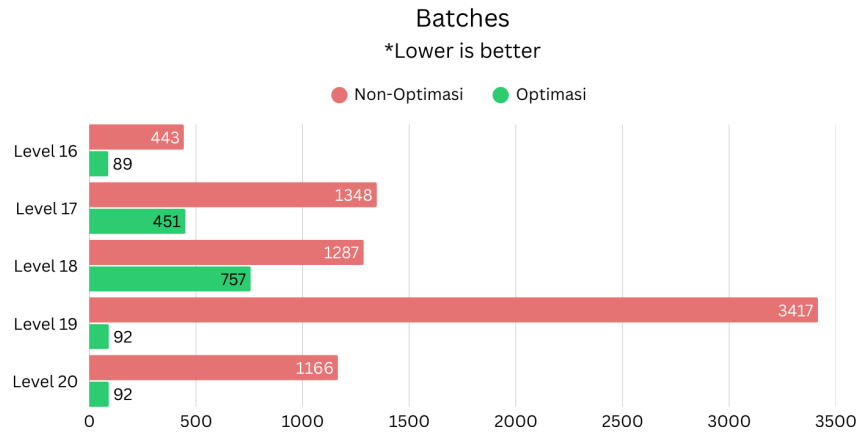
| Variabel | Level 16 | Level 17 | Level 18 | Level 19 | Level 20 |
|---------------|---------------|--------------|-------------|--------------|-------------|
| Draw Calls | 445 | 1400 | 1300 | 3480 | 1186 |
| Batches | 443 | 1348 | 1287 | 3417 | 1166 |
| SetPass Calls | 348 | 1016 | 983 | 2825 | 1089 |
| Triangles | 896.1K | 2.7M | 2.4M | 7.1M | 2.2M |
| Vertices | 834.1K | 1.8M | 1.6M | 4.8M | 1.3M |
| ShadowCaster | 82 | 289 | 258 | 535 | 234 |
| UsedTexture | 30 / 3.8 MB | 28 / 3.3 MB | 27 / 3.3MB | 33 / 3.7 MB | 30 / 3.4MB |
| UsedBuffer | 910 / 15.7 MB | 227 / 4.0 MB | 721 / 4.2MB | 731 / 8.7 MB | 248 / 4.1MB |
| VertexBuffer | 4 / 78 KB | 3 / 0.7 MB | 3 / 0.6MB | 3 / 1.9 MB | 3 / 0.4MB |
| FPS | 170.7 | 136.8 | 134.1 | 103.9 | 117.8 |

Tabel 4 Data Profile Rendering *Optimasi*

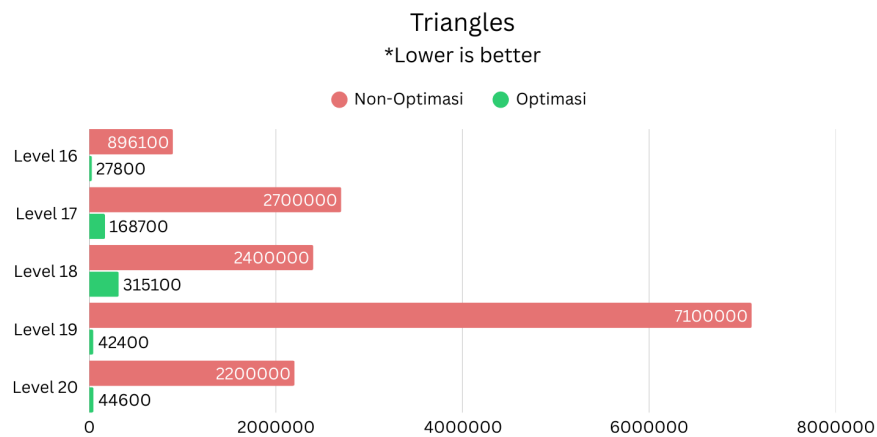
| Variabel | Level 16 | Level 17 | Level 18 | Level 19 | Level 20 |
|---------------|--------------|-------------|-------------|--------------|-------------|
| Draw Calls | 90 | 451 | 759 | 93 | 218 |
| Batches | 89 | 451 | 757 | 92 | 209 |
| SetPass Calls | 12 | 29 | 35 | 10 | 13 |
| Triangles | 27.8K | 168.7K | 315.1K | 42.4 K | 44.6K |
| Vertices | 46.5K | 221.1K | 514.5K | 87.0 K | 70.6K |
| ShadowCaster | 0 | 289 | 456 | 0 | 0 |
| UsedTexture | 27 / 3.2MB | 28 / 2.7MB | 26 / 3.0MB | 21/2.8 MB | 26 / 3.0MB |
| UsedBuffer | 943 / 12.7MB | 272 / 2.9MB | 370 / 5.2MB | 370 / 4.2MB | 309 / 3.8MB |
| VertexBuffer | 3/65.4KB | 2 / 384.8KB | 2 / 263.3KB | 2 / 247.4 KB | 2 / 223.5KB |
| FPS | 213.1 | 261.7 | 250.8 | 277.4 | 215.9 |



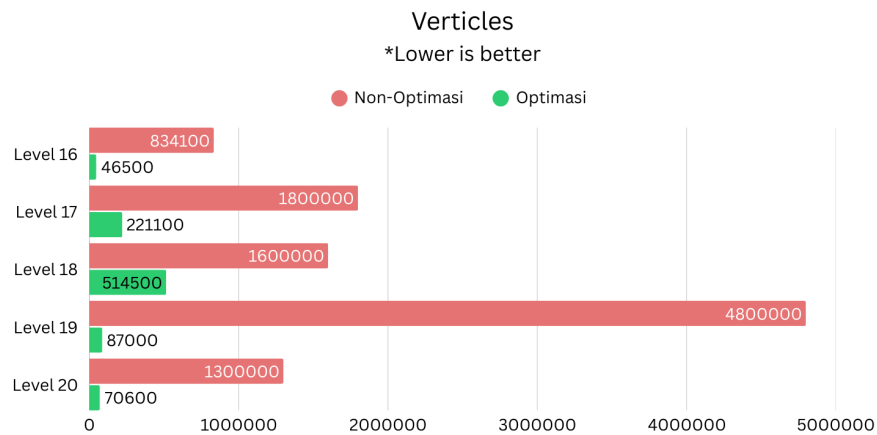
Gambar 8 Grafik perbandingan pada variabel Draw Calls



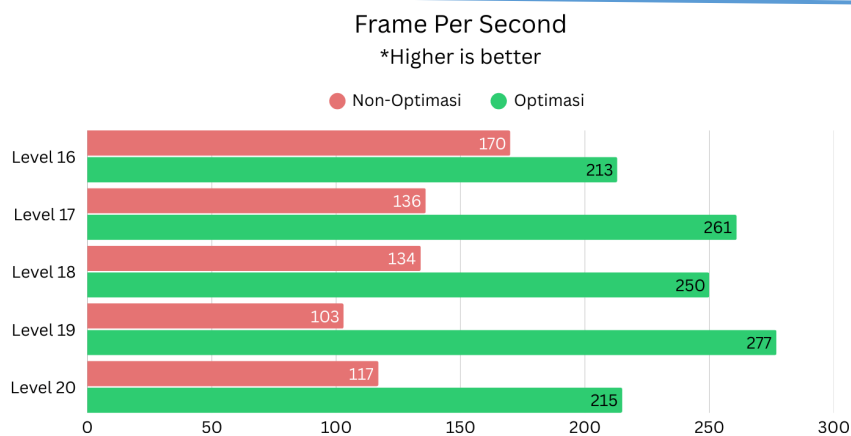
Gambar 9 Grafik perbandingan pada variabel Batches



Gambar 10 Grafik perbandingan pada variabel Triangles



Gambar 11 Grafik perbandingan pada variabel Vertices



Gambar 12 Grafik perbandingan pada variabel FPS

Berdasarkan hasil pengujian, teknik Occlusion Culling dibantu dengan Render Scale dan optimasi Aset 3D dengan Vertex Decimation memberikan peningkatan kinerja yang signifikan dibandingkan kondisi Non Optimasi. FPS naik dapat yang paling rendah terjadi peningkatan sebesar 43 FPS (+24,86%) pada level 16 hingga yang tertinggi peningkatan sebesar 174 FPS (+167,05%) pada level 19, peningkatan secara signifikan ini terjadi karena menunjukkan peningkatan kelancaran permainan secara jelas.

Secara umum, mayoritas parameter rendering mengalami penurunan beban yang cukup signifikan sehingga membuktikan bahwa Occlusion Culling efektif dalam mengoptimalkan penggunaan CPU dan GPU dengan dampaknya permainan menjadi lebih stabil, responsif, dan berjalan lebih lancar, sesuai dengan hasil dari penelitian sebelumnya[2] serta terjadi peningkatan yang lebih baik yaitu sekitar lebih dari 7% pada tiap data pengujian.

4. Kesimpulan

Dari hasil penelitian yang dilakukan, dapat disimpulkan bahwa teknik optimasi dengan Occlusion Culling yang dibantu oleh Render Scale dan Optimasi Aset 3D (Vertex Decimation) memberikan peningkatan performa yang signifikan. Peningkatan ini selaras dengan penelitian sebelumnya[2], [3], dan terbukti memberikan peningkatan lebih dari 7% pada semua parameter kunci yang dianalisis, yaitu Draw Calls, Batches, SetPass Calls, Triangles, Vertices, UsedTexture, UsedBuffer, VertexBuffer, dan FPS. Hasil ini mengindikasikan bahwa penggunaan kombinasi teknik optimasi tersebut efektif dalam meningkatkan performa game dan mengurangi beban proses rendering pada CPU dan GPU yang ditandai dengan meningkatnya angka Frame Per Second dan membuat game lebih lancar saat dimainkan. Hal ini membuat teknik penerapan Occlusion Culling, Render Scale, dan Vertex Decimation sangat disarankan dalam pengembangan game, Namun, penelitian ini masih terbatas pada pengujian pada satu jenis perangkat, satu game engine yaitu Unity game engine, dan satu metode upscaling yaitu FSR 1.0, sehingga penelitian selanjutnya disarankan untuk melakukan pengujian pada spesifikasi perangkat yang berbeda serta menggunakan engine lain, maupun metode upscaling lain untuk dapat menguji konsistensi hasil.

Daftar Pustaka

- [1] Nicolette Orispää, "Nicolette Orispää Optimization in Mobile Game Development: Leveraging 2.5D Visuals for Performance and Aesthetic Appeal," Apr. 2025.
- [2] A. Prima Bayu Kusuma, R. Riwinoto, T. Rekayasa Multimedia, and P. Negeri Batam, "Analisis Perbandingan Teknik Optimasi Occlusion Culling & LOD Group Pada Game VR Mahakarya Vokasi Dengan Perangkat VR Oculus Quest 2," 2024. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAMN>

- [3] F. Anderson Simatupang and D. Amalia Purnamasari, "Optimasi Level of Detail dan Occlusion Culling dalam Game Aaron Lost in the Jungle," *Journal of Applied Multimedia and Networking (JAMN)*, vol. 8, no. 2, 2024, [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAMN>
- [4] K. Huang, X. Yin, T. Gu, and W. Gao, "Perceptual-Centric Image Super-Resolution using Heterogeneous Processors on Mobile Devices," in *ACM MobiCom 2024 - Proceedings of the 30th International Conference on Mobile Computing and Networking*, Association for Computing Machinery, Inc, Dec. 2024, pp. 1361–1376. doi: 10.1145/3636534.3690698.
- [5] D. Aldi Tri Buana putra, A. K. Adisusilo, and N. I. Prasetya, "Optimasi Aset dan Karakter Permainan 3D Berbasis Tematik Sekolah Dasar," *Journal of Information System, Graphics, Hospitality and Technology*, vol. 4, no. 01, pp. 1–6, Mar. 2022, doi: 10.37823/insight.v4i01.165.
- [6] Unity Technologies, "Unity - Manual: Occlusion Culling," Unity Documentation. Accessed: Dec. 09, 2025. [Online]. Available: <https://docs.unity3d.com/6000.3/Documentation/Manual/OcclusionCulling.html>
- [7] Unity Technologies, "Unity - Manual: The Rendering Statistics window."
- [8] T. Jiang, S. Huang, and Z. Yan, "Rendering Optimization of Building Information Models Based on Occlusion Culling," 2025, pp. 358–368. doi: 10.2991/978-94-6463-656-7_35.
- [9] D. Ugur, I. Amer, and M. Hefceda, "Decoupling Video Upscaling from Rendering for Cloud Gaming," in *MMSys 2025 - Proceedings of the 16th ACM Multimedia Systems Conference*, Association for Computing Machinery, Inc, Mar. 2025, pp. 68–78. doi: 10.1145/3712676.3714439.
- [10] D. Ugur, "Transparent Video Upscaling for Cloud Gaming," 2024.
- [11] Advanced Micro Devices (AMD), "AMD FIDELITYFX SUPER RESOLUTION 1.0 2 OVERVIEW," 2021. Accessed: Jan. 08, 2026. [Online]. Available: <https://gpuopen.com/fidelityfx-superresolution/>
- [12] M. Berg, "Energy implications of upscaling and frame generation in games."
- [13] B. Bissell, M. Morris, E. Shaffer, M. Tetzlaff, and S. Berrier, "Vessel: A cultural heritage game for entertainment," in *Archiving 2021 - Final Program and Proceedings*, Society for Imaging Science and Technology, 2021, pp. 1–6. doi: 10.2352/issn.2168-3204.2021.1.0.2.
- [14] B. Jiménez, "Degree title Bachelor of Culture and Arts Author(s) Georgi Babushkin Thesis title Analysing pipeline for a realistic 3D game-ready weapon Commissioned by-Year 2023 Pages 40 pages."
- [15] P. Vuoristo, "Degree title Bachelor of Culture and Arts Author(s) Ville Ruti Thesis title Optimization of 3D Models for Mobile Games: Balancing Quality and Performance Year 2024 Pages 40 pages, 5 pages of appendices."