

## Systematic Literature Review: Analisis Teknik Optimasi terhadap Kinerja Game

Aditya Dwi Erlangga<sup>1</sup>, Jusuf Viandra Prasetyoadi<sup>2</sup>, Deva Afandi Maulana<sup>3</sup>

<sup>1,2,3</sup> Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Purwokerto, Indonesia  
surel: <sup>1</sup>[22sa31a012@mhs.amikompurwokerto.ac.id](mailto:22sa31a012@mhs.amikompurwokerto.ac.id); <sup>2</sup>[22sa31a014@mhs.amikomputwokerto.ac.id](mailto:22sa31a014@mhs.amikomputwokerto.ac.id),  
<sup>3</sup>[22sa31a033@mhs.amikompurwokerto.ac.id](mailto:22sa31a033@mhs.amikompurwokerto.ac.id)

### Info Artikel

#### Sejarah artikel:

Diterima 25-01-2026

Revisi 08-02-2026

Diterima 18-02-2026

#### Kata kunci:

Optimasi Game

Kinerja Game

Systematic Literature Review

Rendering

FPS

### ABSTRAK

Perkembangan industri game dalam lima tahun terakhir mendorong peningkatan kualitas grafis dan kompleksitas lingkungan permainan, namun hal ini juga menimbulkan tantangan dalam menjaga performa agar tetap stabil pada berbagai perangkat, termasuk yang memiliki keterbatasan sumber daya. Masalah seperti penurunan frame rate, meningkatnya waktu rendering, serta tingginya beban kerja CPU dan GPU menjadi hambatan utama dalam proses pengembangan game. Berbagai teknik optimasi seperti Level of Detail (LOD), Occlusion Culling, pengurangan draw-call, pemrosesan berbasis GPU, dan multithreading telah diterapkan untuk mengurangi beban render tanpa menurunkan kualitas visual. Namun, penelitian sebelumnya masih bersifat terpisah dan berfokus pada studi kasus tertentu, sehingga belum tersedia gambaran menyeluruh mengenai efektivitas setiap teknik optimasi. Penelitian ini menggunakan metode Systematic Literature Review (SLR) untuk mengumpulkan, menilai, dan merangkum hasil penelitian terkait teknik optimasi game yang diterbitkan pada periode 2020 hingga 2025. Melalui pendekatan ini, diperoleh pemahaman yang lebih terstruktur mengenai dampak berbagai teknik optimasi terhadap kinerja game.

### Penulis yang sesuai:

Aditya Dwi Erlangga

Program Studi Teknologi Informasi Fakultas Ilmu Komputer Universitas Amikom Purwokerto

Email: [22sa31a012@mhs.amikompurwokerto.ac.id](mailto:22sa31a012@mhs.amikompurwokerto.ac.id)

## 1. PENDAHULUAN

Perkembangan industri game dalam lima tahun terakhir menunjukkan peningkatan pesat, baik dari sisi kualitas visual maupun kompleksitas dunia permainan[1]. Game modern menuntut tampilan grafis yang realistis, interaktivitas tinggi, serta lingkungan dinamis yang terus berubah secara real-time. Namun, peningkatan kompleksitas ini membawa tantangan serius dalam menjaga performa permainan agar tetap stabil dan responsif pada berbagai platform, termasuk perangkat dengan keterbatasan sumber daya seperti ponsel dan konsol menengah[2]. Oleh karena itu, optimasi game menjadi aspek penting yang berperan langsung terhadap kenyamanan pengguna dan efisiensi kinerja sistem[3].

Permasalahan utama yang sering muncul dalam pengembangan game adalah penurunan *frame rate*, meningkatnya waktu rendering, dan tingginya beban kerja prosesor serta kartu grafis[2]. Faktor-faktor tersebut umumnya disebabkan oleh jumlah objek 3D yang besar, geometri yang kompleks, dan pengelolaan data visual yang kurang efisien[4]. Untuk mengatasi hal tersebut, pengembang menerapkan berbagai teknik optimasi seperti *Level of Detail (LOD)*, *Occlusion Culling*, *draw-call batching*, serta pemrosesan paralel berbasis GPU. Teknik-teknik ini bertujuan mengurangi beban render tanpa mengorbankan kualitas visual. Misalnya, penelitian oleh Lee *et al.* (2023) menunjukkan bahwa metode *vertex chunk-based object culling* dapat mempercepat proses render pada game dengan jumlah objek yang besar, sedangkan Li dan Nan (2021) membuktikan bahwa penyederhanaan mesh dengan menjaga fitur bentuk dapat mengurangi kompleksitas geometri tanpa menurunkan kualitas tampilan secara signifikan[3][5].

Beberapa penelitian lain juga menunjukkan hasil positif dari penerapan optimasi berbasis GPU dan algoritma culling. Ye mengembangkan metode *3D model occlusion culling* berbasis *WebGPU pipeline* yang mampu meningkatkan efisiensi rendering hingga 40% pada skenario kompleks[6], sementara Wu mengusulkan pendekatan *occluder generation* otomatis yang efektif untuk lingkungan dengan arsitektur padat[7]. Selain itu, Li memperkenalkan algoritma *hybrid occlusion culling* yang memanfaatkan *CPU* dan *GPU* secara bersamaan, menghasilkan peningkatan kinerja yang signifikan pada scene berukuran besar[8]. Meskipun demikian, hasil penelitian sebelumnya masih bersifat terpisah dan berfokus pada studi kasus tertentu, sehingga belum tersedia sintesis sistematis yang membandingkan efektivitas berbagai teknik optimasi terhadap performa game secara menyeluruh.

Meskipun berbagai teknik optimasi seperti *LOD*, *occlusion culling*, *draw-call reduction*, *GPU-based processing*, dan *multithreading* telah banyak dikembangkan, penelitian-penelitian tersebut umumnya masih berdiri sendiri, berfokus pada satu metode atau studi kasus tertentu, serta menggunakan parameter pengujian yang berbeda-beda. Kondisi ini membuat sulit untuk memahami efektivitas setiap teknik secara lengkap, terutama ketika dibandingkan dari segi peningkatan *frame rate*, efisiensi pemrosesan, atau pengurangan beban *GPU* dan *CPU*. Dengan kata lain, belum tersedia penemuan ilmiah yang menyatukan berbagai temuan penelitian tersebut ke dalam bentuk yang terstruktur dan dapat dijadikan rujukan oleh peneliti maupun pengembang game.

Berdasarkan kebutuhan tersebut, penelitian ini menggunakan metode *Systematic Literature Review (SLR)* agar peneliti dapat mengumpulkan, memahami, dan merangkum berbagai penelitian yang sudah ada dengan cara yang tersusun[9]. Metode ini juga membantu peneliti melihat pola, perbandingan, dan hasil dari berbagai pendekatan optimasi game yang telah dilakukan sebelumnya. Dengan *SLR*, penelitian ini tidak hanya mengidentifikasi teknik-teknik optimasi yang digunakan dalam pengembangan game, tetapi juga menilai pengaruhnya terhadap performa seperti *frame rate*, waktu rendering, dan penggunaan *CPU* dan *GPU*. Proses ini dilakukan melalui tahapan perumusan pertanyaan penelitian, pencarian artikel dari database menggunakan kata kunci tertentu, penyaringan artikel berdasarkan kriteria yang telah ditentukan, analisis isi artikel yang lolos seleksi, dan penyusunan hasil akhir[10]. Peninjauan literatur difokuskan pada penelitian yang terbit antara tahun 2020 hingga 2025 dan membahas teknik optimasi terhadap kinerja game.

## 2. METODE

Penelitian ini menggunakan metode *Systematic Literature Review (SLR)* dengan mengacu pada protokol *PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses)* sebagaimana dikemukakan oleh Page[11]. Protokol ini memastikan bahwa artikel yang dianalisis merupakan penelitian yang relevan dan berkualitas tinggi, sehingga hasil kajian mampu merepresentasikan perkembangan terkini terkait penerapan teknik optimasi game dan pengaruhnya terhadap kinerja permainan. Proses pencarian literatur dilakukan melalui Scopus, dengan fokus pada publikasi yang terbit pada rentang tahun 2020 hingga 2025. Kata kunci yang digunakan dalam proses pencarian meliputi *game*, *game optimization*, *rendering*, *game performance*, yang dikombinasikan menggunakan operator Boolean untuk memperluas cakupan pencarian. Dari proses tersebut, diperoleh sebanyak 267 artikel pada tahap awal identifikasi. Setelah proses identifikasi literatur, dilakukan pemeriksaan duplikasi menggunakan Rayyan. Hasil pemeriksaan awal menunjukkan terdapat 5 artikel duplikat, sehingga jumlah artikel yang dilanjutkan ke tahap penyaringan judul dan abstrak adalah 262 artikel. Setelah dilakukan penyaringan berdasarkan judul dan abstrak, jumlah artikel berkurang menjadi 43 artikel. Selanjutnya, penerapan kriteria inklusi dan eksklusi menghasilkan 10 artikel yang memenuhi kriteria untuk dianalisis lebih lanjut. Artikel yang bersifat non-ilmiah, tidak menyertakan hasil



evaluasi kinerja, atau tidak secara langsung membahas optimasi game dikeluarkan dari kajian. Setiap artikel yang terpilih dievaluasi berdasarkan tingkat relevansi terhadap topik optimasi game, kejelasan metodologi penelitian, kesesuaian metrik kinerja yang digunakan seperti *frame rate*, penggunaan CPU dan GPU, serta implikasi praktis dari hasil yang dilaporkan. Analisis data dilakukan secara deskriptif untuk mengidentifikasi pola, tren, serta keterbatasan yang dilaporkan dalam penerapan teknik optimasi game. Hasil analisis selanjutnya dirangkum dalam bentuk tabel dan dibahas untuk menjawab pertanyaan penelitian yang telah ditetapkan.

#### RESEARCH QUESTIONS (RQ)

Tinjauan literatur ini difokuskan untuk menjawab pertanyaan penelitian berikut:

**RQ1:** Teknik atau algoritma optimasi game apa saja yang digunakan dalam pengembangan game?

**RQ2:** Bagaimana pengaruh penerapan teknik optimasi game terhadap kinerja game?

**RQ3:** Apa saja keterbatasan yang dilaporkan dalam penerapan teknik optimasi game terhadap kinerja game?

#### Proses Pencarian

Proses pencarian literatur dilakukan secara sistematis melalui basis data akademik **Scopus**. Pemilihan basis data ini didasarkan pada cakupan publikasi ilmiah yang luas dan berkualitas tinggi di bidang ilmu komputer, dan pengembangan game. Untuk memastikan hasil pencarian yang relevan dan terfokus, pencarian literatur dibatasi pada artikel yang diterbitkan dalam rentang tahun 2020 hingga 2025.

Proses pencarian literatur dilakukan dengan menggunakan kombinasi kata kunci yang dihubungkan menggunakan operator Boolean AND dan OR. Kata kunci yang digunakan disesuaikan dengan topik optimasi game dan kinerja permainan, sehingga artikel yang diperoleh relevan dengan tujuan penelitian.

Proses pencarian literatur dilakukan melalui tahapan berikut:

1. Kata kunci ditentukan berdasarkan topik utama penelitian dan istilah terkait, seperti *game*, *game performance*, *game optimization*, *rendering*.
2. Kata kunci dikombinasikan menggunakan operator Boolean AND serta OR untuk mempersempit dan memfokuskan hasil pencarian.
3. Hasil pencarian awal disaring berdasarkan kesesuaian judul dan abstrak dengan fokus penelitian.
4. Proses seleksi lanjutan dilakukan untuk memastikan bahwa literatur yang dipilih membahas teknik optimasi game dan menyertakan evaluasi kinerja permainan.

Tabel 1. Sumber dan Kata Kunci

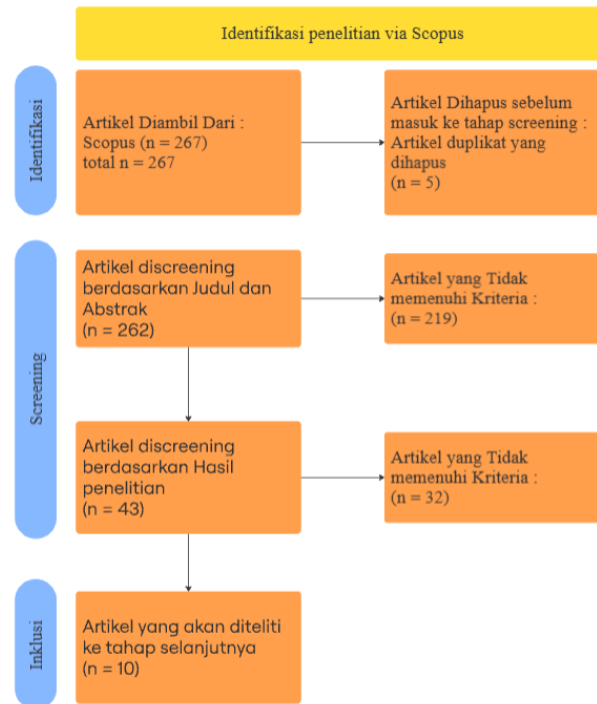
Sumber	Kata Kunci
Scopus	“Game AND game performance AND game optimization OR rendering”

Tabel 2. Kriteria Inklusi dan Eksklusi

Inklusi	Eksklusi
Artikel ilmiah yang diterbitkan pada rentang tahun 2020 sampai 2025	Artikel yang diterbitkan sebelum tahun 2020
Artikel yang membahas teknik optimasi game atau optimasi rendering pada game	Artikel yang tidak membahas optimasi game atau hanya membahas game secara umum
Artikel yang membahas pengaruh optimasi terhadap kinerja game	Artikel yang tidak menyertakan evaluasi kinerja atau metrik performa
Artikel yang menyajikan metode penelitian dan hasil eksperimen secara jelas	Artikel yang tidak menyertakan hasil eksperimen
Artikel berbahasa Inggris atau Indonesia dengan teks lengkap yang dapat diakses	Artikel dengan teks tidak lengkap atau tidak dapat diakses secara penuh



## Diagram PRISMA untuk proses seleksi literatur



Gambar 1. Tabel PRISMA

### 3. HASIL DAN PEMBAHASAN

Hasil analisis terhadap 10 artikel terpilih menunjukkan bahwa teknik optimasi berperan penting dalam meningkatkan kinerja game. Teknik yang paling banyak digunakan berfokus pada optimasi rendering, pemanfaatan GPU, dan pengendalian beban komputasi. Penggunaan optimasi shader, API grafis tingkat rendah seperti OpenGL dan DirectX 12, serta pemrosesan paralel berbasis GPU terbukti mampu meningkatkan nilai frame per second (FPS) dan menjaga stabilitas performa game, bahkan pada skenario dengan kompleksitas tinggi. Selain itu, teknik Dynamic Level of Detail (LOD) membantu mengurangi beban CPU dengan menyesuaikan tingkat detail objek secara otomatis. Optimasi pada kecerdasan buatan, seperti AI berbasis Finite State Machine (FSM), juga berkontribusi dalam menjaga kestabilan FPS. Meskipun demikian, penerapan teknik optimasi masih memiliki keterbatasan, seperti ketergantungan pada perangkat keras tertentu dan potensi penurunan kualitas visual jika tidak diterapkan secara tepat.

#### 3.1. Hasil dan Pembahasan untuk setiap pertanyaan penelitian (RQ)

RQ1 : Teknik atau algoritma optimasi game apa saja yang digunakan dalam pengembangan game?

Hasil menunjukkan bahwa teknik optimasi game sebagian besar berfokus pada peningkatan efisiensi rendering, pemanfaatan GPU, dan stabilitas kinerja real-time. Dari analisis 10 artikel terpilih, teknik optimasi terkait rendering grafis, komputasi paralel GPU, dan manajemen level-of-detail (LOD) merupakan pendekatan yang paling sering diterapkan untuk meningkatkan kinerja game. Teknik-teknik ini banyak digunakan karena dampaknya yang langsung terhadap metrik kinerja kritis seperti frame per detik (FPS), penggunaan CPU, beban kerja GPU, konsumsi memori, dan stabilitas frame.

Optimasi yang berfokus pada rendering, termasuk optimasi shader, pemilihan antarmuka pemrograman grafis (API), dan perbaikan pipa rendering, muncul sebagai kategori dominan. Beberapa studi menekankan penggunaan antarmuka pemrograman grafis tingkat rendah, seperti OpenGL dan DirectX 12, serta optimasi berbasis shader untuk mengurangi beban rendering dan meningkatkan kecepatan bingkai, terutama dalam lingkungan 3D waktu nyata. Selain itu, teknik berbasis GPU yang menggunakan compute shader dan pemrosesan paralel secara luas diterapkan

untuk memindahkan tugas komputasi dari CPU ke GPU, menghasilkan skalabilitas yang lebih baik dan kinerja real-time yang stabil dalam simulasi kompleks seperti sistem partikel dan simulasi kerumunan.

Teknik lain yang sering digunakan meliputi Level of Detail (LOD) dinamis dan optimasi mesh, yang bertujuan untuk mengurangi kompleksitas geometris berdasarkan beban sistem atau kondisi tampilan. Pendekatan ini sangat efektif dalam menjaga stabilitas kinerja dengan menyesuaikan resolusi mesh dan kompleksitas rendering secara dinamis. Selain itu, teknik optimasi yang menargetkan subsistem kecerdasan buatan (AI), seperti AI berbasis Finite State Machine (FSM), diterapkan untuk mengurangi beban komputasi yang disebabkan oleh implementasi AI yang sederhana, sehingga meningkatkan kinerja game secara keseluruhan tanpa mengorbankan perilaku gameplay.

Tabel 3. Distribusi Teknik Optimasi Game yang digunakan

Kategori teknik optimasi	Jumlah Artikel	Penerapan
Optimasi Render dan Shader	4	Real-time 3D rendering, shader performance comparison, graphics API optimization
GPU Parallel Computing and Compute Shaders	3	Particle systems, massive crowd simulation, GPU-based physics and effects
Level of Detail (LOD) and Mesh Optimization	2	Terrain rendering, dynamic mesh adjustment, CPU load-based LOD control
AI and Logic Optimization	1	FSM-based AI for 2D games, NPC behavior optimization
Total	10	

Berdasarkan Tabel 3, teknik optimasi rendering dan shader merupakan kategori yang paling dominan, dengan jumlah artikel terbanyak dibandingkan teknik optimasi lainnya. Dominasi ini menunjukkan bahwa optimasi pada pipeline rendering menjadi fokus utama dalam penelitian kinerja game, mengingat proses rendering merupakan salah satu komponen yang paling banyak mengonsumsi sumber daya sistem. Selain itu, teknik optimasi berbasis GPU juga menunjukkan tingkat penggunaan yang signifikan, yang mencerminkan meningkatnya tren pemanfaatan komputasi paralel GPU untuk menangani perhitungan kompleks secara efisien pada aplikasi real-time. Teknik optimasi Level of Detail (LOD) dan mesh tetap memiliki peran penting, khususnya pada skenario yang melibatkan lingkungan berskala besar atau rendering terrain, di mana pengendalian kompleksitas geometri sangat diperlukan untuk menjaga stabilitas frame per second (FPS). Sementara itu, teknik optimasi yang berfokus pada kecerdasan buatan dan logika game, meskipun jumlahnya lebih sedikit, tetap menunjukkan manfaat kinerja yang terukur, terutama dalam mengurangi penurunan frame rate yang disebabkan oleh implementasi logika game yang tidak efisien. Secara keseluruhan, temuan ini mengindikasikan bahwa penelitian optimasi game modern memprioritaskan teknik yang secara langsung menangani kompleksitas rendering dan beban kerja GPU, dengan optimasi tambahan pada AI dan manajemen aset sebagai pendukung peningkatan kinerja game secara menyeluruh.

RQ2: Bagaimana pengaruh penerapan teknik optimasi game terhadap kinerja game?

Hasil analisis terhadap 10 jurnal terpilih menunjukkan bahwa penerapan teknik optimasi memberikan dampak kuantitatif yang jelas terhadap kinerja game. Dampak tersebut umumnya diukur menggunakan metrik performa seperti frame per second (FPS), stabilitas frame time, penggunaan CPU dan GPU, serta kapasitas sistem dalam menangani kompleksitas game. Setiap penelitian melaporkan peningkatan performa dengan besaran yang bervariasi, tergantung pada jenis teknik optimasi dan konteks implementasinya.

Tabel 4. Dampak Teknik Optimasi terhadap Kinerja Game

Penulis & Tahun	Teknik Optimasi yang digunakan	Dampak terhadap Kinerja Game
Adeniyi et al., 2024[12]	FSM-based AI subsystem	FPS turun hingga 50 FPS pada pendekatan naive, sedangkan pendekatan FSM mempertahankan FPS stabil di 60 FPS
Zheng, 2025[13]	DirectX 12 pada Unreal Engine	Peningkatan FPS dan penurunan latensi dibanding DirectX 11 akibat pengurangan CPU overhead
Zhang et al, 2024 [14]	Mobile cloud gaming berbasis SoC cluster	Meningkatkan kepadatan sesi game hingga 4,5× dengan penurunan performa yang sangat kecil
Maciá-Lillo et al., 2025[15]	Hybrid GPU (dGPU + iGPU)	Peningkatan performa game melalui pemisahan beban rendering dan komputasi AI/physics

Lombardo et al., 2024[16]	GPU parallel computing untuk crowd simulation	Menjaga FPS stabil di 60 FPS saat mensimulasikan hingga 20.000 agen secara real-time
Minamo et al., 2020[17]	Dynamic Level of Detail (LOD) berbasis CPU usage	FPS tetap stabil dengan penurunan beban CPU, korelasi CPU-LOD mencapai 0,92
Jonemaro et al., 2025[18]	OpenGL fragment shader vs Unity surface shader	OpenGL fragment shader menghasilkan FPS lebih tinggi dan CPU usage lebih rendah, dengan GPU usage lebih tinggi
Klein et al., 2023[19]	Frame timing optimization (VFT analysis)	Penurunan smoothness terdeteksi saat variasi frame time 4-12 ms, meskipun FPS relatif konstan
Onufriienko, 2024[20]	Compute shader untuk sistem partikel adaptif	Sistem berbasis GPU mampu menangani lebih banyak partikel dengan FPS lebih tinggi dibanding CPU
Koulaxidis & Xinogalos, 2022[21]	Optimasi dasar Unity (low poly, object pooling)	Versi teroptimasi menunjukkan FPS lebih tinggi serta penurunan jumlah batch dan polygon

Berdasarkan Tabel 4, seluruh jurnal menunjukkan bahwa teknik optimasi memberikan dampak nyata dan terukur terhadap kinerja game. Optimasi rendering dan pemanfaatan GPU secara konsisten meningkatkan FPS dan stabilitas performa, terutama pada game dengan beban grafis dan komputasi tinggi, melalui pengurangan bottleneck CPU dan peningkatan skalabilitas sistem. Sementara itu, optimasi pada AI, Dynamic LOD, dan teknik dasar pada game mobile berkontribusi dalam menjaga kestabilan FPS serta efisiensi penggunaan sumber daya tanpa menurunkan kualitas pengalaman bermain. Hasil ini menegaskan bahwa dampak teknik optimasi terhadap kinerja game dapat diukur secara kuantitatif melalui peningkatan FPS, efisiensi sumber daya, dan stabilitas sistem.

RQ3: Apa saja keterbatasan yang dilaporkan dalam penerapan teknik optimasi game terhadap kinerja game?

Hasil analisis terhadap 10 jurnal terpilih menunjukkan bahwa meskipun teknik optimasi mampu meningkatkan kinerja game secara signifikan, setiap pendekatan memiliki keterbatasan dan tantangan implementasi. Keterbatasan tersebut umumnya berkaitan dengan kompleksitas pengembangan, ketergantungan pada perangkat keras tertentu, trade-off antara kualitas visual dan performa, serta keterbatasan skalabilitas pada konteks game atau platform yang berbeda. Teknik optimasi berbasis GPU dan API grafis tingkat rendah memerlukan dukungan sistem yang spesifik, sementara teknik yang lebih ringan, seperti Dynamic LOD dan optimasi dasar pada game mobile, memiliki dampak performa yang lebih terbatas dan sangat bergantung pada kondisi sistem saat runtime.

Tabel 5. Keterbatasan dan Tantangan dari penelitian

Penulis & Tahun	Teknik Optimasi yang digunakan	Keterbatasan dan tantangan
Adeniyi et al., 2024	FSM-based AI subsystem	Terbatas pada game 2D dan perilaku AI sederhana, kurang fleksibel untuk AI kompleks
Zheng, 2025	DirectX 12 pada Unreal Engine	Implementasi kompleks dan bergantung pada dukungan perangkat keras serta driver
Zhang et al, 2024	Mobile cloud gaming berbasis SoC cluster	Membutuhkan infrastruktur edge yang mahal dan kompleks
Maciá-Lillo et al., 2025	Hybrid GPU (dGPU + iGPU)	Potensi interferensi antar GPU dan kompleksitas sinkronisasi
Lombardo et al., 2024	GPU parallel computing untuk crowd simulation	Ketergantungan tinggi pada GPU, kurang optimal pada perangkat low-end
Minamo et al., 2020	Dynamic Level of Detail (LOD) berbasis CPU usage	Penurunan detail visual saat LOD diturunkan secara agresif
Jonemaro et al., 2025	OpenGL fragment shader vs Unity surface shader	GPU usage lebih tinggi, tidak efisien pada sistem dengan GPU terbatas



Klein et al., 2023	Frame timing optimization (VFT analysis)	Tidak meningkatkan performa gameplay secara langsung, hanya persepsi visual
Onufriienko, 2024	Compute shader untuk sistem partikel adaptif	Tidak semua perhitungan cocok dipindahkan ke GPU, overhead transfer data
Koulaxidis & Xinogalos, 2022	Optimasi dasar Unity (low poly, object pooling)	Dampak performa relatif kecil pada game berskala besar

Berdasarkan Tabel 6, keterbatasan utama dalam penerapan teknik optimasi game terletak pada keseimbangan antara peningkatan kinerja dan kompleksitas implementasi. Teknik optimasi berbasis GPU dan rendering menawarkan peningkatan performa yang tinggi, namun menuntut keahlian teknis serta dukungan perangkat keras yang memadai, sedangkan teknik optimasi yang lebih sederhana lebih mudah diterapkan tetapi memberikan dampak performa yang relatif kecil. Selain itu, beberapa teknik berpotensi menurunkan kualitas visual atau fleksibilitas sistem apabila diterapkan secara agresif, sehingga pemilihan teknik optimasi perlu disesuaikan dengan konteks game, target platform, dan kebutuhan performa yang diinginkan.

#### 4. KESIMPULAN

Berdasarkan hasil *Systematic Literature Review* terhadap 10 artikel terpilih, dapat disimpulkan bahwa teknik optimasi game berpengaruh signifikan terhadap peningkatan kinerja permainan. Teknik yang paling umum digunakan berfokus pada optimasi rendering, pemanfaatan GPU, dan pengelolaan level of detail (LOD), karena memberikan dampak langsung terhadap stabilitas frame per second (FPS) dan efisiensi penggunaan sumber daya. Penerapan teknik optimasi terbukti mampu meningkatkan FPS, mengurangi beban CPU dan GPU, serta menjaga stabilitas performa game. Namun, setiap teknik memiliki keterbatasan, seperti ketergantungan pada perangkat keras tertentu dan potensi trade-off antara kualitas visual dan performa. Oleh karena itu, pemilihan teknik optimasi perlu disesuaikan dengan kebutuhan game dan target platform. Penelitian selanjutnya diharapkan dapat mengembangkan teknik optimasi yang lebih adaptif untuk berbagai jenis game dan perangkat.

#### REFERENSI

- [1] Q. Zhang, "Advanced techniques and high-performance computing optimization for real-time rendering," vol. 0, pp. 14–19, 2024, doi: 10.54254/2755-2721/90/2024MELB0061.
- [2] G. Koulaxidis and S. Xinogalos, "Improving Mobile Game Performance with Basic Optimization Techniques in Unity," pp. 201–223, 2022.
- [3] M. Li and L. Nan, "Feature-preserving 3D mesh simplification for urban buildings," *ISPRS J. Photogramm. Remote Sens.*, vol. 173, no. April 2020, pp. 135–150, 2021, doi: 10.1016/j.isprsjprs.2021.01.006.
- [4] J. He, P. Wang, J. Zhu, and J. Zeng, "Application and Optimization of Mesh Simplification in 3D virtual Reality Scene in a High-performance Environment," 2024, doi: 10.4108/eai.17-11-2023.2342816.
- [5] E. S. Lee and B. S. Shin, "Vertex Chunk-Based Object Culling Method for Real-Time Rendering in Metaverse," *Electron.*, vol. 12, no. 12, pp. 1–12, 2023, doi: 10.3390/electronics12122601.
- [6] L. Ye *et al.*, "3D Model Occlusion Culling Optimization Method Based on WebGPU Computing Pipeline," *Comput. Syst. Sci. Eng.*, vol. 47, no. 2, pp. 2529–2545, 2023, doi: 10.32604/csse.2023.041488.
- [7] K. Wu, X. He, Z. Pan, and X. Gao, "Occluder Generation for Buildings in Digital Games," *Comput. Graph. Forum*, vol. 41, no. 7, pp. 205–214, 2022, doi: 10.1111/cgf.14669.
- [8] F. Li, S. Liu, N. Ma, Y. Liu, G. Xing, and Y. Zhang, "A GPU-friendly hybrid occlusion culling algorithm for large scenes," *Displays*, vol. 80, p. 102533, Dec. 2023, doi: 10.1016/j.displa.2023.102533.
- [9] H. Aïdasso and É. De, "Build Optimization : A Systematic Literature Review," vol. 1, no. 1, pp. 1–36, 2024.
- [10] P. Toukiloglou and S. Xinogalos, "A Systematic Literature Review on Adaptive Supports in Serious Games for Programming," 2023.
- [11] M. J. Page *et al.*, "RESEARCH METHODS AND REPORTING The PRISMA 2020 statement : an updated guideline for reporting systematic reviews and Meta-Analyses," 2021, doi: 10.1136/bmj.n71.
- [12] A. E. Adeniyi *et al.*, "Development of Two Dimension (2D) Game Engine with Finite State Machine (FSM) Based Artificial Intelligence (AI) Subsystem," in *Procedia Computer Science*, 2024, vol. 235, pp. 2996–3006. doi: 10.1016/j.procs.2024.04.283.
- [13] H. Zheng, "Efficient Real-Time Rendering and Optimization Using DirectX12 for Unreal Engine Graphics," in *Advances in Transdisciplinary Engineering*, 2025, vol. 75, pp. 460–470. doi: 10.3233/ATDE250803.
- [14] L. Zhang, S. Wang, and M. Xu, "High-density Mobile Cloud Gaming on Edge SoC Clusters," 2024, pp. 339–347. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85201226607&partnerID=40&md5=57d86e34fd5b0f436373b182dd2054ae>
- [15] A. Maciá-Lillo, A. Jimeno-Morenilla, H. Mora, and E. Duta-Costache, "Hybrid Architecture for AI-Based RTS Games," *IEEE Trans. Games*, vol. 17, no. 3, pp. 686–699, 2025, doi: 10.1109/TG.2025.3533949.
- [16] V. Lombardo, D. Gadia, and D. Maggiorini, "Massive Crowd Simulation with Parallel Computing on GPU," *IEEE Access*, vol. 12, pp.



- 
- 173279–173303, 2024, doi: 10.1109/ACCESS.2024.3501093.
- [17] A. E. Minamo, A. R. Agisna, W. A. Kusuma, W. Suharso, and H. Wibowo, “Optimizing Game Performance with Dynamic Level of Detail Mesh Terrain Based on CPU Usage,” 2020, pp. 93–98. doi: 10.1109/ICoIAS49312.2020.9081835.
- [18] E. M. A. Jonemaro, I. M. Agusti, H. Tolle, M. A. Akbar, and A. Pinandito, “PERFORMANCE IMPROVEMENT IN REAL-TIME 3D RENDERING USING OPENGL FRAGMENT SHADER AND UNITY SURFACE SHADER,” *J. Eng. Technol. Ind. Appl.*, vol. 11, no. 55, pp. 230–237, 2025, doi: 10.5935/jetia.v11i55.2670.
- [19] D. Klein, J. Spjut, B. Boudaoud, and J. Kim, “Variable Frame Timing Affects Perception of Smoothness in First-Person Gaming,” in *IEEE Conference on Computational Intelligence and Games, CIG*, 2024. doi: 10.1109/CoG60054.2024.10645568.
- [20] D. M. Onufriienko, “Using a compute shader for an adaptive particle system,” *Math. Model. Comput.*, vol. 11, no. 1, pp. 1–18, 2024, doi: 10.23939/mmc2024.01.001.
- [21] G. Koulaxidis and S. Xinogalos, “Improving Mobile Game Performance with Basic Optimization Techniques in Unity,” *Modelling*, vol. 3, no. 2, pp. 201–223, 2022, doi: 10.3390/modelling3020014.

