

ABSTRAK

Peningkatan kualitas visual dalam industri game mendorong bertambahnya kompleksitas geometri dan beban komputasi yang harus diproses oleh perangkat keras. Di sisi lain, perkembangan kemampuan perangkat keras tidak selalu mampu mengimbangi peningkatan tuntutan grafis, sehingga optimisasi perangkat lunak menjadi aspek penting dalam pengembangan game. Unreal Engine 5 memperkenalkan teknologi Nanite berbasis virtualized geometry sebagai alternatif terhadap teknik optimisasi konvensional. Namun, efektivitas implementasi Nanite dibandingkan pendekatan Non-Nanite masih bergantung pada karakteristik skenario yang dihadapi. Penelitian ini bertujuan membandingkan performa pipeline rendering geometri Nanite dan Non-Nanite yang menerapkan teknik Level of Detail (LOD), Occlusion Culling, dan Hierarchical Instanced Static Mesh (HISM) berdasarkan metrik frames per second (FPS), frame time, dan draw call, serta mengidentifikasi kondisi yang paling sesuai bagi masing-masing metode.

. Pengujian dilakukan pada empat skenario, yaitu lingkungan desa, hutan, pabrik, dan sci-fi, dengan tingkat kompleksitas geometri yang berbeda. Data dianalisis menggunakan statistik deskriptif, uji Mann-Whitney, dan korelasi Spearman.

Hasil penelitian menunjukkan bahwa Nanite menghasilkan FPS yang lebih tinggi pada skenario dengan kompleksitas geometri tinggi, tetapi menunjukkan performa yang lebih rendah pada skenario berkompleksitas rendah. Selain itu, Nanite menghasilkan frame time yang lebih konsisten serta jumlah draw call yang lebih rendah dibandingkan Non-Nanite. Uji Mann-Whitney menunjukkan perbedaan FPS yang signifikan secara statistik ($p < 0,001$), sedangkan analisis korelasi Spearman menunjukkan hubungan negatif yang kuat antara jumlah draw call dan FPS ($r = -0,76$; $p < 0,001$). Dengan demikian, Nanite lebih sesuai diterapkan pada lingkungan dengan kompleksitas geometri tinggi, sedangkan teknik Non-Nanite tetap efektif untuk skenario yang telah dioptimisasi secara baik sejak tahap perancangan.

Kata kunci: *Nanite, LoD, Unreal Engine 5, draw call, virtualized geometry, optimisasi grafis.*