

BAB V PENUTUP

5.1 Kesimpulan

5.1.1 Implementasi teknik *Geometry Instancing* dan *Nanite*

Implementasi HISM/PLIB dan *Nanite* terbukti meningkatkan *FPS* rata-rata lebih dari sembilan kali lipat pada kondisi terburuk (*preset Epic*), dari 8,32 *FPS* (U-1) menjadi 83,51 *FPS* (O-1), serta menurunkan *DrawPrimitive calls* rata-rata sekitar 95% pada seluruh skenario pengujian. Optimasi dilakukan dengan menerapkan *instancing* pada aset statis atau *actor* menggunakan *Hierarchical Instanced Static Mesh* (HISM) melalui fitur *Packed Level Instance Blueprint* (PLIB) yang disediakan *Unreal Engine* untuk menurunkan jumlah aktor/komponen aktif dan mengurangi *overhead* pengelolaan *scene* di sisi CPU. Selain itu, *Nanite* diterapkan pada aset statis yang memenuhi kriteria untuk meningkatkan efisiensi pemrosesan geometri di sisi GPU pada skenario *real-time*. Implementasi optimasi dilakukan tanpa mengubah skenario uji *benchmark*, sehingga perbedaan hasil merepresentasikan dampak optimasi terhadap performa.

5.1.2 Perancangan dan penerapan sistem *Graphics Scalability*

Sistem *Graphics Scalability* dirancang agar konfigurasi kualitas grafis dapat diterapkan secara konsisten pada setiap skenario pengujian. Implementasi dilakukan melalui *UI settings* yang modular, di mana WBP *CanvasSetting* bertugas menjalankan logika *apply* pengaturan, sedangkan kontrol interaksi dipisahkan menggunakan widget khusus (*toggle/selector/button*) demi menjaga modularitas dan memudahkan pengembangan UI kedepannya. *Preset* kualitas (*Epic, High, Medium, Low*) diterapkan sebagai konfigurasi yang terkontrol sehingga proses *benchmarking* dapat dilakukan secara replikatif. Melalui penerapan *preset* yang terstandarisasi ini, tren performa antar*preset* pada kondisi *optimized* dapat diamati dengan jelas, yaitu *FPS avg* meningkat dari 83,51 (*Epic*) menjadi 182,89 (*Low*), *FPS 1% Low* meningkat dari 70,10 menjadi 146,70, dan *Frame Time* rata-rata turun dari 12,05 ms menjadi 5,51 ms.

5.1.3 Besar peningkatan performa setelah optimasi

Penerapan optimasi menghasilkan peningkatan *FPS* rata-rata yang signifikan pada seluruh *preset*, dengan penurunan *Frame Time* rata-rata yang konsisten. Penurunan *DrawPrimitive calls* rata-rata sekitar 95% disetiap pengujian menunjukkan optimasi efektif mengurangi beban pengiriman *rendering*, sehingga *frame* menjadi lebih stabil dan performa meningkat. Penggunaan VRAM rata-rata menurun pada seluruh pengujian, yang mengindikasikan kondisi *optimized* lebih efisien dalam penggunaan memori grafis pada skenario uji yang sama. Utilisasi GPU rata-rata meningkat pada seluruh pengujian, yang selaras dengan peningkatan *FPS* (GPU bekerja lebih aktif ketika *frame rate* naik). Utilisasi CPU menunjukkan perubahan yang bergantung pada skenario; peningkatan pada Pengujian 1–3 dapat terjadi karena peningkatan *throughput frame* pada kondisi *optimized*, sedangkan penurunan pada Pengujian 4 menunjukkan beban CPU relatif lebih ringan pada *preset* rendah setelah optimasi.

5.2 Saran

1. Pengujian lanjutan disarankan dilakukan pada beberapa kelas perangkat (GPU/CPU berbeda dan kapasitas VRAM berbeda) untuk memetakan perubahan *bottlenecks* pada tiap *preset scalability*.
2. Pengujian sebaiknya dilakukan dengan pengulangan beberapa kali per skenario untuk memperoleh estimasi yang lebih valid dan representatif pada *FPS*, *Frame Time*, serta metrik kestabilan seperti *FPS 1% Low*.
3. Sistem *benchmark* dapat dikembangkan dengan memasukkan metrik *Draw calls* dan *RAM* secara langsung ke output CSV/JSON agar analisis kuantitatif *Draw calls* tidak hanya mengandalkan dokumentasi profiling.
4. Untuk memperkuat validasi metrik VRAM, disarankan dilakukan perbandingan lintas sumber (pembacaan internal aplikasi vs pemantauan sistem) agar interpretasi perbedaan angka penggunaan VRAM lebih akurat.
5. Pengujian selanjutnya sebaiknya membuat fitur *loading compile shaders* supaya proses kompilasi shader terjadi sebelum *benchmark/scene* dijalankan. Dengan begitu, pengujian menjadi lebih konsisten dan valid karena terhindar dari *stutter* atau penurunan *FPS* mendadak akibat shader yang masih dikompilasi saat runtime.
6. Penelitian lanjutan dapat memperluas objek penelitian ke jenis lingkungan lain seperti open-world *outdoor* atau *scene* dengan objek animasi (*Skeletal Mesh*) untuk membandingkan efektivitas HISM dan Nanite pada kondisi yang lebih bervariasi.