



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan tujuan penelitian, hasil yang didapatkan, dan analisis, terdapat beberapa kesimpulan yang dapat ditarik.

Perancangan *pipeline* kerja untuk produksi film animasi 3D “Cybercock” sudah berjalan, tetapi masih belum optimal. Terdapat banyak aspek dalam *pipeline* ini yang dapat dioptimisasi.

Hasil dari eksplorasi penghubungan Maya dengan Unreal Engine yaitu Terdapat lima tahapan utama untuk memastikan pekerjaan di Maya dapat dipindahkan dan di-*render* dengan baik dalam Unreal Engine. Tahapan tersebut berupa penyusunan *directory structure* dan *naming convention* yang baik, persiapan aset, *rigging* dan *animating*, *data export*, dan *scene rebuilding*.

Directory structure yang digunakan pada “Cybercock” adalah struktur *flat*, dengan pendekatan karakter-sentris, yang mengintegrasikan struktur *folder* ‘Cinematics’ pada struktur folder utama. *Naming convention* yang dipakai dibagi menjadi tiga rumus, yaitu aset, animasi, dan *image sequence* hasil render. Dengan ketiga rumus *naming convention* dan *directory structure*, *workflow* produksi animasi menjadi efektif dalam *game engine*.

Aset karakter, properti, dan *environment* dipersiapkan sesuai dengan kebutuhan *game engine*. Sebuah *base model* dibuat di awal sebelum proses *rigging*.

Base model ini akan digerakkan oleh animasi di tahap selanjutnya. *Base model* dan *rig* dalam Maya harus mematikan fitur *segment scale compensate*.

Proses *texturing* dilakukan seperti *pipeline texturing* biasa. Material yang di-assign dalam Maya hanya akan menjadi *material slot*. Persiapan *material* dilakukan dalam UE4 dengan pembuatan *material master* dan *material instance*.

Proses *rigging* dilakukan sesuai dengan tabel kebebasan *joint*. Proses *animating* dilakukan seperti biasa.

Data export mempunyai tiga *pipeline* utama, yaitu *camera*, *FBX joint animation*, dan *Alembic geometry cache*. Ketiga *pipeline* ini membutuhkan *bake animation*, lalu *export*. Proses ini dibantu dengan otomatisasi oleh *export tool*.

Scene rebuilding pada UE4 dijalankan dengan penempatan karakter, properti, dan *environment* sebagai *actor* di dalam *scene*. Karakter dan properti lalu melalui proses *setup anim dynamics* dan *nvcloth*. Pembuatan Anim Blueprint untuk setiap *skeletal mesh* dapat memperbaiki *playback* animasi yang tidak akurat ataupun *stuttering*. Penyusunan aset *environment* dilakukan dengan *pipeline Unreal-centric*. Selanjutnya adalah *assign animation* pada fitur *sequencer*, *lighting*, *editing*, dan akhirnya *rendering*.

5.2. Saran

Penulis mempunyai beberapa saran. Pertama, selain mempelajari Unreal Engine, pembaca juga dapat mencoba mempelajari *real-time rendering* dari *software* lain. Contohnya seperti Unity HDRP, CryEngine, dan Blender Eevee. Selain

mempelajari cara kerja dari *software* tersebut, mempelajari bahasa pemrograman yang dipakai dalam *software* tersebut juga akan sangat membantu.

Skripsi ini adalah penelitian mengenai penerapan Unreal Engine pada *pipeline rendering* animasi secara dasar. Oleh karena itu, peneliti selanjutnya dapat mengembangkan lebih lanjut mengenai topik lain yang mirip, dengan menggunakan skripsi ini sebagai acuan.

Skripsi ini juga terbatas pada penerapan dalam satu kelompok kecil. Penelitian selanjutnya dapat mempelajari penerapan *pipeline* ini dalam studio skala besar dengan jumlah anggota lebih dari 100 orang. Studio dengan skala tersebut tentu akan membutuhkan *render farm*. Di sana peneliti selanjutnya juga dapat mencoba menerapkan *pipeline render* ini menggunakan *render farm*, yang akan meningkatkan efisiensi *render* lebih jauh.

Dengan penggunaan *game engine* seperti Unreal Engine atau Unity, *project* yang dikerjakan berpotensi menjadi lebih bervariasi. Tidak hanya sebatas film animasi *pre-rendered*, tetapi juga *Virtual Reality*, *Augmented Reality*, dan lebih banyak lagi. Tidak hanya *storytelling* satu arah, tetapi juga interaktif. Ini adalah peluang bagi peneliti selanjutnya untuk dapat mempelajari teknik-teknik *storytelling* yang lain.