

## 1.2. TUJUAN PENELITIAN

Tujuan dari dibuatnya laporan skripsi ini adalah untuk memberikan analisa dari proses pembuatan *production tools* pada *workflow* kerja *rigger* di Mosmoss studio. Di balik berbagai kekurangan, penulis berharap laporan skripsi ini bisa juga menjadi bahan referensi ataupun studi kasus kepada pembaca di bidang animasi.

## 2. STUDI LITERATUR

### 2.1. Peran *rigger* di animasi 3D

Dalam industri animasi 3D terdapat banyak posisi yang berkontribusi dalam suatu proyek. Penulis sebagai *rigger* bertugas untuk menyediakan *rig* karakter yang bisa dipakai oleh *animator* dengan menyusun hirarki tulang dan *rig system* (Artaty, 2020).

Pada *workflow* yang dipakai penulis *rig* terbagi menjadi *rig system* dan *skinning* dimana di dalam *rig system* terdapat sistem yang disusun dan dibuat sedemikian rupa yang bersifat untuk mendikte bagaimana *joint* bergerak. Sistem *rig* ini seperti *forward kinematic*, *inward kinematic*, *twist joint*, *global*, *ribbon* dan seterusnya. Sedangkan pada *skinning* adalah koneksi antara seberapa besar pengaruh pergerakan *joint* dengan pergerakan *vertex* objek 3D yang dituju.

### 2.2. Python Programming Language

Aplikasi Autodesk Maya terintegrasi dengan bahasa *scripting* seperti Python dan bahasa khusus MEL (Maya embedded language). Penulis dapat menggunakan sebagian besar fitur aplikasi Autodesk Maya dengan menulis *command* dalam kaidah penulisan Python, sehingga penulis dapat menciptakan *tools* khusus yang dapat menjalankan perintah tertentu dengan prosedur yang diminta.

Kelebihan penggunaan Python sendiri sangat banyak melalui *statement* yang dikutip dari Eric (2019):

Python merupakan bahasa pemrograman yang sangat efisien, dimana program yang dibuat dapat memiliki baris kode yang jauh lebih sedikit dibanding dengan bahasa pemrograman lainnya. Python *syntax* juga memungkinkan *user* untuk menulis program menjadi lebih bersih dan mudah dibaca dan diperbaiki.

Selain dari kemudahan dalam membaca Python, fungsi bahasa ini juga sangat luas yang dapat dipakai untuk *front-end* dan *back-end developing*, *data analytics*, *web development*, *game development* dan *Artificial Intelligence*.

Dalam menggunakan Python, salah satu *library* yang dipakai penulis untuk menjalankan beberapa operasi *basic* Autodesk Maya adalah maya commands. Penulis dapat mengakses seluruh jenis *command* yang dapat dipakai pada *library* maya commands pada *website* [help.autodesk.com](http://help.autodesk.com). Pada *website* ini sudah terlampir secara lengkap dan terstruktur seluruh jenis *command* dengan detail *flag* yang dapat dipakai (Autodesk,2022).

### **2.3. PySide2 Module Framework**

PySide2 merupakan *module* resmi bahasa Python yang sangat mirip dengan *framework* QT versi 5.12 (Python, 2022). Penggunaan PySide2 bisa terintegrasi dengan UI Autodesk Maya mengingat elemen UI yang dipakai Autodesk Maya 2017 ke atas sudah menggunakan QT.

*Module* sendiri merupakan salah satu bentuk *file* yang dapat berisi *class*, *function* dan variabel yang bisa dipakai oleh *user* (Eric, 2019). Dengan adanya *module* akan sangat membantu proses penulisan kode yang repetitif ataupun kompleks, karena di dalam *module* sudah dilengkapi dengan banyak kebutuhan kode yang pada umumnya dipakai secara berulang.

Pertimbangan penulis menggunakan PySide2 dibandingkan *module* lain seperti PyQt5, dikarenakan *module* ini yang sudah ter-*install* pada versi 2020 dan lisensi penggunaan PySide2 adalah LGPL (lesser general public licensing) yang tidak mengharuskan *sharing source code* jika ingin memonetisasi *tools*

Dalam PySide2 terdapat beberapa *class* yang paling sering digunakan penulis, penggunaan ini dibedakan berdasarkan *treatment* yang diberikan *class* terkait terhadap *operation logic* yang akan dilaksanakan program. Beberapa *class* tersebut adalah, QtCore, QtWidgets, dan QtGui.

PySide2 menggunakan beberapa *class* yang berfungsi dalam mengatur *layout* dan beberapa di antaranya adalah QHBoxLayout, QVBoxLayout, QFormLayout, dan QGridLayout (Qt documentation, 2022). Penulis dapat menambahkan *layout* ini sebagai *parent* ataupun *child* dari *layout* lain, sehingga membuat *code* yang lebih *clean* dan terstruktur.

#### **2.4. Global System**

Setiap objek 3D disusun dari *Transformation nodes* dan *shape nodes*, dimana pada *transformation nodes* mengatur seberapa jauh *pivot point* dari suatu objek berpindah, berotasi dan merubah skala. Pada aplikasi Autodesk Maya objek dapat memiliki objek yang lebih tinggi dalam hirarki atau *parent object* dimana selanjutnya sifat transformasi suatu *object child* bisa kita manipulasi dengan prinsip hierarki *parent-child relationship* (O'Hailey, 2018).

Sistem global yang digunakan pada *rigging* membebaskan prinsip *parent-child* untuk mempertahankan posisi dan rotasi suatu objek tanpa pengaruh dari objek *parent* dengan menggunakan *constraint system*. Sistem global ini dapat berupa global khusus untuk rotasi atau posisi maupun keduanya sekaligus, biasanya sistem ini berguna khususnya untuk bagian seperti kepala, bahu, rambut maupun dekorasi tambahan.

Selain dari sistem global seperti *rotation* dan *translation* terdapat juga *global space swap* yang memiliki sifat yang cukup berbeda dibandingkan sistem global di atas. *Global space swap* memiliki sistem dimana objek akan mengikuti posisi dan rotasi dari *parent* yang ditentukan, dengan ini objek bisa memiliki beberapa *parent* sekaligus.

Sistem ini dibantu menggunakan *parent constraint* dimana pada *constraint* tersebut memiliki beberapa *parent* (seperti pada IKFK system). Pada *global space*