

## BAB III

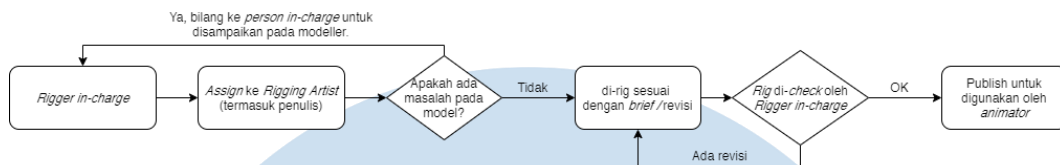
### PELAKSANAAN KERJA MAGANG

#### 3.1. Kedudukan dan Koordinasi

Penulis berkedudukan sebagai *intern rigging artist* selama praktek magang di studio animasi Lumine Studio. *Rigging* merujuk pada suatu pekerjaan menyusun suatu sistem gerak menggunakan kerangka digital pada model karakter 3D (Arshad et al., 2019). Kegiatan *rigging* menjadi sangat penting karena akan menentukan deformasi dari model yang dibuat oleh *modeler* yang secara langsung berpengaruh pada hasil akhir film. Maka, seorang *rigging artist* harus dengan teliti merancang *rig* dengan deformasi yang rapih dan tepat, sekaligus memenuhi kebutuhan *animator* untuk menggerakkan *rig* sesuai *animatic* atau *storyboard*.

Selama praktek magang, ada 4 orang *rigging artist* selain penulis di Lumine Studio. Penulis seringkali dibimbing oleh mereka, terutama oleh Kak Stefi, yang *in-charge* pada *project* serial *Little Babies*, dan Kak Ashariah, yang *in-charge* pada *project-project* lain seperti *games* dan *short movies*. Penulis berperan dalam me-*rig* berbagai *property* dan *character* pada macam-macam *project*. Terkadang penulis diberi tanggung jawab untuk me-*rig character* atau *property* pada *project* serial untuk membantu Kak Stefi. Namun, terkadang penulis juga diberi tanggung jawab untuk me-*rig character* atau *property* pada *project short movies* untuk membantu Kak Ashariah.

Di Lumine Studio, penulis tidak menjalin komunikasi langsung ke *modeler* atau *animator*. Namun, penulis berkomunikasi dengan *rigging artist* yang *in-charge* atau ditugaskan pada *project* tersebut. Jadi, tugas-tugas yang dikerjakan penulis selalu di-*assign* dan kemudian di-*review* oleh *Rigging Artist* yang bertanggung jawab atas *project*. Berikut bagan dan penjelasan lebih detail mengenai proses koordinasi yang penulis temui di Lumine Studio.



Gambar 3.1 Bagan Alur Koordinasi  
(Dokumentasi Pribadi)

Bagan di atas menggambarkan proses koordinasi yang penulis temui selama praktek magang di Lumine Studio. Berikut adalah uraian mengenai detail alur koordinasi dari bagan di atas:

1. Pihak *modelling* mengabarkan apabila *asset-asset* telah bisa mulai untuk di-*rig*.
2. *Rigging artist in-charge*, yaitu *rigging artist* yang bertanggung jawab atas *project* tersebut akan meng-*assign asset-asset* yang perlu di-*rig* kepada penulis dan *rigging artist* lainnya.
3. Apabila penulis atau *rigging artist* menemui kesalahan pada model, seperti model yang tidak simetris, model tidak berada pada titik sumbu pusat, atau dibutuhkannya loop tambahan pada model, maka penulis atau *rigging artist* akan memberitahukan kepada *rigging artist in-charge* untuk kemudian disampaikan kepada *modeller*.
4. Jika tidak ada kesalahan, maka model akan di-*rig* sesuai dengan *brief* yang diberikan oleh *rigging artist in-charge*.
5. *Rig* yang telah selesai di-*upload* pada server Lumine Studio untuk di-*review* oleh *rigging artist in-charge*.
6. Apabila ada kesalahan, maka *rigging artist in-charge* akan memberitahukan kesalahan pada *rigging artist* yang mengerjakan *rig* untuk diperbaiki. Jika tidak, *rig* akan di-*approve*.
7. Selain koordinasi searah seperti di atas, terkadang ada *update-update* yang dilakukan oleh *modeller*, seperti revisi dari *client*. Maka *modeller* akan mengabarkan *rigging artist in-charge* supaya *rig* bisa segera di-*update*.

## 3.2. Tugas dan Uraian Kerja Magang

Penulis diberi kesempatan untuk ikut serta pada berbagai macam *project* selama praktek magang di Lumine Studio. Pada awal praktek magang dilaksanakan, penulis diberikan latihan-latihan untuk menilai pemahaman penulis terhadap *rigging* sekaligus melatih penulis. Kemudian seiring waktu berjalan, penulis diberi kepercayaan untuk mengerjakan *project-project* yang ada. Berikut uraian lebih jelas mengenai semua tugas yang dipercayakan kepada penulis.

### 3.2.1. Tugas yang Dilakukan

Penulis diberi kepercayaan untuk ikut berperan dalam beberapa *project* yang sedang berlangsung di Lumine Studio. Pada *project Mary*, penulis diberi kesempatan untuk *me-rig character Robert*, *character Pigeon*, serta beberapa *property* seperti *pie\_piece* dan *flower\_pot*. Selain *project Mary*, penulis juga diberi kepercayaan untuk ikut serta pada *project serial Little Babies*. Penulis lebih sering terlibat pada *project Little Babies* karena karena padatnya *timeline project serial* tersebut. Selain itu, penulis juga berkesempatan ikut serta dalam pengembangan *script* untuk mempermudah kerja.

### 3.2.2. Uraian Kerja Magang

Dari berbagai macam *project* yang penulis ikut berperan di dalamnya, ada beberapa *project* yang penulis pilih untuk diuraikan secara lebih detail. Beberapa *project* tersebut adalah merancang *rig* sayap untuk *character Pigeon* di *project Mary* dan *men-develop script Playblast to Master* sebagai *Research and Development*. *Project-project* ini penulis pilih karena penulis memiliki peran yang cukup besar dalam mengerjakannya sehingga dapat diuraikan dengan lebih baik.

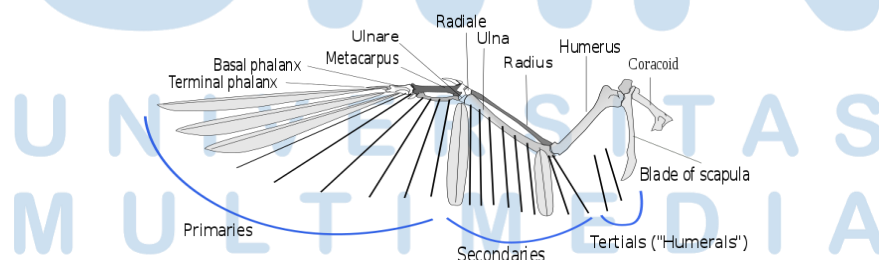
## A. Perancangan rig sayap character Pigeon pada project short movie Mary.

### 1. Briefing

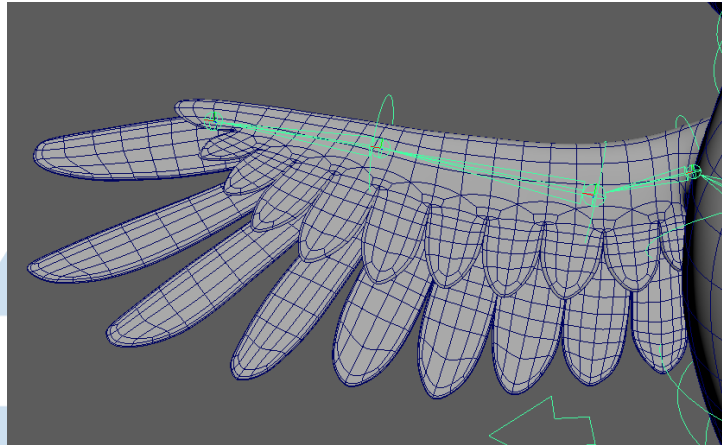
Pada perancangan *rig* sayap *character Pigeon*, penulis di-assign untuk melanjutkan *rig* yang sebelumnya telah dibuat oleh Kak Fajar (*Rigging Artist*). Sebelum melanjutkan *rig*, penulis mengecek dan menganalisa terlebih dahulu *rig character Pigeon*, mengenai sistem *rig* yang digunakan, serta bagaimana sebaiknya penulis melanjutkan *rig* tersebut. Penulis mengecek dan menganalisa *rig* dengan cara mencoba menggerakkan masing-masing *controller* dan menelaah koneksi-koneksi yang ada pada sistem *rig*. Setelah mengecek dan menganalisa, penulis memutuskan ada beberapa hal yang dapat ditingkatkan pada *rig* tersebut. Salah satunya adalah menggunakan sistem *ribbon* pada sayap sehingga bulu-bulu sayap bisa dengan otomatis membuka dan menutup saat sayap digerakkan.

### 2. Perancangan Rig Sayap Character Pigeon

Hal pertama yang penulis lakukan setelah menganalisa *rig*, yaitu memperbaiki *outliner* file maya. Kemudian dengan hati-hati dan sambil memperhatikan koneksi, menghapus rangkaian *joint* dan *controller* bagian sayap hingga menyisakan *joint* dan *controller shoulder* atau bahu. Lalu penulis membuat ulang rangkaian *joint* sayap berdasarkan referensi dari anatomi burung pada gambar 3.2. Tak hanya itu, penulis juga mereplika sistem *controller* yang digunakan pada sistem *rig* awal, sehingga tidak terdapat deformasi yang tidak diinginkan pada hasil akhir.



Gambar 3.2 *The Skeleton of bird's wing*  
(Sumber: L. Shyamal – Own Work)

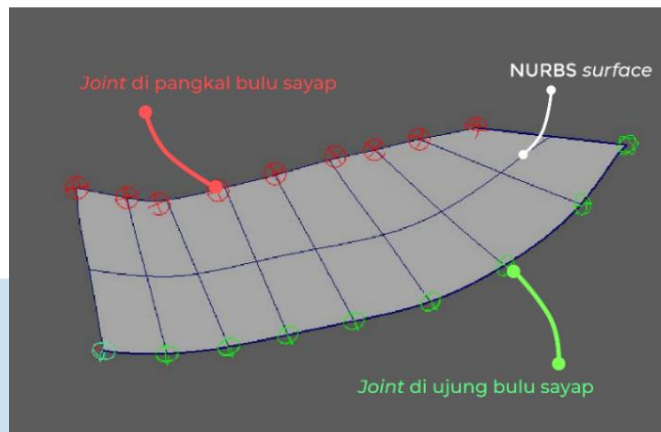


Gambar 3.3 Peletakan *joint* sayap *character pigeon*  
(Sumber: Dokumentasi Pribadi)

### 3. Aplikasi *Ribbon System* Pada Sayap *Character Pigeon*

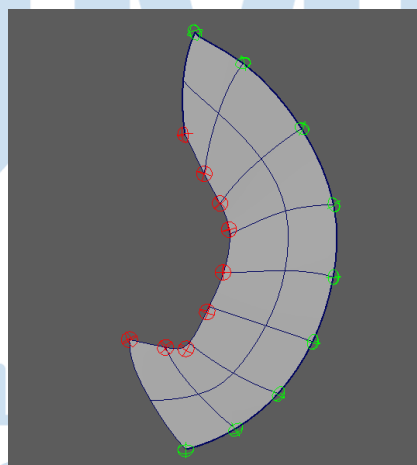
Penulis kemudian mengadaptasi sistem *ribbon* yang belum lama ini penulis pelajari dari kanal *Youtube antCGI*. Sistem *ribbon* sebenarnya dipakai untuk menciptakan *controller bend* pada lengan atau kaki *character* supaya *rig* lebih fleksibel (antCGI, n.d.). Namun penulis menyadari bahwa sistem *ribbon* bisa diadaptasi untuk menciptakan bulu sayap yang dapat mekar dan menutup secara otomatis ketika sayap dikembangkan atau ditutup.

Caranya adalah dengan memanfaatkan *NURBS surface* sebagai *base* dan *follicle* dari Maya nHair. *Follicle* memiliki karakteristik untuk menempel di suatu titik pada *surface* (Tina O’Hailey, 2018), sehingga sangat cocok untuk dijadikan *parent* dari *joint*. Dengan demikian, *joint* tersebut menempel pada *follicle* dan selalu mengikuti deformasi dari *NURBS surface*. Penulis membuat *follicle* dan *joint-joint* yang akan menempel pada *follicle* sesuai dengan jumlah dan posisi masing-masing bulu sayap *character Pigeon* dikali dua, sebaris di pangkal bulu sayap, dan sebaris lagi di ujung bulu sayap seperti pada gambar 3.4.



Gambar 3.4 Peletakkan *joint* dan *follicle* pada sistem *ribbon*  
 (Sumber : Dokumentasi pribadi)

NURBS *surface* kemudian akan di-*skinning* pada *joint* sayap *character Pigeon* supaya dapat mengikuti gerakan apabila *controller* digerakkan. Metode tersebut akan menghasilkan sistem *rig* sayap dengan *joint* tambahan di pangkal dan ujung bulu-bulu sayap. Kemudian, penulis memakai *aim constraint* pada masing-masing *joint* di pangkal bulu sayap, untuk mengarah pada *joint* di ujung sayap. Dengan demikian, *joint* yang ada pada pangkal bulu sayap akan selalu mengarah pada *joint* yang ada pada ujung sayap. Apabila ada deformasi pada NURBS *surface*, *joint-joint* yang ada pada ujung sayap akan mengikuti bentuk deformasi dari NURBS *surface* dan menyebar seperti pada gambar 3.5.

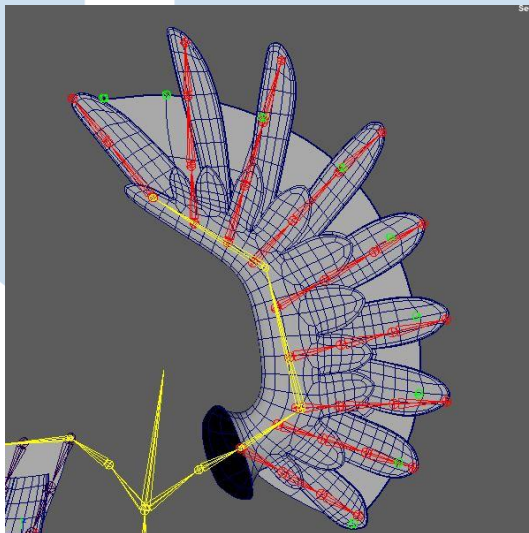


Gambar 3.5 Deformasi sayap saat mengembang.  
 (Sumber: Dokumentasi Pribadi)



#### 4. Pengaplikasian *Skin Weight* Pada Sayap *Character Pigeon*

Pada tahap *skinning*, penulis tidak menggunakan *joint* sayap awal yang ada pada gambar 3.2 untuk di-*skin* ke *geometry* sayap. Penulis menggunakan *joint* tersebut untuk memberi *influence* dan mendeformasi *NURBS surface*. Sedangkan untuk *geometry* sayap dan bulu-bulu sayap, penulis *bind skin* dengan *joint-joint* dari sistem *ribbon*, yaitu *joint-joint* berwarna merah pada gambar 3.4. Terakhir, penulis merapihkan *skinning* pada *geometry* sayap dan bulu-bulunya, hingga mendapatkan hasil akhir seperti pada gambar 3.6.



Gambar 3.6 Hasil akhir rancangan *rig* sayap  
(Sumber: Dokumentasi pribadi)

#### 5. *Feedback*

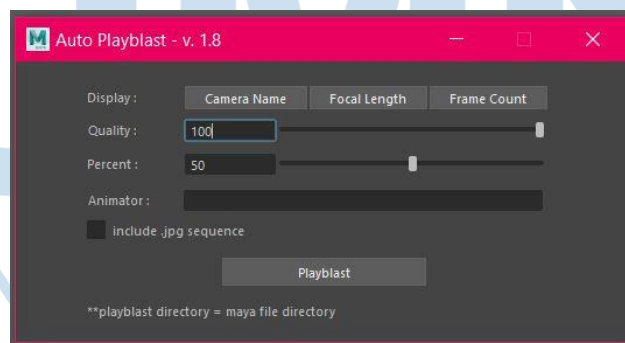
*Rig* sayap rancangan penulis cukup meKak Ashariah sebagai *rigger lead project Mary* cukup puas dengan sistem sayap yang penulis buat. Penulis sempat melakukan kesalahan, yaitu salah menargetkan beberapa *joint* dengan *aim constraint*. Namun kesalahan tersebut dapat diatasi dengan merancang ulang rangkaian *aim constraint* pada *joint* sayap. Demikian uraian *task* perancangan *rig* sayap *character Pigeon*.

## B. Pengembangan script Auto Playblast menggunakan python sebagai bagian dari Research and Development Lumine Studio.

### 1. Briefing

Pada *task* pengembangan *script Auto Playblast*, penulis ditawarkan oleh supervisi untuk mencoba *scripting*. Penulis ditugaskan untuk memperbarui *script Auto Playblast* yang telah ada di Lumine Studio menjadi *Playblast to Master*. Untuk pembaruan, penulis diminta menambahkan beberapa fitur baru pada *script Auto Playblast* yang lama. Fitur-fitur tambahan tersebut yaitu:

- *Script* dapat mendeteksi *file path* dari *file scene maya* dan secara otomatis membuat *save path* dan nama untuk *playblast* yang akan dibuat.
- *Script* dapat mendeteksi versi video *playblast* yang tersimpan dan secara otomatis membuat nama dengan versi terbaru.
- Pengguna dapat memilih untuk meng-*export* hasil *playblast* sebagai *work in progress (WIP)* atau *publish*, masing-masing memiliki *save path* atau *directory* dan penamaan yang berbeda.
- Pengguna dapat memilih format dari *playblast* yang akan dibuat.
- Pengguna dapat memilih tingkat *subdivision* dari *geometry* yang akan di-*playblast*.



Gambar 3.7 Tampilan Awal Script Auto Playblast  
(Sumber: Data Perusahaan Lumine Studio, 2021)

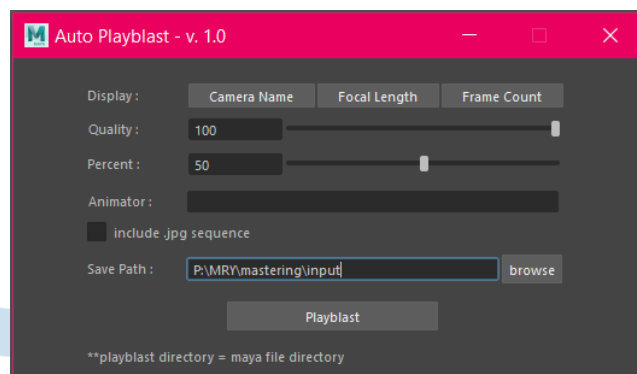


Supervisi penulis menyediakan *file Maya* kosong beserta contoh *directory* tujuan penyimpanan *playblast*. Penulis menggunakan *file* kosong dan *directory* tersebut untuk keperluan mencoba apakah *script* dapat berjalan dengan baik. Supervisi juga menyediakan beberapa contoh *script* yang memiliki fungsi yang sama untuk dipelajari penulis.

## 2. Pengembangan *Script Playblast to Master*

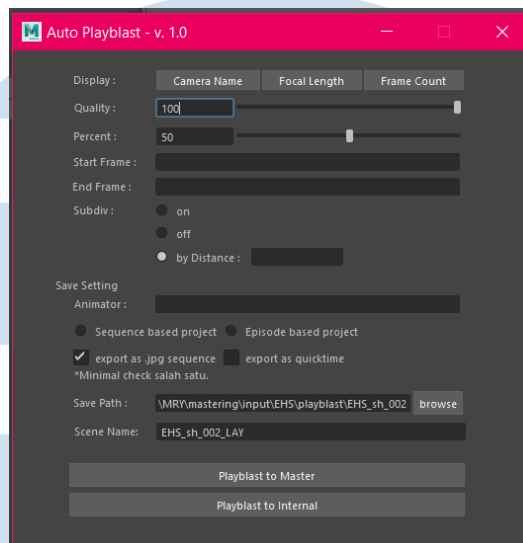
Tahap pertama yang penulis lakukan setelah mempelajari dan memahami tata penulisan *script Auto Playblast* adalah memodifikasi dan menambahkan *UI (user interface)* yang ada. Dalam memodifikasi *UI*, penulis memperhatikan fungsi dari masing-masing elemen yang ditambahkan, dan bagaimana masing-masing elemen tersebut akan bekerja nantinya. Penulis menganalisa bagaimana *script Playblast to Master* akan digunakan dengan membayangkan diri penulis sebagai pengguna. Dengan demikian, penulis dapat mengoreksi dan meningkatkan kualitas *UI script Playblast to Master* bagi penggunaannya. Berikut beberapa gambar *work in progress* dari *script Playblast to Master*:

- a. Tampilan *UI* penambahan fitur *save path*.



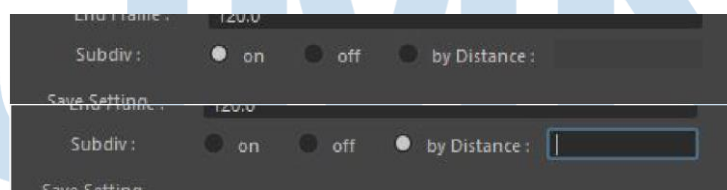
Gambar 3.8 *Work In Progress Script Playblast to Master 1*  
(Sumber: Dokumentasi Pribadi)

b. Tampilan *UI* penambahan fitur lengkap.



Gambar 3.9 *Work In Progress Script Playblast to Master 2*  
(Sumber: Dokumentasi Pribadi)

Penulis kemudian menuliskan *function* tambahan untuk membuat automasi pada *UI* yang telah dibuat. Automasi pada *UI* dibuat agar *script* dapat memberikan interaksi dengan pengguna. Dengan demikian, pengguna dapat menggunakan *script* dengan lebih tepat. Contoh interaksi tersebut yaitu pilihan “*by Distance*” pada fitur pemilihan *Subdiv*. *Dialog box* yang terdapat di sebelah pilihan “*by Distance*” hanya dapat diakses dan diisi oleh pengguna apabila pengguna memilih pilihan “*by Distance*”.



Gambar 3.10 *Dialog box “by Distance”* yang interaktif  
(Sumber: Dokumentasi Pribadi)

Setelah merancang *UI* dan *function* interaktif, penulis melanjutkan mengembangkan *function* utama dari *script Playblast to Master*. Penulis mulai dengan menuliskan *function* untuk membuat *variable* yang dapat meng-*query* atau merekam data yang pengguna *input* melalui *UI window*.

Data-data tersebut digunakan untuk menentukan secara spesifik bagaimana perintah akan dieksekusi. Kemudian, penulis memodifikasi *script* dengan menghubungkan fitur-fitur baru pada *function* yang telah ada. Maka ketika *script* dijalankan, perintah-perintah yang telah dirancang akan dijalankan berdasarkan data-data dan kustomisasi yang di-*input* pengguna pada *UI window*. Misalnya, jika pada *UI window* pengguna mencentang pilihan “*export as quicktime*”, maka *output* video *playblast* akan berupa *file quicktime* dengan kompresi *.mov*.

Terakhir, Penulis mencoba meng-*execute* atau menjalankan *script Playblast to Master*. Percobaan menjalankan *script* dilakukan untuk memastikan *script* berjalan dengan baik pada kondisi , sekaligus memahami keterbatasan *script*. Proses mencoba menjalankan *script* penulis jalankan beberapa kali selama penulisan *script*. Hal ini dilakukan penulis agar *error* dapat terdeteksi sesegera mungkin dan penulis dapat lebih mudah mengidentifikasi penyebab *error*.

### 3. Aplikasi *Script Playblast to Master*

*Script Playblast to Master* akan sering digunakan oleh para *animator* untuk mempersingkat proses pembuatan video *playblast preview* animasi. Pada saat *animator* meng-*export* hasil pekerjaan mereka berupa video *playblast* untuk di-*review*, mereka memerlukan beberapa pengaturan tambahan. Kemudian, mereka harus memindahkan video *playblast* tersebut pada *directory* yang ditentukan. Proses tersebut sangat memakan waktu, tidak efektif, serta rentan menimbulkan kesalahan. Contohnya adalah salah penamaan, salah meletakkan *directory file*, atau salah menimpa *file* penting.

*Script Playblast to Master* berperan untuk menanggulangi masalah-masalah tersebut. *Script Playblast to Master* dirancang untuk mendeteksi *directory* dari *file Maya*. Dengan demikian, *script Playblast to Master* dapat merancang *save path* dan *save name* yang sesuai dengan *directory* dan tipe *project*. Pengguna juga dapat mengkustomisasi beberapa elemen

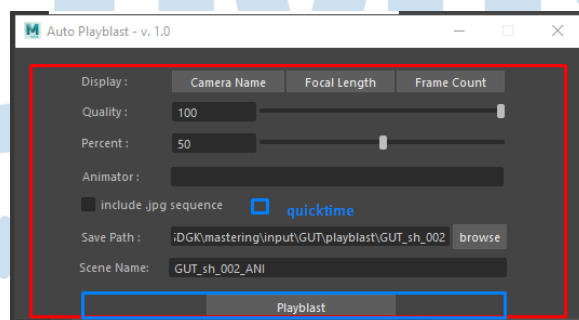
pada *playblast* seperti *quality*, *subdivision*, *format playblast*, serta kemana *playblast* akan di *export*.

*Script Playblast to Master* menyediakan dua opsi *path*. Opsi pertama adalah untuk meng-*export* video *playblast* ke folder *Master*. Opsi ini digunakan ketika animasi telah selesai dan akan digunakan untuk ditunjukkan ke *client*. Opsi kedua adalah untuk meng-*export* video *playblast* folder *review* untuk di *review* secara *internal* perusahaan.

Maka dengan *script Playblast to Master*, *animator* dapat meng-*export* video *playblast* dengan lebih efisien. Para *animator* tidak perlu membuka pengaturan untuk menyetel tampilan *playblast* karena dapat diatur langsung melalui *script*. Lalu, *animator* tidak perlu merapihkan peletakkan *file playblast* dan mengatur penamaannya. Kemudian, *animator* dapat terhindar dari kesalahan-kesalahan yang dapat terjadi karena kekeliruan atau *human error*.

#### 4. *Feedback*

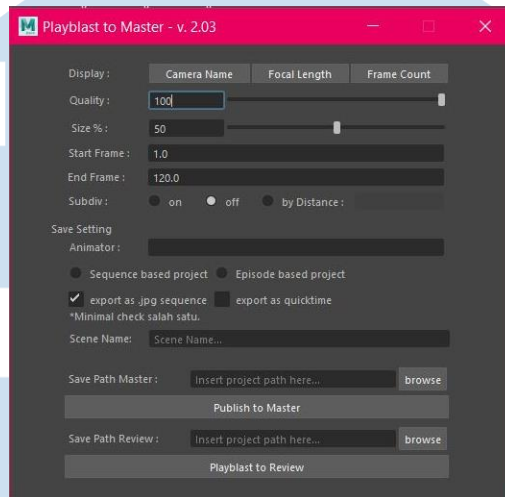
Selama proses pengembangan *script*, penulis mendapat banyak arahan dan masukan dari supervisi. Arahan dan masukan diberikan pada penulis agar pengembangan *script* dapat sesuai *pipeline* kerja Lumine Studio. Masukan dan arahan penulis terima melalui media pesan dan *screenshot* agar dapat lebih mudah dimengerti. Berikut merupakan contoh *feedback* berupa *screenshot* yang diberikan oleh supervisi:



Gambar 3.11 Contoh *Feedback* dari supervisi  
(Sumber: Dokumentasi Pribadi)

## 5. Hasil akhir

Berikut merupakan tampilan *UI* final *script Playblast to Master*:



Gambar 3.12 Tampilan *UI* final *script Playblast to Master*  
(Sumber: Dokumentasi Pribadi)

Pada hasil akhir, *script Playblast to Master* dapat diaplikasikan dan sesuai dengan *pipeline* kerja produksi. *Script Playblast to Master* sudah diaplikasikan pada *project Mary*, yang merupakan salah satu *project* Lumine Studio yang sedang berjalan. Supervisi penulis juga sempat mengkonfirmasi bahwa *script* dapat dijalankan tanpa masalah dan mampu memperbaiki *mastering input*. Maka dapat dikatakan bahwa *script Playblast to Master* berjalan dengan baik.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 3.2.3. Kendala yang Ditemukan

Selama praktek kerja magang, penulis sempat menghadapi beberapa kendala yang sebagian besar penyebabnya karena proses kerja yang dilakukan secara *full online* atau *remote*. Kendala-kendala yang sempat dihadapi penulis, yaitu:

1. Miskomunikasi antar penulis dan *rigging artist* lain rekan kerja karena komunikasi berjalan melalui media *chat* di *social media*.
2. Penulis tidak dapat memanfaatkan fasilitas studio dengan sepenuhnya karena sinyal yang kurang baik. Salah satunya adalah *Computer* yang disediakan oleh studio bagi penulis yang seharusnya diakses melalui aplikasi *Anydesk*. Namun, karena sinyal yang tidak stabil, penulis seringkali mengunduh pekerjaan dan mengerjakan *task* yang diberikan dengan laptop pribadi.
3. Sinyal internet yang tidak stabil seringkali menghambat pekerjaan. Penulis bahkan terkadang tidak dapat mengunduh *asset* model untuk di-*rig* dari *server* studio.

### 3.2.4. Solusi atas Kendala yang Ditemukan

Berdasarkan dari kendala-kendala yang penulis temukan selama proses kerja magang, berikut merupakan solusi yang terpikirkan penulis untuk mengatasi kendala-kendala tersebut, yaitu:

1. Bersifat aktif dan tidak malu untuk bertanya jika ada instruksi yang masih belum jelas. Serta, seringkali mengkonfirmasi instruksi yang diberikan rekan kerja untuk meminimalisir terjadinya salah paham.
2. Menyediakan data atau kuota internet cadangan pada perangkat ponsel supaya dapat digunakan sebagai *hotspot* ketika kecepatan internet *wi-fi* sedang lambat.