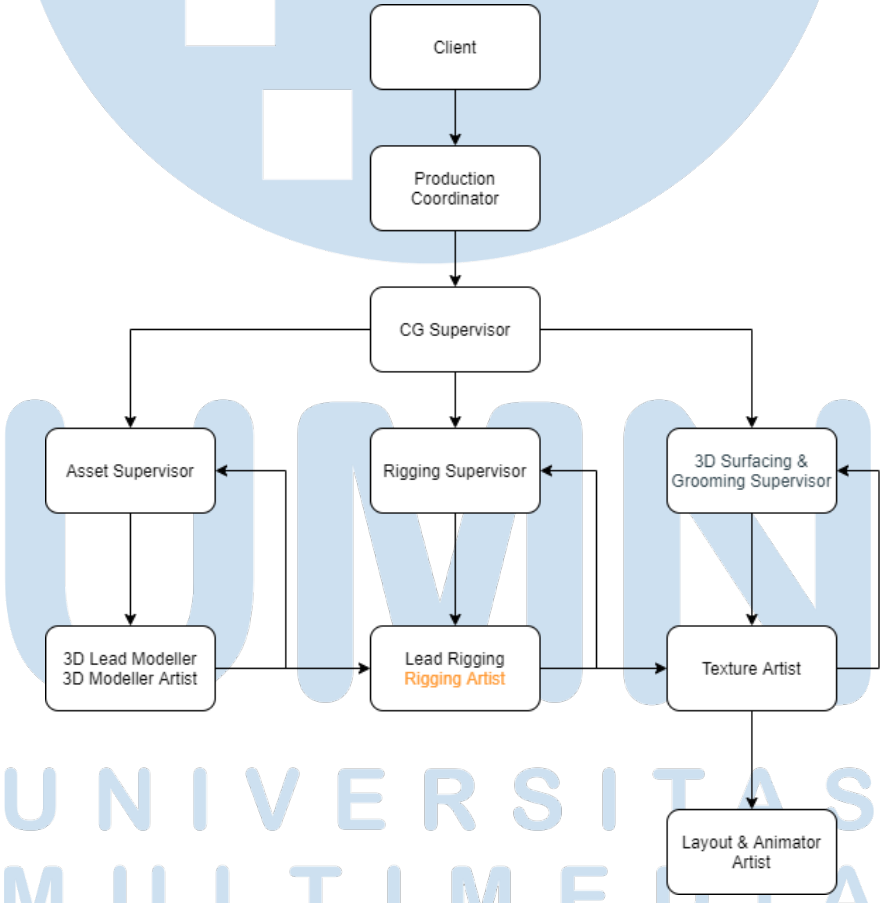
BAB III
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Saat melaksanakan kerja di Infinite Frameworks Studio, penulis memiliki kedudukan sebagai 3D *junior rigger*. Tugas penulis sebagai *rigger* adalah menyiapkan *asset rigging* sesuai dengan arahan yang diberikan oleh *production coordinator* dalam suatu *project*. Selama penulis bekerja di Infinite Frameworks Studios, penulis dibimbing oleh *rigging supervisor*, Res Yudikata. *Rigging supervisor* bertugas untuk memastikan apakah *rig* yang sudah dibuat sesuai dengan standar dan siap untuk dianimasikan oleh animator.



Gambar 3.1 Koordinasi Kerja Penulis di Infinite Frameworks Studios (Dokumentasi pribadi)

Berdasarkan pada gambar koordinasi kerja penulis di Infinite Frameworks Studios, sebuah *project* dapat dimulai ketika studio dan pihak *client* sudah memiliki kesepakatan mengenai *project* yang akan dikerjakan. Pihak *client* kemudian memberikan bahan – bahan yang dibutuhkan dalam proses pembuatan animasi kepada *production coordinator* studio. *Production coordinator* kemudian memberikan *assignment* kepada *production artist* yang terdiri dari *modeling, texturing,* dan *rigging.*

Modeller memiliki tugas untuk menciptakan objek 3D *asset* sesuai dengan permintaan dari *client*. Setelah *modeller* selesai membuat 3D *asset*, 3D *asset* tersebut kemudian diberikan kepada *rigging artist* untuk diberi *rig* agar dapat dianimasikan. Saat *rigging artist* sedang membuat *rig* untuk *asset* tersebut, *texture artist* juga dapat membuat *texture* sesuai dengan keinginan *client.* Ketika setiap divisi sudah selesai dengan tugasnya, hasil tersebut akan diberikan ke masing – masing *supervisor* untuk diperiksa apakah hasil pekerjaan tersebut dapat dilanjutkan ke tahap berikutnya.

3.2 Tugas dan Uraian Kerja Magang

Berikut merupakan tugas dan uraian kerja penulis selama bekerja di Infinite Frameworks Studios.

3.2.1 Tugas yang Dilakukan

Berikut merupakan *list* tabel tugas yang dilakukan oleh penulis selama bekerja di Infinite Frameworks Studios.

Tabel 3.1 Detail pekerjaan yang dilakukan selama magang

NO	Minggu	Proyek	Keterangan
1	1-3	Perkenalan tempat kerja dan adaptasi lingkungan kerja	Mempelejari *rigging pipeline* studio dari penerimaan *asset*, pembuatan *body rig* hingga di *publish* ke animator.
2	4	BFF Project dan latihan rig	Revisi beberapa *asset set* & *dressing* milik rekan kerja, mempelajari cara kerja membuat *facial rig*.

3	5-6	DEP Project	Mengerjakan *rig prop* untuk episode baru sesuai dengan arahan *production coordinator.*
4	6-9	KB & VGS Project	Mengerjakan beberapa *variant character asset* KB dan membuat dua *rig char* VGS.
5	9 - 14	VGS Project	Mengerjakan rig character & props untuk episode baru.

3.2.2 Uraian Kerja Magang

Dalam proses pelaksanaan kerja magang, penulis memiliki status kedudukan sebagai karyawan kontrak dimana penulis tidak dapat memberikan informasi secara *detail* terhadap *project* atau *asset* yang sedang dikerjakan. Oleh karena itu penulis hanya dapat memberikan informasi secara umum tentang proses pekerjaan penulis selama melaksanakan program kerja.

Di Infinite Frameworks Studios, penulis sebagai 3D *rigger* memiliki tugas untuk membuat *rigging asset character*, *properties*,dan *dress.* Selama penulis mengerjakan *asset* tersebut, penulis menggunakan *software autodesk Maya* 2018. Terdapat beberapa tahapan dalam pembuatan rigging di Infinite Frameworks Studios, tahapan tersebut terdiri dari:

A. Sebelum Pembuatan Rigging

Sebelum penulis membuat *rig*, perlu diperhatikan terlebih dahulu beberapa aturan dasar dalam pembuatan *rigging.* Pertama, penggunaan struktur *group* dalam sebuah *asset.* Sebelum mulai membuat *rig*, perlu diperhatikan terlebih dahulu struktur *group* dalam setiap *asset,* pastikan *group* tersebut terdiri dari *group geo* yang terdiri dari seluruh *asset* hasil *modeling* dan group *rig* untuk seluruh hasil *rigging.*

Gambar 3.2 Contoh Struktur Penamaan *Group* dalam *Asset*

(Dokumentasi pribadi)

Kedua, penamaan file. Perlu dipastikan kembali bahwa setiap penamaan asset sudah sesuai dengan standar studio. Di Infinite Frameworks Studios, penamaan file setiap asset selalu dimulai dengan nama *project*, diikuti dengan tipe *asset* tersebut, lalu nama *asset* tersebut, kemudian diakhiri dengan status *asset* tersebut. Dalam contoh kasus ini penulis memiliki tugas untuk membuat *rig* karakter *AngelOfDenial* pada *project* DB, sehingga penamaan yang tepat untuk *file* tersebut adalah "DB_C_AngelOfDenial_RG". Penamaan "DB" untuk nama *project*, "C" untuk penamaan tipe *asset*, "AngelOfDenial" untuk nama karakter, dan "RG" untuk status dari *asset* tersebut.

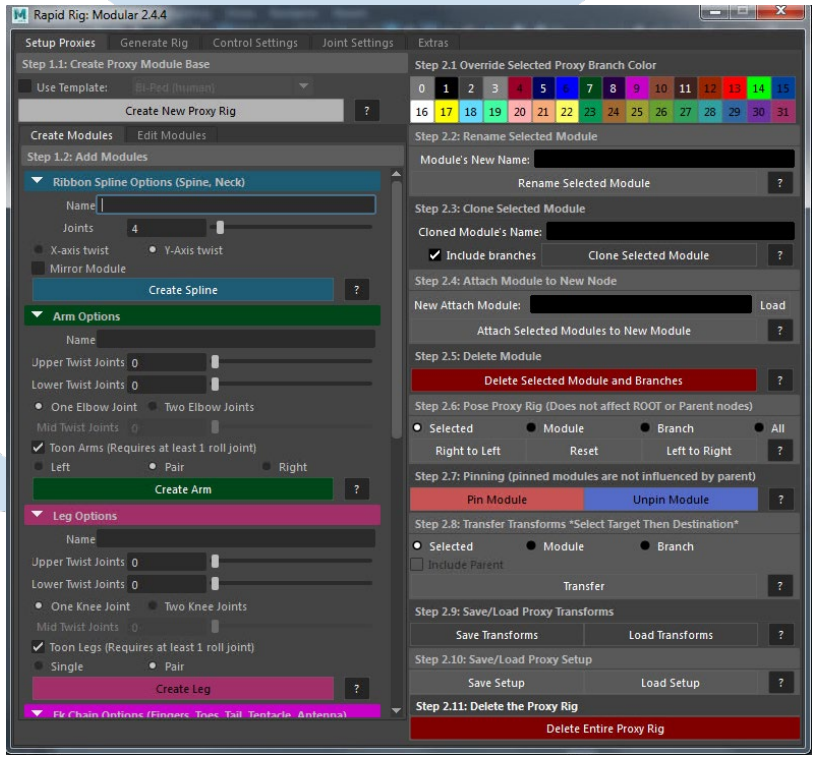
Ketiga, penamaan untuk *asset rigging.* Setiap *asset rig* perlu memiliki penamaan sesuai dengan standar studio agar tidak menimbulkan kebingungan saat membuat *rig* atau terjadi masalah yang tidak diinginkan. Penamaan *rigging* tersebut terdiri dari "(sisi)_(bagian objek)_(angka)_(tipe)". "sisi" dapat menunjukan bagian sisi yang sedang dikerjakan, terdiri dari kiri atau kanan. "bagian objek" dapat berupa penamaan bagian – bagian dari objek seperti *arm*, *finger*, *leg*, *head,* dan lain-lain. "tipe" dapat berupa informasi penamaan dalam objek seperti *Controller* (Ctrl), *Joint Object* (Jnt), *Mesh Object* (MSH), *Locator Object* (Loc). Contoh penamaan untuk *asset rig* dapat berupa "L_Finger_01_Ctrl".

B. *Tool* Pembuatan *Rigging*

Sebelum penulis membuat rigging pada asset, terdapat beberapa tool untuk mempermudah penulis dalam membuat sebuah rig agar dapat meminimalisir penggunaan waktu. Tool tersebut terdiri dari:

1. *Rapid Rig Modular*

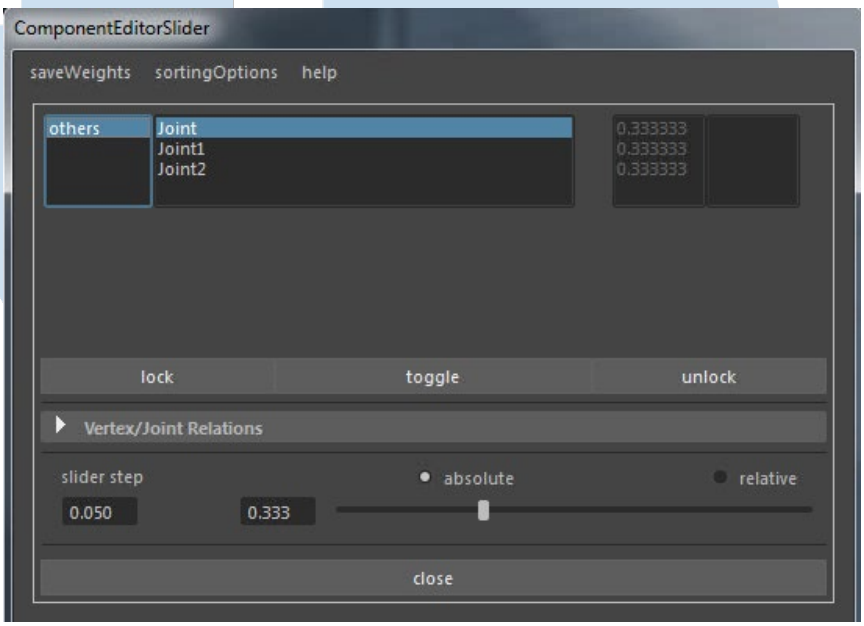
   Rapid Rig Modular (RRM) merupakan *tool script* untuk membuat *rigging* dengan berbagai varian bentuk dari manusia, *humanoid*, binatang, hingga binatang berbentuk monster. *Tool* ini sudah dirancang untuk mempermudah proses pembuatan *rigging* dan meminimalisir penggunaan waktu. Cara kerja *tool* ini adalah *user* hanya perlu membuat *guide rig* sesuai dengan *model* yang diinginkan, kemudian *tool* tersebut akan membuat *rig* secara otomatis sesuai dengan *guide rig* yang dibuat. *Tool* tersebut secara otomatis menciptakan *rig* dari *root*, *spline*, *neck*, *arms*, *legs*, *head*, serta memiliki fitur *IK/FK Switch*, *Squash & Stretch*, *FK Chains*, dan beberapa *attribute* lainnya.



Gambar 3.3 *Menu UI tool Rapid Rig Modular*
(Dokumentasi pribadi)

2. *CarlaH skinWeightsEditor*

   Tool ini bertujuan untuk mempermudah *user* dalam melakukan *skinning*. Pengguna dapat merubah *skin weights* lebih dari satu *joint* dengan menggunakan *slider* yang tersedia. Kemudian *tool* ini secara otomatis menyusun setiap *joint* sesuai penamaannya untuk mempermudah *user* dalam mencari. Selain itu pengguna mendapatkan *real-time feedback* dari perubahan *skin weight* yang dibuat.



Gambar 3.4 *Menu UI tool CarlaH skinWeightsEditor*

(Dokumentasi pribadi)

3. Control Library

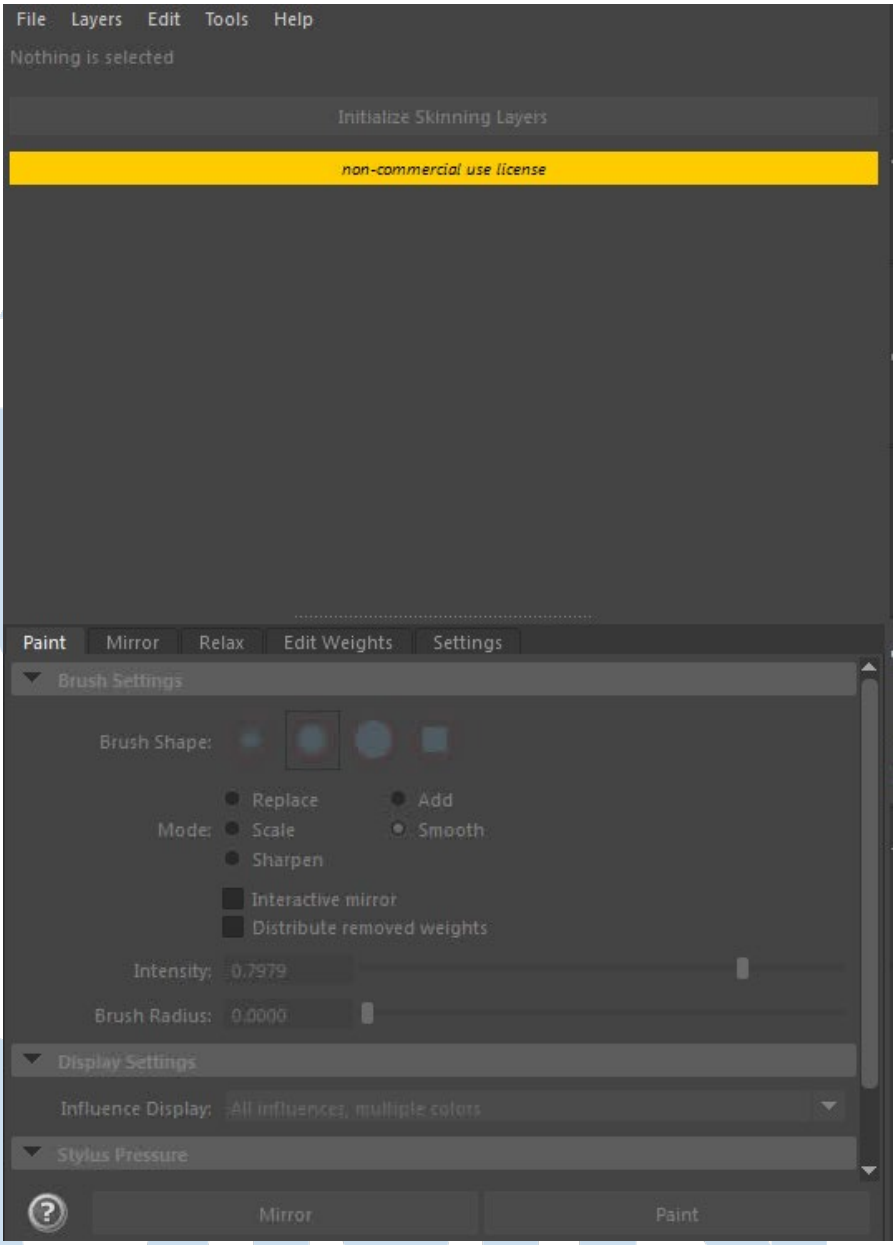
   Tool ini berguna untuk membantu *user* dalam membuat *controller rigging*. *Tool* ini menyediakan berbagai bentuk *controller* dan juga *user* dapat merubah warna *controller* sesuai dengan yang diinginkan. Selain itu *tool* ini juga memiliki fitur tambahan untuk dapat merubah *rotation* dan *scale* dari *controller* yang dipilih.

Gambar 3.5 *Menu UI tool Control Library*

(Dokumentasi pribadi)

4. *NgSkin Tools*

   Merupakan *tool plugin* untuk membantu *user* dalam melakukan *skinning*. *Tool* tersebut memberikan fleksibilitas kepada *user* dengan berbagai fitur seperti *layers skin*, *smooth weights*, *mirroring*, dan masih banyak lagi.



Gambar 3.6 *Menu UI tool NgSkin Tools*

(Dokumentasi pribadi)

5. *Perseus Auto Rig*

   Merupakan tool yang mempermudah *user* dalam melakukan *facial rig*, dalam tahap ini user hanya perlu memberikan *guide rig* sesuai dengan model yang diinginkan. Kemudian *tool* tersebut secara otomatis akan memberikan *joint* serta *controller* sesuai dengan *guide rig* yang dibuat.

Gambar 3.7 *Menu UI tool Perseus Auto Rig*

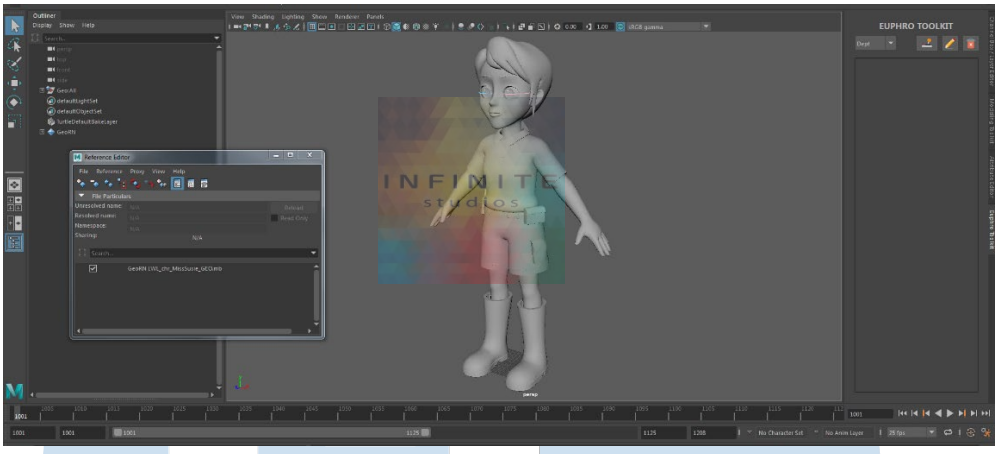
(Dokumentasi pribadi)

C. Proses Pembuatan *Rigging*

Secara keseluruhan, proses pembuatan rigging memiliki tahapan yang sama pada setiap asset yang dikerjakan dalam project di Infinite Frameworks Studios. Dalam hal ini, penulis mencoba menjelaskan proses penulis membuat *rigging* karakter di salah satu *project*.

1. *Reference File Asset*

   Reference asset yang sudah dikerjakan oleh *modeller.*

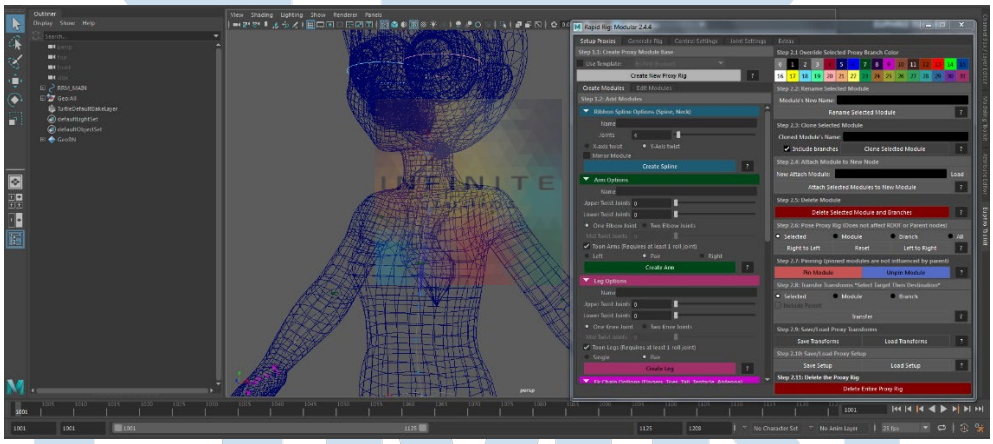


Gambar 3.8 Proses Penulis Melakukan *Reference File Asset*

(Dokumentasi pribadi)

2. *Build Body Rig Using RapidRigModular Tool*

   Kemudian, penulis membuat *guide rig* menggunakan *tool rapid rig modular* dan menyesuaikan setiap *joint* dengan posisi *model*.

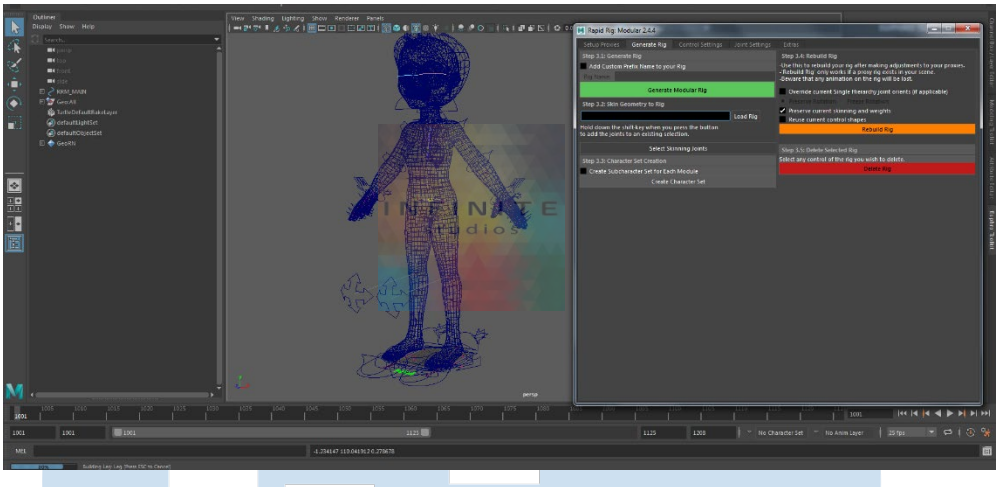


Gambar 3.9 Proses Penulis Membuat *Body Rig*

(Dokumentasi pribadi)

3. *Generate Rig*

   Dengan hanya menekan tombol *generate*, seluruh *rig* akan dibuat secara otomatis dari sistem *hierarchy* hingga fitur fk/ik *switch*.

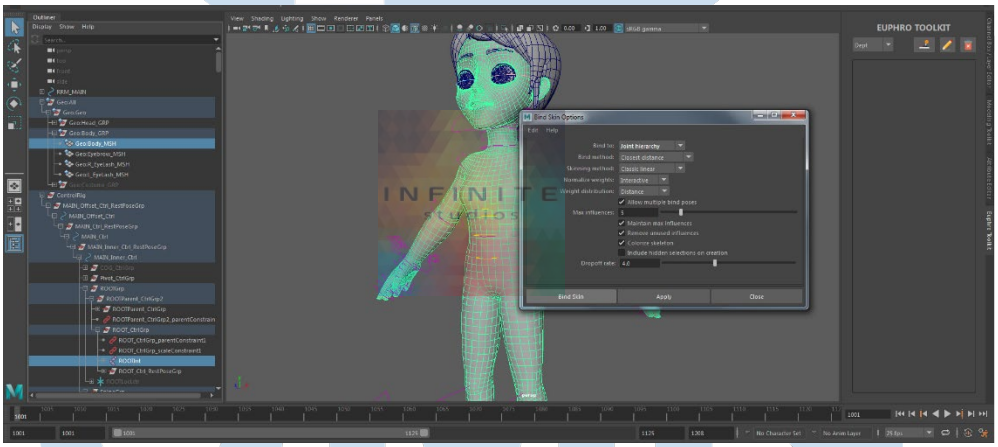


Gambar 3.10 Proses Penulis Melakukan *Generate Rig*

(Dokumentasi pribadi)

4. *Bind Skin*

   Setelah *rig* berhasil di *generate*, *rig* tersebut belum terhubung dengan 3D *model* sehingga perlu dilakukannya sistem *bind skin* dimana hasil *rig* yang sudah dibuat akan terhubung dengan 3D *model*.



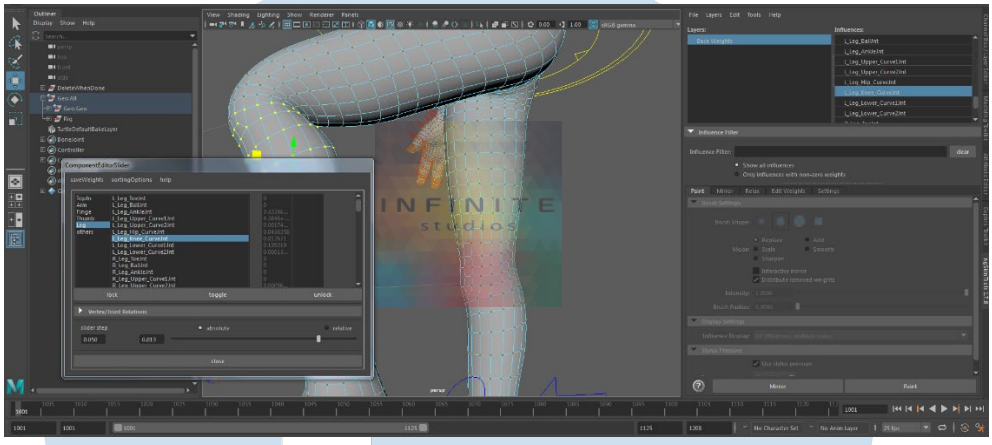
Gambar 3.11 Proses Penulis Melakukan *Bind Skin*

(Dokumentasi pribadi)

5. *Blocking Skin* dan *Smooth Skin*

   Setelah berhasil di tahap *bind skin*, hasil *skinningnya* belum relatif sempurna sehingga perlu melakukan *blocking skin* di *area* tertentu sesuai dengan pergerakannya dengan menggunakan *carlah skinweightseditor*. Setelah hasil *blocking skin* sudah sesuai dengan yang diinginkan, hasil

blocking skin tersebut dapat di *smooth* agar tidak terlihat kaku pada saat melakukan gerakan *extreme* dengan menggunakan *ngskin tool*.

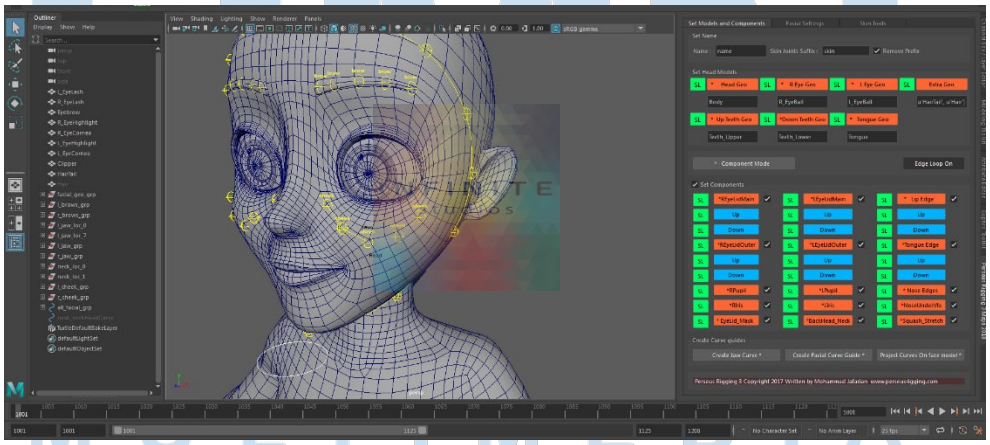


Gambar 3.12 Proses Penulis Melakukan *Blocking* dan *Smooth Skin*

(Dokumentasi pribadi)

6. *Facial Rig Using Perseus Auto Rig*

   Tahap selanjutnya setelah *body rig* adalah *facial rig*. Dalam tahap ini *file asset facial rig* dibuat terpisah. Dengan menggunakan *tool perseus*, penulis hanya perlu menyeleksi *area* tertentu sesuai dengan kebutuhan *tool*. *Perseus* kemudian akan secara otomatis membuat dan merancang setiap *joint* untuk terhubung secara langsung dengan 3D *model*. Setelah berhasil *generate*, penulis hanya perlu merapikan beberapa *area skinning* agar terlihat rapi.

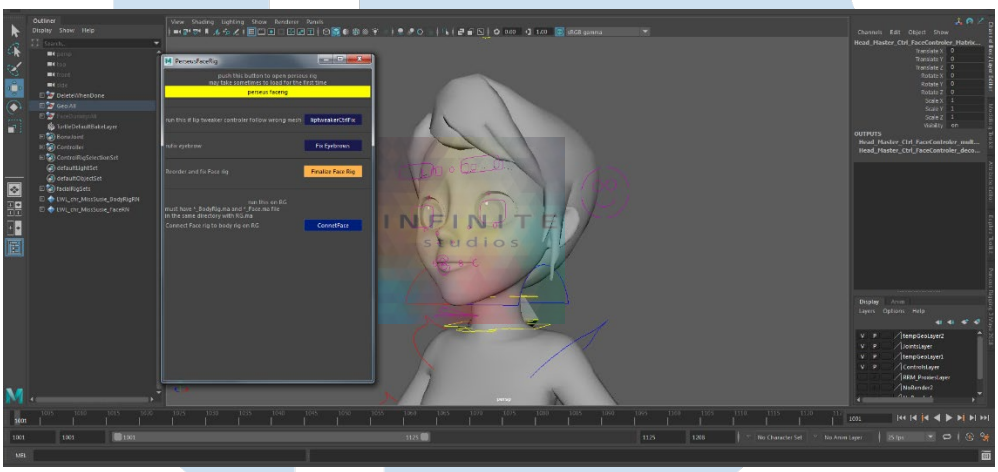


Gambar 3.13 Proses Penulis Membuat *Facial Rig*

(Dokumentasi pribadi)

7. *Combine Body Rig & Facial Rig*

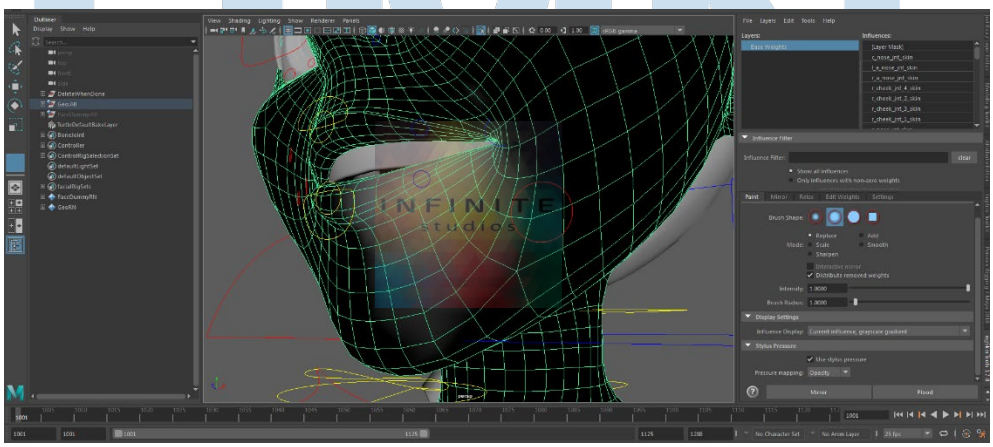
Saat *body rig* dan *facial rig* sudah selesai, penulis hanya perlu menjalankan *script* yang sudah disediakan oleh studio, dan secara otomatis *asset body rig* dan *facial rig* akan digabungkan dalam satu *file Maya*. Secara umum *script* tersebut menggabungkan *asset body rig* dan *facial rig* ke dalam satu *file Maya*, kemudian *script* akan menjalankan sistem *blendshape* yang menghubungkan seluruh *asset facial rig* ke *asset body rig*.



Gambar 3.14 Proses Penulis Menggabungkan *Body Rig* dan *Facial Rig*
(Dokumentasi pribadi)

8. *Smooth* & *Finalize Rig*

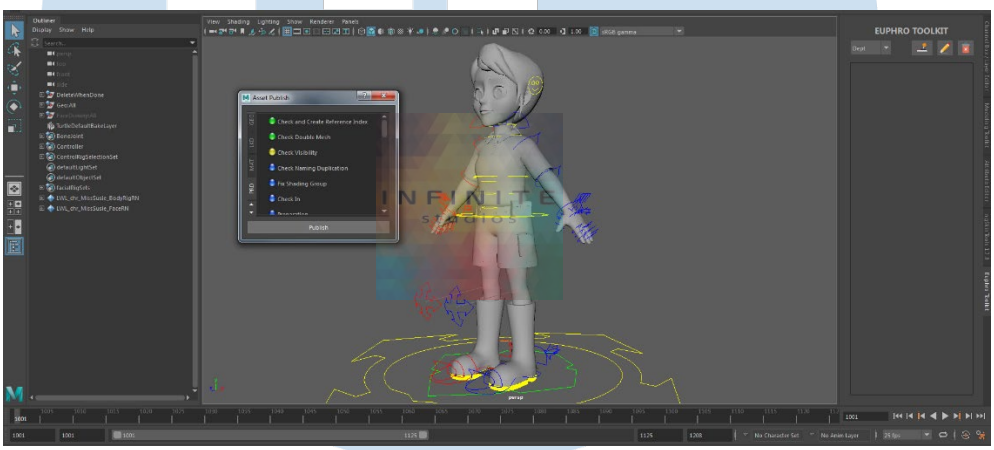
Setelah *body rig* dan *facial rig* sudah berhasil digabungkan, penulis hanya perlu merapikan beberapa bagian *skin* dan memastikan *asset* tersebut siap untuk dianimasikan.



Gambar 3.15 Proses Penulis Melakukan *Smooth* dan *Finalize Rig*
(Dokumentasi pribadi)

9. *Publish Asset*

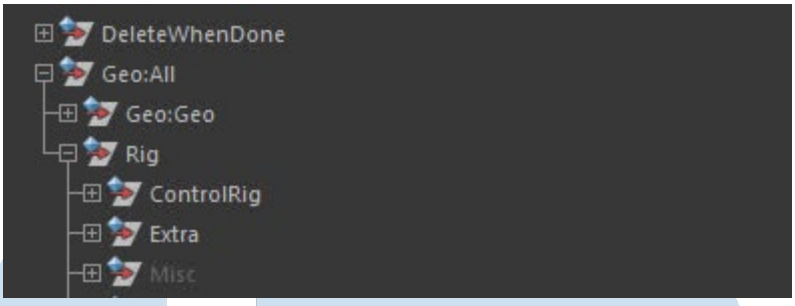
Apabila *asset rig* sudah rapi dan benar sesuai standar studio, *asset* tersebut kemudian akan dipublish menggunakan *tool* yang sudah disediakan oleh studio. Secara umum *tool* tersebut hanya bekerja untuk memastikan tidak ada kesalahan atau hal-hal yang mengganggu dalam *asset rig* tersebut. Setelah *asset* tersebut lolos dalam pemeriksaan *tool* tersebut, *asset rig* tersebut kemudian akan *dipublish* ke *folder* yang sudah ditentukan dan siap untuk dianimasikan.



Gambar 3.16 Proses Penulis Melakukan *Publish Asset*

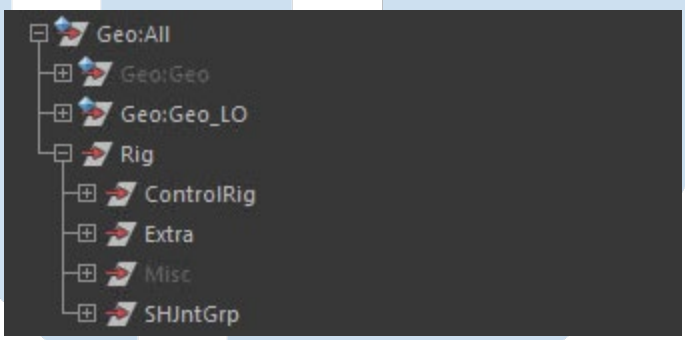
(Dokumentasi pribadi)

Dari berbagai project yang penulis kerjakan selama bekerja di Infinite Frameworks Studios, terdapat satu project yang memiliki alur pengerjaan *rigging* yang berbeda, project tersebut adalah *project* VGS. Pada umumnya ketika melakukan *generate rig*, sistem *hierarchy joint* akan tergabung secara langsung dengan *controller*, namun dalam *project* VGS, *joint hierarchy* terbentuk secara terpisah dengan *controller* serta memiliki *group* tersendiri. Hal tersebut dikarenakan *project* VGS menggunakan *render game engine* sehingga *joint hierarchy* dibuat terpisah agar mempermudah saat ingin melakukan *import* ke *game engine*. Selain itu, dalam project VGS, *file body rig* dan *facial rig* tidak digabung dengan menggunakan *blendshape* melainkan *file facial* akan di *import* ke *file body rig*. Kemudian hasil *skinning facial rig* akan di *copy* ke dalam *body rig*, serta seluruh *controller facial rig* akan di *parent* ke dalam group body rig.



Gambar 3.17 Sistem *Hierarchy Rig* Untuk *Project* Umum

(Dokumentasi pribadi)



Gambar 3.18 Sistem *Hierarchy Rig* Untuk *Project* VGS

(Dokumentasi pribadi)

3.2.3 Kendala yang Ditemukan

Selama penulis melaksanakan program kerja di Infinite Frameworks Studio, penulis mengalami beberapa kendala. Pertama, *server* dan komputer studio sering mengalami gangguan sehingga penulis kesulitan untuk dapat mengakses server dan komputer studio. Kedua, terdapat beberapa bahasa teknis baru yang penulis belum pernah pelajari selama masa perkuliahan sehingga penulis sedikit kesulitan dalam memahami catatan revisi yang diberikan. Ketiga, penulis sering mengalami kendala teknis saat mengerjakan *rigging*. Beberapa kendala teknis yang penulis alami dalam pengerjaan *rigging* antara lain, *error* saat ingin *bind skin joint* ke mesh yang dituju, *asset rigging* berubah ketika *export* ke dalam *game engine*, *reference unknown node* ketika ingin *reference asset*.

3.2.4 Solusi atas Kendala yang Ditemukan

Saat penulis sedang mengalami beberapa kendala tersebut, penulis menemukan solusi atas kendala yang dihadapi. Solusi atas gangguan *server* dan komputer, untuk dapat mengatasi kendala tersebut penulis memberitahu kepada *supervisor* yang kemudian akan disampaikan kepada *production coordinator* agar *production coordinator* dapat meminta pihak IT untuk segera memeriksa *server* dan komputer tersebut. Solusi atas catatan revisi yang sulit dimengerti, untuk dapat mengatasi masalah tersebut penulis mencoba untuk membaca ulang catatan revisi untuk memahaminya secara perlahan atau penulis mengkonfirmasi catatan revisi tersebut kepada supervisi. Untuk solusi kendala teknis, penulis dapat berkomunikasi secara langsung dengan supervisi menggunakan aplikasi *Basecamp* atau *meeting* melalui *Google Meeting.*