ASEECT
Homepage
https://jurnal.polibatam.ac.id/index.php/ASEECT

Journal of Applied Sciences, Electrical Engineering and Computer Technology

Vol 1 No 1 2020

Dynamic Rigging Menggunakan *Expression* pada Maya 3D

Selly Artaty Zega[1]*, Zulfiqar Naufal Shibghatullah[1]
[1]Politeknik Negeri Batam, Jurusan Teknik Informatika, Batam

**E-mail: selly@polibatam.ac.id*

Received: 03-04-2020 *Accepted: 29-04-2020* *Published: 30-04-2020*

Abstrak

Penelitian ini mencakup pembuatan dan penggunaan *tool rigging* otomatis pada aplikasi 3D Maya. Tujuan penelitian ini adalah menghasilkan *tool dynamic rigging* untuk membangun *rig* serta mengontrol gerakan dinamik pada animasi. Penggunaan *dynamic rigging* dapat menurunkan performa *software*, namun pada Maya terdapat *expression* yang dapat mengkalkulasi gerakan menggunakan bahasa MEL, sehingga dapat meningkatkan kinerja *software*. Penelitian terbagi dalam tahapan merancang logika *dynamic rig, script,* dan antarmuka, implementasi pembuatan *script*, serta implementasi *tool* yang dihasilkan pada objek 3D. *Script* Phyton untuk menghasilkan *UI tool* dan komponen *dynamic rig*, sedangkan *script* MEL pada *node expression* untuk menghasilkan gerakan rambut. Dalam penelitian ini, penulis menggunakan 3 objek yaitu tentakel, rantai, dan rambut. Gerakan dinamik dihasilkan dari proses kalkulasi rumus Fisika Dasar yaitu kecepatan dan percepatan yang dijadikan dalam bahasa MEL pada *node expression*. Uji coba pada 3 objek tersebut berhasil memperlihatkan gerakan dinamik yang diperlukan dalam pembuatan animasi.

Kata kunci: *Rigging, Otomatis, Dynamic, Expression*

Abstract

This research covers the manufacture and use of automated rigging tools in 3D Maya applications. The purpose of this study is to produce a dynamic rigging tool to build a rig and control dynamic motion in animation. The use of dynamic rigging can reduce software performance, but in Maya, some expression can calculate movements using the MEL language, to improve software performance. The research is divided into stages of designing dynamic rig logic, scripts, and interfaces, implementing script creation, and implementing tools that are generated on 3D objects. Python scripts to produce UI tools and dynamic rig components, while MEL scripts on expression nodes to produce hair movements. In this study, the authors used three objects namely tentacles, chains, and hair. Dynamic motion is produced from the calculation process of the Basic Physics formula, namely the speed and acceleration used in the MEL language on the node expression. Trials on these 3 objects succeeded in showing the dynamic movements needed in making animation.

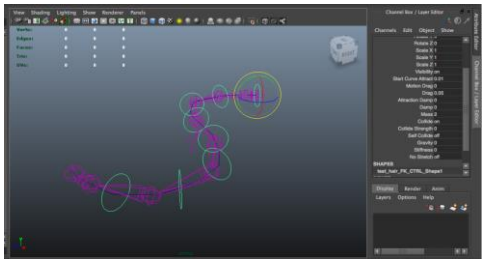
Keywords: *Rigging, Otomatis, Dynamic, Expression*

Pendahuluan

Pergerakan rambut yang realis pada animasi menjadi komponen penting dalam menampilkan gerakan virtual suatu penokohan karakter, hal ini disampaikan Maryann Simmons dalam penelitiannya yang berjudul *Directing Hair Motion on Tangled* [1]. Dalam studi tersebut, Maryann Simmons memaparkan pendekatan *hair dynamics* dengan mengontrol *key-frame* menggunakan *interleaving animation and simulation*, yaitu menyatukan pendekatan animasi dan simulasi. Studi tersebut menghasilkan gerakan yang dinamis pada 500 helai rambut.

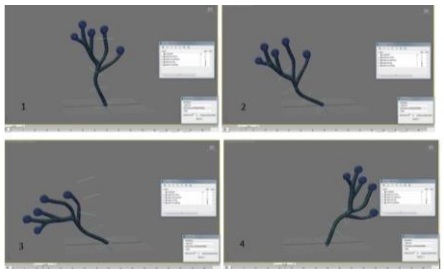
Namun, Menglei Chai berpendapat bahwa kendala yang ditemukan dalam menghasilkan gerakan rambut yang realis adalah komputasi yang dibutuhkan sangat tinggi. Dalam penelitiannya, Menglei Chai mengusulkan pengurangan jumlah rambut yang berinteraksi. Simulasi diberikan pada *guide hair* yang memimpin pergerakan sisa helai rambut lainnya [2]. Namun, selain keberhasilan pergerakan rambut oleh kedua penelitian tersebut, Wyndham Batton dalam thesisnya menyampaikan bahwa masalah yang umum muncul pada gerakan rambut menggunakan komputer adalah adanya perbaikan gerakan yang konstan melalui beberapa iterasi [3]. Untuk itu, dibutuhkan suatu *tool* yang dapat mengautomasi gerakan rambut dengan kebebasan pengguna dalam mengontrol gerakannya. Sehingga diusulkanlah penelitian ini dengan tujuan utama menghasilkan *tool dynamic rigging* untuk membangun *rig* serta mengontrol gerakan dinamik pada animasi.

Gerakan dinamik diterapkan menggunakan prinsip seperti *dynamic rig* yang dibuat oleh Batton, W. dalam thesisnya pada tahun 2016 berjudul *A Dynamic Hair Rigging System for Maya* [3]. Berdasarkan thesis yang dibuat oleh Batton, *dynamic rig* dibuat menggunakan kurva seperti yang dapat dilihat pada gambar 1, sebagai penentuan posisi untuk *joint* serta penentu gerakan dari sistem dinamik yang dibuatnya. Selain itu, dinamik yang dibuat menggunakan sistem nHair Maya sehingga simulasi perlu dijalankan terlebih dahulu untuk memperlihatkan hasil gerakan dinamik. *Dynamic rig* yang dibuat oleh penulis menggunakan rumus perhitungan yang dibuat oleh Brown Bag Films, sehingga tidak perlu menggunakan kurva untuk membuat *dynamic rig* bergerak [3].



Gambar 1. *Dynamic Hair Rig* (Sumber: dok. penulis)

Brown Bag Films dalam websitenya yang berjudul *Setting up a Dynamics Rig System – 3D Studio Max* telah membuat tool sistem dinamik untuk 3D Studio Max serta dapat digunakan pada Maya dengan menggunakan *node expression* sebagai kalkulasi dinamik [4]. *Node expression* pada Maya dapat melakukan perhitungan matematika secara efesien [5].



Gambar 2. BBF Dynamics Rig System (Sumber, www.brownbagfilms.com)

Berdasarkan Gambar 2, terlihat gerakan dinamik dimana objek yang bergerak ke kiri dan ke kanan mengalami pergerakan melewati lingkaran yang berfungsi sebagai pengendali objek tersebut. *Script* yang dibuat oleh Brown Bag Films dijadikan penulis sebagai acuan pembuatan *script* pada penelitian ini.

Animasi merupakan mekanisme penerapan efek gerakan dinamis pada objek statis yang tidak bergerak terhadap ruang dan waktu sehingga menjadikan objek tersebut seolah-olah bergerak [6]. Animasi terbagi menjadi animasi 2D di mana terdapat sumbu x dan y dan animasi 3D yang memiliki sumbu x, y, dan z. Animasi 3D terbagi menjadi 8 proses produksi yaitu *modelling, shading, texturing, rigging, layout, animasi, visual effect,* dan *render*. Untuk membuat animasi 3D terdapat berbagai software seperti Blender, 3D Studio Max, Maya 3D dan sebagainya. Software Maya 3D salah satu perangkat yang digunakan untuk membuat animasi 3D yang dibuat oleh Autodesk memiliki banyak fitur yang sudah disediakan. Namun dengan bantuan script menggunakan MEL (*Maya Embedded Language*) dan Python dapat

meningkatkan produktivitas proses pembuatan animasi 3D.

Rigging yaitu proses desain dan implementasi manipulasi artikulasi dari karakter 3D yang dianimasikan [7]. Selama proses *rigging*, sistem hirarki dari gerakan dan kontrol artikulasi untuk animasi dibuat pada model 3D. Rigging merupakan salah satu bagian dari proses pembuatan animasi 3D dimana memberikan tulang yang diperlukan pada model 3D supaya dapat melakukan suatu gerakan. Dalam proses pembuatan *rigging*, terdapat fungsi yang disebut *dynamic rig* dimana dari suatu *joints* bisa bergerak secara dinamis tanpa perlu dikendalikan menggunakan *controller*. Hal ini mempermudah animator dalam membuat animasi gerakan seperti rambut dan rantai. Untuk membuat gerakan dinamik dapat menggunakan fitur nHair yang disediakan oleh Maya 3D. Namun, dengan menggunakan *expression node* yang memiliki fungsi yang serupa dengan nHair, dapat menambah kelancaran ketika animator membuat animasi gerakan pada objek tertentu.

Simulasi dapat digunakan dalam pembuatan animasi dinamik seperti pada rambut. Jika simulasi dilakukan langsung dari *expression* akan mempermudah kinerja pembuatannya. Dalam hal ini, simulasi yang dilakukan tidak menggunakan simulasi dinamik yang terdapat pada Maya, namun menggunakan kalkulasi pada *expression*, sehingga perhitungan gerakan selalu berubah sesuai dengan perubahan *frame*. Hal ini masih perlu dibuktikan menggunakan 3 model yang telah disediakan penulis yaitu tentakel, rantai dan rambut. Penggunaan 3 objek ini untuk melihat hasil dari *dynamic rigging* jika menggunakan expression dalam segi pembuatan animasi 3D dan pembuatan *rigging*.

Sehingga, studi tentang *dynamic rigging* menggunakan *expression* pada Maya 3D ini ditujukan untuk membahas cara pembuatan *dynamic rigging* pada tiga jenis objek, yaitu rambut, antenna dan rantai. Proses pembuatan menggunakan expression dibuat dengan bahasa pemrograman MEL dan Python. Dengan metode ini, dapat mempermudah proses pembuatan rigging dan animating pada pembuatan animasi 3D di software Maya 3D.

Metode Penelitian

Tahapan metode penelitian mengumpulkan data, tahap perancangan, pembuatan *script* dan implementasi pada model 3D.

A. Pengumpulan Data

Diawali dengan mengumpulkan data yang dibutuhkan untuk pembuatan animasi 3D, *dynamic rigging* dan persamaan dinamik. Dalam pembuatan animasi 3D pergerakan rambut, untuk mencapai gerakan dinamis seperti dapat berkibas dengan rapi, animator memerlukan waktu yang lama tergantung dengan kerumitan gerakan dan banyaknya rambut. Dengan adanya *dynamic rigging* pada rambut, proses pembuatan animasi menjadi lebih efisien terhadap sumber daya dan waktu.

Literatur pembuatan gerakan rambut antara lain disampaikan Maryann Simmons menggunakan pendekatan *hair dynamics* dengan mengontrol *key-frame* menggunakan *interleaving animation and simulation* [1]. Menglei Chai mengusulkan pengurangan jumlah rambut yang berinteraksi. Simulasi diberikan pada *guide hair* yang memimpin pergerakan sisa helai rambut lainnya [2]. Serta studi Wyndham Batton yang menggunakan rumus perhitungan yang dibuat oleh Brown Bag Films dalam menghasilkan gerakan rambut yang diterapkan pada Maya *software* [3].

Dalam pembuatan sistem dinamik memerlukan persamaan untuk menghasilkan gerakan yang dinamis. Persamaan diambil dari beberapa sumber yaitu persamaan menghitung kecepatan dan percepatan, dinamik Brown Bag Films, dan dari jurnal Batton, 2016 [3], [4]. Berikut adalah persamaan *dynamic* yang akan diimplementasikan:

$$OT = T + IT \quad \text{.....(1)}$$

Persamaan (1) Persamaan Inisialisasi Out Translate

$$A = (-1DV) + (S(T - OT) + G \quad \text{.....(2)}$$

Persamaan (2) Percepatan

$$V = V + A \quad \text{.....(3)}$$

Persamaan (3) Persamaan Kecepatan

$$OT = OT + V \quad \text{.....(4)}$$

Persamaan (4) Persamaan Penambahan Out Translate

$$OT = (S)(OT) + ((1 - S)\,T) \quad \text{.....(5)}$$

Persamaan (5) Persamaan Nilai Akhir Out Translate

Keterangan:

S = *State* (kondisi apakah status dinamik aktif atau tidak. Memiliki nilai 0 hingga 1)

V = *Velocity* (Kecepatan yang didapat dari perpindahan frame)

T = *Translate* (Lokasi perpindahan)

IT = *InitialTranslate* (Lokasi sebelum perpindahan)

OT = *OutTranslate* (Lokasi perpindahan berikutnya.)

S = *Stiff* (*Input* ketegangan dari dinamik. Memiliki nilai 0 hingga 1)

D = *Damp* (*Input* efek basah atau tidaknya pada dinamik. Memiliki nilai 0 hingga 1)

G = *Gravity* (nilai grafitasi)

A = *Acceleration* (percepatan yang didapatkan)

Persamaan (1), (2), (3) dan (4) akan digunakan pada node expression pada Maya 3D dan dikonversi sebagai bahasa MEL sehingga perhitungan tersebut bisa menghasilkan gerakan dinamik. Gerakan dinamik dihitung mulai dari dibacanya nilai T dan IT untuk menghitung perpindahan lokasi yang terjadi. Kemudian memperhitungkan A yang didapat dari perhitungan D, V, S, T, OT, dan G. Nilai OT yang telah selesai diperhitungkan akan menjadi *output node expression* sehingga *rig* bergerak secara dinamis.

B. Perancangan *Script* dan *UI*

Perancangan *script dynamic rigging* dituangkan dalam *pseudocode* sedangkan antar muka dalam bentuk diagram antar muka. Berikut adalah rancangan tahapan implementasinya:

1. Objek tentakel, rantai, atau rambut di-*import* kemudian diletakkan pada koordinat yang diinginkan. Selain melakukan *import*, objek lain yang diperlukan juga dapat dibuat langsung di Maya.

2. Menjalankan *script* Python melalui *script editor* di Maya 3D untuk membuka *window dynamic rig tool*. *Window* ini berfungsi sebagai UI untuk membuat *dynamic rig*.

3. Melakukan *input* nama dan jumlah *locator* pada kotak *input* yang terdapat pada *window dynamic rig tool*. Tipe data nama berupa *string* dan tipe data jumlah locator berupa *integer*.

4. Membuat *locator* dengan menekan tombol *"Create Locator"* dan diletakkan sesuai dengan kebutuhan objek *polygon*. Jumlah dan nama *locator* yang tercipta adalah sesuai dengan nilai *input* pada tahap 3. *Locator* yang dibuat perlu diposisikan pada tempat yang sesuai dengan kebutuhan.

5. Menjalankan *script* membuat dinamik dengan menekan tombol *"Create Dynamic"*. Setelah diposisikan, maka perlu memilih group locator terlebih dahulu kemudian menekan tombol "Create Dynamic". Hal ini dilakukan untuk mendapatkan jumlah pengulangan.

6. Membuat group utama untuk merapikan dengan menampung semua objek *locator, joint,* dan *controller*. *Script* akan melakukan pengecekan apakah ada *duplicate group* atau tidak. Jika ada, maka *script* akan menghapus yang sudah ada terlebih dahulu.

7. Membuat *controller* pada locator yang telah diposisikan. *Controller* yang terbuat dari *curve* akan dibuat sejumlah *locator* yang telah di-*input*. *Controller* pertama akan diberikan atribut tambahan untuk mengatur dinamik, ketegangan, kaku, dan *frame* awal dinamik yang dibuat. Atribut tambahan tersebut diperlukan sebagai *input dynamic rig* dan akan menentukan hasil dari gerakan itu sendiri. Sedangkan *controller* lainnya hanya memiliki atribut dasar yaitu *translate, rotate,* dan *scale*.

8. Membuat *joint* pada posisi *locator* yang telah diletakkan. *Joint* diperlukan untuk *skinning* pada objek. Hal ini dilakukan untuk menggerakkan objek. Selain itu, posisi pergerakan akan ditentukan oleh perhitungan *expression* sehingga menghasilkan gerakan dinamis.

9. Membuat *locator* baru pada *locator* yang telah diletakkan. *Locator* baru akan digunakan sebagai pengarah hadap *joint* yang sebelumnya dibuat. *Locator* baru mendapatkan *input* dari hasil

kalkulasi dinamik. Sehingga ketika *joint* bergerak, arah yang menghadap keluar dapat ditentukan.

10. Membuat *nodes* yang diperlukan untuk melakukan kalkulasi gerakan. *Nodes* yang dibuat adalah *plus-minus average*, *aim constraint*, *orient constraint*, dan *expression*. *Plus-minus average* digunakan untuk menghitung berapa hasil penjumlahan dari 2 *controller*. *Aim* dan *orient constraint* digunakan untuk menggerakkan bagian *controller* lain. Sedangkan *expression* yang memiliki rumus dinamik dalam bentuk bahasa MEL digunakan untuk kalkulasi gerakan dinamis.

11. Menyambungkan seluruh koneksi *controller, joint,* dan *node* yang telah dibuat sehingga *controller* bisa menggerakkan *joint*.

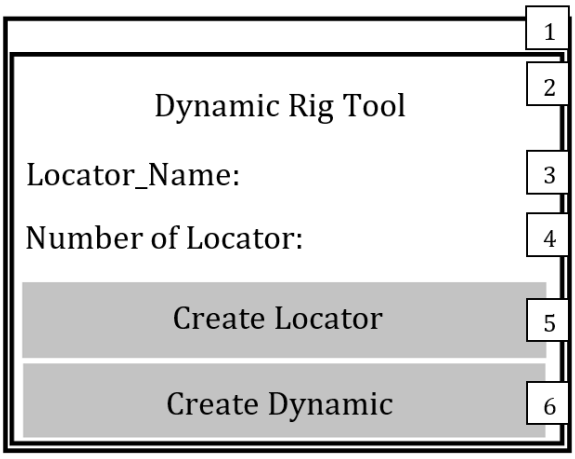
12. Merapikan semua yang telah dibuat ke group utama. Tahap yang terakhir yaitu menyambungkan seluruh begian sehingga *controller* bisa menggerakkan *joint.*

Berikut tahapan rancangan script dalam bentuk algoritma pembuatan *dynamic rigging*:

```
Initial State
Import object or create object
//comment bagian di atas adalah tahap
1
OPEN window "dynamic rigging tool"
//comment bagian di atas adalah tahap
2
If Button "Create Locator" is pressed
        Get locator name from textbox
        input "Locator Name" by user
        Get number of locator from
        textbox input "Number of Loca-
        tor" by user
        //comment bagian di atas ada-
        lah tahap 3
        Set default Locator Y is 0
        FOR each input number of loca-
tor DO
                Create Locator at coor-
                dinate vector (0,Y,0)
                ADD Locator Y is Y + 2
        //comment bagian di atas ada-
        lah tahap 4
IF Button "Create Dunamic" is pressed
//comment bagian di atas adalah tahap
5
        IF group object "dynamic" is
exists
                Delete existing group
object dynamic
        Create group object dynamic
        //comment bagian di atas ada-
        lah tahap 6
        FOR all selected locator DO
                Create controller at
        locator position
                //comment tahap 7
                Create joint at locator
position
                //comment bagian di
        atas adalah tahap 8
                Create newlocator at
locator position
                //comment bagian di
        atas adalah tahap 9
                Add attribute vector
                gravity, vector out-
                translate, vector ve-
                locity, vector initial-
                translate
                IF locator is first the
        locator
                        Add attribute
                        float dynamic,
                        float stiff,
                        float damp,
                        float frame to
                        controller.
                                Set at-
tribute dynamic to 0
                                Set at-
tribute stiff to 0.5
                                Set at-
tribute damp to 0.5
                                Set at-
tribute frame to 1
                Add attribute vector
                initialtranslate, vec-
                tor outtranslate, vec-
                tor gravity to control-
                ler.
                Create node orientcon-
        straint
                Create node aimcon-
        straint
                Create node plus-
        minusaverage
                Create node expression
                Set expression's mel
                command textbox with
                following string:
                        OT = T + IT
                        A = (-1 x D x
                V) + (S(T - OT)+G
                        V = V + A
                        OT = OT + V
                        OT = (S x OT) +
                ((1 – S) x T)
                //comment bagian di
        atas adalah tahap 10
                Connect node orientcon-
                straint to joint to
                orient controller
                Connect node aimcon-
                straint to joint to aim
                newlocator
                Connect node plus-
                minusaverage between
                controller and newloca-
                tor
                Connect node expression
        between controller and joint
        //comment bagian di atas ada-
        lah tahap 11
Final State
Locator to Guide the position of dy-
namic created
Group dynamic created
Controller dynamic for each locator
created
Joint dynamic for each for each loca-
tor created
Locator dynamic for each locator cre-
ated
```

```
    Constraint dynamic for each locator
    created
Expression for each locator created
Grouping all created object
          //tahap 12
```

Gambar 3 merupakan rancangan antar muka *tool dynamic rigging*.



Gambar 3. Desain *Window User Interface Dynamic Rig Tool* (Sumber, Penulis)

Berikut ini penjelasan desain antar muka pada gambar 3:

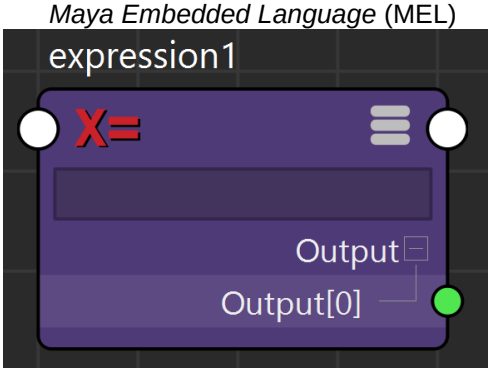
1. *Tool Window* sebagai *window* utama digunakan menampung semua *interface*.
2. *Name Tool Window.*
3. *Text Field Locator Name* digunakan untuk *input* penamaan *locator* dan *dynamic rig* yang akan dibuat.
4. *Text Field Number of Locator* digunakan untuk *input* jumlah locator dan *dynamic rig* yang akan dibuat.
5. *Button Create Locator* digunakan untuk membuat *locator* yang dapat *user* gunakan untuk penentu posisi *dynamic rig*.
6. *Button Create Dynamic* digunakan untuk membuat *dynamic rig* berdasarkan tempat *locator*.

C. Pembuatan *Script*

Bahasa pemrograman yang digunakan adalah MEL dan Python. MEL akan digunakan pada *node expression* dan Python digunakan untuk membangun antar *tool dynamic rigging*.

Berikut rincian dari kebutuhan *script* yang diperlukan dalam pembuatan *dynamic rigging.*

1. *Maya Embedded Language* (MEL)



Gambar 4. *Node Expression* (Sumber, Penulis)

Bahasa pemrograman MEL dalam pembuatan *dynamic rig* akan diimplementasikan pada *node expression* seperti pada gambar 4 dimana fungsi dari *node* tersebut sebagai proses *input* dan *output* dari hasil rumus yang dibuat. Hasil dari rumus tersebut memberikan rotasi pergerakan *joint* sehingga polygon dapat bergerak dan menampakan pergerakan.

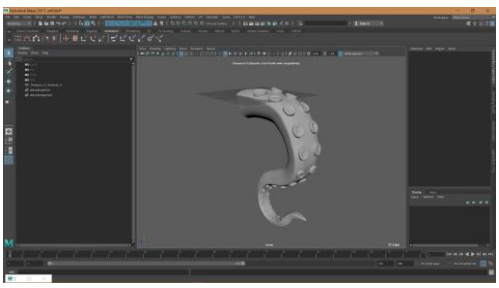
2. Python

Bahasa pemrograman Python dalam pembuatan *dynamic rig* menjadi *script* utama untuk pembuatan pembuatan objek *locator, controller, joint,* dan *node*. *Script* ini akan menampung rumus yang dibuat untuk diimplementasikan pada *node expression*. *Script* yang sudah dibuat kemudian diimplementasikan ke objek 3D.

D. Implementasi pada Model 3D

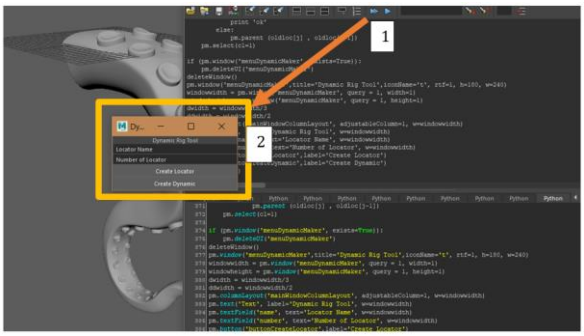
Berikut adalah tahapan implementasi pada model 3D:

1. User melakukan *import* model



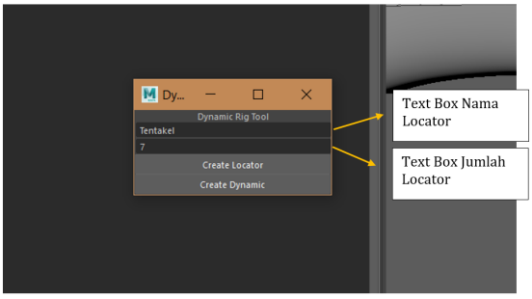
Gambar 5. *Import* Model Tentakel

2. *User* menjalankan *script dynamic rig tool*



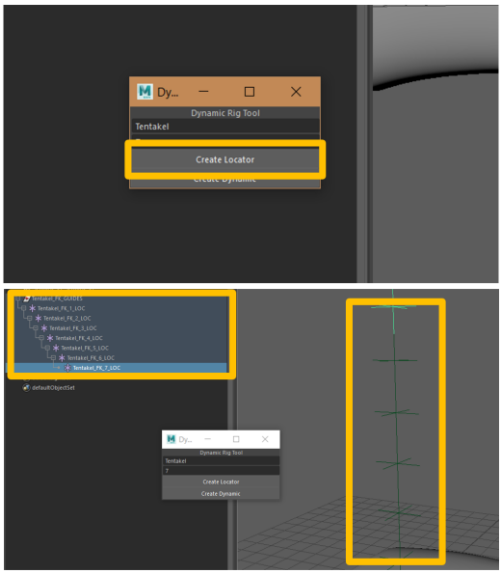
Gambar 6. Menjalankan *script dynamic rig tool*

3. *User* melakukan *input* pada *dynamic rig tool*



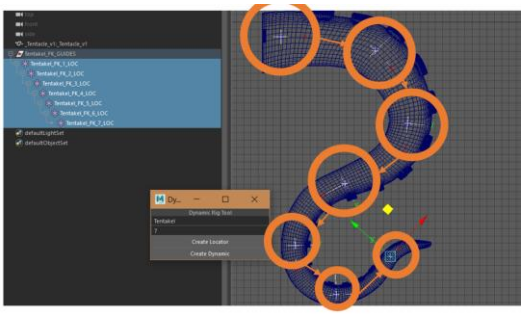
Gambar 7. *Input* Nama dan Jumlah *Locator*

4. Tekan tombol *create locator*



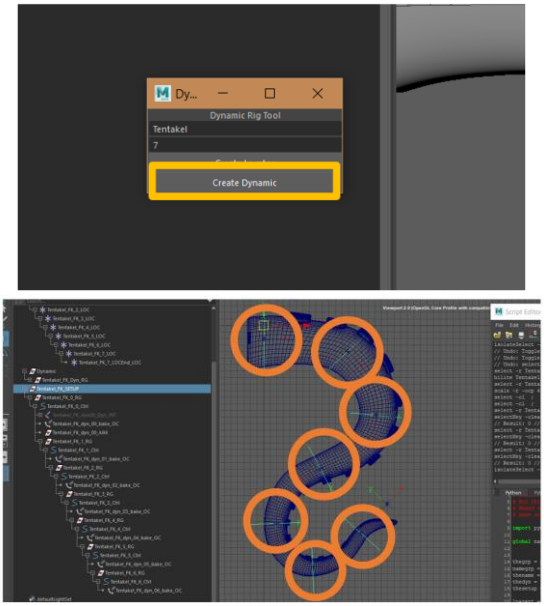
Gambar 8. Menciptakan *Locator*

5. Memposisikan *locator*



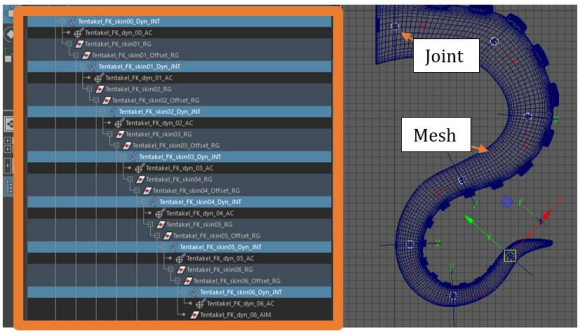
Gambar 9. Mengatur Posisi *Controller*

6. Tekan tombol *create dynamic*



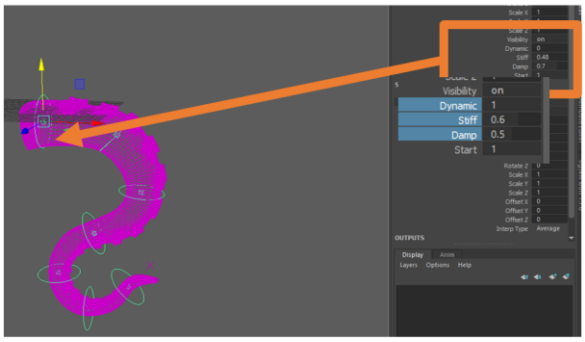
Gambar 10. Menciptakan Gerakan *Dynamic*

7. *Binding* dan *skinning*



Gambar 11. Menempelkan *Mesh* pada *Joint* (*Binding and Skinning)*

8. Menggerakkan *dynamic*

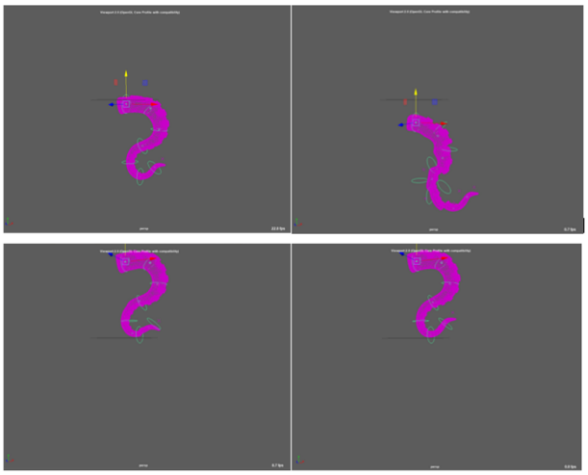


Gambar 12. Pengatura Atribut

Untuk menjalankan *dynamic, user* perlu mengubah nilai pada channel box seperti pada gambar 12. Atribut *dynamic* menjadi 1 dimana awalnya bernilai 0. Atribut tersebut hanya terdapat pada controller pertama dari *dynamic rig* yang berfungsi sebagai *parent*. Atribut *stiff* memiliki fungsi sebagai ketegangan, jika nilai tersebut mendekati 0 akan memberikan efek pergerakan tali dan jika mendekati 1 akan memberikan efek pergerakan seperti besi. Atribut *damp* jika nilai mendekati 0 akan memberikan efek pergerakan spons basah dan jika mendekati 1 akan memberikan efek pergerakan seperti spons kering. Atribut start berfungsi sebagai penentu *frame* ke berapa dinamik mulai bergerak.

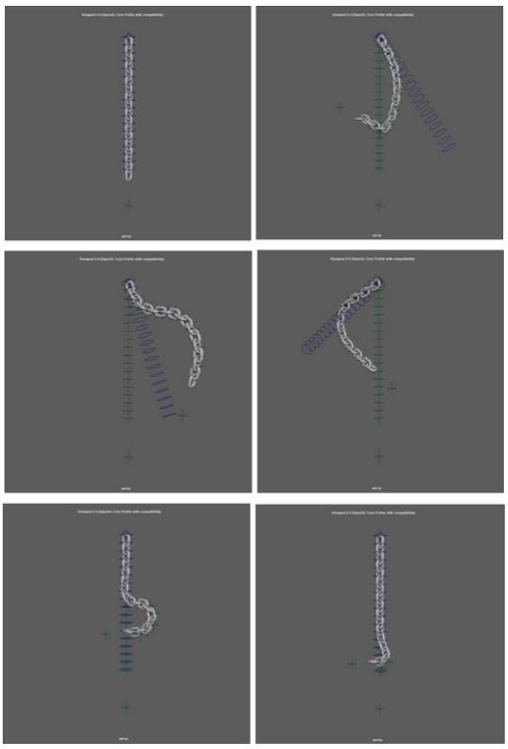
Hasil dan Diskusi

Penerapan *dynamic rigging* pada objek tentakel berhasil diciptakan yang disajikan pada gambar 13.



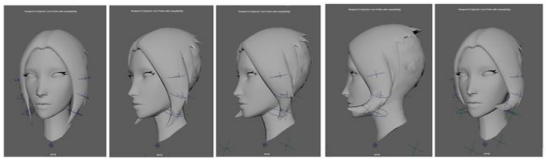
Gambar 13. *Dynamic Rigging* Tentakel (Sumber, Penulis)

Nilai atribut *stiff* dan *damp* masing-masing adalah 0.6 dan 0.5. Hal ini memberi efek memantul atau kenyal pada tentakel. Gerakan tersebut sesuai dengan sifat tentakel yang elastis atau lentur. Dari analisa di atas, maka untuk pembuatan dinamik yang menghasilkan gerakan lentur seperti tentakel memerlukan nilai *stiff* yang lebih besar dari *damp*.



Gambar 14. *Dynamic Chain Frame 1 to 6*

Gambar 14 menunjukkan penerapan *dynamic rigging tool* pada objek rantai. Pergerakan dilakukan pada sisi pangkal atau bagian atas rantai yang dapat memberikan gerakan dinamis seperti rantai pada umumnya. *Controller* utama pada rantai tersebut memiliki pengaturan atribut *stiff* 0.5, *damp* 0.7, dan *start* 1. Objek mesh yang sudah di-*bind skin* dengan *joint* terlihat keluar dari *controller* yang memiliki bentuk bulat. Hal ini terjadi karena kalkulasi pergerakan yang dilakukan dari atas rantai sehingga bagian bawah rantai memiliki nilai *damp* dan *stiff* kurang dari 1. Hasil analisa untuk benda seperti rantai akan memberikan kesan lebih nyata jika nilai *stiff* rendah dari pada *damp*.



Gambar 15. *Dynamic Hair Frame 1 to 5* (Sumber, Penulis)

Pada gambar 15, dynamic rig dapat membuat pergerakan rambut dengan menggerakkan bagian kepala, sehingga rambut akan mengikuti sesuai pergerakan kepala tersebut. Posisi dinamik yang berada pada kiri dan kanan akan memberikan perubahan pergerakan yang berbeda sehingga dapat digunakan tidak hanya titik tengah dari objek, namun bisa juga pada bagian kiri dan kanan. Setiap induk dari dinamik memiliki pengaturan atribut stiff 0.5, damp 0.5, dan start 1. Dikarenakan diatur tepat di antara 0 dan 1, serta rambut tidak telalu panjang atau pendek, efek pergerakan yang diberikan juga tidak terlalu kaku ataupun terlalu relaks. Berdasarkan hasil analisa tersebut objek seperti rambut bisa diberikan nilai stiff dan damp yang sama atau mendekati satu dengan yang lainnya tergantung kepanjangan rambut.

Berdasarkan implementasi di atas, script yang dibuat oleh Brown pada aplikasi 3D Studio Max dapat diterapkan pada aplikasi 3D Maya. Hal ini juga sesuai dengan penelitian Wyndham Batton yang mengadaptasi script tersebut pada aplikasi 3D Maya dan diterapkan pada objek rambut dan rantai [3]. Sedangkan pada penelitian ini, objek penerapan ditambah dengan adanya tentakel. Penggunaan 3 jenis objek pada penelitian ini menunjukkan hasil pergerakan dynamic yang berbeda-beda sesuai dengan karakteristik objek, yang diatur pada atribut dynamic rigging tool, seperti atribut stiff dan damp.

Kesimpulan

Dengan menggunakan prinsip dinamik yang telah dibuat Batton, W. dan Brown Bag Films serta memanfaatkan *node expression* yang digunakan sebagai kalkulasi gerakan dinamis dapat memberikan hasil yang serupa dengan dinamik yang telah dibuat oleh Brown Bag Films. Percobaan pada tiga jenis objek yaitu tentakel, rantai dan rambut memberikan efek yang berbeda sesuai dengan pengaturan nilai *stiff* dan *damp.*

Ucapan Terima Kasih

Ucapan terima kasih kepada seluruh tim yang telah berkontribusi dalam studi ini, secara khusus kepada Res Yudikata dan Mohammad Esa Sangkala sebagai tim rigging Infinite Framework Studios.

Daftar Pustaka

[1]. Simmons, M., Ward, K., Yosumi, H., Leo, H., & Zhao, X. (2011, April 16). Directing Hair Motion on Tangled. *SIGGRAPH '11: Special Interest Group on Computer Graphics and Interactive Techniques Conference.* New York: Association for Computing Machinery. Retrieved from ACM Digital Library.

[2]. Chai, M., Zheng, C., & Zhou, K. (2014). A Reduced Model for Interactive Hairs. *ACM Transactions on Graphics.* ACM Journal

[3]. Batton, W. (2016). *A Dynamic Hair Rigging System for Maya.* South Carolina: TigerPrints Clemson University.

[4]. Brown Bag Labs. (2013, May 21). *Setting up a Dynamics Rig System – 3D Studio Max*. Retrieved from Brown Bag Films: https://www.brownbagfilms.com/labs/entry/setting-up-a-dynamics-rig-system-3d-studio-max

[5]. Lanier, L. (2016). *Advanced Maya Texturing and Lighting.* Canada: Wiley.

[6]. Patil, G. V., & Deshpande, S. L. (2016). Distributed rendering system for 3D animations with Blender. *IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)* (pp. 91-98). Pune, India: IEEE.

[7]. Arshad, M. R., Yoon, K. H., Manaf, A. A., & Mohamed, M. A. (2019). Physical Rigging Procedures Based on Character. *International Journal of Recent Technology and Engineering (IJRTE)*, 4138-4147.

.