

BAB III

METODOLOGI

3.1. Gambaran Umum

Perancangan yang akan dibuat penulis adalah perancangan sistem *rig* untuk membuat variasi 3D *model* tubuh manusia dengan dukungan *MAXscript*. Perancangan ini menjadi salah satu komponen penting dalam proses produksi dari film animasi pendek yang berjudul “*Light Up*”. Film ini menggunakan tema pertunjukan, oleh karena itu penonton dalam jumlah banyak akan diperlukan agar mampu memperlihatkan perubahan suasana selama cerita berlangsung. Objek penelitian dalam perancangan yang dilakukan penulis adalah variasi tubuh manusia dan sistem *rig* yang didukung dengan *MAXscript*. Proses perancangan dilakukan dengan menggali serta mengobservasi berbagai macam data mengenai tubuh manusia dan acuan teknis untuk merealisasikannya dalam bentuk variasi 3D *model* manusia.

3.1.1. Sinopsis

Film pendek animasi “*Light Up*” bercerita mengenai kisah Aruna, putri dari seorang penari. Tragedi yang menimpa orang tuanya dan kecacatan yang diterima pasca kerusuhan di Kota Tua membuatnya kehilangan impian sebagai seorang penari yang sempurna. Terkadang Aruna mencoba untuk terus mengejar impiannya namun situasi lingkungan yang tidak mendukung hanya mengikis harapannya semakin tipis. Impiannya runtuh ketika ia dicaci maki dan dilempar batu ketika ia berjalan di alun-alun kota. Dititik semua harapan dan impian

tertutup aruna menemukan sebuah topeng mistis yang terbaring diantara reruntuhan. Aruna yang termakan rasa penasaran memakai topeng tersebut dan dengan seketika topeng tersebut mengabulkan semua impian yang ia harapkan yaitu kesempurnaan. Dengan dukungan topeng tersebut Aruna berlari dan menari kembali di alun-alun kota. Masyarakat yang melihat keindahan Aruna bersemangat dan menyoraki Aruna. Aruna yang menari dengan bersemangat tidak sadar akan topengnya yang sedikit demi sedikit terlepas dan pada akhirnya terjatuh pecah. Hilangnya semua kekuatan topeng tersebut mengembalikan Aruna ke kondisinya yang tidak sempurna. Ditengah-tengah kepanikan dalam benak Aruna penonton sedikit demi sedikit memberi semangat untuk kembali menari. Pada titik ini Aruna sadar bahwa kesempurnaan bukanlah hal yang selama ini ingin ia dapatkan melain jati dirinya sebagai seorang penari.

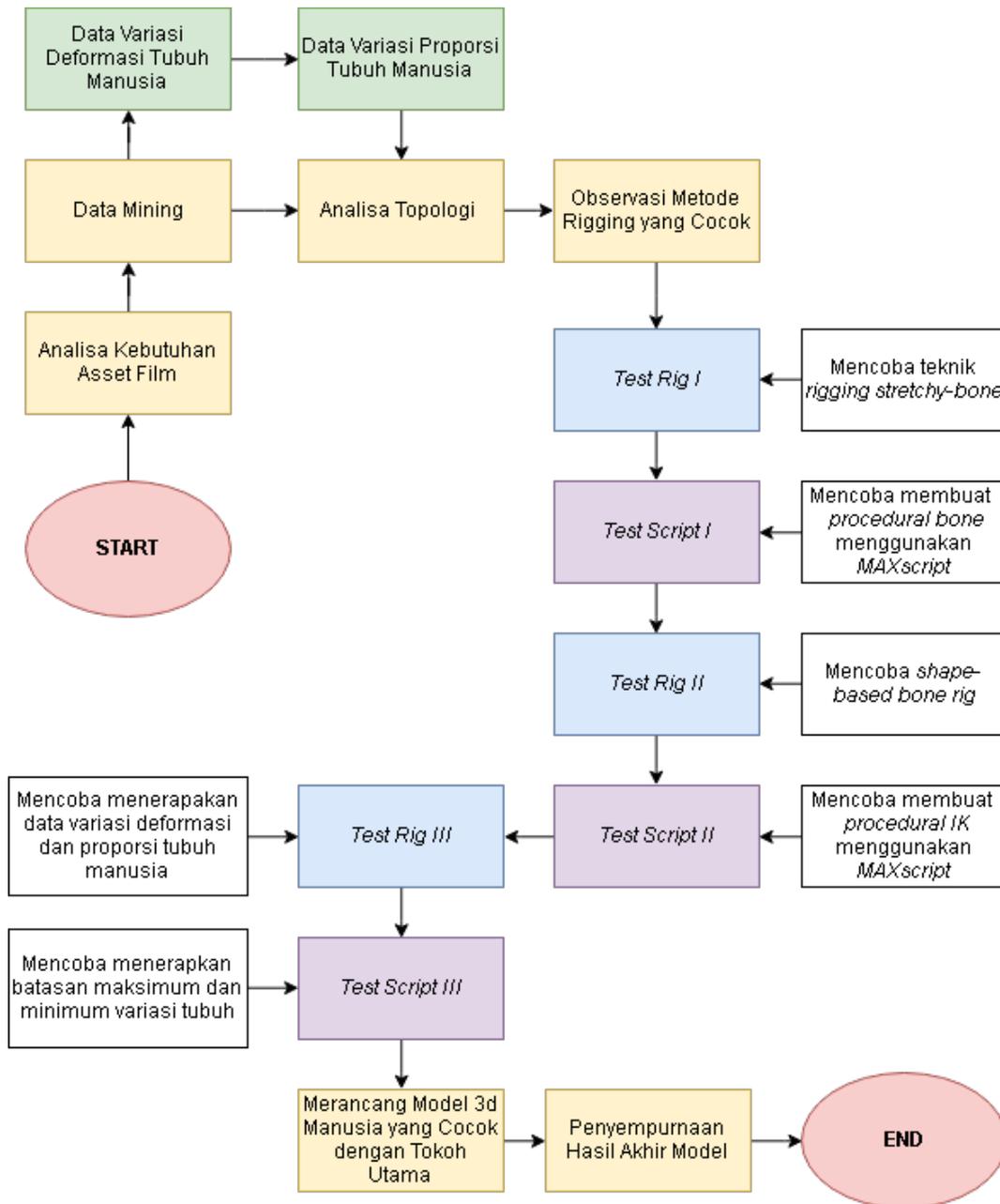
3.1.2. Posisi Penulis

Selama proses pembuatan film animasi pendek “*Light up*”, penulis berperan sebagai *director*, *technical artist*, *character modeler* dan *rigger*. Tanggung jawab penulis adalah sebagai *rigger* dengan menitik beratkan *crowd production* dan *rig*.

3.2. Tahapan Kerja

Selama Proses perancangan sistem *rig* yang didukung dengan *MAXscript*, penulis perlu menyiapkan serangkaian tahapan. Tahapan tersebut dimulai dari pencarian dan observasi hal-hal yang berkaitan variasi tubuh manusia, pemberian *bone* beserta *control* dan *skinning*. Semua tahapan diatas akan didukung dengan *MAXscript* dengan tujuan pembuatan sistem otomatis dan *user interface (UI)*. Selain tahapan-tahapan diatas penulis juga melakukan eksperimen dengan metode

lain untuk mencapai tujuan yang sama agar mampu membandingkan kelebihan dan kekurangan masing-masing metode.



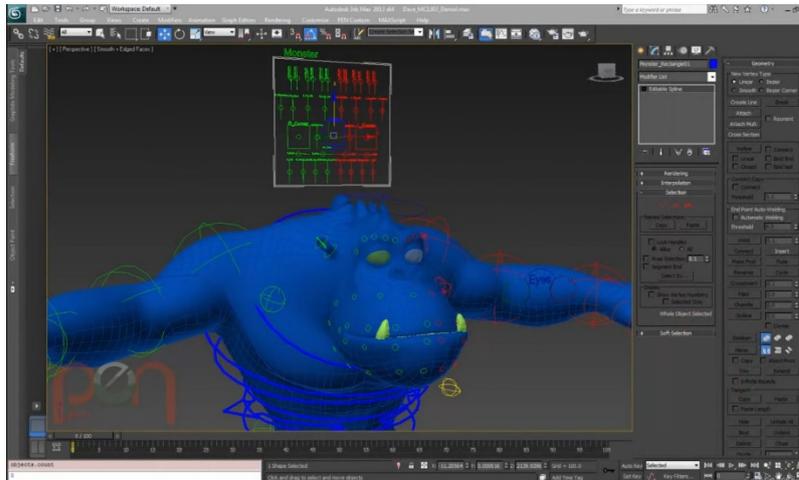
Gambar 3.1. Skematika Perancangan
(sumber: dokumentasi pribadi)

3.3. Acuan

Untuk menghasilkan sebuah sistem *rig* yang mampu membuat variasi 3D *model* manusia, penulis menggunakan beberapa acuan. Acuan yang dimaksud antara lain acuan teknik *rigging*, acuan hasil yang ingin dicapai serta acuan mengolah variasi dan proporsi.

3.3.1. Point Based Rigging

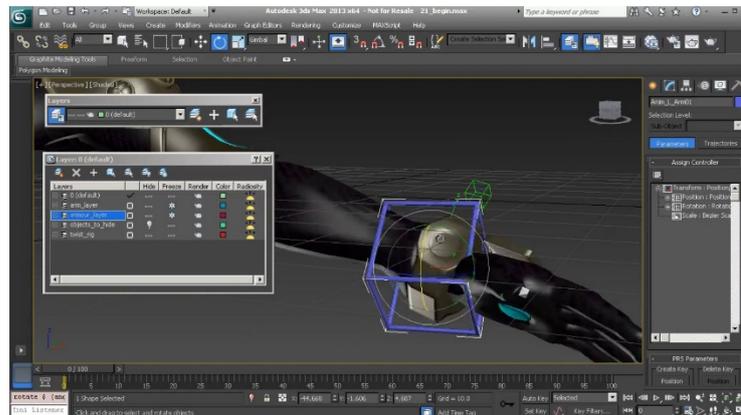
Acuan teknis *rigging* diambil dari *rigging* yang dibuat oleh Paul Neale dari Pen Production dan Delano Athias dari Pluralsight. *Rig* Paul Neale menggunakan teknik penempatan *bone* dan *skinning* menggunakan *points* dan *shapes* dengan penambahan *modifier poly select* pada *shapes*. Pengamplikasian pada *rig*, terdapat pada deformasi muka *monster*. Teknik ini dipilih penulis dengan alasan mengurangi kerumitan penempatan *bone* secara banyak. Selain itu teknik ini juga memberikan kebebasan lebih pada *control* karena *controller* mampu di-*resize*, *rotate* dan *translate*. Acuan ini menjadi cocok dengan tujuan penulis karena dengan teknik ini penempatan *bone* dalam bentuk *shape* dapat diletakkan dengan mudah dan bebas pada bagian-bagian tubuh yang ingin di deformasi.



Gambar 3.2. *Rigging* oleh Paul Neale
(sumber: <http://paulneale.com/tutorials/>)

3.3.2. Rigging

Acuan teknis *rigging* lainnya yaitu pembuatan *rig* oleh Delano Athias dari Pluralsight. *Rig* yang dijelaskan oleh Athias merupakan sistem dasar *rigging* di *3DS Max* yakni penerapan *IK*, *FK*, *constraint*, penempatan *bone* serta skinning. Alasan penulis memilih acuan teknik *rig* ini ialah karena tokoh *crowd* pada film animasi yang akan dibuat tidak perlu memiliki *rig* yang terlampaui kompleks. Hal ini sesuai dengan acuan *rig* dari Athias yang menjelaskan dasar-dasar dari *rigging* di *3DS Max*. Menggunakan acuan ini penulis mampu menerapkan sistem *rig* dasar pada sistem yang dibuat penulis.

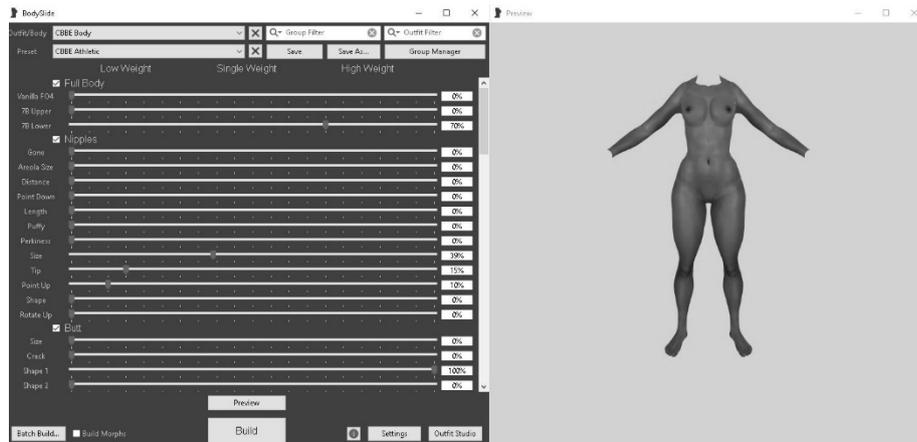


Gambar 3.3 *Rigging* oleh Delano Athias

(<https://www.pluralsight.com/>)

3.3.3. User Interface

Acuan karya akhir yang ingin dicapai diambil dari sebuah *game modification* yang bernama *Caliente's Body Slider*. Acuan ini memiliki puluhan parameter yang mampu memberikan variasi pada tubuh tokoh utama dalam *game* tersebut. Variasi ini dapat diaplikasikan pada berbagai macam *meshes* serta mampu mengubah warna pigmen dan menambahkan beberapa fitur tambahan. Penerapan dari *modification* ini terdapat pada *game Fallout 4* dimana didalamnya juga terdapat acuan untuk melakukan variasi diluar dari *parameter* yang sudah dibuat. Dengan kata lain, variasi dapat diubah oleh pemain sesuai dengan preferensi yang diinginkan.



Gambar 3.4. *Caliente's Body Slider*

(sumber: dokumentasi pribadi)

3.3.4. Pemrosesan Data

Acuan pemrosesan variasi serta proporsi tubuh, pada 3D *model* manusia yang akan dibuat menjadi acuan penulis dalam merancang karya ini. Hal ini dikarenakan data proporsi serta variasi yang didapat dalam dunia nyata tidak cocok dengan prinsip-prinsip dasar animasi yang baik. Acuan ini menjadi penting agar antara desain tokoh utama, masyarakat disekitarnya dan lingkungan terlihat berbaur. Dengan demikian acuan yang penulis gunakan antara lain film “Moana” oleh Walt Disney Studio, *Overwatch Animated shorts*, serta iklan milik usaha konsultan *Better Help* yang berjudul, “*Crumbling Beneath Your Feet*”. Acuan-acuan ini dipilih dengan alasan memiliki gaya *modeling* dan *render* yang mirip dengan apa yang ingin dicapai di film animasi pendek “*Light Up*”.



Gambar 3.5. Iklan “Crumbling Beneath Your Feet” oleh *Better Help*

(sumber: <https://www.youtube.com/watch?v=KCFsK61FDzA>)

3.3.5. Data Anthropometri Tubuh

Referensi data batasan maksimum dan minimum untuk variasi bentuk tubuh. Data ini merupakan data pengukuran bagian tubuh dari 377 orang Indonesia yang didalamnya terdiri dari 245 pria dan 132 wanita. Dengan adanya referensi data ini penulis memiliki dasar batasan yang ingin dicapai.

Table 3
Anthropometric data for Indonesian males and females (all dimensions in cm, body weight in kg).

| Dimension | Male citizens | | | | Male Chinese | | | | Female citizens | | | | Female Chinese | | | |
|------------------------------------|---------------|------|-------|-------|--------------|------|-------|-------|-----------------|------|------|-------|----------------|------|-------|-------|
| | 5th | 50th | 95th | SD | 5th | 50th | 95th | SD | 5th | 50th | 95th | SD | 5th | 50th | 95th | SD |
| 1. Stature | 162 | 172 | 183 | 6.23 | 165 | 171 | 180 | 4.81 | 150 | 159 | 169 | 5.76 | 151 | 159 | 166 | 5.06 |
| 2. Eye height | 151 | 160 | 172 | 6.3 | 153 | 160 | 169 | 5.08 | 139 | 148 | 158 | 6.12 | 137 | 146 | 158 | 6.73 |
| 3. Shoulder height | 134 | 143 | 155 | 6.41 | 134 | 143 | 151 | 5.05 | 123 | 132 | 141 | 5.91 | 123 | 132 | 139 | 5.43 |
| 4. Elbow height | 99 | 107 | 114 | 5.12 | 99 | 106 | 112 | 4.29 | 91 | 99 | 108 | 6.4 | 92 | 98 | 107 | 5.35 |
| 5. Hip height | 83 | 95 | 105 | 6.76 | 81 | 94 | 103 | 6.48 | 78 | 88 | 97 | 5.91 | 79 | 90 | 96 | 5.68 |
| 6. Knuckle height | 68 | 75 | 82 | 4.75 | 69 | 74 | 80 | 5.13 | 63 | 70 | 78 | 4.37 | 64 | 69 | 77 | 3.89 |
| 7. Fingertip height | 58 | 64 | 71 | 4.82 | 59 | 64 | 70 | 5.13 | 54 | 60 | 65 | 3.67 | 53 | 60 | 68 | 3.99 |
| 8. Sitting height | 80 | 89 | 96 | 5.24 | 85 | 90 | 96 | 6.55 | 78 | 83 | 90 | 4.7 | 79 | 84 | 88 | 2.97 |
| 9. Sitting eye height | 69 | 76 | 84 | 4.58 | 72 | 78 | 85 | 6.54 | 67 | 73 | 80 | 5.83 | 68 | 72 | 79 | 3.64 |
| 10. Sitting shoulder height | 52 | 59 | 67 | 6.27 | 55 | 61 | 72 | 7.15 | 51 | 56 | 63 | 4.94 | 52 | 57 | 64 | 3.67 |
| 11. Sitting elbow height | 19 | 24 | 30 | 4.74 | 19 | 25 | 31 | 7.13 | 19 | 25 | 32 | 5.19 | 21 | 24 | 30 | 3.24 |
| 12. Thigh thickness | 12 | 16 | 22 | 3.59 | 13 | 16 | 20 | 2.76 | 11 | 15 | 19 | 3.22 | 12 | 15 | 19 | 2.81 |
| 13. Buttock-knee length | 48 | 56 | 64 | 4.89 | 49 | 57 | 64 | 4.83 | 45 | 53 | 60 | 4.81 | 48 | 53 | 60 | 4.06 |
| 14. Buttock-popliteal length | 40 | 46 | 54 | 4.82 | 38 | 47 | 56 | 5.36 | 37 | 43 | 51 | 4.21 | 39 | 44 | 52 | 3.97 |
| 15. Knee height | 46 | 54 | 62 | 5.21 | 44 | 53 | 61 | 5.65 | 43 | 50 | 60 | 5.27 | 42 | 49 | 60 | 5.38 |
| 16. Popliteal height | 38 | 44 | 49 | 3.78 | 36 | 44 | 50 | 5.36 | 38 | 44 | 50 | 3.92 | 36 | 43 | 47 | 3.85 |
| 17. Shoulder breadth (bideltoid) | 36 | 45 | 52 | 4.66 | 38 | 45 | 50 | 4.6 | 37 | 43 | 53 | 5.43 | 40 | 44 | 53 | 4.97 |
| 18. Shoulder breadth (biacromial) | 31 | 37 | 43 | 3.61 | 33 | 38 | 44 | 3.83 | 33 | 38 | 44 | 3.56 | 34 | 38 | 44 | 3.18 |
| 19. Hip breadth | 28 | 35 | 43 | 4.41 | 30 | 35 | 44 | 4.09 | 29 | 35 | 45 | 7.22 | 30 | 34 | 42 | 4.21 |
| 20. Chest (bust) depth | 16 | 21 | 27 | 3.5 | 17 | 22 | 27 | 4.02 | 17 | 21 | 28 | 3.38 | 19 | 23 | 28 | 3.61 |
| 21. Abdominal depth | 15 | 21 | 29 | 4.46 | 15 | 21 | 30 | 5.19 | 14 | 18 | 25 | 3.44 | 15 | 20 | 26 | 3.93 |
| 22. Shoulder-elbow length | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA |
| 23. Elbow-fingertip length | 42 | 47 | 56 | 4.55 | 41 | 46 | 53 | 4.27 | 37 | 43 | 50 | 4.27 | 37 | 42 | 47 | 3.72 |
| 24. Upper limb length | 68 | 76 | 84 | 6.39 | 68 | 75 | 85 | 5.06 | 62 | 70 | 77 | 4.69 | 64 | 68 | 74 | 3.92 |
| 25. Shoulder-grip length | 56 | 65 | 73 | 6.29 | 59 | 66 | 74 | 5.13 | 54 | 60 | 68 | 4.3 | 54 | 60 | 68 | 4.64 |
| 26. Head length | 17 | 20 | 24 | 2.21 | 17 | 20 | 24 | 2.58 | 15 | 18 | 22 | 3.95 | 15 | 19 | 22 | 2.13 |
| 27. Head breadth | 15 | 18 | 22 | 2.06 | 15 | 18 | 21 | 1.89 | 14 | 17 | 21 | 2.48 | 14 | 18 | 21 | 2.11 |
| 28. Hand length | 17 | 19 | 22 | 1.64 | 15 | 19 | 22 | 2.42 | 16 | 18 | 20 | 1.72 | 17 | 18 | 20 | 2.16 |
| 29. Hand breadth | 7 | 9 | 11 | 1.09 | 8 | 9 | 11 | 0.89 | 6 | 8 | 10 | 4.85 | 6 | 8 | 9 | 0.73 |
| 30. Foot length | 22 | 25 | 29 | 2.58 | 11 | 25 | 28 | 4.43 | 21 | 23 | 26 | 2.63 | 21 | 23 | 26 | 2.3 |
| 31. Foot breadth | 8 | 10 | 12 | 3.96 | 8 | 10 | 12 | 1.16 | 7 | 9 | 11 | 2.2 | 7 | 9 | 10 | 1.08 |
| 32. Span | 158 | 172 | 186 | 8.5 | 155 | 171 | 182 | 8.73 | 146 | 156 | 170 | 7.61 | 150 | 159 | 168 | 6.52 |
| 33. Elbow span | 78 | 86 | 96 | 5.97 | 79 | 87 | 94 | 4.36 | 73 | 79 | 89 | 5.38 | 73 | 81 | 88 | 4.53 |
| 34. Vertical grip reach (standing) | 192 | 206 | 221 | 10.54 | 197 | 206 | 222 | 7.74 | 174 | 186 | 204 | 9.1 | 176 | 189 | 202 | 8.07 |
| 35. Vertical grip reach (sitting) | 112 | 122 | 136 | 7.9 | 116 | 123 | 130 | 5.18 | 101 | 113 | 124 | 7.2 | 106 | 115 | 128 | 10.25 |
| 36. Forward grip reach | 64 | 73 | 81 | 5.89 | 66 | 74 | 81 | 4.7 | 61 | 67 | 76 | 4.39 | 60 | 67 | 74 | 4.76 |
| 37. Body weight (kg) | 50 | 63 | 89.25 | 13.19 | 53.05 | 63 | 93.45 | 13.35 | 39.80 | 53 | 80 | 11.68 | 41.90 | 55 | 70.40 | 9.49 |

Gambar 3.6. Data Anthropometri Manusia

(sumber: *International Journal of Industrial Ergonomics*)

Untuk memperkuat data diatas penulis juga menggunakan data lingkaran tubuh. Data yang digunakan ialah hasil riset korelasi *Body Mass Index (BMI)* dengan lingkaran bagian tubuh 193 wanita dengan jenjang umur 20 tahun hingga 83 tahun. Menggunakan rumus “Keliling = $2 \times \pi \times \text{jari-jari}(r)$ ” penulis mendapatkan data dalam bentuk jari-jari dari lingkaran tubuh yang terdapat dalam data. Jari-jari ini kemudian dikali untuk menemukan diameter lingkaran tubuh. Diameter ini yang digunakan penulis sebagai salah satu referensi penentu batasan dalam *slider*.

TABLE 1: Anthropometric parameters of the study population.

| Participants | All (n = 193) | Normal (n = 61) | Overweight (n = 68) | Obese (n = 64) |
|-------------------------------|------------------------|------------------------|------------------------|------------------------|
| Parameter | | | | |
| Age | 48.9 (46.8–51.0) | 44.5 (41.2–48.1) | 52.9 (49.8–56.1) | 49.1 (45.0–53.7) |
| Weight (kg) | 71.1 (69.2–73.1) | 58.3 (56.9–59.7) | 69.5 (67.9–71.0) | 88.1 (85.3–91.1) |
| Height (cm) | 160.0 (159.2–160.8) | 161.4 (160.1–162.6) | 159.1 (157.8–160.4) | 159.7 (157.9–161.5) |
| BMI | 27.8 (27.0–28.6) | 22.4 (21.9–22.9) | 27.5 (27.1–27.8) | 34.6 (33.7–35.5) |
| Circumference of (cm) | | | | |
| Neck | 33.8 (33.5–34.1) | 31.9 (31.5–32.2) | 33.6 (33.3–34.0) | 35.9 (35.4–36.5) |
| Waist | 91.1 (89.3–92.9) | 79.8 (78.0–81.7) | 89.6 (87.9–91.4) | 105.0 (102.4–107.6) |
| Hip | 107.1 (105.6–108.6) | 97.8 (96.7–98.9) | 105.0 (103.8–106.3) | 119.2 (116.8–121.6) |
| Arm | 26.5 (26.1–27.0) | 23.8 (23.5–24.2) | 26.5 (26.1–26.9) | 29.4 (28.7–30.2) |
| Forearm | 24.90 (24.6–25.2) | 23.1 (22.8–23.5) | 24.8 (24.5–25.1) | 26.8 (26.3–27.5) |
| Wrist | 16.0 (15.8–16.2) | 15.1 (14.9–15.4) | 15.9 (15.7–16.2) | 16.9 (16.5–17.2) |
| Thigh | 45.3 (44.6–46.1) | 41.5 (40.6–42.4) | 44.4 (43.6–45.3) | 50.4 (49.2–51.7) |
| Ankle | 23.2 (22.9–23.5) | 21.8 (21.4–22.2) | 23.1 (22.7–23.5) | 24.8 (24.3–25.4) |
| Waist/hip ratio | 0.85 (0.84–0.86) | 0.82 (0.80–0.83) | 0.85 (0.84–0.87) | 0.88 (0.86–0.90) |
| Mean circumference of (cm) | | | | |
| ABC ^a | 46.1 (45.5–46.7) | 41.9 (41.5–42.4) | 45.4 (45.0–45.9) | 51.2 (50.4–52.0) |
| UBC ^b | 38.5 (38.0–39.1) | 34.8 (34.3–35.3) | 38.1 (37.7–38.6) | 42.9 (42.1–43.6) |
| LBC ^c | 58.6 (57.8–59.4) | 53.8 (53.2–54.4) | 57.6 (56.9–58.2) | 64.9 (63.7–66.1) |
| (N + W + H + A + Th)/5 | 60.9 (60.0–61.7) | 55.0 (54.4–55.7) | 59.9 (59.3–60.5) | 68.1 (67.0–69.2) |
| (Waist + hip + arm + thigh)/4 | 68.3 (67.1–69.6) | 60.8 (60.1–61.6) | 66.5 (65.8–67.2) | 76.1 (74.8–77.5) |
| (Neck + waist + hip)/3 | 77.4 (76.3–78.6) | 69.9 (69.0–70.8) | 76.2 (75.4–77.0) | 86.8 (85.3–88.3) |
| (Waist + hip)/2 | 99.2 (97.6–100.8) | 88.9 (87.6–90.2) | 97.4 (96.3–98.6) | 112.2 (110.2–114.3) |

Normal = women with BMI ≤ 24.9 . Overweight = women with BMI 25–29.9. Obese = women with BMI ≥ 30 .

^aABC, average body circumference, average of (neck + waist + hip + arm + forearm + wrist + thigh + ankle).

^bUBC, upper body circumference, average of (neck + waist + arm + forearm + wrist).

^cLBC, lower body circumference, average of (hip + thigh + ankle).

N = neck, W = waist, H = hip, A = arm, F = forearm, Wr = wrist, Th = thigh, and Ank = ankle. Geometric means (95% CI) are presented.

Gambar 3.7. Data Lingkar Tubuh Orang Indonesia

(sumber: research Article, Hindawi Publishing Corporation)

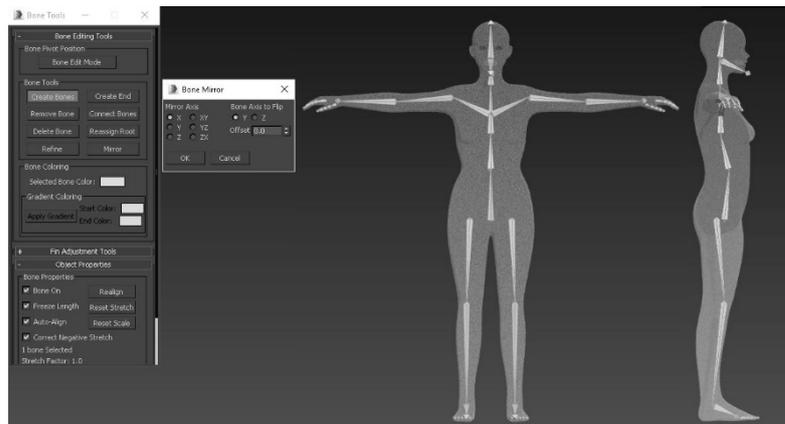
3.4. Proses Perancangan

Dalam merancang sistem *rig* untuk menciptakan variasi pada *3D model* manusia, penulis perlu melewati rangkaian proses. Proses yang dimaksud adalah, penempatan *bone*, penempatan *control*, dan *skinning*. Selanjutnya penulis

menyiapkan komponen-komponen menggunakan *reaction manager* yang nantinya menjadi data masukan bagi *MAXscript*. Terakhir ialah perancangan *script* dengan bahasa *MAXscript* untuk mengolah semua data dan *rig* yang dibuat untuk menghasilkan sebuah *3D model* manusia yang siap untuk dianimasi. Proses ini juga mencakup pembuatan *user interface* dari sistem *rig*.

3.4.1. Penempatan Bone

Penulis melakukan perancangan peletakan *bone* dasar yang akan digunakan untuk keperluan animasi. Penempatan *bone* yang dibuat mengikuti acuan *rig* yang dibuat oleh Delano Athias dimana, *bone* pertama kali dibuat dari pinggang hingga ke kepala dan rahang, lalu dilanjutkan pembuatan *bone* untuk tangan dan kaki. Karena tangan dan kaki merupakan bagian tubuh *appendicular* seperti yang dijelaskan oleh Assefa dan Tsige, maka *bone* yang dibuat perlu dicerminkan menggunakan tombol *mirror* yang berada pada *bone tool*. Setelah semua *bone* selesai disiapkan, masing-masing *bone* ditempatkan pada bagian tengah *geometry*. Nantinya khusus untuk bagian kaki dan tangan akan digerakan menggunakan teknik *orientation constraint* dan *position constraint* terhadap *bone Forward Kinematic (FK)* dan *Inverse Kinematic (IK)*.



Gambar 3.8. Penempatan *Bone*

(sumber: dokumentasi pribadi)

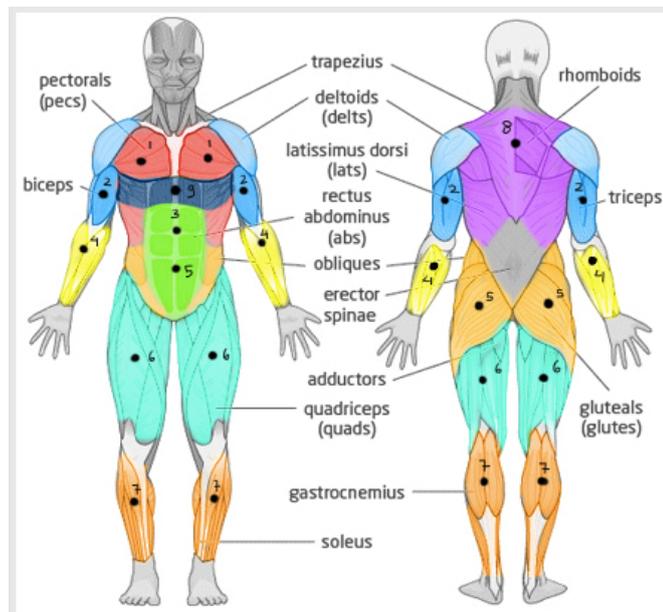
Setelah perancangan *bone* dasar selesai, penulis menyiapkan *bone FK* yang berfungsi untuk animasi dan juga variasi proporsi tubuh. *Bone* ini hanya dibuat pada bagian lengan atas, lengan bawah, paha dan betis dan memiliki posisi sama persis dengan *bone* dasar. Hal ini dilakukan penulis berdasarkan eksperimen yang dilakukan, bahwa ketika penempatan posisi dan orientasi *bone* dasar dan *bone FK* tidak sesuai, maka translasi *bone dasar* terhadap *bone FK* maupun *IK* menjadi kurang akurat dan menyebabkan *rig* yang dibuat tidak mampu melakukan sistem *IK FK switch* yang akan digunakan untuk keperluan animasi. Selain itu hal ini menjadi penting dikarenakan kesamaan posisi dan orientasi merupakan syarat pemberian *look at constraint* untuk keperluan deformasi proporsi nantinya.

Setelah *bone* dasar dan *bone FK* selesai, penulis tidak membuat *bone IK*. Keputusan ini diambil penulis atas dasar eksperimen untuk mengetahui kecocokan sistem *stretchy bone* pada *bone FK* dengan sistem *IK* terhadap *bone dasar*. Melalui proses tersebut penulis menemukan bahwa *bone* dasar mampu menyesuaikan perubahan apapun terhadap *bone FK* namun tidak mampu

berpindah ke sistem *IK* dengan masalah serupa diatas yaitu perpindahan yang tidak akurat karena posisi ketiga *bone* tersebut tidak sama. Maka dari itu nantinya pembuatan *bone IK* dibuat menggunakan bantuan *MAXscript*.

3.4.2. Analisa Penempatan Shape-based Bone pada Tubuh

Sebelum melanjutkan ke penempatan *shape-based bone*, penulis melakukan analisa bagian-bagian tubuh yang akan diberikan variasi. Didasari teori dari Assefa dan Tsige, bahwa otot merupakan komponen penggerak tubuh dan juga sebagai pembentuk tubuh selain lemak. Maka penulis membagi bagian-bagian tubuh berdasarkan kelompok otot *superficial* yang berarti otot-otot, yang dapat dirasakan tepat dibawah kulit. Kemudian penulis mencoba mengelompokan otot-otot manusia berdasarkan bagian-bagian tubuh serta menandai bagian-bagian yang akan dideformasi.



Gambar 3.9. Kelompok Otot Superficialis

(sumber: dokumentasi pribadi)

Tabel 3.1. Nomor Kelompok dan Nama Otot Tiap Variasi

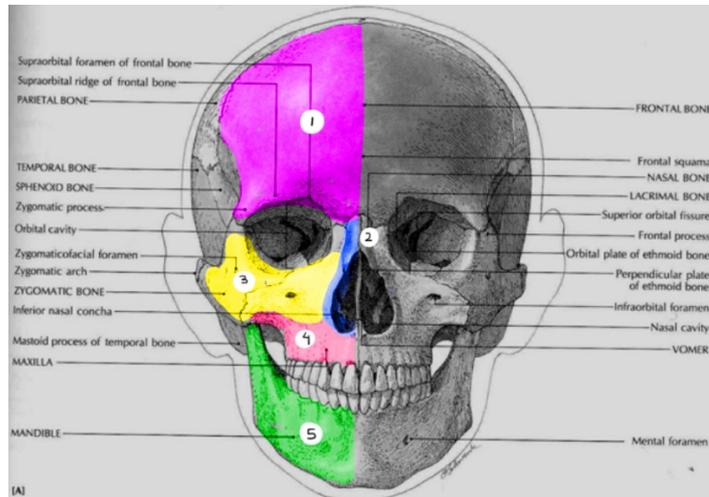
| Nomor Kelompok | Nama Otot | Deskripsi |
|----------------|---|---|
| 1 | <ul style="list-style-type: none"> • Pectoral Major | Kelompok otot ini akan dinamai sebagai variasi <i>chest thickness</i> |
| 2 | <ul style="list-style-type: none"> • Triceps brachii • Biceps brachii • Deltoid | Kelompok otot ini akan dinamai sebagai variasi <i>upper arm thickness</i> |
| 3 | <ul style="list-style-type: none"> • Rectus abdominis | Kelompok otot ini akan dinamai sebagai variasi <i>belly</i> |
| 4 | <ul style="list-style-type: none"> • Extensor carpi radialis longus • Extensor carpi ulnaris • Brachioradialis • Flexors of wrist and fingers | Kelompok otot ini akan dinamai sebagai variasi <i>lower arm thickness</i> |
| 5 | <ul style="list-style-type: none"> • Gluteus medius • Gluteus maximus | Kelompok otot ini akan dinamai sebagai variasi <i>hip width</i> |
| 6 | <ul style="list-style-type: none"> • Sartorius • Rectus femoris • Vastus lateralis | Kelompok otot ini akan dinamai sebagai variasi <i>thigh thickness</i> |

| | | |
|---|--|--|
| | <ul style="list-style-type: none"> • Vastus medialis • Iliopsoas • Pectineus • Adductor longus • Adductor magnus • Gracilis • Semitendinosus • Vastus lateralis • Semimembranosus • Biceps femoris | |
| 7 | <ul style="list-style-type: none"> • Tibialis anterior • Peroneus longus • Gastrocnemius • Soleus | Kelompok otot ini akan dinamai sebagai variasi <i>calf thickness</i> |
| 8 | <ul style="list-style-type: none"> • Teres minor • Teres Major • Trapezius • Latissimus dorsi | Kelompok otot ini akan dinamai sebagai variasi <i>upper back thickness</i> |
| 9 | <ul style="list-style-type: none"> • Serratus anterior • Latissimus dorsi | Kelompok otot ini akan dinamai sebagai variasi <i>abdomen thickness</i> |

Dilanjutkan dengan analisa komponen pembentuk wajah manusia.

Menurut Assefa dan Tsige, tulang berperan untuk pembentukan wajah dan juga

sebagai tempat melekatnya otot. Maka dari itu penulis mengamati tulang manusia pada bagian wajah dan menandai komponen pembentuk wajah manusia.



Gambar 3.10. Faktor Pembentuk wajah Manusia
(sumber: Human Anatomy and Physiology, Assefa & Tsige)

Tabel 3.2. Daftar Tulang Pembentuk Wajah

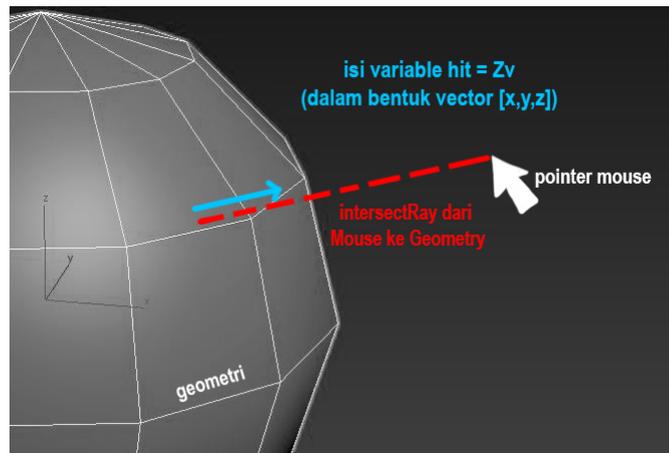
| Nomor | Nama Tulang | Deskripsi |
|-------|--|---|
| 1 | Frontal | Frontal merupakan tulang yang membentuk bagian kening dan nantinya akan menjadi variasi <i>Eye Size</i> |
| 2 | <ul style="list-style-type: none"> • Naval • Lateral, Septal, Aral cartilage | Naval merupakan tulang yang memberikan bentuk dari jembatan hidung. Sedangkan hidung akan |

| | | |
|---|-----------|---|
| | | dibentuk dari tulang lunak atau cartilage bagian ini akan menjadi deformasi <i>nose size</i> . |
| 3 | Zygomatic | Zygomatic merupakan tulang yang membentuk bagian kening dan nantinya akan menjadi variasi <i>Eye Size</i> dan <i>cheek size</i> |
| 4 | Maxilla | Maxilla akan menentukan bentuk dari rahang atas dan nantinya mendasari variasi <i>jaw width</i> dan <i>size</i> . |
| 5 | Mandible | Mandible menjadi dasar utama variasi <i>jaw width</i> dan <i>size</i> . |

3.4.3. Penempatan Shape-based Bone pada Tubuh

Setelah mengetahui bagian-bagian tubuh yang akan diberi deformasi, penulis merancang penempatan *bone* untuk melakukan deformasi pada sistem, berupa *shape* yang diberi *modifier poly select*. Pemberian *modifier poly select* terhadap *shape* ini menjadi penting karena tanpa memberikannya, proses *skinning* tidak dapat dilanjutkan karena ketidakmampuan *modifier skin* untuk membaca *shape* sebagai *bone*. *Shape-based bone* ini kemudian ditempatkan sesuai dengan pengelompokan otot *superficialis*.

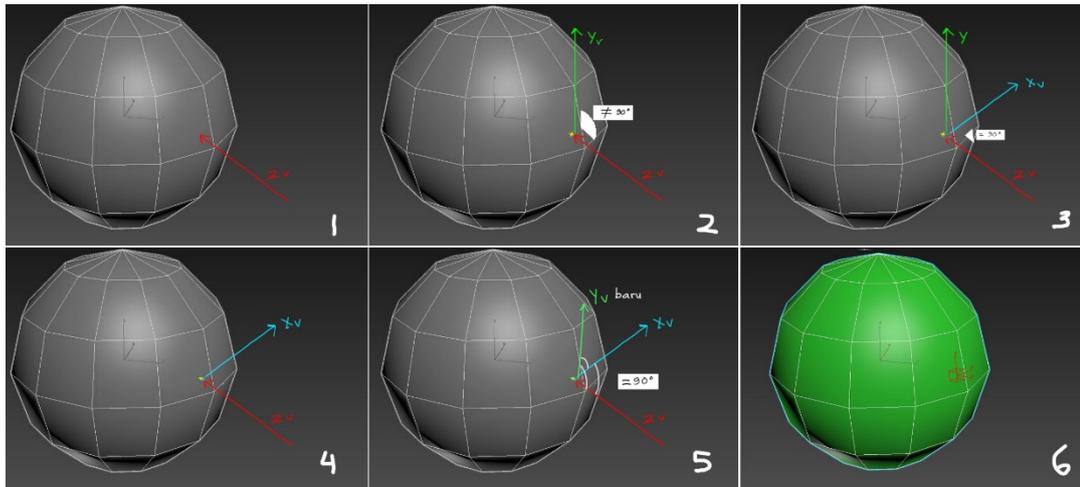
Untuk mempermudah serta meningkatkan keakuratan penempatan *shape-based bone* untuk wajah, penulis menulis *script* yang mengacu pada *script* yang diciptakan oleh Paul Neale. Fungsi dari *script* ini ialah menempatkan *helper* dan *shape-based bone* dengan *pivot point* yang mengikuti dari arah *normal* dari *face polygon* yang di pilih. Nantinya digunakan sebagai media pemberian sistem deformasi dan *facial rigging*.



Gambar 3.11. Visualisasi Kode *IntersectRay*

(sumber: dokumentasi pribadi)

Cara kerja *script* ini ialah menyiapkan *pivot point* yang dengan arah *vector* yang sesuai dengan *face polygon* tersebut. Hal ini dapat dicapai dengan memilih *face polygon* yang ingin diberikan *shape-based bone*, kemudian menembakan *ray* ke *polygon* tersebut. Arah balik yang didapat akan menjadi *vector* sumbu Z. Sumbu Z ini akan menjadi acuan bagi *vector* Y dan X dibentuk. Langkah selanjutnya penulis membuat *vector* Y kearah atas. Tujuan pembuatan *vector* Y ini hanya untuk mendapatkan titik perpotongan dimana pangkal *vector* sumbu Z menyentuh *face polygon* yang dipilih, selebihnya *vector* Y pertama ini tidak digunakan lagi. Posisi perpotongan ini digunakan penulis untuk mendapatkan *vector* X dengan memberikan perintah *cross* pada *script* dengan begitu *vector* X akan dibentuk 90 derajat dari *vector* Z. Menggunakan cara yang sama penulis mampu menemukan *vector* Y baru. Ketiga *vector* ini menjadi orientasi *pivot point* dari *shape-based bone* yang akan diberikan pada geometri.



Gambar 3.12. Proses Pembuatan Pivot *Point* Sesuai *Normal Face*
(sumber: dokumentasi pribadi)

3.4.3.1. Script

```

struct pointBasedRig
(
    targetFace=undefined,
    controlSize=2,

```

Gambar 3.13. *Struct*
(sumber: dokumentasi pribadi)

Script diawali dengan pembuatan *struct* yang gunanya untuk membuat *container* untuk menyimpan kelas-kelas, *variable*, *command* ataupun *function* tertentu. Penggunaan *struct* dapat dipanggil menggunakan nama yang diberikan penulis. Dalam kasus ini adalah *pointBasedRig*. Didalam *struct* pada *line* 3 adalah pembuatan *variable* baru yaitu *targetFace* dan *controlSize*. *TargetFace* adalah *variable* untuk menentukan *polygon face* dari geometri yang akan ditempelkan *shape-based bone* Untuk sementara diberikan *value undefined* yang berarti dalam

keadaan kosong. *ControlSize* merupakan *variable* untuk memberikan ukuran dari *shape* yang akan dibuat dalam skala *point*.

```
fn makeControl hit=  
(  
  if hit!=undefined do  
  (  
    =  
    pt=point box:true cross:false axisTripod:true centerMarker:false size:(controlSize/2)  
    wireColor:red name:(uniqueName"PT_ControlRoot")  
    Zv=hit.dir  
    Yv=[0,0,1]  
    Xv=normalize (cross yV Zv)  
    Yv=normalize (cross Zv Xv)  
    pt.transform=matrix3 Xv Yv Zv hit.pos  
  
    pt2=point box:true cross:false axisTripod:true centerMarker:false size:(controlSize)  
    wireColor:green name:(uniqueName "PT_ControlPos")  
    pt2.transform=pt.transform  
    pt2.parent=pt  
  
    cnt=circle radius:(controlSize) wireColor:blue name:(uniqueName "CNT_Face")  
    cnt.transform=pt.transform  
    cnt.parent=pt2  
  
    xf=xform()  
    addModifier cnt xf  
    xf.gizmo.pos.z=controlSize  
    convertToSplineShape cnt  
  )  
)
```

Gambar 3.14. *Function makeControl*

(sumber: dokumentasi pribadi)

Function makeControl terletak didalam *struct*. *Function* ini berfungsi untuk membuat 3 object. Objek pertama ialah sebuah *point* bernama “PT_ControlRoot” dengan ukuran *controlSize* yang dibagi dua. *Point* ini digunakan sebagai posisi dasar. Objek berikutnya adalah *point* bernama “PT_ControlPos” yang di-*parent* ke *point* pertama. *Point* “PT_ControlPos” ini berfungsi sebagai tempat pemberian *slave reaction manager*, *position constraint* dan *orientation constraint*. Objek terakhir adalah *circle shape* dengan nama “CNT_face” yang di-*parent* ke *point* “PT_ControlPos”. *Shape* ini menjadi cikal bakal *shape-based bone* yang di *skin* ke geometri. Seluruh objek yang dibuat pada *function* ini memiliki *pivot point* sesuai dengan arah normal *face polygon* geometri yang dipilih. Untuk

mempermudah dalam melakukan seleksi *shape-based bone* penulis memberikan *modifier transform* untuk mengubah posisi *shape* tanpa mengubah posisi *pivot point*.

```

fnrunOnClick=
(
  tool mouseHit
  (
    on mousePoint clickNumber do
    (
      if clickNumber>1 do
      (
        r=(mapScreenToWorldRay mouse.pos)
        hit=intersectRay $GEO_W002 r
        makeControl hit
        if queryBox "mirror control??" title:"Hey" do
        (
          hit=intersectRay $GEO_W002 (ray(r.pos*[-1,1,1])(r.dir*[-1,1,1]))
          makeControl hit
        )
      )
    )
  )
  startTool mouseHit
)
)

```

Gambar 3.15. *Function runOnClick*

(sumber: dokumentasi pribadi)

Function runOnClick terletak didalam *struct*. *Function* ini berfungsi untuk mendapatkan data *vector* yang dihasilkan dari perpotongan *ray* dengan geometri. Pada *function* ini dibuat *tool* yang bernama *mouseHit*, yang berfungsi untuk melakukan berbagai macam tindakan berdasar *mouseclicks*. Data yang didapatkan melalui *mouseclicks* tersebut kemudian diproses dengan cara memanggil *function makeControl* dan memasukan *variable hit* kedalamnya. Penulis menambahkan fitur *mirror* dalam *script* ini dengan tujuan mencerminkan *shape-based bone* pada posisi satu dengan arah sebaliknya.

```
pointBasedRig=pointBasedRig()  
pointBasedRig.targetFace=${GEO_W002  
pointBasedRig.runOnClick()  
)
```

Gambar 3.16. Target Objek

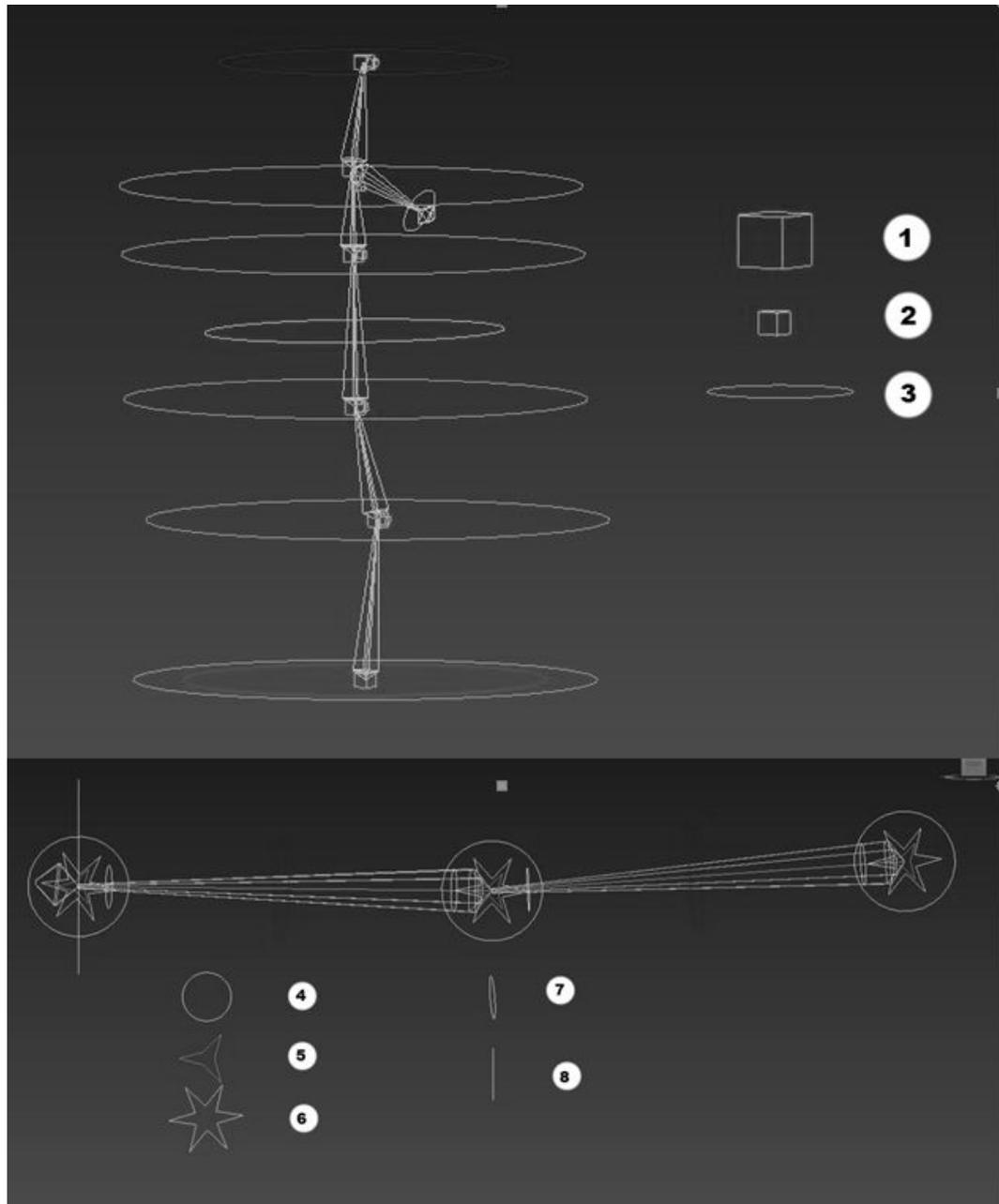
(sumber: dokumentasi pribadi)

Terakhir *variable targetFace* diisi dengan geometri yang ingin diberikan *shape-based bone*. Ketika semua *variable* dilengkapi dan *function* dijalankan maka penulis mampu membuat *shape-based bone* pada titik-titik tulang yang menjadi landasan pembentukan wajah untuk dideformasi dan animasi.

3.4.4. Penempatan Controller

Selanjutnya penulis melakukan perancangan penempatan *controller* untuk masing-masing *bone*. Penempatan *controller* dilakukan dengan meletakkan *shape* pada titik pangkal *bone dasar* dan *bone FK*. Hal ini dilakukan penulis agar mempermudah mengendalikan *bone*. Penulis membagi perancangan *controller* menjadi 2 bagian yaitu perancangan *controller* untuk variasi proporsi tubuh dan perancangan *controller* untuk variasi bentuk tubuh. Hal ini dilakukan penulis dengan tujuan mempermudah pembagian masing-masing peran *controller*.

3.4.4.1. *Controller Variasi Proporsi tubuh*



Gambar 3.17. Penempatan Kontrol *rig* tubuh

(sumber: dokumentasi pribadi)

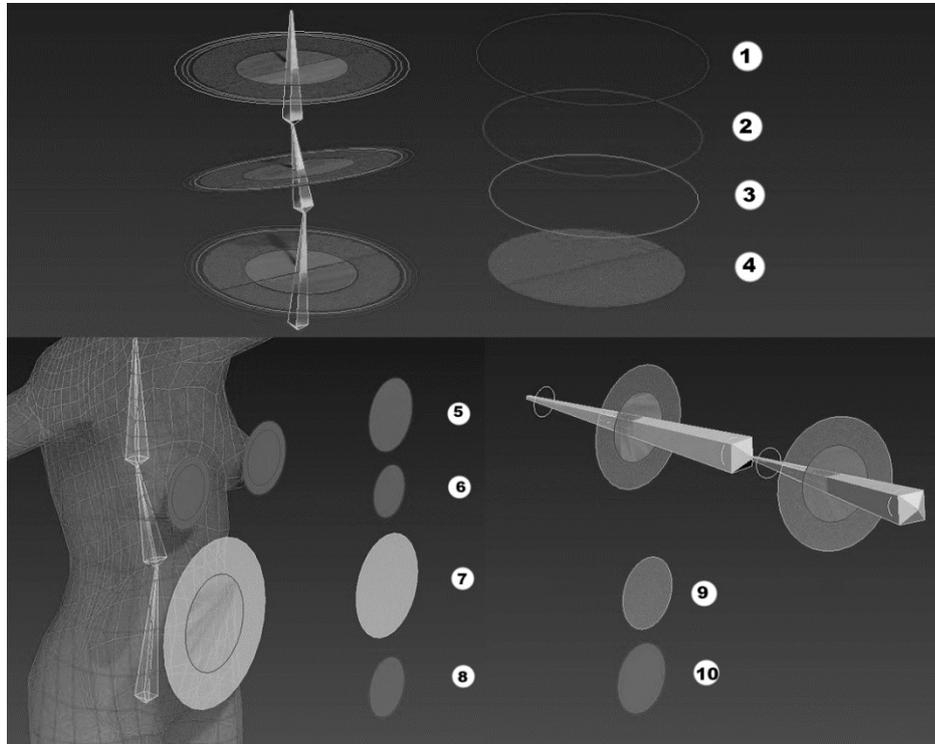
Tabel 3.1. Tabel Deskripsi Sistem *Rig*

| Nomor | Body Part | Nama Controller | Deskripsi |
|-------|-----------|-----------------|--|
| 1 | Torso | helper_LookAt_ | Peletakan <i>point</i> “helper_LookAt_” diletakan pada pangkal <i>bone</i> dasar. <i>Point</i> ini berguna sebagai <i>lookAt Target</i> untuk pembuatan variasi proporsi torso <i>model</i> . |
| 2 | Torso | upnodeSpine_ | Peletakan <i>point</i> “upnodeSpine” digunakan sebagai upnode dari <i>look at constraint</i> yang nantinya akan diberikan pada <i>bone</i> dasar <i>spine</i> . |
| 3 | Torso | CTRL_body_ | Peletakan <i>shape</i> “CTRL_body” digunakan sebagai kontrol dari rotasi <i>bone</i> untuk keperluan animasi. |
| 4 | Limb | FK_Ctrl_ | Peletakan <i>shape</i> “FK_Ctrl” diletakan pada pangkal <i>bone</i> FK. <i>Shape</i> ini berguna sebagai <i>lookAt Target</i> untuk pembuatan variasi proporsi torso <i>model</i> . Setelah proses pemberian |

| | | | |
|---|------|------------------|--|
| | | | <p>variasi selesai <i>controller</i> ini digunakan untuk keperluan animasi.</p> |
| 5 | Limb | MAIN_Ctrl_ | <p>Peletakan <i>shape</i> “MAIN_Ctrl” diletakan pada pangkal <i>bone</i> dasar. <i>Shape</i> ini berfungsi untuk memberikan acuan posisi dari <i>bone</i> dasar ketika melakukan variasi proporsi. Nantinya <i>shape</i> ini akan berperan dalam sistem <i>IK FK Switch</i>.</p> |
| 6 | Limb | <i>point_IK_</i> | <p>Peletakan <i>shape</i> “<i>point_IK_</i>” diletakan pada pangkal <i>bone</i> dasar. <i>Shape</i> ini memegang kunci utama pembuatan <i>procedural IK bone</i> karena memegang peran sebagai acuan penempatan <i>IK</i>. <i>Shape</i> ini di <i>parent</i> ke <i>shape</i> “<i>FK_Ctrl_</i>”. Hal ini dilakukan agar panjang <i>procedural IK bone</i> yang nanti dibuat dapat menyesuaikan dengan panjang <i>bone FK</i> yang telah diubah.</p> |

| | | | |
|---|------|----------------|--|
| 7 | Limb | helper_Stretch | <p>Peletakan <i>shape</i> “helper_Stretch_” diletakan pada dengan posisi sedikit lebih jauh dari pangkal <i>bone</i>. Penempatan ini digunakan sebagai <i>position target</i> dari <i>controller</i> deformasi. Dengan demikian apabila <i>bone</i> di-stretch posisi <i>shape-based bone</i> dan <i>controller</i>-nya tetap berada ditengah relatif dari posisi antara “helper_Stretch” satu dengan yang lain.</p> |
| 8 | Limb | CTRL_IK_ | <p><i>Shape</i> “CTRL_IK” diletakan pada pangkal <i>end bone</i> dasar. Fungsinya ialah mengendali kan IK yang dibuat secara <i>procedural</i> ketika <i>procedural IK bone</i> dibuat.</p> |

3.4.4.2. Controller Variasi Bentuk Tubuh



Gambar 3.18. Penempatan Kontrol *deformer*.

(sumber: dokumentasi pribadi)

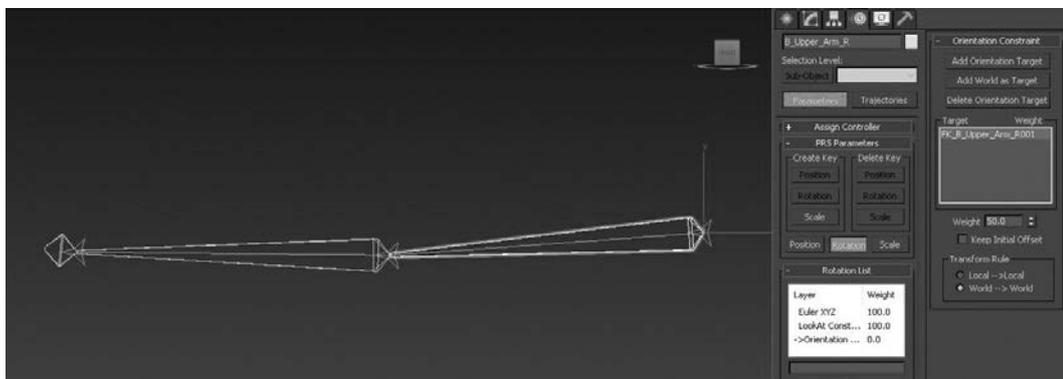
| Nomor | Body Part | Nama Controller | Deskripsi |
|-------|-----------|----------------------|--|
| 1 | Torso | Ctrl_AllDeformer_ | <i>Controller</i> ini adalah <i>parent</i> dari <i>controller</i> 2 dan 4. <i>Controlller</i> ini berfungsi untuk mengendalikan <i>deformer child</i> serta sebagai acuan posisi <i>deformer child</i> agar selalu berada di tengah relatif dengan posisi “ <i>helper_LookAt</i> ” satu dengan yang lain dengan menggunakan <i>position constraint</i> . |
| 2 | Torso | Ctrl_CustDeformer01_ | <i>Controller</i> di <i>parent</i> ke “Ctrl_All Deformer” dengan fungsi mengendalikan “RM_Deformer”. |
| 3 | Torso | RM_Deformer_ | RM_Deformer merupakan <i>controller</i> yang dibuat untuk mengendalikan kedua <i>controller</i> nomor 4 melalui Reaction Manager. |

| | | | |
|---|-------|----------------------|---|
| 4 | Torso | Ctrl_CustDeformer02_ | Dalam nomor ini terdapat 2 <i>controller</i> yaitu <i>controller</i> untuk mengendalikan <i>shape-based bone</i> yang di <i>skin</i> ke geometri. Satu <i>Controller</i> mengendalikan deformasi pada bagian depan dan satunya untuk bagian belakang. |
| 5 | Torso | Ctrl_CustDeformer_ | <i>Controller</i> ini berfungsi untuk mengendalikan <i>shape-based bone</i> yang di <i>skin</i> ke geometri |
| 6 | Torso | SHP_Deformer_ | <i>Shape Based Bone</i> yang di <i>Skin</i> ke geometri. Objek ini diatur melalui <i>reaction manager</i> . |
| 7 | Torso | Ctrl_CustDeformer_ | <i>Controller</i> ini berfungsi untuk mengendalikan <i>shape-based bone</i> yang di <i>skin</i> ke geometri. |
| 8 | Torso | RM_Deformer_ | <i>Shape Based Bone</i> yang di <i>Skin</i> ke geometri. Objek ini diatur melalui <i>reaction manager</i> . <i>Bone</i> |

| | | | |
|----|------|--------------------|---|
| | | | ini di parent ke- <i>controller</i> nomor 7. |
| 9 | Limb | Ctrl_CustDeformer_ | <i>Controller</i> ini adalah parent dari <i>controller 10</i> . <i>Controlller</i> ini berfungsi untuk mengendalikan <i>deformer child</i> serta sebagai acuan posisi <i>deformer child</i> agar selalu berada di tengah relatif dengan posisi “helper_Stretch” satu dengan yang lain dengan menggunakan <i>position constraint</i> . |
| 10 | Limb | RM_Deformer_ | <i>Shape Based Bone</i> yang di Skin ke geometri. Objek ini diatur melalui <i>reaction manager</i> . <i>Bone</i> ini di parent ke- <i>controller</i> nomor 9. |

3.4.5. Perancangan Sistem IK dan FK

Pemberian sistem *stretchy bone* dilakukan agar panjang *bone* FK maupun *bone* dasar dapat diberikan variasi setelah diberikan *modifier skin*. Perancangan ini dilakukan pada *bone-bone* yang nantinya akan dibuat *procedural IK Bone* yaitu *Upper Arm, Lower Arm, Thigh* dan *Calf*. Semua bagian tubuh tersebut memiliki perancangan sistem yang sama.

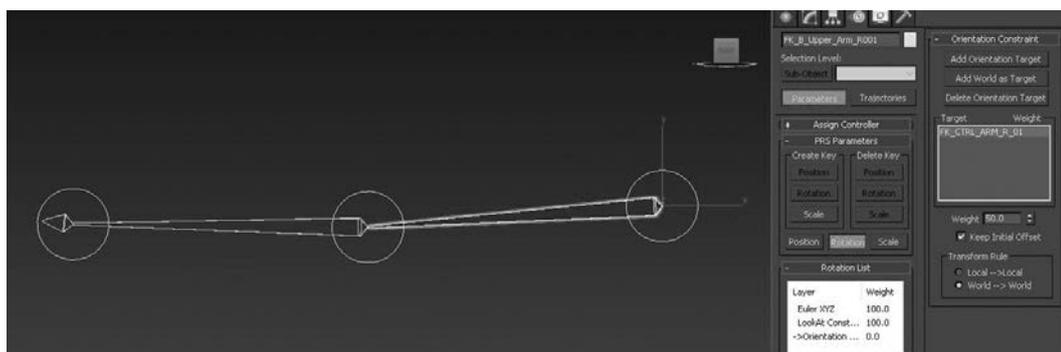


Gambar 3.19. *stretchy bone*

(sumber: dokumentasi pribadi)

Pada kasus ini penulis menggunakan bagian lengan atas kanan sebagai contoh. Pada *bone* dasar perancangan dimulai dengan memberikan *look at constraint* dan menargetkannya ke-*controller* “Main_Ctrl_LowerArm_R”. Hal ini menyebabkan orientasi *bone* mengikuti posisi *controller*. Setelah ini *bone* dasar diberikan *position constraint* terhadap “Main_Ctrl_UpperArm_R”. Dengan begitu posisi pangkal *bone* dasar akan mengikuti posisi *controller* “Main_Ctrl_UpperArm_R” namun memiliki orientasi mengarah ke *controller* “Main_Ctrl_LowerArm_L”. Apa bila sistem ini diterapkan pada bagian lengan bawah kanan juga, maka penulis akan mendapatkan *bone* dasar yang mampu memanjang atau memendek, tergantung dari penempatan posisi *controller*

“Main_Ctrl_”. Kemudian masih pada *bone* dasar yang sama penulis memberikan *orientation constraint* terhadap *bone FK* dengan status *weight = 0*. Hal ini menjadi kunci finalisasi dari sistem *rig* yang dirancang. Karena setelah proses variasi selesai dan memasuki tahap animasi, maka *orientation constraint* akan di tetapkan sebagai aktif dengan *weight = 100*, sedangkan *look at constraint* dibuat tidak aktif dengan *weight = 0*. Proses finalisasi yang nantinya dibantu dengan *MAXscript*, juga akan menambahkan target *orientation constraint*. Target yang ditambahkan adalah *procedural IK bone* yang dibuat bersamaan proses finalisasi berlangsung. Dengan begitu, setelah proses finalisasi selesai, *output* yang dihasil mamliki sistem *IK/FK switch* didalamnya.

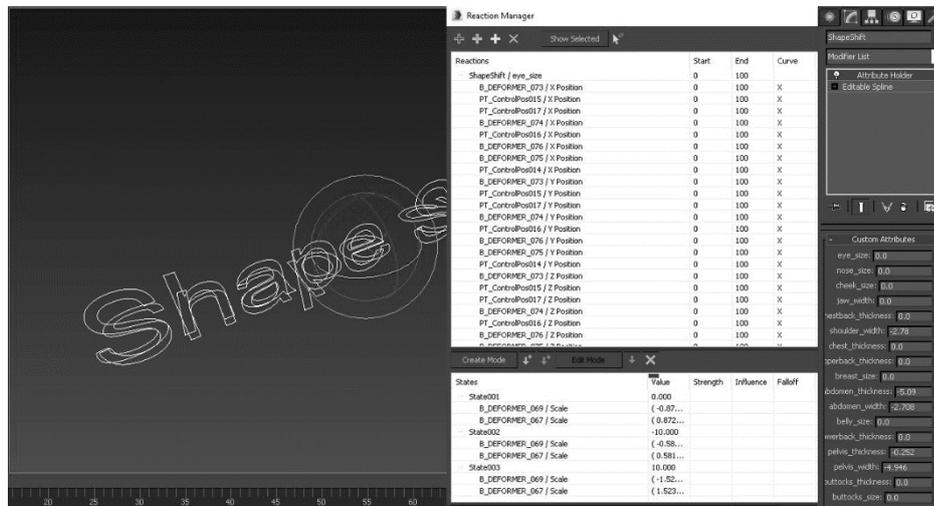


Gambar 3.20. *Orientation Constraint*

(sumber: dokumentasi pribadi)

3.4.6. Reaction Manager

Sebelum memasuki proses perancangan *Reaction Manager* terlebih dahulu membuat objek dengan atribut yang mampu menjadi *master* terhadap *slave* yang lain. Dalam perancangan ini objek yang dibuat penulis ialah sebuah *shape* yang diberikan *attribute holder*. Kemudian penulis memberikan *parameter* sesuai dengan jumlah variasi bentuk yang dirancang sebelumnya.



Gambar 3.21. Reaction Manager

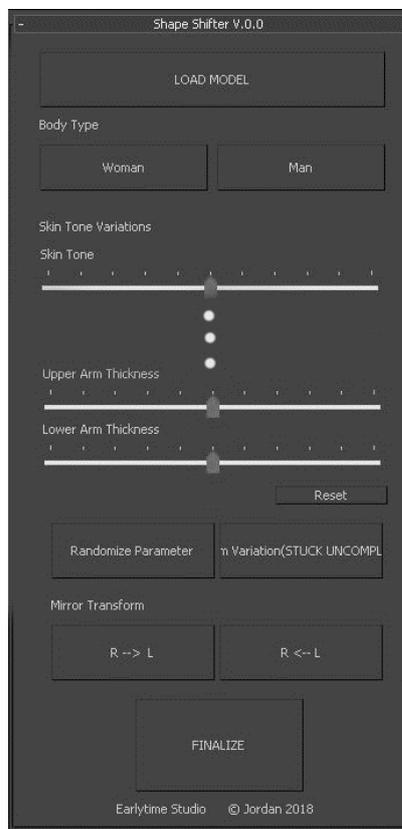
(sumber: dokumentasi pribadi)

Setelah itu penulis membuat hubungan *master-slave* menggunakan *reaction manager* terhadap masing-masing *shape-based bone* yang di *skin* ke *model* terhadap parameter yang dibuat. Dilanjutkan dengan menentukan batas maksimum serta batas minimum yang dapat diubah. Batasan ini didasari data rata-rata lingkar bagian tubuh wanita berumur 20 tahun hingga 83 tahun yang penulis dapat dari *Hindawi Publishing Corporation Article*. Dengan menggunakan perhitungan keliling lingkaran sederhana penulis mampu mengkonversi data lingkar bagian tubuh menjadi diameter. Untuk menentukan batas lebih jauh dari batasan variasi dunia nyata penulis menggunakan batasan teknik. Yang dimaksud dari batasan teknik adalah batasan dimana deformasi *model* mencapai titik yang merusak sistem *rig*. Batasan ini dirancang oleh penulis agar mampu memperluas variasi bentuk tubuh manusia dalam segi animasi.

3.4.7. Finalisasi

Selanjutnya penulis merancang *user interface* untuk mempermudah pengguna sistem *rig* serta merancang proses akhir untuk membuat *procedural IK bone* dan memindahkan sistem *rig* dari *mode* variasi menjadi *mode* animasi. Proses ini sepenuhnya dirancang menggunakan *MAXscript* dan *Visual MAXScript*.

3.4.7.1. Visual MAXscript



Gambar 3.22. Visual *MAXScript*

(sumber: dokumentasi pribadi)

Sebelum masuk kedalam *scripting*, penulis menyiapkan *user interface* dengan menggunakan *Visual MAXscript*. Disini *UI* dibuat dengan cara melakukan *drag and drop*. Perancangan *UI* ini dibuat dengan acuan *modification game Fallout 4 "Caliente's Body Slider"*. Pada bagian atas

penulis membuat tombol “Load Model” yang berfungsi untuk membuka *file* tubuh yang telah diberi sistem. Setelah itu terdapat pemilihan tipe tubuh antara laki-laki atau perempuan. Berlanjut ke komponen utama *UI* yang terdiri dari *slider* yang berguna untuk mengatur hubungan *master-slave* yang dibuat pada *reaction manager*. Pada akhir *slider* terdapat tombol *reset*, *Randomize Parameter*, *Custom variation*, *mirror transform* dan terakhir *finalize*. Masing-masing tombol akan menjalankan perintah yang dibuat penulis dalam *MAXscript*. Setelah desain *UI* penulis menyimpan data desain tersebut dalam format (.ms) agar penulis mampu melakukan perubahan lebih jauh melalui *MAXscript*.

3.4.7.2. Perancangan MAXscript

```
rollout ShapeShifter "Shape Shifter V.0.0" width:352 height:3200
  (
```

Gambar 3.23. Kode *Rollout*

(sumber: dokumentasi pribadi)

Membuat *rollout* dengan nama “*Shape Shifter V.0.0*”.Hal ini bertujuan untuk membuat *window* baru dari *UI* yang telah dibuat sebelumnya.

```
slider 'eye_size' "Eye Size" pos:[8,832] width:320 height:44 range:[-10,10,0] align:#left
slider 'nose_size' "Nose Size" pos:[8,880] width:320 height:44 range:[-10,10,0] align:#left
slider 'cheek_size' "Cheek Size" pos:[8,928] width:320 height:44 range:[-10,10,0] align:#left
slider 'jaw_size' "Jaw Size" pos:[8,976] width:320 height:44 range:[-10,10,0] align:#left
slider 'chin_size' "Chin Size" pos:[8,1024] width:320 height:44 range:[-10,10,0] align:#left
```

Gambar 3.24. Kode *Visual MAXScript*

(sumber: dokumentasi pribadi)

Didalam *rollout* terdapat perintah untuk membuat komponen-komponen *UI*. Komponen ini terdiri dari *slider*, *button*, *label*, dan *checkboxbutton*. Masing-masing komponen memiliki *caption*, *name* dan

koordinat penempatan. Setiap komponen akan diberi perintah sesuai dengan nama yang ditentukan penulis.

```
on eye_size changed val do
(
  $CTRL_DEFORMER_HOLDER.modifiers[#Attribute_Holder].Custom_Attributes.eye_size = val
)
on nose_size changed val do
(
  $CTRL_DEFORMER_HOLDER.modifiers[#Attribute_Holder].Custom_Attributes.nose_size = val
)
```

Gambar 3.25. Kode untuk Mengontrol *Parameter*

(sumber: dokumentasi pribadi)

Kode “*on (name) changed val do..*” berfungsi untuk menyatakan kondisi dari *slider* tertentu. Apabila kondisi tersebut dipenuhi maka perintah akan dijalankan. Dalam kasus ini perintah yang diberikan penulis pada masing-masing *slider* ialah melakukan perubahan pada *custom attribute* yang didalamnya berisi data *reaction manager* dimana *value* perubahan pada parameter *custom attribute* sama dengan *value slider*. Dengan begitu apabila *value* pada parameter diubah maka parameter dalam *custom attribute* turut berubah.

```
on randomize pressed do
(
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.eye_size = (random -5 5)
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.nose_size = (random -5 5)
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.cheek_size = (random -5 5)
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.jaw_width = (random -5 5)
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.chestback_thickness = (random -5 5)
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.shoulder_width = (random -5 5)
)
```

Gambar 3.26. Kode Mengacak *Parameter*

(sumber: dokumentasi pribadi)

Kode “*on randomize pressed do ()*” berarti kondisi ketika tombol *randomize* ditekan maka tombol akan menjalankan serangkaian perintah. Perintah yang tentukan penulis untuk tombol ini terbagi menjadi 3 bagian. Bagian pertama merupakan perintah untuk menentukan *value* dari parameter *custom attribute* dengan pemberian kode *random* pada jenjang

tertentu. Dengan begitu ketika tombol ini ditekan *value* pada parameter akan berubah secara acak dalam jenjang *value* yang ditentukan.

```
pelvisThickness.value = $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.pelvis_thickness
abdomenWidth.value = $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.abdomen_width
buttocksThickness.value = $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.buttocks_thickness
pelvisWidth.value = $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.pelvis_width
abdomenThickness.value = $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.abdomen_thickness
lowerBackThickness.value = $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.lowerback_thickness
```

Gambar 3.27. Kode Menyamakan *Value* dengan *Slider*

(sumber: dokumentasi pribadi)

Setelah itu penulis menyamakan *value* yang berubah pada parameter dengan *value* pada *slider* hal ini dilakukan penulis agar *value* antara *slider* dan *parameter* sinkron. Hal ini dapat dicapai dengan memberikan kode “<slidername>.value = \$.modifier[#Attribute_Holder].Custom_Attributes.<parameter>”.

```
$FK_CTRL_ARM_R_02.pos.x = -(random 15 19)
$FK_CTRL_ARM_L_02.pos.x = (random 15 19)

$FK_CTRL_ARM_R_03.pos.x = -(random 24 29)
$FK_CTRL_ARM_L_03.pos.x = (random 24 29)
```

Gambar 3.28. Kode Mengacak Proporsi Tubuh

(sumber: dokumentasi pribadi)

```
on resetProportions pressed do
(
$FK_CTRL_LEG_L_02.pos.z = -13.503
$FK_CTRL_LEG_R_02.pos.z = -13.503

$FK_CTRL_ARM_R_02.pos.x = -15.938
$FK_CTRL_ARM_L_02.pos.x = 15.938

$FK_CTRL_ARM_R_03.pos.x = -25.6204
$FK_CTRL_ARM_L_03.pos.x = 25.6204

$FK_CTRL_LEG_L_03.pos.z = -31.49
$FK_CTRL_LEG_R_03.pos.z = -31.49

$Circle013.pos.z = 15.354

lowerArmLength.value = 25.6204
upperArmLength.value = 15.938
thighLength.value = -13.503
calfLength.value = -31.49
chestLength.value = 15.354
```

Gambar 3.29. Kode Mengembalikan Proporsi ke Kondisi Awal

(sumber: dokumentasi pribadi)

Penulis juga memberikan variasi untuk proporsi dengan metode yang sama dengan variasi bentuk tubuh. Hal ini dicapai dengan memindahkan masing-masing *controller* “FK_Ctrl_” ke posisi yang diinginkan. Dengan adanya sistem *stertchy bone* yang sebelumnya disiapkan, penulis mampu memanjangkan maupun memendekkan bagian tubuh *model*.

Button “resetProportion” dirancang untuk mengembalikan semua variasi proporsi yang dibuat ke posisi awal. Penulis mengembalikan posisi awal *controller* “FK_Ctrl_” dengan kode “\$.pos = value”. Value yang ditetapkan penulis merupakan posisi awal dari *controller* dalam sistem koordinat

```

on resetVariations pressed do
(
  allSlider = #(pelvisThickness,abdomenWidth,buttocksThickness,pelvisWidth,abdomenThickness,
  lowerBackThickness,chestThickness,upperbackThickness,upperbackThickness,buttocksSize,
  shoulderWidth,thighThickness,calfThickness,upperArmThickness,lowerArmThickness,jawWidth,
  eyeSize,noseSize,cheekSize,skinTone)
  for i in allSlider do
    i.value = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.eye_size = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.nose_size = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.cheek_size = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.jaw_width = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.chestback_thickness = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.shoulder_width = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.chest_thickness = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.upperback_thickness = 0
  $ShapeShift.modifiers[#Attribute_Holder].Custom_Attributes.breast_size = 0
  $FK_CTRL_LEG_L_02.pos.z = -13.503
  $FK_CTRL_LEG_R_02.pos.z = -13.503
  $FK_CTRL_ARM_R_02.pos.x = -15.938
  $FK_CTRL_ARM_L_02.pos.x = 15.938
  lowerArmLength.value = 25.6204
  upperArmLength.value = 15.938
  thighLength.value = -13.503
  calfLength.value = -31.49
  chestLength.value = 15.35
)

```

Gambar 3.30. Kode Mengembalikan Semua Parameter yang Diubah

(sumber: dokumentasi pribadi)

Button “resetVariation” berfungsi untuk mengembalikan semua perubahan bentuk tubuh maupun proporsi ke posisi awal. Dalam

mengembalikan posisi *slider* ke posisi awal penulis melakukan *looping* dengan *condition* “*for<expression> in <variable> do*”. Perintah yang dirancang penulis ialah “*value=0*”. Diikuti dengan perintah untuk mengembalikan semua *value* di *custom attribute* menjadi 0.

```
on skinTone changed val do
(
  $ShapeShifter.modifiers[#Attribute_Holder].Custom_Attributes.skin_tone = val
)
```

Gambar 3.31. Kode untuk Mengontrol Warna Kulit

(sumber: dokumentasi pribadi)

Button “*skinTone*” berfungsi untuk memberi variasi warna kulit.

Dicapai dengan mengubah *value* dari *custom attribute* yang mengatur tekstur warna kulit.

```
on women pressed do
(
  $GEO_W002.morpher[1].value = 0
)
on man pressed do
(
  $GEO_W002.morpher[1].value = 100
)
```

Gambar 3.32. Kode untuk Mengontrol Jenis Kelamin

(sumber: dokumentasi pribadi)

Button “*women*” dan “*man*” berfungsi untuk mengubah *base* geometri menjadi wanita ataupun pria. Hal ini dilakukan dengan mengubah *value morpher* yang disiapkan oleh penulis.

```
on load pressed do
if queryBox "Are you sure to continue?" do
(
  sceneName = "D:\Shape Shifter V.0.1\Shape Shifter V.0.1.max"
  loadMaxFile (sceneName)
)
```

Gambar 3.33. Kode untuk Memasukan Data ke Dalam Scene

(sumber: dokumentasi pribadi)

Tombol “*load*” dibuat penulis untuk memasukan *model* baru ketika *model* pertama selesai dibuat menggunakan kode “*loadMaxFile*”.

```

on finalize pressed do
  if queryBox "the process CANNOT BE UNDONE, are you sure to continue?" title:"Finalization Process" do
    L_IKarm_pos01=$POINT_IK_L_ARM01.pos
    L_IKarm_pos02=$POINT_IK_L_ARM02.pos
    L_IKarm_pos03=$POINT_IK_L_ARM03.pos
    L_IKarm_pos04=$POINT_IK_L_ARM03.pos

    ikBoneShoulder=BoneSys.createBone L_IKarm_pos01 L_IKarm_pos02 [0,-1,0]
    ikBoneShoulder.name = uniqueName "IK_BONE_L_arm_0"
    ikBoneShoulder.width =0.393701
    ikBoneShoulder.height =0.393701
    ikBoneShoulder.parent = $MAIN_CTRL_ARM_L_01
    ikBoneElbow=BoneSys.createBone L_IKarm_pos02 L_IKarm_pos03 [0,-1,0]
    ikBoneElbow.name = uniqueName "IK_BONE_L_arm_0"
    ikBoneElbow.height = 0.393701
    ikBoneElbow.width = 0.393701
    ikBoneElbow.parent = ikBoneShoulder
    ikBoneNub=BoneSys.createBone L_IKarm_pos03 L_IKarm_pos04 [0,-1,0]
    ikBoneNub.height = 0.393701
    ikBoneNub.width = 0.393701
    ikBoneNub.name = uniqueName "IK_BONE_L_arm_0"
    ikBoneNub.parent = ikBoneElbow

    ikArm=IKSys.ikChain ikBoneShoulder ikBoneNub "IKHISolver"
    ikArm.name = uniqueName "IK_ARM_L"
    $IK_ARM_LO01.transform.controller.VHTarget = $SWIVEL_L_ARM

```

Gambar 3.34. Kode Pembuatan *bone IK*

(sumber: dokumentasi pribadi)

Tombol *finalize* merupakan tombol terpenting untuk membuat *procedural IK bone*, memberikan target orientasi, *parenting*, memberikan *wire parameter*, serta memindahkan mode variasi menjadi mode animasi. Kode *queryBox* digunakan penulis sebagai penetapan kondisi yang berhubungan dengan pengguna. Ketika tombol ini ditekan maka pengguna *script* perlu menentukan apakah setuju dengan kondisi yang diberikan. Bila setuju maka tombol akan melakukan serangkaian perintah. Perintah pertama adalah pembuatan *procedural IK bone*. Tahap pertama penulis membuat *variable* “L_IKarm_pos00” untuk menentukan posisi “*Point_Ik_*” yang sudah dibuat dalam proses perancangan *controller*. Lalu *bone* dengan kode “*Bonesys.createBone*” berdasarkan posisi-posisi ini dan di-

parent ke *bone* sebelumnya. *Bone* yang dibuat diberi nama “*IK_Bone_L_arm_*”. Dilanjutkan dengan memberikan *IK* dengan kode “*ikSys.IkChain*” pada *bone* lengan atas ke *end bone*. Terakhir *IK chain* diberi nama dan diberikan target *swivel*. Perintah ini diterapkan pada semua bagian lengan dan kaki.

```
$IK_ARM_LO01.parent = $CTRL_IK_ARM_L
$IK_LEG_LO01.parent = $CTRL_IK_LEG_L
$IK_ARM_RO01.parent = $CTRL_IK_ARM_R
$IK_LEG_RO01.parent = $CTRL_IK_LEG_R

$CTRL_IK_ARM_L.parent =undefined
$CTRL_IK_LEG_L.parent =undefined
$CTRL_IK_ARM_R.parent =undefined
$CTRL_IK_LEG_R.parent =undefined
```

Gambar 3.35. Kode Perubahan hubungan *Parent-Child*

(sumber: dokumentasi pribadi)

Kemudian masing-masing *IK chain* akan di-*parent* ke *controller* “*CTRL_IK_*” dengan kode “*parent*”. Hubungan *parenting* antara *Controller* “*CTRL_IK_*” dengan *controller* “*FK_Ctrl*” dilakukan karena *controller* “*CTRL_IK_*” tidak perlu lagi mengikuti perpindahan *controller* “*FK_Ctrl*” dan siap digunakan untuk keperluan animasi. Maka dari itu *value* di tetapkan penulis sebagai *undefined*.

```
$B_Lower_Arm_L.rotation.controller.Orientation_Constraint.appendTarget $IK_BONE_L_arm_002 50
$B_Upper_Arm_L.rotation.controller.Orientation_Constraint.appendTarget $IK_BONE_L_arm_001 50

$B_Lower_Arm_R.rotation.controller.Orientation_Constraint.appendTarget $IK_BONE_R_arm_002 50
$B_Upper_Arm_R.rotation.controller.Orientation_Constraint.appendTarget $IK_BONE_R_arm_001 50
```

Gambar 3.36. Kode Menambahkan Target Orientasi

(sumber: dokumentasi pribadi)

Perintah selanjutnya ialah memberikan target baru pada *orientation constraint*. Perintah ini berlaku untuk semua *bone dasar* yang diberikan

sistem *IK FK switch*. Target orientasi yang diberikan adalah *procedural IK bone* yang dibuat sebelumnya.

```
paramWire.connect $B_Upper_Arm_L.rotation.controller.Orientation_Constraint.controller  
[#Orientation_Weight_0] $B_Upper_Arm_L.rotation.controller.Orientation_Constraint.controller  
[#Orientation_Weight_1] "100-Orientation_Weight_0"
```

Gambar 3.37. Kode Pembuatan *Wire Parameter*

(sumber: dokumentasi pribadi)

Untuk mempercepat proses perpindahan IK dan FK nantinya, penulis juga menyertakan perintah pembuatan *wire parameter*. Parameter yang dihubungkan adalah *weight* orientasi FK dan IK dengan *expression* “*100-Orientation_Weight_0*”. Dengan begitu maka didapat persamaan sebagai berikut

$$\text{Value Weight IK} = 100 - \text{Value Weight FK}$$

Dengan begitu bila *weight FK* diisi dengan *value* 100 maka *weight IK* akan mendapat *value* 0. Bila diisi 73 maka *weight IK* mendapat *value* 27 dan seterusnya.

```
$B_Upper_Arm_R.rotation.controller.Orientation_Constraint.controller.weight[1] = 0  
$B_Lower_Arm_R.rotation.controller.Orientation_Constraint.controller.weight[1] = 0  
$B_Upper_Arm_L.rotation.controller.Orientation_Constraint.controller.weight[1] = 0  
$B_Lower_Arm_L.rotation.controller.Orientation_Constraint.controller.weight[1] = 0  
$B_Thigh_R.rotation.controller.Orientation_Constraint.controller.weight[1] = 0  
$B_Calf_R.rotation.controller.Orientation_Constraint.controller.weight[1] = 0  
$B_Thigh_L.rotation.controller.Orientation_Constraint.controller.weight[1] = 0  
$B_Calf_L.rotation.controller.Orientation_Constraint.controller.weight[1] = 0
```

Gambar 3.38. Kode Me-Nonaktifkan *Look-At Constraint* dan mengaktifkan *Orientation Constraint*

(sumber: dokumentasi pribadi)

Setelah itu penulis memindahkan semua sistem ke IK. Dengan bantuan *wire* parameter yang dibentuk sebelumnya, penulis cukup

menentukan *value* dari salah satu *weight*. Dengan kata lain apabila *weight* IK memiliki *value* 100 maka sistem yang dipakai adalah IK namun apabila *value* dinyatakan sebagai 0 maka sistem yang berlaku adalah FK.

```
$B_Calf_R.rotation.controller.weight[3] = 0
$B_Calf_R.rotation.controller.weight[2] = 100

$B_Calf_L.rotation.controller.weight[3] = 100
$B_Calf_L.rotation.controller.weight[2] = 0
```

Gambar 3.39. Kode Memindahkan ke *IK System*

(sumber: dokumentasi pribadi)

Selanjutnya penulis melakukan perpindahan orientasi dari *look at constraint* ke *orientation constraint*. Hal ini dilakukan agar perubahan proporsi tidak lagi berubah karena perubahan proporsi lebih lanjut akan merusak sistem IK. Maka dari itu *weight* dari *look at constraint* ditetapkan sebagai 0 dan *weight orientation constraint* ditetapkan sebagai 100.

```
$MAIN_CTRL_ARM_R_01.parent = $B_Shoulder_R
$MAIN_CTRL_ARM_R_02.parent = $B_Upper_Arm_R
$MAIN_CTRL_ARM_R_03.parent = $B_Lower_Arm_R
```

Gambar 3.40. Kode Memindahkan Hubungan *Parent-Child* Lengan dan Kaki

(sumber: dokumentasi pribadi)

Penulis mengubah sistem *parent* dari *controller* “*MAIN_Ctrl_*”. Hal ini dicapai dengan memberikan perintah untuk melakukan *parenting* ke target baru. Penulis memberi kode tersebut karena *controller* “*MAIN_Ctrl_*” tidak perlu mengikuti posisi *controller* “*FK_Ctrl_*” lagi, melainkan mengikuti *bone* dasar. Kode ini menutup proses perubahan mode variasi menjadi mode animasi.

```
nf = newRolloutFloater "Shape Shifter V.0.0" 352 500  
addRollout ShapeShifter nf
```

Gambar 3.41. Kode untuk Membuat *Rollout* Baru

(sumber: dokumentasi pribadi)

Script ditutup dengan memberikan kode “*newRolloutFloater*” dan memasukan UI yang dibuat didalamnya dengan kode “*addRollout*”.

3.4.8. Membandingkan Keefektifan Waktu

Pembuatan Rig yang Menyerupai Hasil Rig yang Dirancang

Penulis melakukan perbandingan lama pembuatan *rig* menggunakan model tokoh utama dari film “*Light Up*” dengan pembuatan sistem yang dirancang penulis. Penulis membuat ulang *rig* yang menyerupai system *rig* yang dirancang dengan model Tokoh utama sebagai media pengaplikasian *rig*. *Perhitungan* waktu terhitung dari keadaan kedua *model* 3D yang sudah disiapkan sebelumnya hingga *model* mencapai tahap siap untuk dianimasikan. Setiap *Model* dikerjakan selama 8 jam perharinya hingga selesai. Lama waktu yang terhitung semenjak mulai pengerjaan hingga mencapai tahap siap animasi akan menjadi pembanding tingkat keefektifan waktu dari *script* yang dibuat.