



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

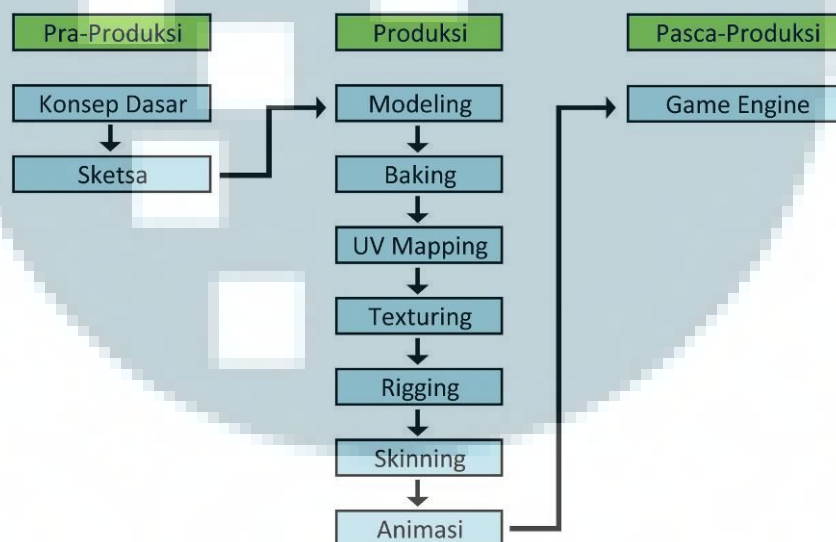
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI TUGAS AKHIR

3.1. Gambaran Umum

Baik dalam film, animasi, atau *game*, semuanya melewati proses pembuatan. Proses itu terbagi menjadi 3, yaitu praproduksi, produksi, dan pascaproduksi. Berdasarkan bagan yang dikutip dari buku Franson dan Thomas (2007) dan *workflow* penulis, maka dibuatlah bagan sebagai berikut.



Gambar 3.1. *Workflow* Pembuatan *Game*

3.2. Praproduksi

Sebuah *game* selalu berawal dari cerita. Ide cerita bisa diberikan oleh desainer *game* atau pihak luar. Contoh yang sering ditemukan adalah sekuel atau *spin-off* dari *game* yang sudah ada atau simulasi dari kejadian di dunia nyata. Setelah mendapatkan konsep dasar, penulis dan seniman bekerja sama menciptakan

storyboard sebagai bentuk visual dari cerita dan sebagai referensi untuk penulis, artist, dan desainer seperti halnya dalam film.

3.2.1. Sinopsis Cerita

Cerita *game* “*Indictus*” berada di bumi parallel tahun 19xx dimana manusia pada masa itu sedang mengalami zaman *steampunk*, sebuah zaman berlatar belakang fantasi *sci-fi* yang mengambil tema *Victoria* dalam masa industri. Cerita *game* berpusat pada karakter utama, seseorang yang memiliki peran seorang ilmuwan (*scientist*). Ia sering ditugaskan ke luar kota dan bertemu dengan seorang wanita yang kemudian menjadi kekasihnya. Ia bekerja di kotanya untuk membangun kembali kotanya dan membuat hidup penduduknya lebih baik.

Suatu hari tokoh utama mendapat sebuah panggilan untuk menghadiri sebuah konverensi di kota lain. Ia kemudian segera pergi dengan menggunakan sebuah kapal udara untuk pergi ke kota itu. Dalam perjalanannya di tengah laut, kapalnya dihantam oleh badai yang sangat hebat yang berakhir kapal tersedot masuk ke dalam sebuah tornado. Lolos dari tornado, kapal mengalami gangguan mesin dan tidak bisa dikendalikan. Kapal menukik turun menuju sebuah pulau yang dapat mengakibatkan kematian bagi tokoh utama. Tapi karakter segera terjun keluar dari kapal terbang dengan menggunakan parasut dan mendarat di pantai pulau itu.

Tokoh utama kemudian menyusuri pantai tersebut untuk mencari bantuan tapi tidak menemukan satu tanda bangunan atau arsitektur yang ia kenal. Ia pun masuk ke dalam pulau untuk mencari bantuan tapi malah menemukan hewan aneh yang sedang memegang secarik kertas. Hewan tersebut kemudian mengejek tokoh

utama dan kabur. Tanpa pikir panjang, tokoh utama mengejar hewan tersebut yang akhirnya terpancing menuju sarangnya dimana ada beberapa hewan lain menunggu dan menyerangnya. Tokoh utama kemudian mengalahkan hewan-hewan tersebut dan melihat kertas yang tadi dipegang oleh hewan itu.

Ternyata kertas tersebut adalah sebuah halaman dari jurnal seseorang. Disebutkan bahwa pulau tersebut tidak berpenghuni, berbeda dari dunia yang sebenarnya, dan jurnalis sendiri sedang mencari jalan keluar dari pulau itu. Segera tokoh utama melanjutkan perjalanannya, masuk lebih dalam lagi ke dalam pulau untuk mencari jalan keluar dari pulau, sekaligus mengumpulkan catatan-catatan yang ditulis oleh jurnalis untuk membantunya dalam perjalanannya mencari jalan keluar dari pulau. Cerita ini penulis kutip dari Dave (2013).

3.2.2. Penyusunan Konsep

Penulis bersama seorang rekan melakukan *brainstorming* untuk mencari ide cerita. Setelah mendapat cerita dasar, tim melakukan riset mengenai tema yang akan dibuat dan mencari referensi dari *game-game* yang mempunyai *genre* yang sejenis. Sketsa kasar *Player Character* yang bertema *Steampunk* dibuat dengan mengumpulkan desain-desain karakter maupun dari *cosplay* dengan tema tersebut. Untuk *Non Player Character*, penulis mengambil gambar bermacam-macam hewan dari internet kemudian mencoba menggabungkan bagian-bagian tubuh hewan agar terlihat menarik. Dalam pembuatan konsep, penulis mengambil gaya gambar Jepang yang biasanya terlihat dalam *anime*, ataupun *game* buatan Jepang. Dalam game ini, keseluruhan karakter yang penulis kerjakan berjumlah tiga

karakter dimana *Player Character* terdiri dari satu orang dan *Non Player Character* berjumlah dua ekor.

3.2.2.1. *Player Character*

Dalam *game Indictus*, pemain akan mengendalikan satu orang *PC* dari awal cerita sampai akhir *game*. Berikut ini adalah penjabaran *PC* secara fisiologi, sosiologi, dan psikologinya. Karakter sengaja tidak diberi nama agar pemain bisa menempatkan dirinya dalam *game*.

1.) Fisiologi

Karakter ini berasal dari ras Kaukasia, berkulit putih, berambut hitam, berusia 24 tahun dengan tinggi badan 180 cm. Dalam merancang karakter ini, penulis mengambil aliran gambar Jepang yang memiliki ciri khas seperti mata yang besar dan rambut yang cenderung *spiky*, contohnya karakter *Final Fantasy*. Wajah karakter cenderung berbentuk kotak untuk menampilkan sisi maskulinnya, yang biasa terdapat pada pria dewasa. Penulis mengambil referensi dari karakter Kaburagi T. Kotetsu dari anime *Tiger and Bunny* dan Ryutaro Dojima dari *game Persona 4*.



Gambar 3.2 Karakter Kaburagi T. Kotetsu
(<http://cdn.myanimelist.net/images/characters/10/113442.jpg>)



Gambar 3.3 Karakter Ryutaro Dojima
(<http://cdn.myanimelist.net/images/characters/13/46870.jpg>)

Pakaian yang dikenakan karakter merupakan pakaian khas pria yang terdapat pada masa *Steampunk*, yaitu kemeja lengan panjang, vest (rompi), celana panjang, *gaiters* (pelindung betis yang biasa terbuat dari kain, kulit, atau poliester). Tidak lupa penulis menyertakan aksesoris khas *Steampunk* seperti *google* yang biasa digunakan oleh para ilmuwan. *Pouch* yang terdapat pada celana karakter diberikan tambahan-tambahan mesin kecil untuk menambah kesan era mesin yang merupakan ciri khas *Steampunk*. Untuk menambahkan kesan petualangan, penulis membuat pakaian karakter menjadi compang camping, menghilangkan dasi dan merubah kemeja lengan panjang menjadi lengan pendek untuk mengurangi kesan formal. Penulis juga menghilangkan *lab coat* untuk menambahkan kesan keeluasaan gerak. Diceritakan bahwa *lab coat* tersebut sudah sangat lusuh sehingga *PC* merubahnya menjadi perban.



Gambar 3.4. Referensi Pakaian *Steampunk*
(<http://lissrant.blogspot.com/2011/07/steampunk-hat.html>)

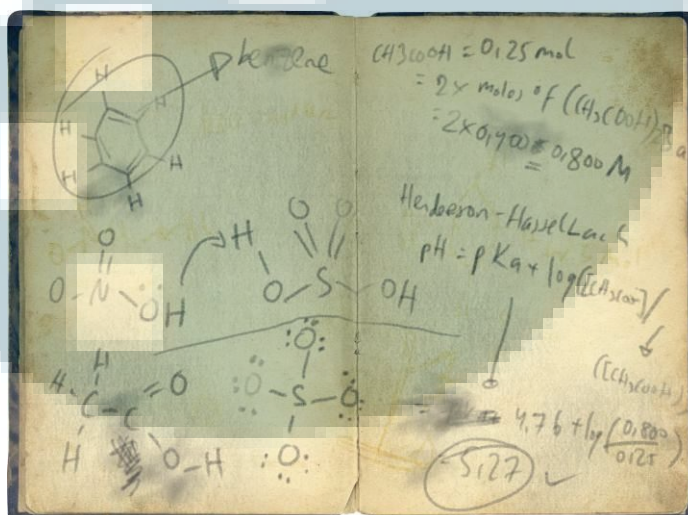
2.) Sosiologi

Sehari-hari *PC* bekerja sebagai ilmuwan di suatu kota kecil. Ia gemar melakukan eksperimen-eksperimen dan menghadiri konferensi-konferensi ilmuwan. Karakter memiliki dua karakter pendukung utama dalam game. Yang pertama adalah kekasih karakter dan yang kedua adalah seorang tua yang juga pernah terdampar di pulau yang sama dengan karakter utama. Keduanya memiliki peran yang berbeda dimana kekasih sang karakter berperan untuk memberikan alasan dan motivasi kenapa karakter harus pulang. Sementara orang tua yang juga pernah terdampar di pulau berfungsi sebagai mentor bagi karakter yang akan memberikan tips selama *game* dimainkan. Bagian ini penulis kutip dari Dave (2013).

3.) Psikologi

PC memiliki karakteristik yang ceria, berapi-api, dan bersemangat yang biasa disebut dengan istilah *hot-blooded*. Sebagai seorang ilmuwan, ia memiliki rasa keingintahuan yang tinggi. Dengan berbekal pengetahuan

tentang ilmu kimia, ia mencoba membuat beberapa ramuan dengan hasil alam dari pulau *Indictus*. Ia juga selalu membawa sebuah catatan kecil yang disimpan di dalam *pouch* untuk mencatat rumus-rumus dan hal-hal yang berhubungan dengan pekerjaannya. Kebiasaan ini kemudian diterapkan saat ia terdampar dalam pulau *Indictus* dimana catatan tersebut akan digunakan sebagai *GUI* untuk menu seperti *inventory system*, *monster data*, dan lain sebagainya. Penulis memilih tekstur kertas yang sedikit usang karena *PC* selalu membawa catatan ini sejak sebelum terdampar di pulau *Indictus*.

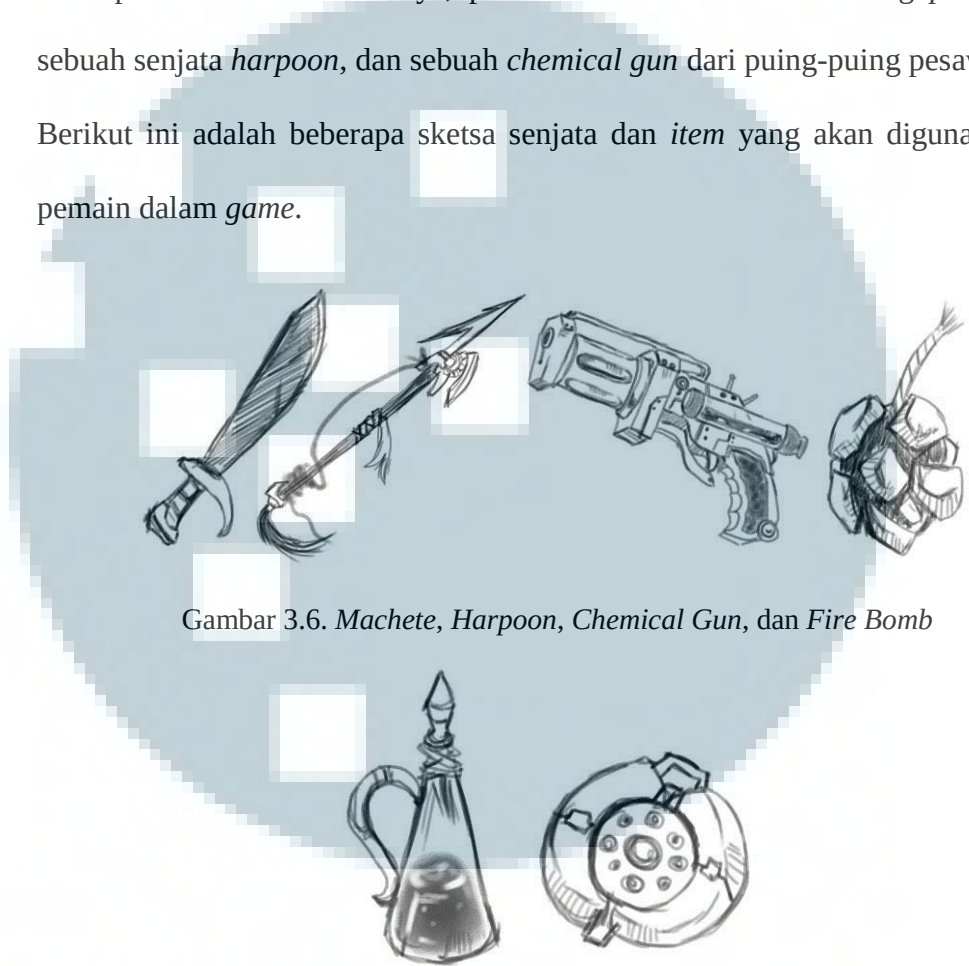


Gambar 3.5. Catatan yang Digunakan oleh *Player Character*

4.) *Items and Weapons*

Saat *PC* berhasil mendarat di pulau *Indictus*, hampir seluruh barang bawaannya hancur dalam pesawat yang jatuh. Beruntung sebuah *box* ikut terbawa oleh *PC*. *Box* ini berisi beberapa alat yang menjadi *item* dan senjata bawaan dalam *game*. Senjata standar yang digunakan dalam *game* adalah sebuah *machete* (sejenis golok) dengan panjang 55-60 cm. Karena *PC*

adalah seorang ilmuwan yang bekerja dalam bidang kimia, senjata dan *item* seperti *bomb* atau *potion* semuanya didapat dan diracik dari sumber daya alam pulau *Indictus*. Nantinya, pemain akan menemukan tabung *potion*, sebuah senjata *harpoon*, dan sebuah *chemical gun* dari puing-puing pesawat. Berikut ini adalah beberapa sketsa senjata dan *item* yang akan digunakan pemain dalam *game*.



Gambar 3.6. *Machete, Harpoon, Chemical Gun, dan Fire Bomb*

Gambar 3.7. *Med Health Potion dan Fire Trap*

3.2.2.2. *Non Player Character*

1.) *Underling (Snap Turtle)*

a.) *Fisiologi*

Secara fisik, hewan adalah makhluk *exoskeleton* yang mengambil bentuk dasar hewan kura-kura. Untuk mendapatkan keunikan tersendiri, penulis

menggabungkannya dengan *angler fish*. Cangkangnya merupakan hasil modifikasi penulis untuk menyesuaikan habitatnya di rawa-rawa.



Gambar 3.8. *Angler Fish*
(<http://swqg.files.wordpress.com/2011/02/anglerfish.jpg>)

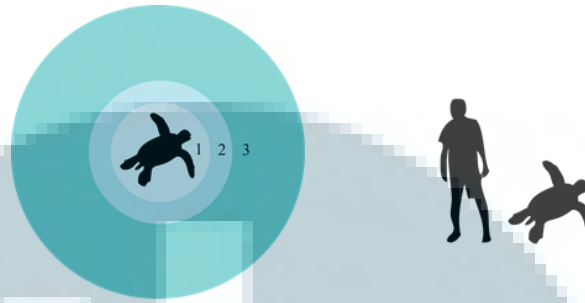
b.) Sosiologi

Hewan ini biasanya berburu secara individual. Apabila berkelompok, biasanya terdiri dari 2-3 ekor. Apabila seekor diserang, yang lainnya juga akan ikut menyerang *player*.

c.) Psikologi

Snap Turtle sebenarnya tidak termasuk hewan yang agresif. Apabila *player* tidak menyerangnya lebih dahulu maka ia akan mengacuhkan *player*.

d.) Perilaku



Gambar 3.9. Radius Serangan *Snap Turtle*
Dave (2013)

Snap Turtle memiliki radius seperti pada gambar diatas. Radius ini tidak akan muncul ketika pemain tidak menyerang snap turtle. Penjelasan sifat berdasarkan radius adalah sebagai berikut:

- 1.) Radius 1 adalah radius dimana karakter ada dalam jarak terdekat dengan *Snap Turtle*.
- 2.) Radius 2 adalah radius dimana *Snap Turtle* akan mengejar karakter sampai ia berada dalam radius untuk menggigit. Jika selama 5 *Snap Turtle* tidak menggigit pemain, maka ia akan melakukan serangan jarak jauh yang mengakibatkan pengurangan *Health Point (HP)* karakter.
- 3.) Radius 3 adalah radius dimana *Snap Turtle* akan menyerang dengan serangan jarak jauh yang mengakibatkan pengurangan *HP* karakter. *Snap Turtle* tidak akan mendekat, tapi jika karakter keluar dari radius 3, maka *Snap Turtle* akan berhenti menyerang pemain.

2.) *Boss (White God)*

a.) Fisiologi

Boss merupakan gabungan antara beberapa hewan seperti bison, gorila, tyrannosaurus, dan burung. Tanduk pada bison sengaja diperbesar sebagai simbol keberanian, kehormatan, dan kekuatan, sama hal nya dengan jaman dahulu helm peperangan dihiasi dengan tanduk. Pernyataan ini dikutip dari Nielsen (2010) dalam artikel berjudul *Animal Horn Symbolism*. Beberapa referensi yang penulis gunakan adalah dewa-dewa Mesir dan Celtic seperti Banebdjedet *the God of Fertility*, Bat, dan Cernunnos *the Horned One*.



Gambar 3.10. Banebdjedet *the God of Fertility*
(<http://aznewage.org/ram5.jpg>)



Gambar 3.11. Dewi Bat
(<http://www.egyptianmyths.net/images/bat.jpg>)



Gambar 3.12. Cernunnos *the Horned One*
(<http://celticawitch.files.wordpress.com/2012/09/cernunnos-on-gundestrup-cauldron.jpg>)

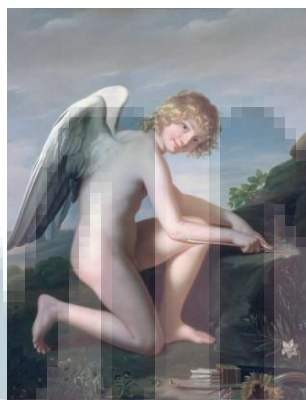
Untuk menekankan kekuatannya penulis menggambar lengan gorilla, terinspirasi dari film *King Kong* karya Peter Jackson dimana *King Kong* adalah hewan yang kuat dan destruktif. Sebagai senjata tambahan, penulis membuat senjata berbentuk sabit di lengannya. Kaki *T-rex* dan ekor yang panjang dan besar digunakan sebagai keseimbangan, agar *boss* terlihat berdiri kokoh. Sayap pada *boss* tidak digunakan untuk terbang, tetapi dimaksudkan untuk memberikan kesan spiritual dan magis, seperti yang dikutip dari Samira (2008) dalam artikel berjudul *Wing Symbols*. Sayap pada *White God* mengambil beberapa referensi dari mitologi Yunani dan Roma, seperti Boreas *the Greek God of North Wind*, Nike *the Winged Goddess of Victory*, dan Cupid *the God of Desire*.



Gambar 3.13. Boreas the Greek God of North Wind
(http://www.theoi.com/image/img_boreas.jpg)

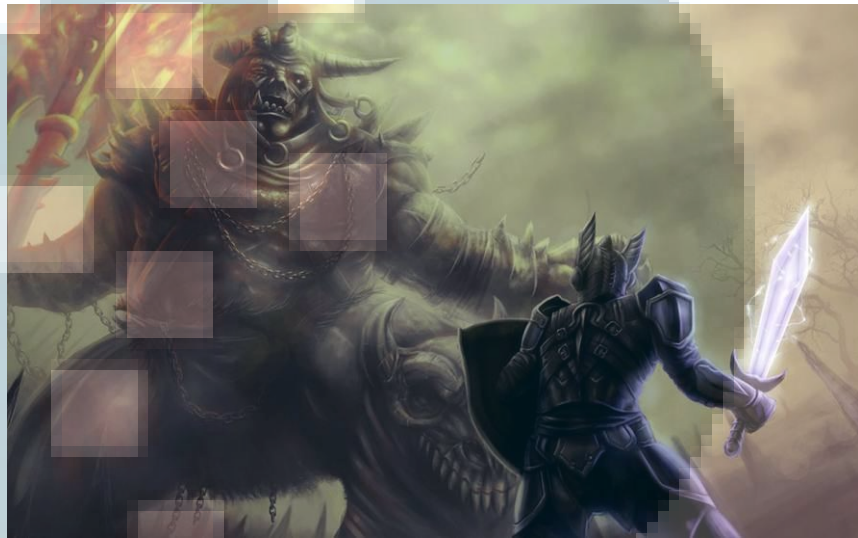


Gambar 3.14. Nike the Winged Goddess of Victory
(<http://thanasis.com/store/nike2.jpg>)



Gambar 3.15. Cupid the God of Desire
(<http://images.fineartamerica.com/images-medium-large/cupid-sharpening-his-arrows-robert-lefevre-.jpg>)

Pada saat pemain melawan *Boss* dalam *game*, kamera secara otomatis akan menyorot *Player Character* dan *Boss* dari bawah yang dikenal dengan sebutan *worm eye view*. Hal ini dilakukan untuk menambahkan kesan tinggi dan kuat pada *White God*. Berikut ini adalah referensi yang menggambarkan penempatan kamera tersebut.



Gambar 3.16. Referensi Sudut Pandang *Worm Eye*
(<http://www.wallpapershunt.com/user-content/uploads/wall/o/4/Hero-Fighting-Forest-Giant-Widescreen-Wallpaper.jpg>)

b.) Sosiologi

Dalam pulau tidak berpenghuni, terdapat lima penjaga pulau yang harus dihadapi *player* agar bisa menamatkan game dimana *Boss* adalah salah satunya. *Boss* tinggal di dunia yang berbeda dengan manusia. Tugasnya adalah melindungi kunci portal yang ada dalam pulau tersebut.

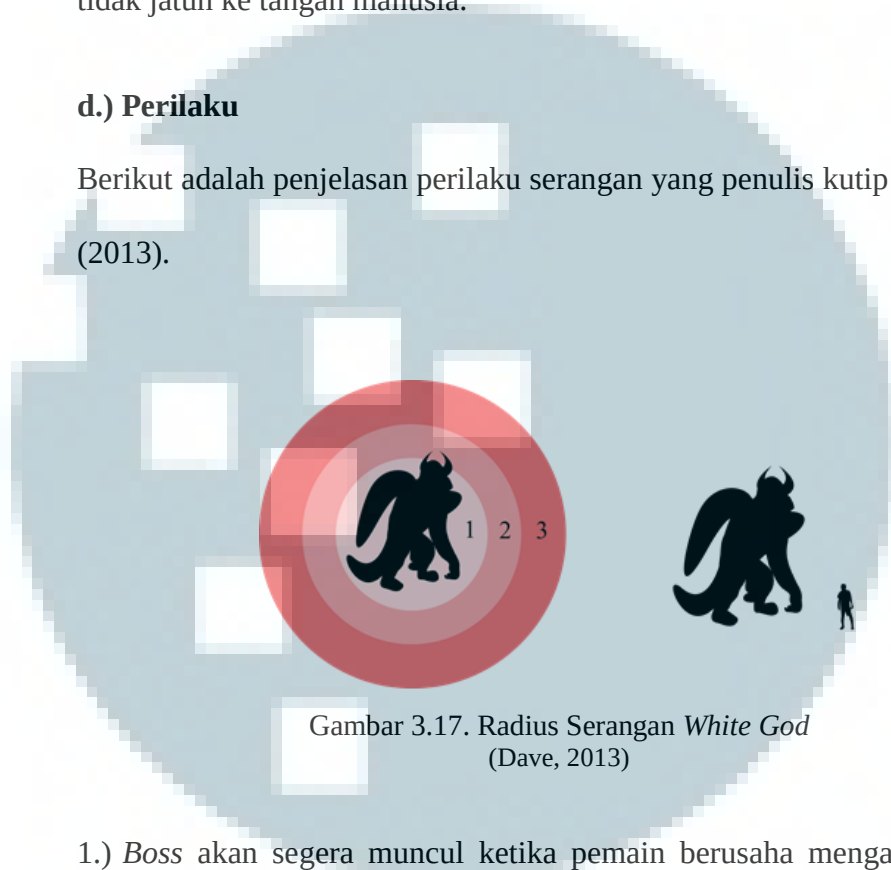
c.) Psikologi

Nama yang diberikan oleh *PC* kepada *Boss* adalah *White God*, karena kesan pertama yang ia dapatkan adalah makhluk suci raksasa berwarna

putih. *Boss* memiliki karakter yang tenang, terlihat dari wajahnya yang tidak beringas, namun ia akan sekuat tenaga menjaga kunci portal agar tidak jatuh ke tangan manusia.

d.) Perilaku

Berikut adalah penjelasan perilaku serangan yang penulis kutip dari Dave (2013).



Gambar 3.17. Radius Serangan *White God* (Dave, 2013)

- 1.) *Boss* akan segera muncul ketika pemain berusaha mengambil batu sakral berelemen besi. *Boss* memiliki sifat agresif dan tidak akan berhenti menyerang sampai ia terbunuh. Penjelasan sifat berdasarkan radius adalah sebagai berikut.
- 2.) Radius 1 adalah radius terdekat antara *Boss* dengan karakter. Pada radius ini, *Boss* akan menyerang dengan cara memukul karakter dari atas dengan kedua tangannya.
- 3.) Dalam radius 2, *Boss* akan berusaha menghempaskan pemain dengan tangannya.

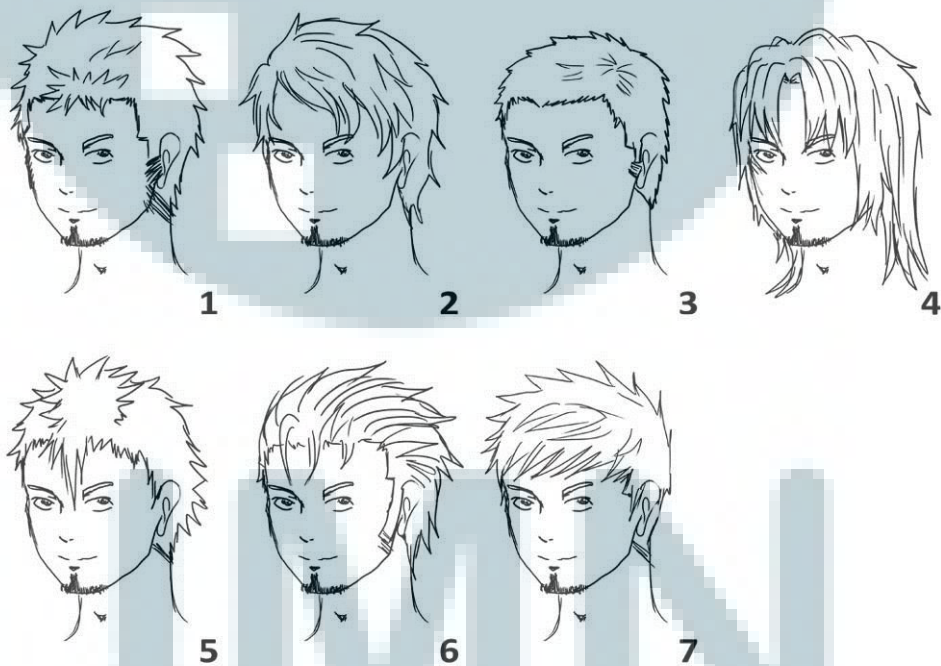
4.) Dalam radius 3, *Boss* akan memutar badannya dan menyerang dengan ekornya.

5.) Di luar radius 3, *Boss* akan menyerang dengan cara berlari dan menghantam karakter dengan badannya.

3.2.3. Sketsa Kasar

3.2.3.1. Sketsa *Player Character*

Berdasarkan konsep karakter yang telah disusun bersama, penulis mencoba membuat sketsa kasar dengan beberapa alternatif.



Gambar 3.18. Variasi Rambut Karakter

Setelah melakukan perundingan dengan memilih mana model yang cocok dengan tipe *hot-blooded* namun terlihat cukup rapi, tim sepakat memilih nomer 6. Sebelum menentukan pakaian yang digunakan karakter *in*

game, penulis merancang set pakaian yang digunakan karakter dalam kesehariannya berupa referensi pakaian pada jaman *Victoria*. Pada jaman itu para pria biasa mengenakan topi, kemeja panjang, rompi, *coat*, dasi kupu-kupu maupun *cravats* (kain yang dipilin dan dikalungkan di bawah kerah seperti dasi), celana panjang, dan sepatu berlapis kain yang disebut *spats*.



Gambar 3.19. Variasi Pakaian Karakter

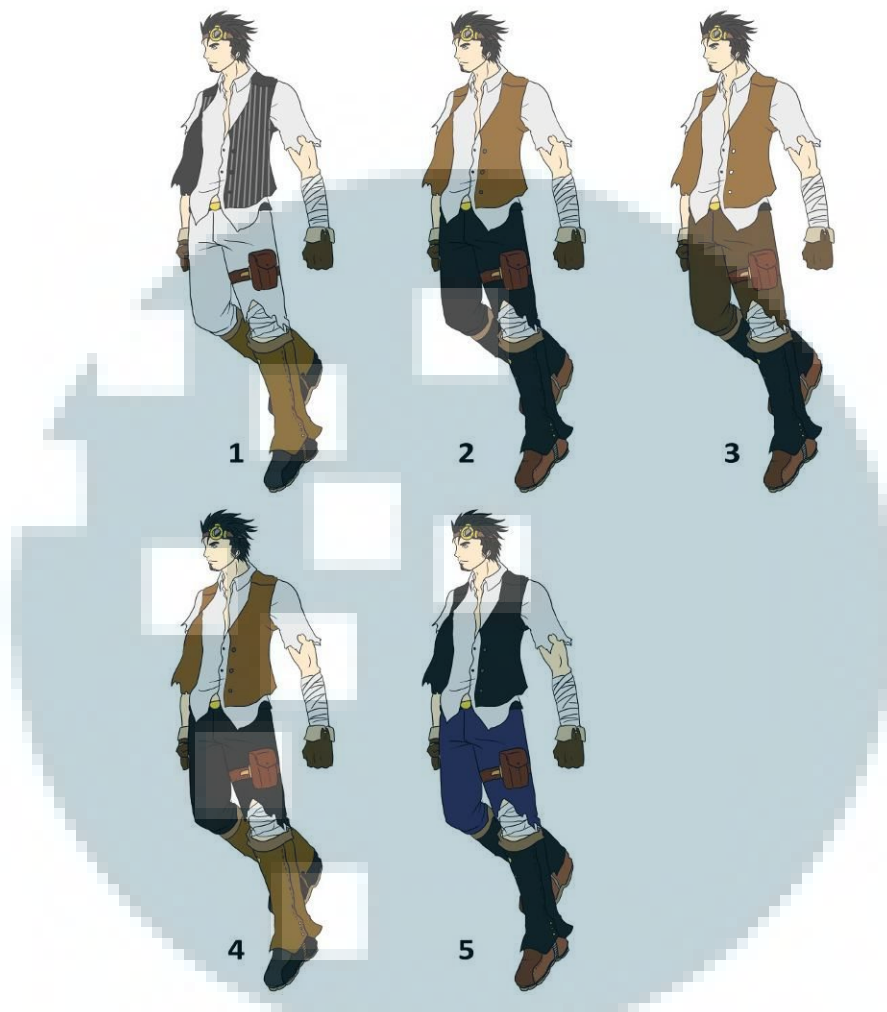
Namun setelah menelusuri lebih lanjut penulis menemukan bahwa *top hats* bisanya dikenakan oleh kalangan atas dan *coat* kurang merepresentasikan ilmuwan. Penulis kemudian menghapus *top hats* dan mengganti coat menjadi *lab coat*.



Gambar 3.20. Pakaian Karakter yang Telah Direvisi

Penulis melanjutkan lebih jauh untuk pakaian *in game*, dimana pakaian *PC* menjadi lusuh dan compang-camping setelah sehari-hari terdampar di dalam pulau tak berpenghuni. Penulis juga menyertakan beberapa alternatif warna sebagai bahan pertimbangan penulis.

UMN



Gambar 3.21. Variasi Warna Pakaian Karakter

Menurut Beyer (2011) dalam website berjudul *Steaming Apparel*, tidak ada batasan warna dalam fashion *Steampunk*, hanya saja terdapat perbedaan kelas antara warna cerah dan warna gelap. Warna cerah biasanya digunakan oleh kalangan atas, dimana harga pewarna kain yang kuat lebih mahal dan memberikan kesan bersih. Sebaliknya warna gelap biasa ditemukan pada kalangan biasa atau pekerja. Genre *Steampunk* sendiri identik dengan warna coklat karena sebagian besar kostum *Steampunk*

memakai bahan kulit (*leather*). Warna lain yang sama identiknya adalah putih dan hitam.

3.2.3.2. Sketsa *Non Player Character*

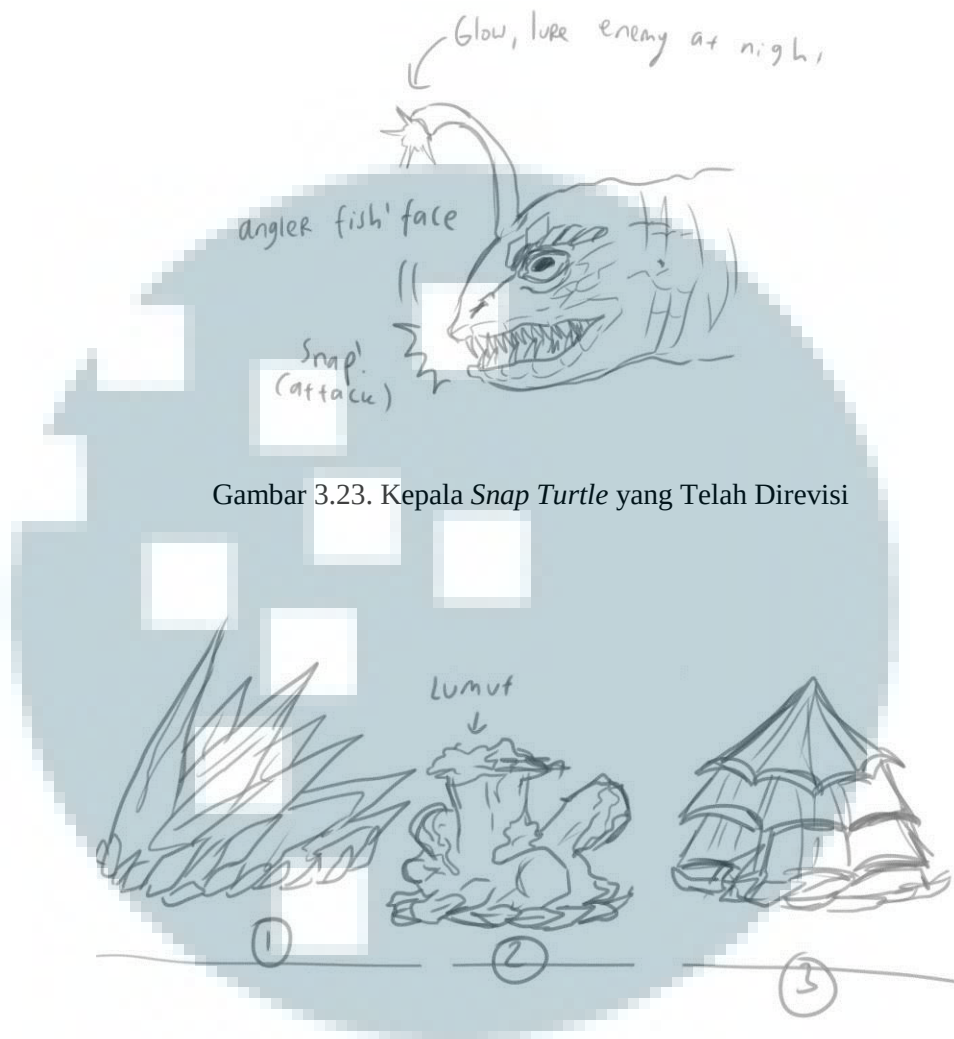
Awalnya, penulis membuat empat macam jenis *Enemy* dan satu *Boss*. Namun karena keterbatasan waktu, penulis menyederhanakannya menjadi satu *Enemy* dan satu *Boss*. Seperti yang telah dijelaskan pada bab 2 mengenai *silhouette painting*, penulis menerapkannya pada pencarian bentuk monster.

a. *Enemy*



Gambar 3.22. *Silhouette Painting Snap Turtle*

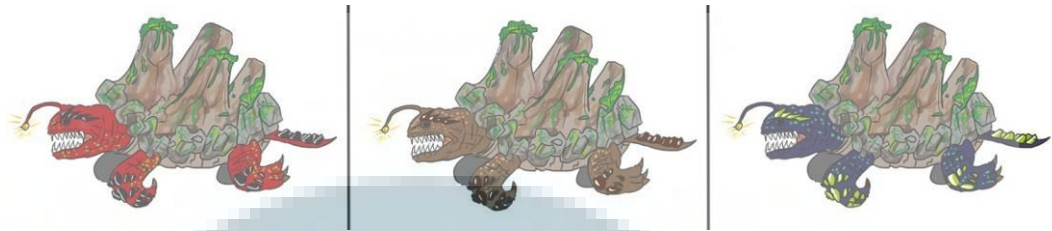
Penulis mencoba memberikan bentuk yang unik dengan merubah moncong kura-kura menjadi *angler fish*. Batu-batu berlian tajam pada kura-kura dinilai terlalu berlebihan sehingga penulis membuat tiga alternatif untuk didiskusikan dengan rekan.



Gambar 3.23. Kepala *Snap Turtle* yang Telah Direvisi

Gambar 3.24. Alternatif Cangkang *Snap Turtle*

Kemudian tim sepakat untuk memilih nomer dua dengan alasan batu-batuan akan tersamar dengan baik di rawa-rawa.



Gambar 3.25. Alternatif Warna *Snap Turtle*

Penulis memilih warna dasar merah untuk menunjukkan keagresifannya. Selain itu, warna merah sering dipakai oleh karakter berwujud iblis, misalnya *Hellboy*. Dalam hal ini, wajah *Snap Turtle* menyeramkan seperti iblis.

b. Boss



Gambar 3.26. Sketsa Awal *White God*

Karena dianggap terlalu datar, penulis menambahkan sedikit *exoskeleton* dan senjata pada tubuhnya.



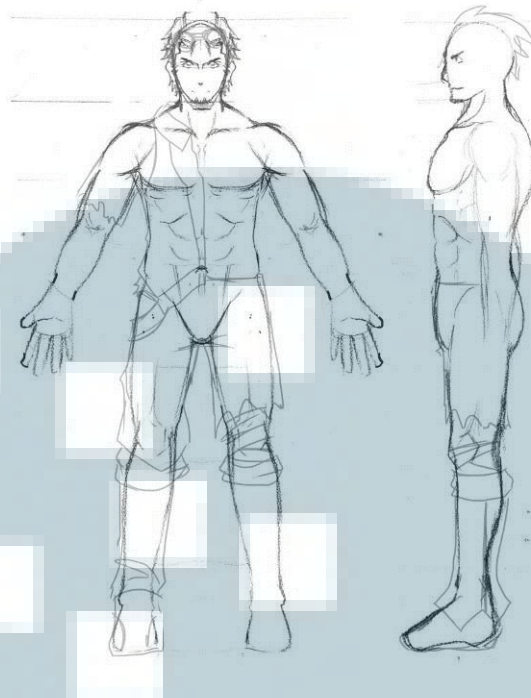
Gambar 3.27. Sketsa Akhir Karakter *White God*

3.2.4. *Model Sheet*

Sebuah *model sheet* adalah guide dalam melakukan *modeling*. Agar dapat melakukan *modeling* wajah dengan lebih detail, penulis juga membuat *model sheet* kepala karakter dalam tampak dan samping. Nantinya model kepala dan tubuh akan dilakukan secara terpisah baru kemudian disatukan dengan *weld*.



Gambar 3.28. *Model Sheet* Kepala *Player Character*



Gambar 3.29. *Model Sheet Tubuh dan Pakaian Player Character*

3.3. Produksi

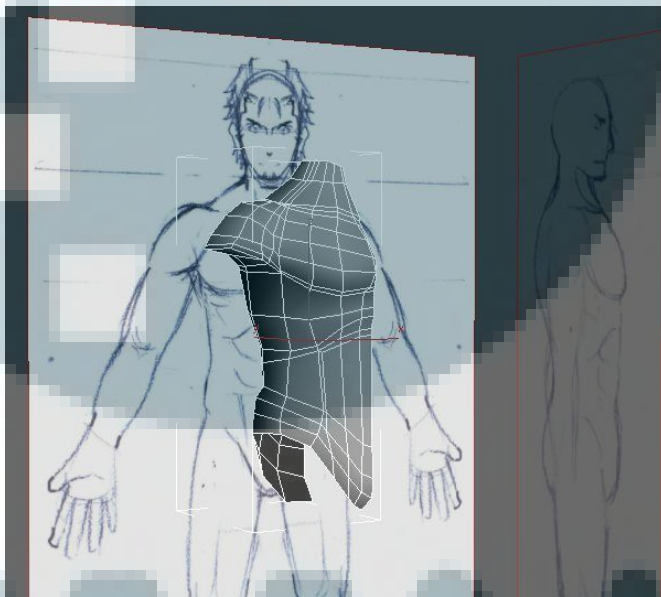
Pada tahap produksi, sketsa yang telah dibuat ditransformasi menjadi bentuk 3D yang bisa dikendalikan yang dikenal dengan istilah *modeling*. Nantinya akan dilanjutkan dengan *baking*, *texturing*, *rigging*, dan penganimasian karakter.

3.3.1. *Modeling*

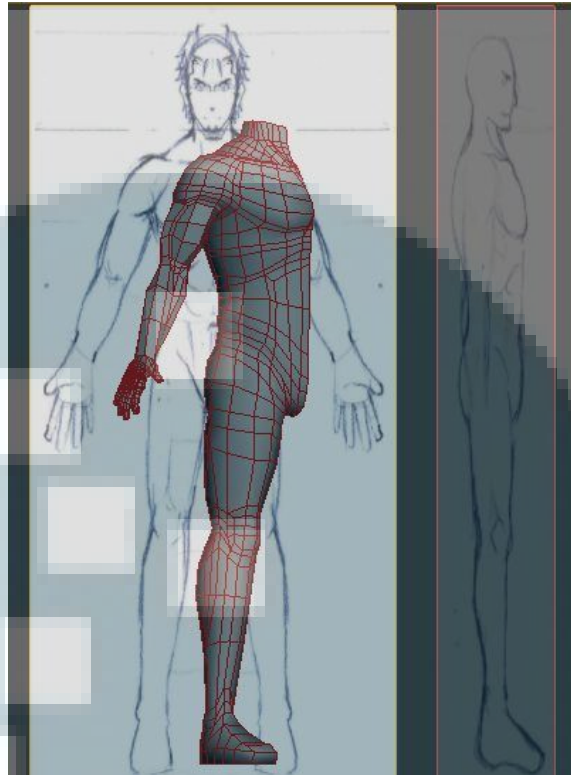
Dalam pengerjaan Tugas Akhir ini, penulis menggunakan program *3ds Max 2011* untuk membuat model dasar sebelum dilanjutkan ke *digital sculpting*. Penulis menggunakan teknik *Edge modeling* dan *Box modeling* untuk menyelesaikan *Player Character*.

3.3.1.1. Bagian Tubuh

Penulis memutuskan untuk membuat tubuh karakter lebih dulu karena dirasa lebih mudah. Dengan teknik *Box Modeling*, penulis membuat *base* tubuh karakter sebelum diberikan pakaian dan aksesoris. *Box* tersebut nantinya akan dimodifikasi menggunakan teknik seperti *extrude*, *connect*, *cut*, *weld*, dan lain sebagainya. Penulis hanya membuat setengah dari bagian tubuh karena *modifier symmetry* akan secara otomatis akan membentuk bagian yang setengahnya.

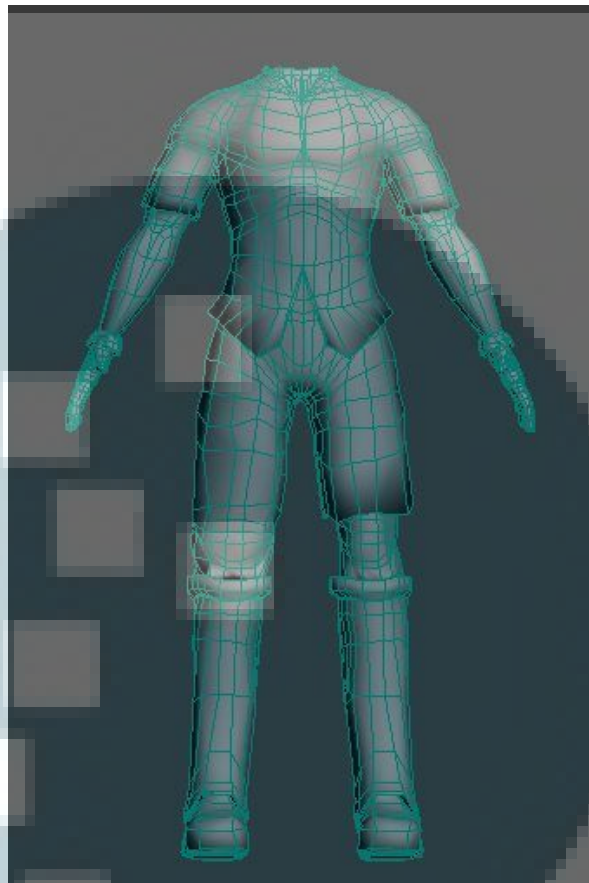


Gambar 3.30. Proses Pembuatan Tubuh dengan Teknik *Box Modeling*



Gambar 3.31. Keseluruhan *Base Model* yang Telah Selesai

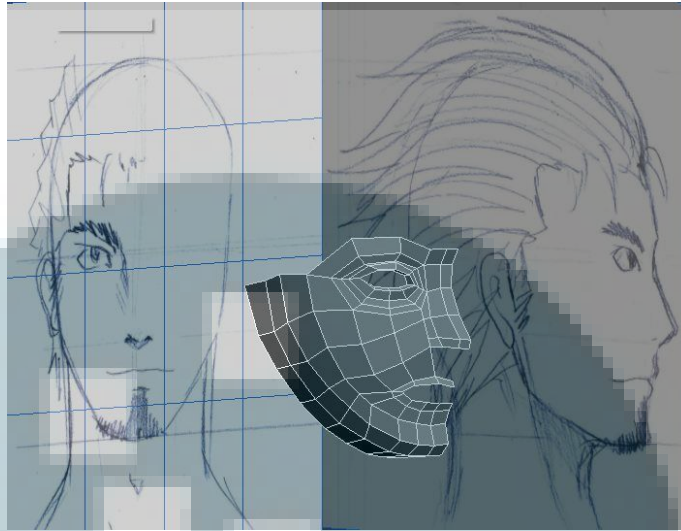
Mempertimbangkan bahwa celana panjang karakter tidak simetris, penulis melakukan *collapse* pada *modifier symmetry* agar penulis dapat melakukan *edit* pada kedua sisi model. Dengan teknik *extrude* penulis mulai membuat pakaian karakter satu persatu.



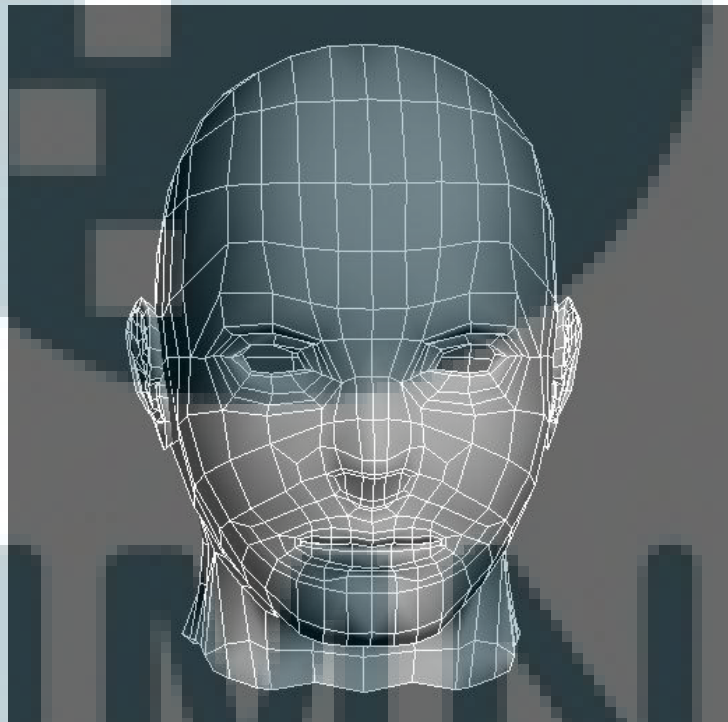
Gambar 3.32. Hasil Akhir Tubuh Karakter

3.3.1.2. Bagian Kepala

Wajah adalah salah satu bagian penting dalam penampilan karakter. Karena itu, penulis melakukan *Edge Modeling* agar dapat membentuk topologi wajah dengan lebih tepat dan detil. Sama halnya dengan pembuatan tubuh model, penulis cukup membuat setengah dari kepala model.



Gambar 3.33. Proses Pembuatan Wajah dengan *Edge Modeling*

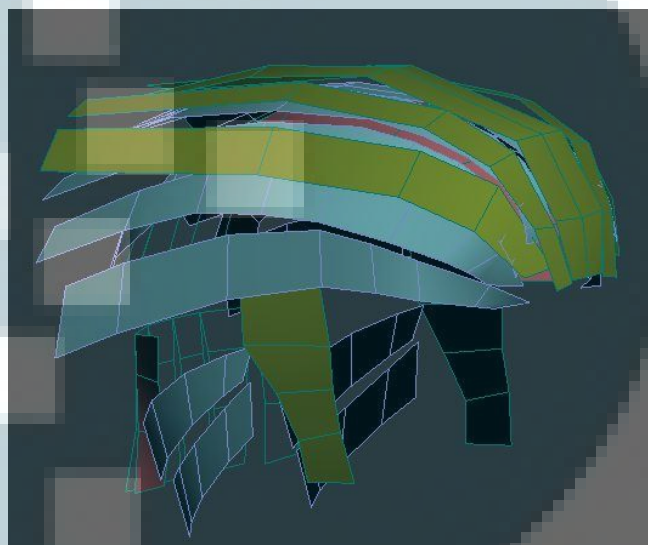


Gambar 3.34. Model Kepala Final

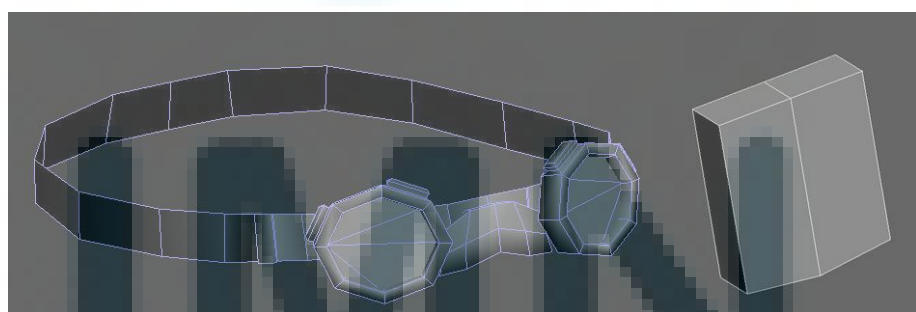
3.3.1.3. Rambut dan Aksesoris

Rambut adalah salah satu bagian yang membutuhkan *poly* dalam jumlah sangat banyak yang tentu saja tidak bisa diaplikasikan ke dalam *game*. Trik

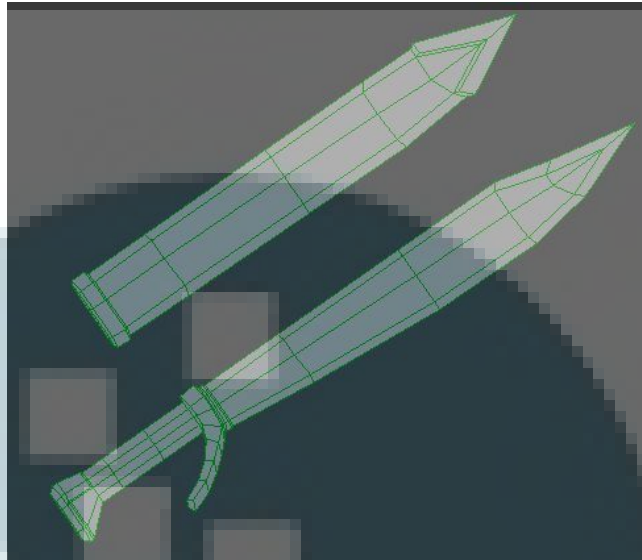
yang biasa digunakan agar karakter memiliki rambut yang natural adalah dengan membuat *plane* yang disusun berlapis-lapis kemudian ditekstur dan diberikan *opacity map*. Teknik ini akan dibahas lebih lanjut pada subbab *texturing*. Untuk *google* dan *pouch*, penulis menggunakan *standart primitives* seperti *tube*, *plane*, dan *box*.



Gambar 3.35. *Plane* Berlapis Sebagai *Hair*



Gambar 3.36. *Google* dan *Pouch*

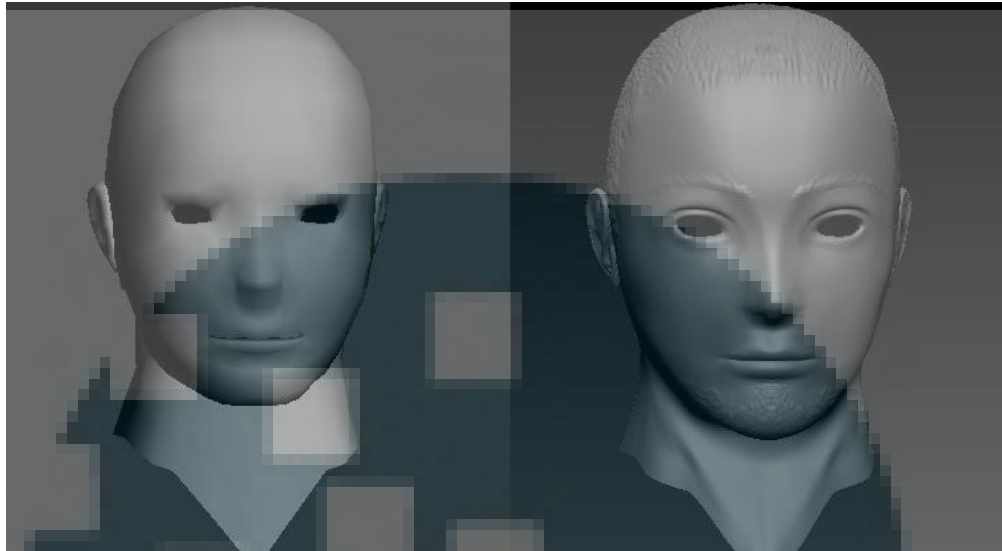


Gambar 3.37. Senjata *Machete*

Untuk memudahkan proses *digital sculpting*, model *low poly* kemudian dipisah per bagian dengan cara *detach*. Bagian-bagian itu adalah kepala, kemeja, lengan, celana, lutut, betis, dan sepatu. Dengan demikian penulis dapat melakukan *sculpting* per bagian tanpa khawatir tidak sengaja merusak bagian lainnya.

3.3.2. Proses *Digital Sculpting*

Seperti yang telah dijelaskan pada subbab *Digital Sculpting*, model *low poly* telah diimpor menggunakan *GoZ* dan diberikan *SDiv*. Program *ZBrush* memberikan banyak jenis pilihan *brush* untuk memberikan efek yang berbeda-beda pada setiap goresannya. Penulis banyak menggunakan *standart*, *rake*, *clay*, *flatten*, dan *pinch* untuk membuat detail. Pada bagian-bagian tertentu, penulis menggunakan *Active Symmetry* untuk membuat detail yang sama persis antara bagian kanan dan kiri.



Gambar 3.38. Perbandingan Kepala *Low Poly* dan *High Poly*

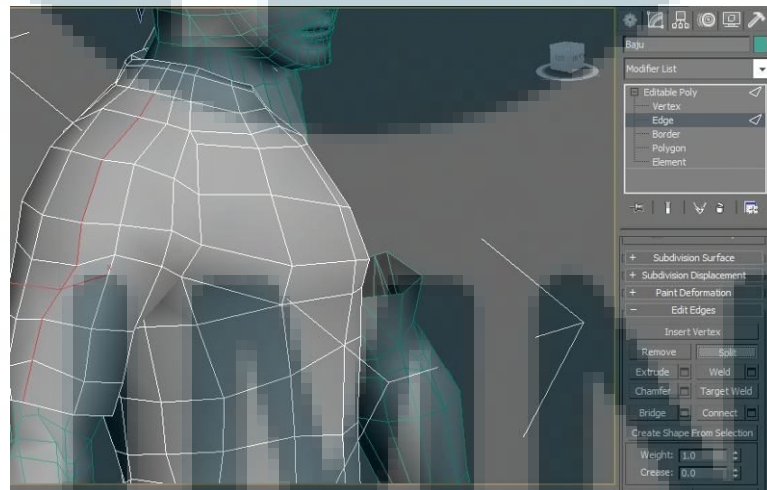
Untuk membuat *vest* penulis membuat masking berbentuk rompi pada bagian kemeja. Dengan *tool Extract*, *ZBrush* akan membentuk *subtool* baru dengan ketebalan tertentu. Penulis menilai cara ini lebih mudah dan cepat daripada melakukan *modeling* pada *3ds Max* karena penulis harus mengatur kembali *vertex* agar mendapat bentuk yang diinginkan.

Guna mengurangi jumlah *polygon* yang membengkak pada saat pengeksportan kembali ke *3ds Max* agar tidak terjadi *crash*, penulis menggunakan plugin *Decimation Master*, sebuah plugin yang dapat mengurangi jumlah *polygon* dalam jumlah yang besar namun tetap mempertahankan detailnya. Dengan *plugin* ini, model *high poly* yang berjumlah 779.856 *poly* dapat direduksi menjadi 650.144 *poly*.

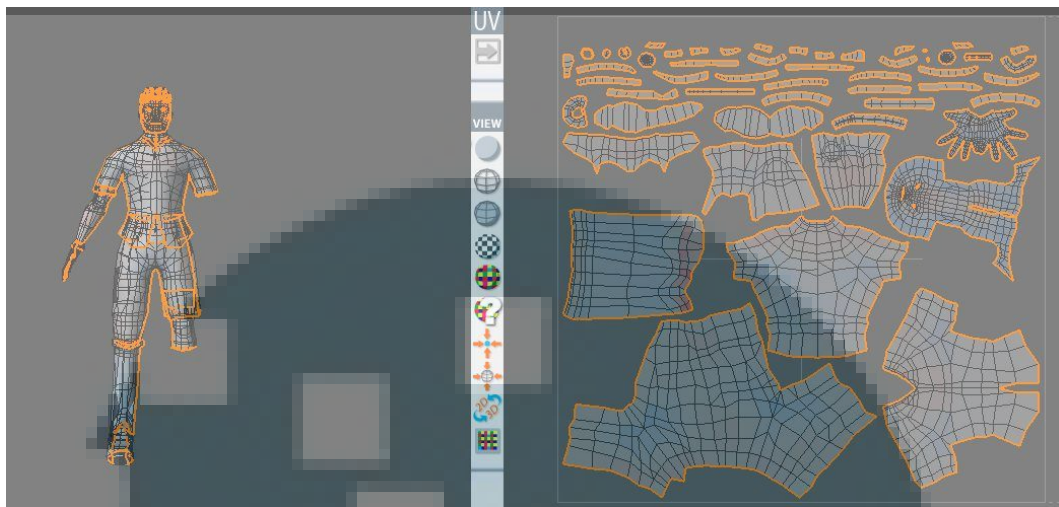
3.3.3. *UV Mapping*

UV Mapping adalah proses pemetaan *polygon* dari karakter 3D menjadi bentuk 2D untuk selanjutnya ditekstur. Sebelum melakukan *baking*, model *low poly* harus

sudah memiliki *UVW Map* sehingga pada saat melakukan proyeksi, model *low poly* sudah memiliki *template* yang benar untuk menampung detail dari model *high poly*. Penulis menggunakan program *Unfold3D* sebagai alat bantu proses *unwrapping*. Ada dua cara melakukan *unwrapping* dengan *Unfold3D*. Cara pertama adalah dengan menggunakan *point-to-point seam* pada program *Unfold* itu sendiri untuk selanjutnya dilakukan *cut* dan *unfold*. Cara kedua adalah dengan melakukan *Edge Split* dalam program *3ds Max*. Keduanya mempunyai prinsip yang sama, yaitu membuat potongan-potongan pada *polygon* untuk direntangkan sebagai *UVW Texture*. Penulis menggunakan teknik kedua karena tidak terbiasa dengan kontrol dalam program *Unfold3D*. Teknik yang penulis gunakan didapatkan dari video tutorial berjudul *Quick UV Unwap & Layout Using Unfold 3D* oleh Putra (2011).



Gambar 3.39 *Edge Split*



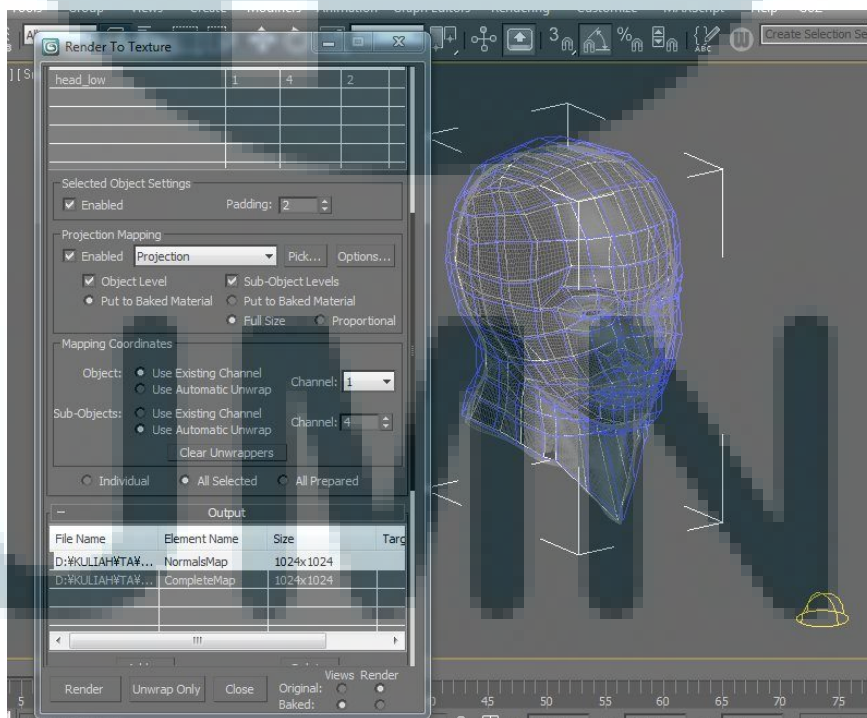
Gambar 3.40 Hasil *Unwrap* dengan *Unfold3D*

Model yang telah di *unwrap* kemudian diekspor kembali sebagai *OBJ* dan diimpor ke dalam *3ds Max*. Pada *Unfold3D*, masih terjadi *stretching* pada beberapa area yang ditandai dengan warna merah. Penulis memperbaikinya dengan menggunakan *relax tool* pada *modifier Unwrap UVW*. Terakhir model diberikan tekstur *checker* untuk menandakan bahwa sudah tidak ada *stretching*. Agar saat melakukan *texturing* penulis mendapat hasil yang maksimal, hasil *unwrap* yang ada kemudian diatur secara manual sehingga menyisakan ruang kosong yang sesedikit mungkin, kecuali pada bagian tubuh karena akan diisi dengan senjata. Wajah merupakan bagian dari karakter yang paling sering diperhatikan sehingga penulis memisahkannya dari anggota tubuh lainnya. Rambut dan aksesoris kepala juga dipisahkan agar terlihat lebih detail. Beberapa bagian tubuh hanya memiliki satu map karena memiliki tekstur yang simetris. Masing-masing map memiliki resolusi 1024X1024 *pixel*.

3.3.4. Baking

Penulis menggunakan fitur *3ds Max* yaitu *Render to Texture* untuk memindahkan detail ke dalam model *low poly*. Model *high poly* kemudian didempetkan dengan model *low poly*. *Baking* dilakukan per bagian, dimana setiap bagian akan menghasilkan map tersendiri yang nantinya akan digabung menggunakan program *Photoshop CS4*. Sumber cahaya pada proses ini menggunakan *Skylight* dengan *sample 30* dengan *render scanline*.

Map yang dihasilkan dari proses ini adalah *Complete Map* dan *Normal Map*. Penulis tidak menggunakan *Lighting Map* karena akan dirender secara *real time* pada *game engine*. Hasil dari proses *baking* ini akan dipetakan pada *UVW Template*. Teknik ini penulis peroleh dari video *tutorial Game Character Creation in 3ds Max and ZBrush* oleh Moraes (2012).



Gambar 3.41. Contoh Proses *Baking*

Sebuah *cage* (garis biru) adalah sebuah geometri *non-renderable* yang mensimulasikan cahaya dan bayangan. Untuk menghasilkan *normal map* yang sempurna, titik-titik pada *cage* tidak boleh saling tumpang tindih ataupun menembus obyek yang akan dilakukan *bake*. Penulis merapikan *cage* dengan cara mereset *cage* kemudian menskala *cage* sehingga menutupi obyek. Apabila masih ada titik-titik yang saling tumpang tindih, penulis merapikan *cage* secara manual.

3.3.4.1. *Complete Map*

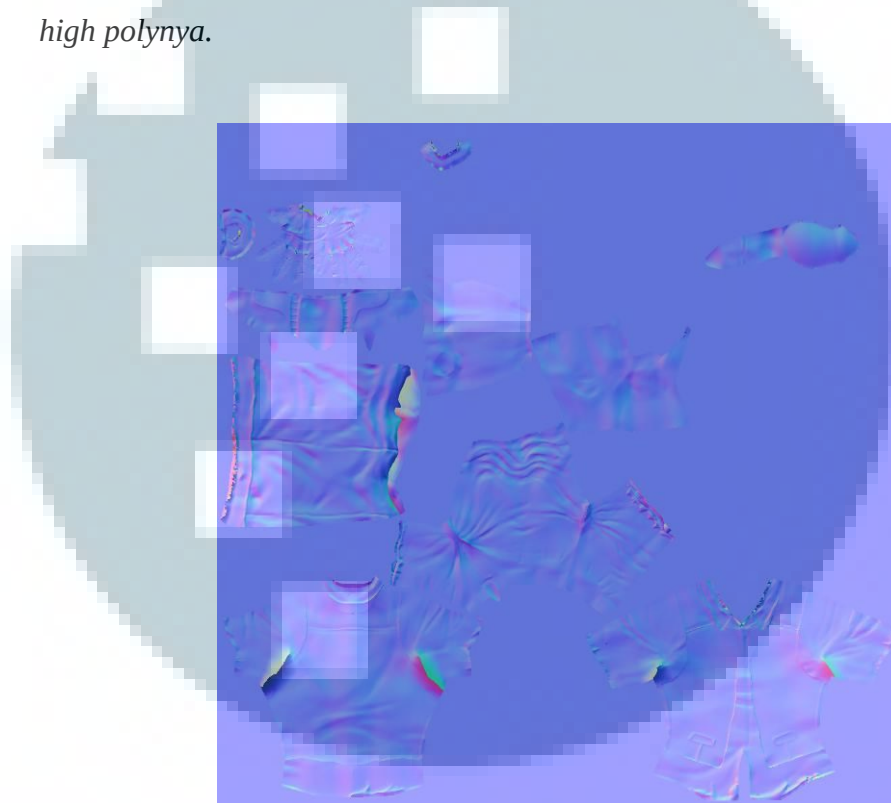
Complete Map digunakan untuk menyimpan pemetaan bayangan yang dihasilkan dari sumber cahaya. Penulis menggunakan warna gradasi abu-abu karena akan digunakan dalam proses *texturing* secara *digital painting*. Bercak-bercak berwarna hitam dan merah pada map akan diperbaiki secara manual dalam program *Photoshop*. Penulis menyimpan *Complete Map* dengan format *TARGA*.



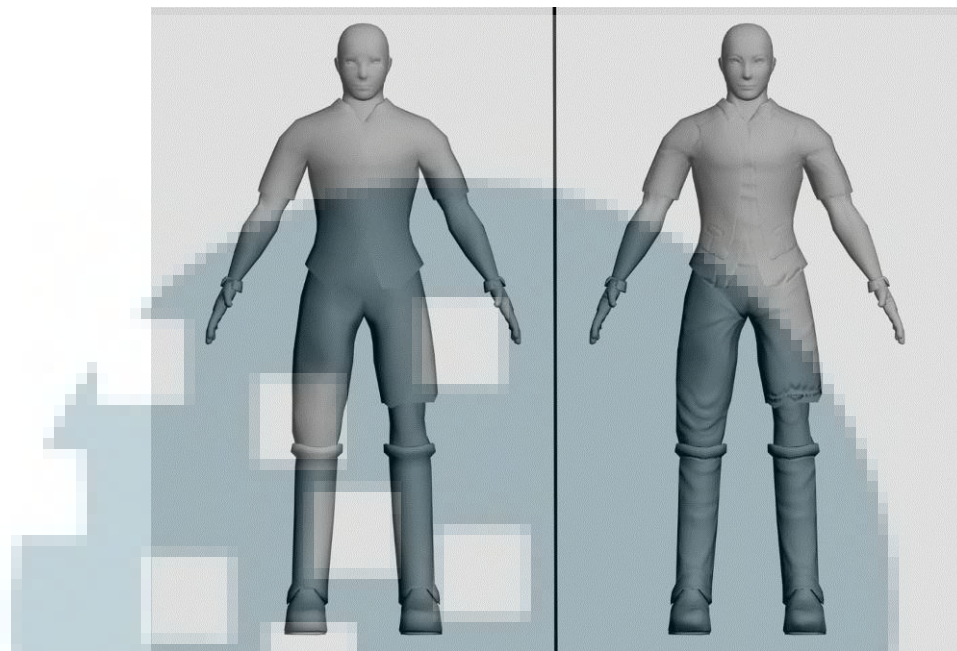
Gambar 3.42. Contoh *Complete Map* pada *Player Character*

3.3.4.2. *Normal Map*

Normal Map akan digunakan sebagai bump untuk model *low poly*. Model yang telah diberi *normal map* nantinya akan mempunyai ilusi kedalaman. Dengan cara ini sebuah model *low poly* bisa terlihat detail seperti model *high polynya*.



Gambar 3.43. Contoh *Normal Map* yang Telah Disatukan



Gambar 3.44. Perbandingan Model *Low Poly* Tanpa dan dengan *Normal Bump*

3.3.5. *Texturing*

Pada proses teksturing, penulis menggunakan program *Photoshop CS4* dan *graphic tablet* untuk mempermudah kerja. Sementara melakukan *digital painting*, penulis mencoba melihat hasilnya apabila diaplikasikan ke dalam model. Hasil akhir yang diperoleh dari proses ini adalah *Complete Map*, *Opacity Map*, dan *Specular Map*. Menurut panduan pengguna program *Unity*, tekstur yang ideal sebaiknya memiliki *power of two*, yaitu 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 atau 2048 pixel dimana tekstur yang tidak memiliki *power of two* hanya akan digunakan untuk *GUI (Graphical User Interface)*.

3.3.5.1. *Diffuse Map*

Dengan memanfaatkan *Complete Map* yang sudah memiliki *shading* atau *value*, penulis hanya menambahkan warna dan detail sesuai kebutuhan.

Teknik ini biasa disebut dengan *Grayscale Painting*. Untuk menambah kesan realistis, penulis memberikan tambahan tekstur seperti kotor dan lusuh. Penulis bereksperimen dengan bermacam-macam *brush* agar bisa mendapat tekstur sesuai keinginan.



Gambar 3.45. *Diffuse Map* Tubuh Karakter

3.3.5.2. *Opacity Map*

Opacity Map menentukan bagian mana yang akan terlihat pada saat *render*. Sebuah model tanpa *opacity map* akan dirender seluruhnya. *Opacity map* terdiri dari warna putih dan hitam. Apabila terdapat gradasi warna putih ke abu-abu, pada saat *rendering* akan memberikan efek transparansi. *Opacity map* akan digabung ke dalam *diffuse map* dalam bentuk *alpha*

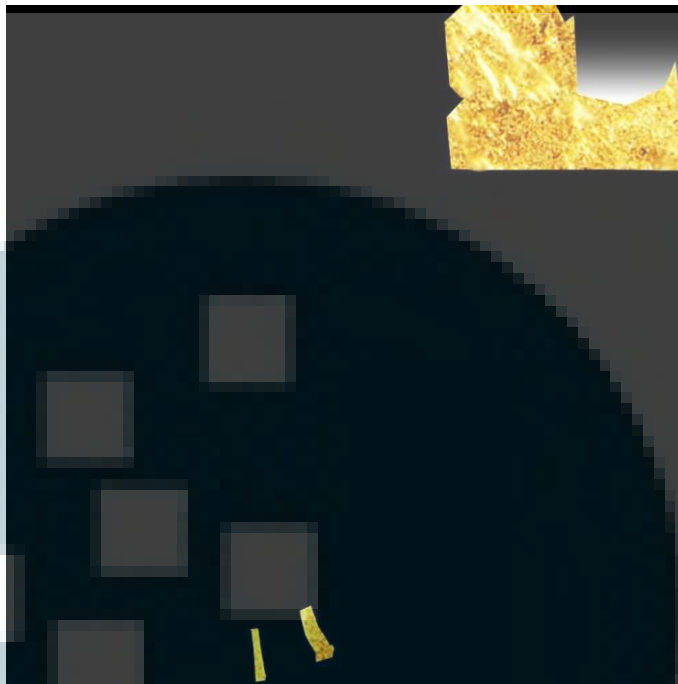
channel yang disimpan dalam format *PNG* atau *TARGA*. Pada karakter, *opacity map* digunakan untuk rambut dan sobekan pada pakaian.



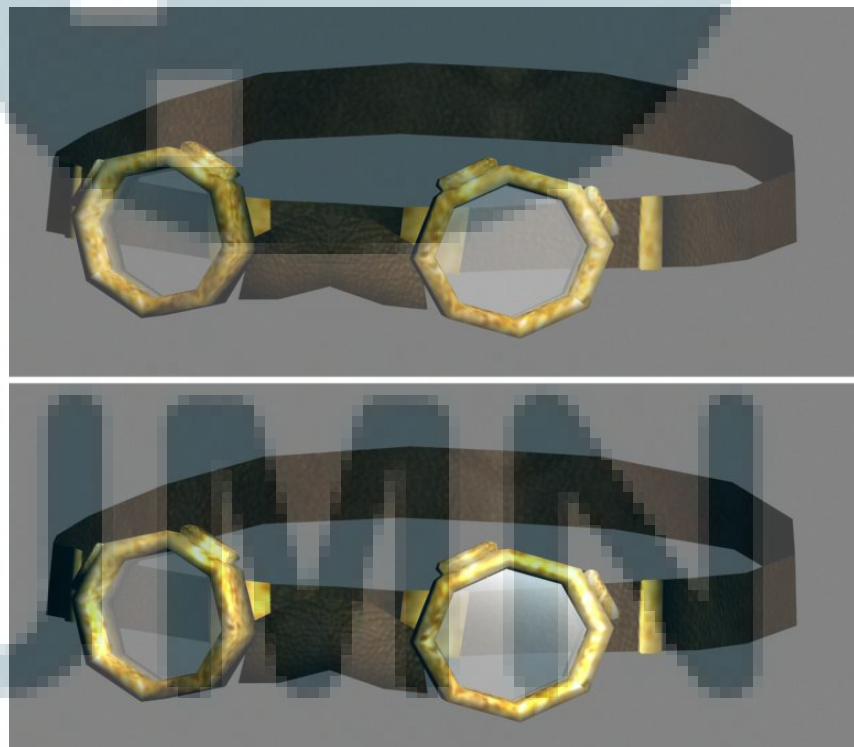
Gambar 3.46. Perbandingan Model Tanpa dan Dengan *Opacity Map*

3.3.5.3. *Specular Map*

Specular map berfungsi untuk memberikan efek mengkilat pada *mesh*. Sama seperti *opacity map*, bagian yang tidak mengkilat diberikan warna hitam. Warna spekular bisa bermacam-macam, tidak harus putih. Penulis dapat mengatur tingkat seberapa terang cahaya melalui *glossiness* yang terdapat pada kotak dialog material. Teknik ini penulis peroleh dari video *tutorial Game Character Creation in 3ds Max and ZBrush* oleh Moraes (2012). Pembuatan *Specular Map* dilakukan dalam program *Photoshop*.



Gambar 3.47. Contoh *Specular Map*



Gambar 3.48. Perbandingan *Google* Tanpa dan Dengan *Specular Map*

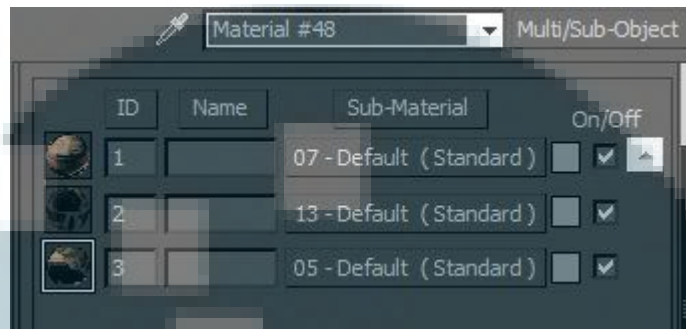


Gambar 3.49. Hasil Akhir *Player Character*

3.3.6. *Material ID*

Pada model 3D yang penulis kerjakan memiliki tekstur yang terpisah, contohnya tekstur badan dan *google*. Hal ini dapat menimbulkan masalah karena setelah semua bagian dilakukan *attach*, maka penulis hanya bisa memberikan material untuk satu bagian saja, misalnya badan. Oleh karena itu, tiap bagian harus diberikan *Material ID*. Menurut panduan pengguna dari *3ds Max*, *Material ID* adalah angka yang menentukan sub-material mana yang akan diberikan kepada sub-obyek tertentu. Setelah itu, penulis menggunakan *Multi/ Sub-Object Material* agar dapat memasukkan seluruh tekstur ke dalam model 3D yang telah disatukan.

Untuk *Player Character*, penulis menggunakan 3 *Material ID* karena model memiliki 3 tekstur yang terpisah.



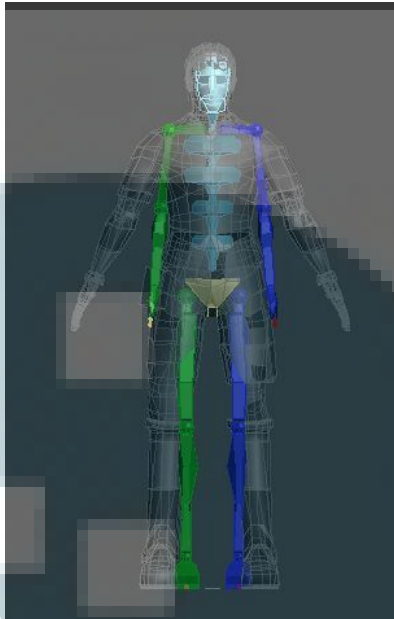
Gambar 3.50. Penggunaan *Material ID* dan *Multi/Sub-Object*

3.3.7. Rigging

Rigging adalah proses memposisikan tulang pada model untuk digunakan pada proses animasi. Dalam proses rigging, penulis menggunakan biped untuk *Player Character* dan Bones untuk *Non Player Character*.

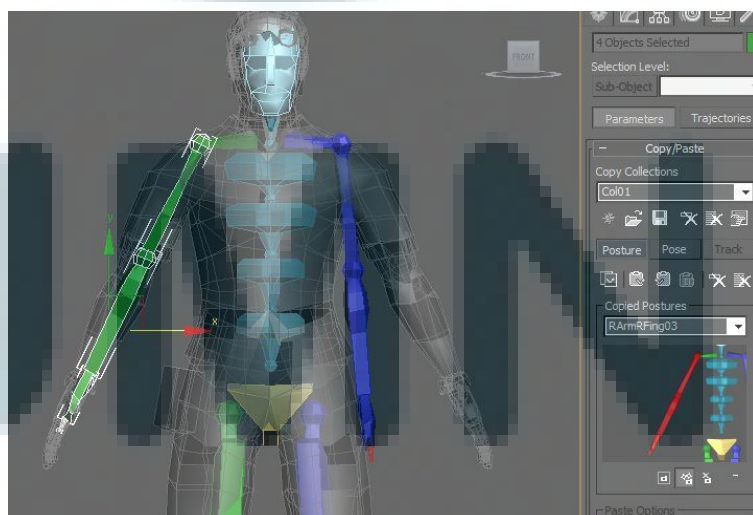
3.3.7.1. Biped

Pada *Player Character*, penulis menggunakan *biped* karena *biped* mempunyai struktur rangka menyerupai manusia. Tinggi *biped* disesuaikan dengan tinggi model, yaitu sekitar 180 cm. Selanjutnya *biped* diposisikan di dalam tubuh model. Agar dapat menyesuaikan dengan proporsi tubuh model, *biped* kemudian dilakukan edit dengan menggunakan *figure mode*. Mode ini memungkinkan penulis untuk merubah struktur *biped* seperti jumlah ruas jari tangan, ruas tulang punggung, dan lain sebagainya. Penulis juga melakukan skala pada beberapa bagian *biped* seperti lengan, betis, dan lain sebagainya.

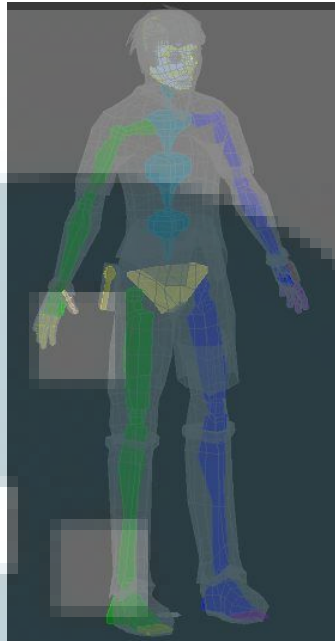


Gambar 3.51. *Biped* yang Telah Diposisikan dalam *Mesh*

Salah satu fitur yang dimiliki *biped* adalah *Copy /Paste* dimana penulis hanya melakukan edit pada setengah bagian tubuh, lalu penulis memilih bagian yang akan dilakukan *copy*. Dengan *paste opposite* maka bagian tubuh yang berlawanan akan memiliki dimensi dan pose yang sama.



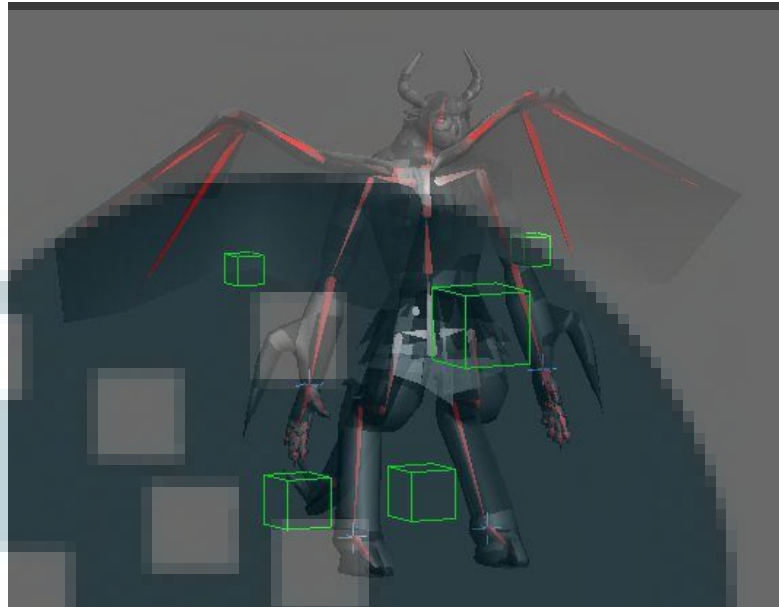
Gambar 3.52. Area yang Dipilih Ditandai dengan Warna Merah



Gambar 3.53. *Biped* yang Telah Mengikuti Proporsi Model

3.3.7.2. *Bones*

Berbeda dengan *biped* yang memiliki struktur manusia, *bones* memungkinkan pengguna untuk membuat struktur sendiri. Penulis menggunakan *bones* untuk kedua *Non Player Character* karena lebih mudah untuk membuat struktur hewan fantasi sesuai keinginan. Dalam *bones*, penulis menggunakan *mirror* untuk menduplikasi setengah bagian kerangka model. Penulis menggunakan modifier *skin* untuk menghubungkan rangka dan model 3D.

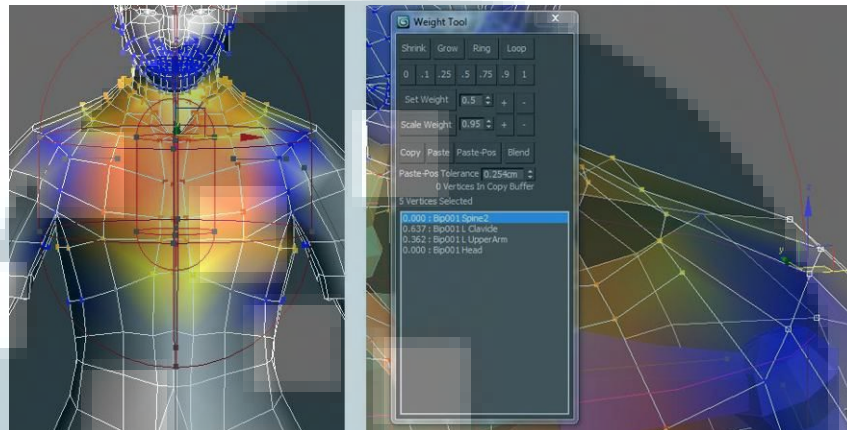


Gambar 3.54. *Bones* yang Telah Diposisikan dalam Tubuh Model

3.3.8. *Skinning*

Skinning adalah proses menghubungkan antara rangka (*biped* atau *bones*) dengan *mesh* agar dapat digunakan dalam proses penganimasian. Masing-masing bagian *biped* mewakili bagian tubuh dimana ia berada. Apabila kita memilih suatu bagian *biped* akan muncul *envelope* yang berbentuk seperti kapsul, yaitu ruang lingkup sampai sejauh mana suatu *bone* akan mengendalikan *vertex* sekitarnya. Seringkali *vertex* yang tidak berhubungan juga akan ikut tertarik. Untuk memperbaiki hal tersebut, penulis mengecilkan area *envelope* dan menggunakan *weight tool*, yang menentukan seberapa kuat suatu *vertex* akan terpengaruhi oleh *bone* tertentu dengan *range* antara 0-1. Warna biru menandakan *vertex* tersebut ditarik dengan lemah sampai ke warna merah berarti ditarik paling kuat. Selain itu kita juga dapat melihat oleh *bone* mana sajakah sebuah *vertex* terpengaruh. Sesuai saran

pembimbing, untuk menghindari beban lebih pada *game engine* maka satu *vertex* hanya boleh dipengaruhi oleh maksimal dua *bone*.



Gambar 3.55. *Envelope* dan *Weight Tool*



Gambar 3.56. Karakter Setelah Proses *Skinning*

3.3.9. Animasi Karakter

Dalam tahap ini, seluruh karakter yang telah diberikan *skinning* telah siap untuk dianimasikan. Pada *Player Character*, penulis mengunduh sekumpulan animasi

hasil *motion capture* yang bersifat *free royalty* dari internet dengan format *.bip*. nantinya gerakan animasi ini akan mengalami penyesuaian dengan proporsi karakter, misalnya bahu yang terlalu turun harus dinaikkan agar terlihat lebih natural. Sedangkan untuk karakter *Underling* dan *Boss*, tim membuat sendiri animasinya karena sangat sulit untuk mencari animasi untuk hewan fantasi sesuai keinginan penulis.

3.3.9.1. Animasi Player Character

Terdapat tujuh jenis gerakan untuk *Player Character*, yaitu:

1.) Idle

Idle adalah gerakan diam apabila pemain tidak menekan tombol apapun di dalam game. Tubuh *Player Character* terlihat berdiri dengan santai sementara bagian dada bergerak naik turun seperti manusia bernapas. Kepalanya bergerak dari arah depan ke kiri seperti sedang melihat keadaan sekitar.

2.) Run Cycle

Berlari akan menjadi gerakan standar saat pemain menekan tombol W, A, S, dan D dalam *game engine*. Gerakan ini bersifat *looping* sehingga pose dalam frame pertama dan terakhir harus sama. Saat penulis memasukkan animasi ke dalam *biped*, PC akan berlari di tempat. Dalam *game engine*, seberapa jauh ia berguling akan diatur oleh *script*.

3.) Walk Cycle

Gerakan berjalan akan dipakai dalam kondisi tertentu saja apabila pemain ingin bergerak dengan kecepatan yang sangat lambat.

4.) *Jump*

Lompat adalah gerakan yang dilakukan apabila pemain ingin melewati rintangan ataupun pindah ke tempat yang lebih tinggi. Sebelum melompat, *Player Character* akan mengambil ancang-ancang berupa tolakan dengan kaki kanan, kemudian melompat dan mendarat dengan kedua kaki dengan pose agak berjongkok, setelah itu berdiri tegak kembali.

5.) *Attack*

Gerakan menyerang berupa *slashing* akan dilakukan pemain untuk melindungi dirinya dari serangan musuh. *PC* akan menyabetkan pedangnya kedepan.

6.) *Die*

Pada saat seluruh *Health Point* pemain mencapai 0, maka *PC* secara otomatis akan mati. Gerakan tubuh *PC* seolah terdorong kebelakang, kemudian jatuh dengan bertumpu pada kedua lututnya, kemudian roboh ke tanah.

7.) *Rolling*

Berguling adalah gerakan untuk menghindari serangan musuh karena karakter tidak memiliki perisai. Penulis sengaja melakukan edit sehingga saat berguling *PC* tidak maju ke depan karena dalam *game engine*, seberapa jauh ia berguling akan diatur oleh *script*.

3.3.9.2. Animasi *Snap Turtle* (*Underling*)

Terdapat empat jenis gerakan untuk *Snap Turtle*, yaitu:

1.) *Idle*

Idle adalah posisi *Snap Turtle* saat tidak terganggu oleh pemain. Ia akan menggerak-gerakkan kepala, ekor, dan sulurnya secara perlahan.

2.) *Attack 1 (Short Range)*

Pada saat pemain berada dekat dengan *Snap Turtle*, maka *Turtle* akan menggigit pemain.

3.) *Attack 2 (Long Range)*

Ketika jarak pemain cukup jauh, maka *Turtle* akan menembakkan gelembung air ke pemain dengan menggoyangkan kepalanya.

4.) *Walk Cycle*

Walk cycle digunakan saat *Snap Turtle* berjalan-jalan dalam habitatnya.

3.3.9.3. Animasi *White God* (*Boss*)

Terdapat tiga jenis animasi untuk *Boss*, yaitu:

1.) *Idle*

Idle dilakukan *Boss* saat pemain belum memasuki radius serangan.

2.) *Walk Cycle*

Walk Cycle digunakan *Boss* untuk mengejar pemain dan menyerangnya.

Boss akan terus mengejar pemain untuk melukai pemain.

3.) *Attack*

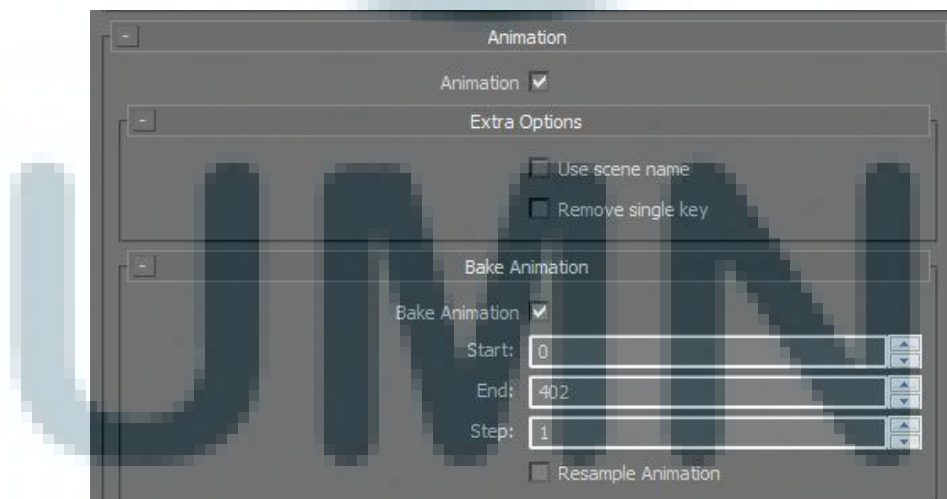
Untuk menyerang pemain, Boss akan mengangkat kedua tangannya tinggi-tinggi kemudian menghantamnya ke tanah.

3.4. Pascaproduksi

Pascaproduksi adalah tahap terakhir dimana model 3D yang sudah dibuat selanjutnya diaplikasikan ke dalam *game engine*, yaitu *Unity*. *Game* ini menggunakan *3rd Person View* dimana pemain dapat melihat karakternya dalam *game*.

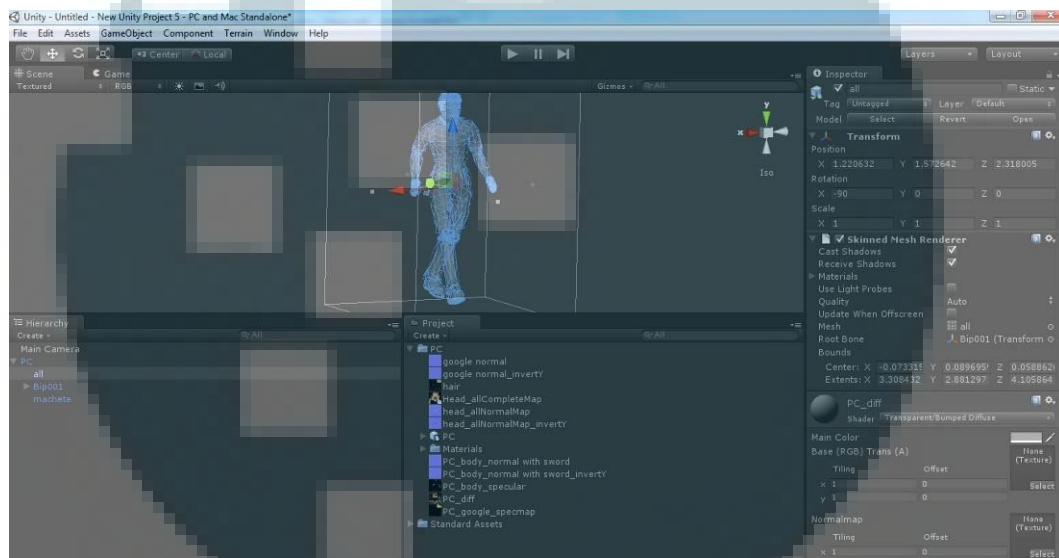
3.4.1. Ekspor dan Impor

Model 3D yang telah ditekstur dan dianimasikan kemudian diekspor ke dalam format *FBX* sebelum masuk ke dalam program *Unity*. Hal yang perlu diperhatikan adalah *Bake Animation* yang harus dicentang agar animasi karakter dapat berjalan dalam *Unity*. *FBX* hanya akan membawa *mesh* dari model sehingga material harus dipasang secara manual dalam program *Unity*.



Gambar 3.57. Pilihan *Bake Animation* dalam Dialog Ekspor

Seluruh file model (*FBX*) dan material (*TARGA*) kemudian dikopi pada tab *Project* dalam *Unity*. Agar lebih sistematis, penulis membuat folder untuk masing-masing karakter. File *mesh* karakter kemudian ditarik kedalam jendela *Scene* untuk selanjutnya dilakukan *edit* berupa pemasangan material, pengaturan animasi, dan lain sebagainya.



Gambar 3.58. Tampilan dalam *Game Engine Unity*

3.4.2. Mengaplikasikan Material

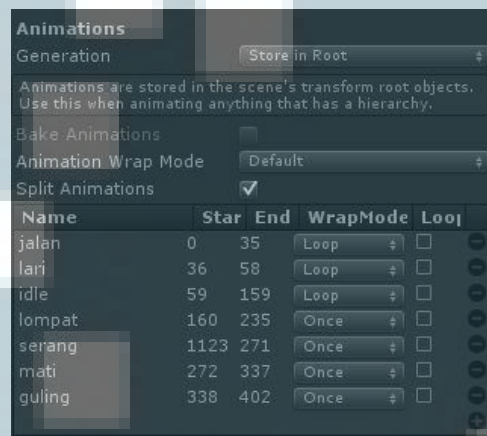
Dalam program *Unity*, setiap model telah diberikan slot kosong sebagai tempat pemasangan material. Slot ini secara otomatis akan mengikuti jumlah *Material ID* yang telah disusun dalam program *3ds Max*.

Tiap slot material dapat diberikan shader yang berbeda tergantung kebutuhan. Pada bagian wajah *Player Character*, penulis menggunakan shader *Bumped Diffuse* untuk memuat tekstur *diffuse* dan *normal map*nya. Pada bagian tubuh dan rambut, penulis menggunakan shader *Transparent /Cutout /Bumped Diffuse* karena tidak semua area tekstur akan ditampilkan. *Opacity Map* yang

penulis gunakan dalam *3ds Max* tidak dimasukkan ke dalam *Unity* karena *game engine* akan membaca *alpha channel* yang terdapat dalam format *TARGA*.

3.4.3. *Split Animation*

Di dalam program *Unity*, animasi yang dimiliki karakter akan dipisahkan sesuai frame nya menggunakan *split animation*. Sebelumnya, telah penulis mencatat pembagian frame dan sifat perulangannya terlebih dahulu lalu melakukan input manual dalam tabel animasi.



Gambar 3.59. Jendela *Split Animation*

1.) *Player Character*

Tabel 3.1. Animasi pada *Player Character*

Animasi	Frame	WrapMode
Walk	0-35	Loop
Run	36-58	Loop
Idle	59-159	Loop
Jump	160-235	Once
Attack	236-271	Once
Die	272-337	Once
Roll	338-402	Once

2.) *Snap Turtle*

Tabel 3.2. Animasi pada *Snap Turtle*

Animasi	<i>Frame</i>	<i>WrapMode</i>
<i>Idle</i>	1-100	<i>Loop</i>
<i>Squirt</i>	109-161	<i>Once</i>
<i>Bite</i>	166-186	<i>Once</i>
<i>Walk</i>	191-251	<i>Loop</i>

3.) *White God*

Tabel 3.3. Animasi pada *White God*

Animasi	<i>Frame</i>	<i>WrapMode</i>
<i>Attack</i>	1-50	<i>Once</i>
<i>Walk</i>	85-125	<i>Loop</i>
<i>Idle</i>	1-100	<i>Loop</i>