

FIRST-PERSON SHOOTER 3D “GAMASHOOT” DENGAN BLENDER DAN UNITY 3D

Muhammad Haki Fauzi¹⁾, Rodiah²⁾

^{1,2)} Teknik Informatika Universitas Gunadarma

Jl. Margonda Raya No.100, Pondok Cina Depok

email : haki.fauzi@yahoo.com¹⁾, rodiah@staff.gunadarma.ac.id²⁾

Abstrak

Semakin tingginya minat masyarakat akan permainan (*game*) menuntut para *designer* dan *developer* industri kreatif untuk mengembangkan ide dan kemampuannya. *Game* diminati mulai dari anak-anak sampai dewasa baik pada papan permainan (*boardgame*), konsol, arcade, handheld, maupun komputer. *Game* pada saat ini yang lebih menarik dimainkan adalah *game* yang sudah berbasis 3D karena objek yang terdapat pada *game* tersebut terlihat lebih nyata (*real*). Dari sekian banyak *game* 3D yang ada, hanya sedikit *game* 3D yang dibuat oleh *designer* dan *developer* dari Indonesia karena pemrograman *game* dapat dikatakan cukup sulit. Pada penelitian ini dikembangkan modul kursus baru untuk merancang dan membuat *First-Person Shooter 3D Game Design and Development* yang berciri khas Indonesia, seperti: bentuk fisik karakter, penggunaan bahasa, dan lingkungan *game* (*mapping*) dengan Blender untuk modeling objek 3D dan 3D Unity sebagai *Game Engine*.

Kata kunci :

Blender, Gama Shoot, Game, Unity 3D

1. Pendahuluan

Industri kreatif di Indonesia sudah mulai berkembang sejak beberapa tahun yang lalu. Perkembangan industri kreatif diikuti dengan tingginya minat masyarakat. Tingginya minat tersebut seakan menuntut para *designer* dan *developer* industri kreatif untuk mengembangkan ide dan kemampuannya. Salah satu industri kreatif yang berkembang sangat pesat dan diminati banyak orang adalah *game*. *Game* diminati mulai dari anak-anak sampai dewasa baik pada papan permainan (*boardgame*), konsol, arcade, handheld, maupun komputer. Tipe *game* (*genre*) sudah banyak bermunculan, mulai dari *Action Shooter*, *Action Adventure*, *Adventure*, *Role-Playing*, *Simulation*, sampai pada *game Strategy*.

Setiap tipe *game* sudah memiliki dua sifat visualisasi yaitu 2D dan 3D. *Game* pada saat ini yang lebih menarik dimainkan adalah *game* berbasis 3D karena objek yang terdapat pada *game* tersebut terlihat lebih nyata. Dari sekian banyak *game* 3D yang ada, hanya sedikit *game* 3D yang dibuat oleh *designer* dan

developer dari Indonesia karena pemrograman *game* dapat dikatakan cukup sulit. Penelitian ini bertujuan untuk merancang dan membuat *First-Person Shooter 3D Game Design and Development*. Pembuatan *game* ini khususnya pada pemrograman yang diberikan pada karakter orang pertama/pemain diperlukan beberapa hal, dimulai dari model karakter orang pertama, mendaftar perilaku yang dilakukan oleh orang pertama/pemain beserta *script*-nya (misalnya : berlari, menembak, terkena serangan).

Metode pengembangan yang digunakan dalam penyusunan penelitian ini adalah pengumpulan model karakter beserta suara, penambahan *script* pada model karakter orang pertama/pemain untuk mengontrol karakter agar dapat berjalan sesuai yang diharapkan, dan pengisian suara. Langkah terakhir dalam pembuatan FPS 3D *game* ini adalah uji coba dengan cara menjalankan dan memainkan *game* yang telah dibuat serta mencari/memperhatikan bug yang ada dalam *game*.

Penelitian ini dikontribusikan pada bidang *game technology* khususnya untuk membuat karakter pemain yang berciri khas Indonesia, seperti: bentuk fisik karakter, penggunaan bahasa, dan lingkungan *game* (*mapping*) dengan Blender untuk membuat model objek 3D dan 3D Unity sebagai *Game Engine*.

2. Tinjauan Pustaka

Game merupakan sesuatu yang dimainkan dengan aturan tertentu. Klasifikasi *game* berdasarkan alat/tampilan *game* terbagi menjadi 2 bagian diantaranya *Board game* dan *Video game*. *Video game* adalah permainan yang divisualisasikan melalui suatu media elektronik yang memerlukan interaksi dari pemainnya menggunakan alat tertentu, sedangkan *Board game* adalah permainan yang memerlukan papan permainan tertentu sebagai area permainannya [2].

Platform mengacu pada kombinasi yang spesifik dari perangkat keras komputer dengan perangkat lunak, yang nantinya akan menjalankan suatu *game*. Salah satu contoh platform yang ada saat ini adalah PC *game*. PC *game* berarti suatu format media yang melibatkan pemainnya berinteraksi dengan sebuah komputer yang dihubungkan ke video monitor dengan resolusi tinggi. *Game* komputer memiliki beberapa jenis kategori yang disebut *genre*. Banyak faktor yang dapat mengkategorikan suatu *game* menjadi *genre* tertentu

seperti metode permainan, sudut pandang permainan, perangkat pengontrol yang digunakan dan tampilan grafis.

Genre komputer sudah berkembang sangat pesat sampai saat ini. Peningkatan *hardware* mempengaruhi munculnya jenis-jenis *game* yang baru, Salah satu diantaranya adalah *shooting game* (tembak-tembakan). Ada dua jenis *shooting game* jika dilihat dari sudut pandang permainan yaitu *first person view* dan *third person view*. *First person view* adalah sudut pandang orang pertama dimana pemain dijadikan seperti karakter yang dimainkan sehingga pemain melihat dari sudut pandang karakter yang dimainkan (contoh : Counter Strike, Point Blank, HALO), sedangkan *third person view* adalah sudut pandang orang ketiga dimana pemain dapat melihat keseluruhan tubuh karakter dalam *game* (contoh : Ragnarok Online, FATE) [2].

Pada dasarnya proses pengembangan *game* terbagi menjadi tiga fase, antara lain : pre-produksi, fase produksi, dan fase uji coba. Fase Pre-Produksi sering disebut fase desain dimana para *game designer* akan mencoba mengembangkan ide-ide yang ada kedalam bentuk yang lebih detail. Penelitian pasar juga dilakukan untuk mendapat gambaran tentang minat dan *trend* industri *game* dimasa mendatang. Hal ini perlu dilakukan mengingat proses pembuatan dapat membutuhkan waktu berbulan-bulan. Besar sekali kemungkinan ide yang dimiliki sekarang ini menjadi kuno pada saat *game* tersebut dirilis. Penelitian mengenai perkembangan teknologi *game* menjadi sangat penting mengingat bahwa ada kemungkinan dalam kurun waktu yang relatif singkat, terobosan-terobosan teknologi baru sering bermunculan. Setelah semua informasi tersebut terkumpul maka akan dibuat sebuah rancangan jangka panjang, kapan sebaiknya *game* tersebut dirilis. Setelah itu semua estimasi, dokumen teknis, serta referensi disiapkan. Setelah semua rencana disusun, maka dibentuk tim sesuai dengan yang dibutuhkan. Semua perangkat baik berupa perangkat keras dan perangkat lunak disiapkan. Kerjasama dengan studio lain juga dibutuhkan, terutama untuk menunjang performa tim yang diinginkan[4].

Pada fase produksi *engine* serta *asset game* mulai dibuat. Proses produksi biasanya dibagi menjadi beberapa tahap. Tahap pertama adalah pembuatan bentuk dasar dari *game* yang dibuat, biasanya hanya menampilkan latar dan sebuah karakter dengan kontrol yang sangat sederhana. Tujuannya untuk melihat apakah grafik yang ditampilkan sudah sesuai dengan yang diharapkan. Setelah itu dilanjutkan dengan pembuatan *first playable*. Pada tahap ini *game* sudah mulai dapat dimainkan. Beberapa animasi telah selesai dikerjakan dan fungsi-fungsi utama mulai diperkenalkan. Tahap selanjutnya adalah versi delta, dimana semua fungsi *game* telah siap dan berjalan sesuai desain. Sebagian *asset* telah selesai dikerjakan. Tahap yang terakhir adalah tahap dimana seluruh isi *game* telah selesai dikerjakan dan siap memasuki fase uji coba[4].

Pada fase uji coba (*testing*) para penguji *game* dikerahkan untuk mencari kesalahan yang ada dalam *game*. Kesalahan sendiri dibedakan menjadi 4 kelas. Kelas A adalah kesalahan yang menyebabkan sebuah *game* tidak bisa dimainkan atau diselesaikan. Sebagai contoh *game* menjadi berhenti saat mengeksekusi. Kelas B adalah kesalahan yang menyangkut masalah fungsi yang tidak dapat berjalan dengan benar misalnya karakter tidak bisa melompat dengan benar. Kelas C bersifat *audio visual*, dimana menyangkut masalah kualitas *game* (Contoh : adanya animasi, tampilan grafis, maupun suara yang buruk kualitasnya). Kesalahan kelas D bersifat usulan-usulan dari para *tester* agar bagian-bagian dari *game* tersebut lebih baik secara keseluruhan [4]. Tim pengembangan dalam pembuatan *game* biasanya terdiri dari tiga orang atau lebih yang mencakup *Project Manager*, *Animator*, dan *Programmer*. Didalam pengembangan sebuah *game* yang cukup kompleks, orang-orang yang dilibatkan tentu lebih banyak [5].

3. Metode Penelitian

3.1. Algoritma Pembuatan “GamaShoot”

Gambar 1 merupakan bagan alur kerja secara umum pada proses pembuatan “GamaShoot”



Gambar 1. Bagan alur pembuatan “GamaShoot”

Berikut adalah algoritma pada penelitian ini dalam proses pembuatan “GamaShoot” :

1. Mengembangkan prolog dan epilog menjadi sebuah skenario. Berikut adalah skenario dalam permainan “GamaShoot”:

- Terdapat dua tokoh yang akan dilibatkan pada *game*, yaitu tokoh protagonis dan antagonis. Gambar 2 merupakan karakter protagonis “Gama” dalam permainan “GamaShoot”.



Gambar 2. Karakter protagonis (Gama)

tersebut sudah berjalan sesuai dengan yang diharapkan dan dinyatakan selesai.

3.2. Object pada pembuatan “GamaShoot”

Beberapa *object* yang digunakan dalam perncangan game ini antara lain :

▪ Senjata

Senjata adalah inti dari permainan yang berjenis *shooter* untuk menembak lawan. Untuk menambahkan senjata pada *First Person Controller* diperlukan model senjata yang telah dibuat sebelumnya menggunakan *software 3D modeling*. Tipe file model senjata yang digunakan penulis adalah *.blend. Gambar 6 adalah model senjata beserta lengan karakter agar pada saat dilihat dari *Main Camera* terlihat bahwa senjata di genggam oleh pemain.



Gambar 6. Senjata

Senjata merupakan *asset* yang di-*import* ke *unity project* yang nantinya akan ditambahkan pada *hierarchy*. Untuk meng-*import* senjata terdapat pada menu *Assets > Import New Asset*, kemudian pilih file *.blend yang dimaksud. Senjata merupakan bagian dari objek karakter pemain yang selalu dibawa oleh karakter pemain, sehingga senjata harus selalu mengikuti *First Person Controller*. Pada *hierarchy*, senjata ditempatkan dalam *object Main Camera* yang berada pada *First Person Controller* seolah menjadi satu kesatuan sehingga senjata akan selalu mengikuti kemanapun controller bergerak. Pada saat pemain membidik sasaran bisa saja lawan yang dibidik berada pada jarak yang cukup jauh sehingga pemain kesulitan untuk menembak karena sasaran terlihat sangat kecil. Untuk itu, diperlukan *scope* untuk memperbesar ruang tembak pada sasaran. *Scope* dapat dibuat pada senjata sehingga seolah senjata memiliki kekeran sungguhan.

Secara logika terdapat dua pandangan berbeda yaitu pandangan normal dan pandangan membidik sehingga diperlukan sebuah kamera baru sebagai pandangan membidik. Kamera ditempatkan sangat dekat tepat dibelakang bidikan senjata dan menghadap ke depan seperti dapat dilihat pada gambar 7.



Gambar 7. Pandangan pada kekeran senjata

▪ Kamera

Jenis kamera yang digunakan sama dengan jenis *Main Camera*, namun jarak pandang (*zoom*) diperbesar sehingga mendapatkan kesan membidik. sama halnya

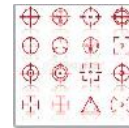
seperti senjata. Pada *hierarchy* kamera *scope* ditempatkan dalam *Main Camera* pada *First Person Controller* agar kamera kekeran tetap mengikuti kemanapun *controller* bergerak seperti dapat dilihat pada gambar 8.



Gambar 8. *Hierarchy* kamera *scope*

▪ Crosshair

File yang dibutuhkan untuk membuat *crosshair* pada project game FPS adalah desain *crosshair* yang bertipe *.png sebagai *asset*. *Asset crosshair* diinput ke dalam *project unity*. Untuk menambahkan *crosshair* pada *hierarchy* hanya memerlukan sebuah *script* dengan nama file *crosshair.js* yang ditempatkan pada *First Person Controller*, sedangkan *asset* desain *crosshair* di simpan pada *variable TextureToDisplay*. *Crosshair* biasanya berbentuk dua garis menyilang atau tanda tambah "+" yang digunakan sebagai kursor atau titik bidikan senjata agar mempermudah mendapatkan arah tembakan senjata tepat pada sasaran. Gambar 9 menunjukkan beberapa contoh bentuk *crosshair*.



Gambar 9. Contoh-contoh *crosshair*

(sumber : www.istockphoto.com, Tanggal Akses : 7 Juni 2012)

▪ Muzzle Flash

Muzzle Flash adalah efek yang sangat perlu digunakan pada sebuah game tembak menembak. Efek ini berupa percikan api yang muncul pada mulut senjata apabila peluru ditembakkan. Dengan adanya *muzzle flash* akan membuat efek dari game terlihat sangat nyata seperti dapat dilihat pada gambar 10.



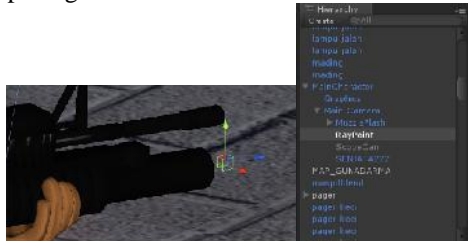
Gambar 10. *Muzzle Flash*

(sumber : <http://www.slowbox.org>, Tanggal akses : 7 Juni 2012)

Asset yang didapatkan telah berupa *prefab*, sehingga cukup mudah untuk membuat efek *muzzle flash*. *Prefab muzzle flash* ditempatkan tepat didepan mulut senjata pada *scene*, sedangkan pada *hierarchy* akan meneempatkan *muzzle flash* didalam *Main Camera* pada *First Person Controller*. Berbeda dengan *spark*, *prefab* ini digunakan tanpa aturan penempatan objek baik pada *scene* ataupun pada *hierarchy*, karena percikan api yang

ditimbulkan adalah berdasarkan pantulan peluru dengan objek sasaran. Untuk menambah kesan nyata, *spotlight* dapat digunakan sebagai cahaya yang dihasilkan oleh *muzzle flash*.

Penulis menggunakan tiga buah *spotlight* disekitar senjata. Sebuah peluru yang ditembakkan memiliki kecepatan yang sangat cepat sehingga tidak dapat dilihat oleh mata. Untuk itu, dalam pembuatan peluru dapat memanfaatkan *empty object* seperti dapat dilihat pada gambar 11.



Gambar 11. *New gameObject* sebagai peluru

4. Hasil dan Pembahasan

4.1. Analisa Fungsi Karakter

Setiap karakter memiliki beberapa animasi yang dilakukan pada saat *game* dijalankan. Misalnya animasi pada karakter orang pertama atau pemain yaitu berjalan, melompat, menembak, dan sebagainya. Berikut adalah analisa fungsi karakter orang pertama:

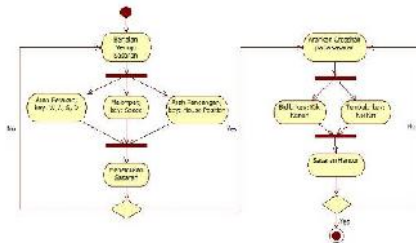
- 1. Arah Pergerakan.** Karakter pemain dapat berlari ke segala arah (maju, mundur, kanan, kiri) untuk mengejar musuh yang akan ditembaknya. Untuk menggerakkan karakter, pemain dapat menekan tombol yang ada pada keyboard. Tombol-tombol yang digunakan adalah “W” untuk maju, “S” untuk mundur, “D” untuk bergerak ke samping kanan, “A” untuk bergerak ke samping kiri dan “SPACE” untuk melompat.
- 2. Arah Pandangan.** Arah pandangan lebih cenderung kepada pemain dapat berbelok ke kanan dan ke kiri. kemudian pemain dapat melihat ke segala arah dengan menggeserkan mouse kedepan untuk melihat keatas, mouse ke belakang untuk melihat kebawah, mouse ke kiri untuk melihat kekiri, dan mouse ke kanan untuk melihat ke kanan.
- 3. Menembak.** Pemain mengarahkan *crosshair* ke badan robot menggunakan mouse. Apabila posisi *crosshair* sudah terarah makan pemain dapat melakukan penembakan agar mengenai sasaran.
- 4. Mengisi Peluru Senjata.** Senjata yang digunakan pemain, memiliki nilai maksimum peluru yang tersedia. jika peluru pada senjata telah habis ditembakkan maka dapat diisi lagi selama persediaan peluru masih ada, hali ini disebut *reload*. Tekan tombol "R" untuk melakukan *reload* senjata. Jumlah peluru maksimal pada senjata adalah 30 butir. Apabila pemain melakukan reload sebelum peluru pada senjata habis maka sisa peluru pada senjata akan

tetap diakumulasikan. Hal ini berpengaruh pada jumlah peluru persediaan sebelumnya yaitu 90 butir. Apabila peluru pada senjata telah habis maka senjata akan di reload secara otomatis.

- 5. Mengambil Peluru.** Keterbatasan peluru yang disediakan di awal permainan memungkinkan pemain membutuhkan peluru tambahan lagi. Untuk itu, pemain dapat mengambil peluru yang ada disekitar robot yang telah dihancurkan. Cara untuk mendapatkan peluru adalah hanya dengan mendekati/melangkahkan badan karakter pemain pada peluru yang ada. Peluru yang tersedia disekitar robot yang telah hancur berjumlah 30 butir.

4.2. Hasil

Penulis melanjutkan *project* yang telah menjadi sebuah peta (*map*) lokasi permainan. *Map* yang digunakan adalah wilayah Kampus Depok Universitas Gunadarma. Gambar 12 memperlihatkan activity diagram untuk melakukan uji coba *project game FPS*.



Gambar 12. *Activity* diagram uji coba

Pada saat pertama kali *game* dijalankan, *user* dapat melakukan pencarian robot yang akan dihancurkan dengan berjalan menggunakan kontrol arah gerakan, melompat, dan arah pandangan, jika *user* telah menemukan robot maka arahkan *crosshair* pada robot. *User* dapat menggunakan bidikan untuk memperbesar ruang tembak apabila sasaran berjarak sangat jauh dari *user*. Setelah *crosshair* tepat mengenai robot, *user* mulai menembakkan peluru sampai robot hancur. Gambar 13 menunjukkan bahwa karakter pemain sedang berjalan dikawasan loket baak Universitas Gunadarma kampus Depok.



Gambar 13. Berjalan

Gambar 14a menunjukkan bahwa pemain sedang menghadapkan pandangannya ke atas dengan menggeser mouse ke depan. Sebaliknya gambar 14b menunjukkan pemain sedang menghadapkan pandangannya ke bawah dengan menggeser *mouse* ke belakang.



(a) (b)

Gambar 14.(a). Arah Pandangan Pemain Posisi Atas, (b)
Arah Pandangan Pemain Posisi Bawah

Dalam menembak sasaran, pemain harus mengarahkan *crosshair* pada sasaran dengan tepat. Pemain mengalami kesulitan karena sasaran berjarak terlalu jauh dari posisi karakter pemain. Pemain dapat juga menggunakan bidikan yang terdapat pada senjata dengan menekan tombol mouse kanan seperti yang tampak pada gambar 15. Robot memiliki daya tahan sebanyak 9 kali tembakan sehingga robot dapat bertahan dari tembakan peluru sebanyak 9 kali, dan akan hancur pada tembakan ke-10.



Gambar 15. Membidik Robot

Gambar 16 menunjukkan ledakan dari robot yang berhasil dihancurkan oleh pemain. Pada tembakan ke-10 robot hancur yang divisualisasikan dengan munculnya gelombang api seperti bom bersamaan dengan mengeluarkan kembang api ke segala arah yang diikuti dengan asap tebal berwarna hitam.



Gambar 16. Ledakan robot

5. Kesimpulan dan Saran

First-Person Shooter 3D game yang dibuat dengan Unity 3D dan Blender telah berhasil menghasilkan suatu permainan dengan karakter berupa arah pergerakan, arah pandangan, menembak, mengisi peluru, dan membidik. Semua fungsi tersebut disatukan pada sebuah karakter pemain dengan sebuah game engine yaitu Unity 3D. Pembuatan karakter pemain menghasilkan suatu permainan yang dapat melakukan aktifitas sesuai yang diharapkan. Untuk pengembangan *game* ini dapat ditambahkan animasi objek dalam melakukan suatu fungsi, audio, dan *graphical user interface* (GUI).

Suatu animasi perlu sekali disertakan agar pemain merasa terkesan saat memainkannya, seperti animasi mengisi amunisi (*reload*). Pengembangan juga dapat dilakukan dari sisi audio seperti suara berjalan, suara ledakan, dan lain-lain. Penambahan dari fasilitas senjata-senjata pelengkap lainnya seperti bom, pisau, dan tameng juga dapat menjadi pengembangan selanjutnya pada *game* ini.

Daftar Pustaka

- [1] Andry A, 2003, *Pengantar JavaScript*, Ilmu Komputer.
- [2] Dawang M, 2005. *Desain Game*, <http://gameworldamazing.wordpress.com>, Tanggal akses : 7 Juni 2012
- [3] Dharwiyanti Sri, 2003, *Pengantar Unified Modeling Language (UML)*, www.ilmukomputer.com, Tanggal akses : 6 Juni 2012
- [4] Lindsay G, 2005, *Game Type and Game Genre*, http://aii.lgrace.com/documents/Game_types_and_genres.pdf, tanggal akses : 6 Juni 2012
- [5] Soekahar F.J, 2004, *Open Source 3D Animation: Blender Publisher*, Jakarta.

Biodata Penulis

Muhammad Haki Fauzi, saat ini sebagai mahasiswa semester 7, Jurusan Teknik Informatika Universitas Gunadarma. Pernah bekerja di PT Logika Interaktif, sebuah perusahaan Indonesia yang bergerak di bidang produksi *game*.

Rodiah, memperoleh gelar Sarjana Teknik Informatika (ST), Jurusan Teknik Informatika Universitas Gunadarma, lulus tahun 2003. Tahun 2006 memperoleh gelar Magister Manajemen Sistem Informasi (MMSI) dari Magister Sistem Informasi Universitas Gunadarma. Program Doktor pada Teknologi Informasi Universitas Gunadarma, lulus tahun 2012. Saat ini sebagai Staf Pengajar Teknik Informatika Universitas Gunadarma.