SIMULASI *VIRTUAL REALITY* PADA RUMAH SAKIT GRAHA AMERTA SURABAYA

Ahmad Hoirul Basori
Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember Surabaya
Jl.Raya ITS, Kampus ITS, Sukolilo, Surabaya
Telp 031-5939214, Fax 031- 5913804
e-mail: hoirul_basori@yahoo.com

Abstract

In this research, we would like to present virtual reality application that will visualize Graha Amerta Surabaya hospital into Virtual environment. This 3D simulation is for helping people on remembering hospital area. There 3 different categories: road way around hospital, lobby reception and VIP room for patient. Guidance process involve Dijkstra algorithm for looking shortest path to specific location. The result of simulation still not perfect because of the coordinate of building still not precise. In other ways, there also difficulty of customize the building became look similar with the real, it's because of the addition on building will increase the number of polygon, there fore memory needs will also increase.

Keywords: *Virtual reality, 3D simulation, Dijkstra*

Abstrak

Paper ini membahas tentang pembangunan aplikasi virtual reality untuk memvisualisasikan rumah sakit Graha Amerta Surabaya ke dalam virtual environment. Hal ini ditujukan untuk membantu proses pengenalan rumah sakit secara cepat. Ada 3 kategori yang divisualisasikan dalam penelitian ini, yaitu: pengenalan jalan menuju rumah sakit, ruang lobby, dan ruang VIP. Proses simulasi dapat berjalan dengan baik pada graphics card yang memenuhi standard visualisasi 3D. Proses pemanduan dengan menggunakan algoritma Dijkstra masih belum mencapai hasil yang sempurna dikarenakan proses pemberian koordinat masih kurang detail. Selain itu, penambahan informasi detail pada ruangan ruangan VIP dan lobby akan menambah jumlah polygon, sehingga memori yang dibutuhkan juga akan semakin besar.

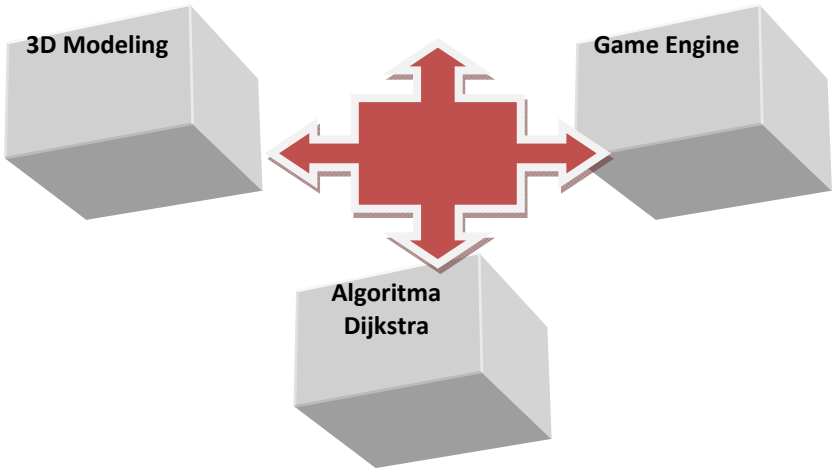
Kata kunci*: Virtual reality,Visualisasi 3D, Dijkstra*

1. PENDAHULUAN

Simulasi 3 dimensi (3D) merupakan topik yang sangat menarik di jaman sekarang ini. Ada beberapa jenis teknologi yang digunakan salah satunya adalah teknologi 3D yang membuat proses simulasi mendekati kenyataannya. Selain itu dengan hadirnya 3D, *user* bisa mendapatkan dunia yang hampir realistis. Semua keterbatasan dunia 2D seperti dicantumkan di atas bisa diatasi. *User* bisa bergerak ke mana saja, dapat melihat ke kiri dan kanan untuk melihat situasi sekitar, serta merubah posisi arah sorotan kamera. Contohnya: Seorang pemain *game racing* jadi mampu melakukan kecurangan seperti: memotong jalan, memutar balik kendaraan, kemudian berjalanan berlawanan arus, Pemain *game* pesawat dapat bergerak ke kiri, kanan, atas, bawah, serta melakukan manuver-manuver seperti yang dilakukan pilot-pilot pesawat tempur sesungguhnya. Tidak hanya itu, untuk menambah tingkat realistis suatu *game*, seorang pemain dapat mengubah posisi kamera ke dalam kokpit, sehingga bisa merasakan pengalaman seperti layaknya seorang pilot.

2. METODOLOGI PENELITIAN

Virtual reality yang dibangun pada penelitian ini melibatkan beberapa aspek dalam pembangunannya, yaitu: *3D Modeling*, *Game Engine*, dan penggunaan algoritma Dijkstra (Gambar 1).



Gambar 1. Diagram kotak rancangan sistem

2.1. Pemodelan 3D (*3D Modeling*)

Dengan munculnya teknologi-teknologi 3D yang berteknologi tinggi seperti: Open GL dan Direct X, pembangunan aplikasi 3D pada *PC Dekstop* dapat disesuaikan dengan kondisi teknologi graphics card yang terbaru, sehingga game yang dihasilkan akan mendekati kondisi nyata. Selain itu, seiring dengan meningkatnya *processing power* serta kemampuan *video card*, jumlah data yang diproses dalam gambar 3D meningkat setiap hari setiap tahunnya, untuk menghasilkan tingkat kedetailan gambar yang tinggi. Namun untuk menghasilkan tingkat kedetailan seperti di atas diperlukan amat banyak kalkulasi matematis untuk menciptakan image per detiknya. Karena waktu dan pemrosesan terbatas, maka semua aspek yang membentuk scene 3D seperti geometri, animasi, lighting/ pencahayaan, dan material harus dioptimasikan [1].

Konsep *"low poly"* dalam 3D modeling berarti membangun objek dengan jumlah polygon yang rendah. Tidak ada ukuran yang pasti seberapa rendah jumlah polygon yang harus digunakan sebab jumlah ini bergantung pada platform yang dipakai (PC, konsol, serta *mobile*) [1].

Dalam kasus kita yakni *mobile device*, disarankan jumlah polygon total dipakai (termasuk karakter, dan objek-objek sekelilingnya) tidak melebihi 1500 polygon [MOT-2004].

Ada beberapa teknik pemodelan yang harus dihindari seperti *NURBS*, pengunaan *surface tools*, atau penggunaan *Mesh Smooth* sebab akan menghasilkan objek dengan jumlah polygon yang banyak [1].

Dalam dunia 3D, semua objek dibentuk dari segitiga. Jumlah polygon yang terpakai berarti jumlah polygon yang sudah dipergunakan untuk membentuk objek tersebut. Dalam 3dsmax, objek dapat berupa *Editable Mesh* (mesh terdiri dari segitiga, sehingga dimanipulasi pada level segitiga), dan Editable Poly (mesh terdiri dari kotak, yang tersusun dari 2 segitiga). Sebagai contoh, objek kubus terdiri dari 6 sisi (kotak). Jika diconvert ke *Editable Mesh* maka kubus tersebut terdiri dari 12 polygon. Sebab satu kotak terbentuk dari 2 segitiga yang digabungkan [1].

Oleh karena itu, pada waktu melakukan pemodelan, pemodel harus selalu memperhatikan jumlah polygon yang terpakai. Untuk melakukan hal ini, pemodel dapat menggunakan *Polygon Counter*, salah satu *tool* yang terdapat pada 3dsmax. Namun sebelum menggunakan *Polygon Counter*, semua objek hendaknya terlebih dahulu dikonversi ke *Editable Mesh*, sehingga *Polygon Counter* memberikan report yang benar mengenai jumlah polygon

yang terpakai. Sebab jika tidak Polygon Counter akan memberikan report yang salah yaitu jumlah polygon dihitung lebih kecil dari sebenarnya disebabkan karena objek berupa *Editable Poly* dan bukan *Editable Mesh* [1].

Dalam dunia nyata, terdapat jumlah polygon yang tak terhitung jumlahnya. Namun dalam aplikasi 3D, semua objek terbentuk dari segitiga [1].

2.2. Game Engine

Suatu aplikasi Virtual reality tidak cukup hanya dimodelkan akan tetapi perlu digerakkan dan dianimasikan.Untuk melakukan animasi pada suatu aplikasi virtual reality diperlukan suatu library khusus yang sekarang lebih dikenal dengan nama *Game engine*. *Game engine* ini sudah banyak beredar di pasaran ada yang open source ada juga yang license.di dalam penelitian ini digunakan *game engine 3d state* yang mempunyai student license. *Game engine 3D state* merupakan *game engine* yang sangat mudah digunakan, selain itu bahasa yang didukung juga banyak.dalam aplikasi ini bahasa yang dipakai adalah Bahasa Pemrograman C# .NET.

Beberapa *game engine* yang sering digunakan di pasaran:

a. **Irrlicht *Game engine***
Merupakan *game engine* yang open source dan dapat digunakan secara bebas.Bahasa pemrograman yang digunakan adalah C++.Pada dasarnya irrlicht mendukung beberapa tipe file 3D seperti : Quake 2, Quake 3,Direct X dan *file* 3ds (Gambar 2).

Gambar 2. Irrlicht *game engine*

b. *Game engine* (3D State)
3D state *game engine* merupakan *game engine* yang license.akan tetapi 3d State juga mengeluarkan versi student license yang daapt didownload dan digunakan untuk keperluan akademik (Gambar 3).

Gambar 3. 3D State *Game engine*

c. XNA *game engine*
Xna merupakan framework engine terbaru dari Microsoft yang digunakan untuk pembuatan game pada X box.Frame work ini gratis dari Microsoft dengan bahasa pemrograman C# sebagai bahasa pembangunnya (Gambar 4).

Gambar 4. XNA Framework untuk *Game engine* pada XBOX

2.3. Algoritma Dijkstra

Algoritma ini sering kali digunakan untuk mencari rute terpendek atau *shortest path*.Algoritma ini memerlukan suatu bentuk graf berarah dengan bobot yang tidak bernilai negative.Input algoritma Dijkstra adalah sebuah graf yang berarah yang berbobot(*Weighted Directed Graph*) G dan sebuah vertex s dalam G dan V merupakan himpunan semua vertex dalam graph G.Setiap sisi dari graf ini adalah pasangan vertices(*u,v*) yang melambangkan hubungan dari vertex u ke vertex v.

Bobot (Wieghts) dari semua sisi dihitung dengan fungsi:

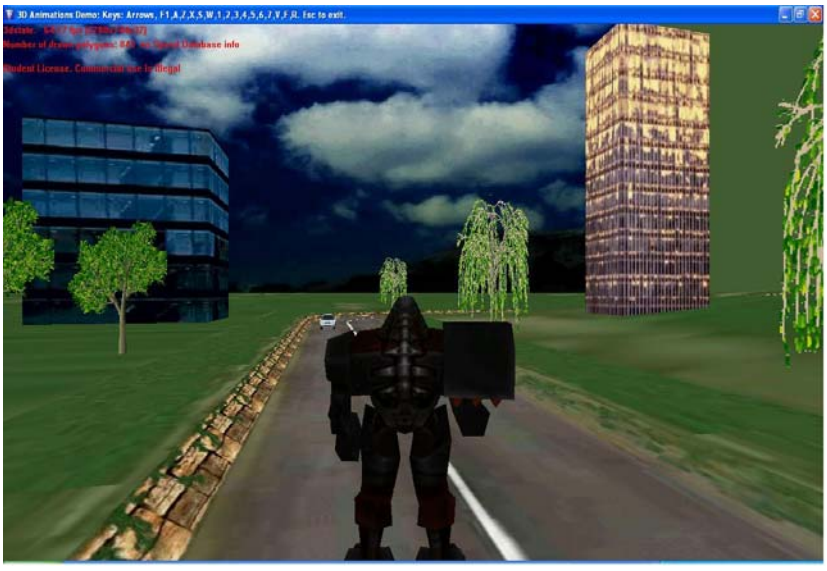
$$w: E \rightarrow [0, \infty)$$

3. HASIL DAN PEMBAHASAN

Seperti yang telah dibahas sebelumnya aplikasi virtual reality pada penelitian ini pada dasarnya dibagi menjadi tiga kategori: pengenalan jalan menuju rumah sakit, ruang bobby dan ruang VIP.

3.1. Pengenalan jalan menuju rumah sakit

Proses simulasi ini ditujukan untuk mempermudah pasien untuk melakukan akses ke rumah sakit dari berbagai jalan.serta jalan jalan yang dapat digunakan sebagai alternative rute tercepat untuk mencapai akses ke Rumah sakit Graha Amerta.



Gambar 5. Proses Visualisasi jalan menuju rumah sakit

Proses ini dapat berjalan lancar, akan tetapi fitur gedung gedung masih kurang lengkap, hal ini dikarenakan penambahan entitas gedung akan membuat penambahan polygon sehingga akan memberatkan aplikasi.selain dipengaruhi oleh banyaknya polygon, aplikasi juga sangat dipengaruhi oleh tekstur yang digunakan.kombinasi tekstur yang beragam juga akan membutuhkan memori yang besar.

Proses simulasi jalan ini menggunakan 2 karakter yaitu robot yang menggambarkan user pengguna simulasi serta mobil yang berlalu lalang sebagai simulasi mobil yang sedang berlalu-lintas.

3.2. Ruang Lobby

Ruang lobby adalah salah satu bagian dari aplikasi ini yang mensimulasikan keadaan dalam ruangan.keadaan lobby didesain untuk mendekati keadaan sebenarnya, akan tetapi masih ada keterbatasan dari segi tekstur dan bentuk aslinya.

Gambar 6. Proses Visualisasi Lobby Rumah Sakit

Lobby merupakan bagian yang sarat dengan polygon, hal ini dikarenakan ruangan lobby mempunyai banyak obyek di dalamnya.ruangan lobby yang ada di simulasi 3D ini masih belum sempurna karena masih banyak objek yang belum dimasukkan, seperti perabotan lobby,telpon,dan tempat duduk pasien.

Selain itu dalam aplikasi ini sudah dilengkapi dengan suara –suara music (audio) untuik memberikan efek yang lebih nyata.akan tetapi masih belum bisa dioptimalkan karena terkendala dengan fasilitas yang diberikan oleh *game engine* 3d State.

3.3. Ruang VIP

Ruang VIP merupakan bagian yang lebih detail dalam simulasi ini.Ruangan VIP 3D hanya terdiri dari satu ruangan yang dilengkapi dengan beberapa perabotan seperti kursi , meja dan tempat tidur.

Gambar 7. Ruang VIP

Ruang VIP dalam virtual environment ini sudah dioptimalkan untuk mendekati keadaan aslinya dengan melakukan penambahan tekstur dan pewarnaan pada setiap komponen-komponennya.dalam proses simulasi ini hanya dibuat 2 kamar VIP sebagai prototype, untuk kedepannya akan disempurnakan agar jumlah ruangan 3D VIP jumlahnya sama dengan aslinya dan perabotan yang ada juga bisa ditambahkan pada ruangan 3D VIP.

Karakter robot yang digunakan juga bisa disesuaikan skalanya terhadap tempat dimana dia berada,jika berada di jalan raya, bisa diperbesar, kemudian jika di dalam lobby , bisa diperkecil dan jika di dalam ruangan seperti VIP , karakter robot bisa diperkecil lagi sehingga sesuai dengan ukuran ruangan tersebut.

4. SIMPULAN

Aplikasi yang dibuat ini sudah dapat membuat virtual environment untuk rumah sakit graha amerta Surabaya.proses simulasi juga sudah dapat berjalan dengan baik.akan tetapi masih ada kekurangan dari segi teknologi audio dan fitur- fitur rumah sakit juga masih banyak yang belum lengkap.

DAFTAR PUSTAKA

[1] ....., "**MOTOCODER: Developing 3D Applications for Mobile Devices**", MOTOROLA, 2004.

[2] Kuester, F., "**A Fantasy Adventure Game as a Learning Environment**", Proceedings of the Innovation and Technology in Computer Science Education (ITiCSE), ACM Press, New York, 114-116, 2005.

[3] Disessa, A. A., "**A Practical Study on the Usage of a Commercial *Game engine* for the development of Educational Games**". *Proceedings of 2nd Games and Digital Entertainment Workshop*, Brazilian Computer Society, 2005.

[4] C. Friedman, J. S. and Disessa, A. A., "**What students should know about technology: The case of scientific visualization**", *Journal of Science Education and Technology 8*, 3, 1999.

[5] Thutchinson, T. C., and Kuester, F., "**Hardware architecture for a visualization classroom: Vizclass**", *Computer Applications in Engineering Education 12*, 4, 232-241, 2004.