

Dodgeball: A Virtual Reality Sport Game Using Kinect And XNA Game Studio

Erick Leonardo¹, Liliana², Kartika Gunadi³

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-mail: leoxerick@gmail.com¹, lilian@petra.ac.id², kgunadi@petra.ac.id³

ABSTRAK

Perkembangan teknologi telah merambah pada seluruh aspek kehidupan manusia, terlebih pada dunia perkembangan *video game*. Banyak pengembang *game* mencari inovasi terhadap sistem permainan yang menarik, namun juga memberi nilai lebih pada permainan tersebut. *Virtual reality game* menjadi salah satu alternatif untuk mengembangkan sisi *gameplay*. Pemain dibuat seolah – olah menyatu dengan dunia *game* sehingga keterlibatan pemain sangat tinggi. Skripsi ini bertujuan untuk membuat *virtual reality game* bertemakan olahraga *Dodgeball* yang memanfaatkan Kinect dengan *integrated development environment XNA Game Studio*.

Penggunaan *Kinect* dalam *game Dodgeball* dimaksudkan untuk mengontrol *character game* agar sesuai dengan kenyataan. *Character* bergerak sesuai dengan pergerakan pemain. Dengan demikian, permainan terasa lebih nyata, dan pemain merasakan sensasi bermain *Dodgeball* yang sebenarnya. Selain itu, pemain juga tidak membutuhkan *controller* dalam bermain. Pergerakan pemain juga lebih natural karena tidak terkendala dengan *game controller*.

Game Dodgeball mengalami beberapa penyesuaian diantaranya, jumlah pemain masing – masing tim adalah 4, dimana maksimum 2 diantaranya adalah pemain manusia. Bola yang digunakan dalam permainan hanya 2, dan pada awal permainan bola diberikan melalui *coin toss*. *Game Dodgeball* juga dikemas dengan menggunakan fisika dan *artificial intelligence* sederhana untuk menambah pengalaman bermain *game Dodgeball*.

Kata Kunci: *Dodgeball, Virtual Reality, Game, Kinect, XNA Game Studio.*

ABSTRACT

Technology development, particularly in video game, has reached to all human life aspects. Many game developers seek not only for innovations to a more interesting gaming system, but also for added values to the game itself. Virtual reality game becomes one of the alternatives to develop the gameplay aspect. Players are highly immersed in the video game world, and convinced as if they were in the game itself. This thesis aims to make a virtual reality game with Dodgeball sport theme using Kinect and XNA Game Studio integrated development environment.

The Kinect used in this Dodgeball game is meant to control a game character to follow the real one. The character moves in the same way in real life. Therefore, players will find the game more realistic, and they will feel the sensation of playing the real Dodgeball. Besides, players do not need to use a game controller.

Players' movement also become more natural in the absence of the game controller.

Our Dodgeball game has gone through several adjustments, such as, each team's players are limited to 4 people, which at most 2 of them are human players. The number of balls used in this game is reduced to 2, and before the game begins, the balls are assigned to teams by coin tosses. Dodgeball game is also packed with physics and simple artificial intelligence to enhance players' gaming experiences in playing Dodgeball game.

Keywords: *Dodgeball, Virtual Reality, Game, Kinect, XNA Game Studio.*

1. LATAR BELAKANG

Perkembangan teknologi telah merambah pada seluruh aspek kehidupan manusia. Terlebih pada dunia perkembangan *video game*. Dunia perkembangan *game* saat ini telah jauh berbeda dengan perkembangan *game* pada era ketika *video game* diperkenalkan. Secara tampilan grafis, musik, efek suara, cerita, alur permainan, jenis *game*, maupun konsep permainan telah banyak diteliti oleh para pengembang *game* dalam meriset pangsa pasar. Hasil observasi tentu menjadi sebuah nilai lebih ketika *game* yang telah dibuat dirilis dan dibeli oleh masyarakat.

Perkembangan *game* tentu saja tidak hanya disambut secara positif. Banyak kalangan yang mengasumsikan bahwa *game* memberikan pengaruh negatif terhadap pemain. Sebagian besar dari kalangan tersebut menyalahkan *game* yang membuat pemainnya duduk di depan komputer maupun televisi selama berjam – jam dengan adegan penuh kekerasan dan hal – hal yang kurang pantas disajikan dalam sebuah *video game*. Akibat dari opini umum tersebut, banyak pengembang *game* yang mulai mencari inovasi terhadap sistem permainan yang menarik, namun juga memberi nilai lebih pada permainan tersebut.

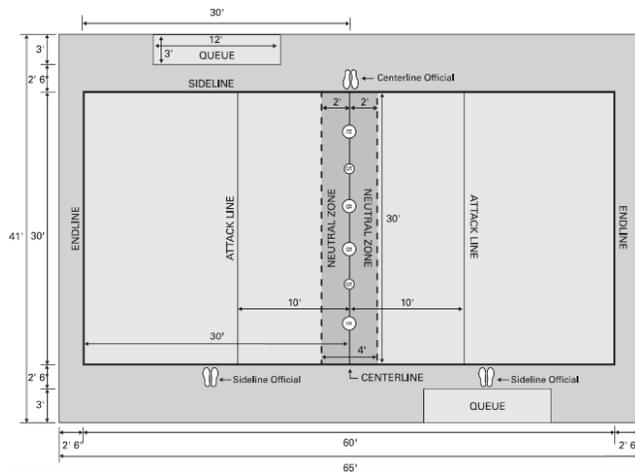
Virtual reality game menjadi salah satu alternatif untuk mengembangkan sisi *gameplay*. Pemain dibuat seolah – olah menyatu dengan dunia *game* sehingga keterlibatan pemain sangat tinggi. Konsep *virtual reality game* masih baru dan masih terus dikembangkan. Contoh penelitian yang dilakukan adalah membuat *fighting game* menggunakan *Kinect* dan *XNA Game Studio* yang dilakukan oleh Aris Semertzidis [6]. *Game* dibuat dengan mengintegrasikan *Kinect* sehingga dapat mengontrol prototipe model. Hasil dari penelitian tersebut adalah bahwa permainan yang dibuat sangat baik dalam sistem *motion*, namun kurang dalam grafik *game* yang bukan menjadi fokus utama penelitian.

Dari permasalahan yang dijabarkan di atas, skripsi ini bertujuan untuk membuat *virtual reality game* bertemakan *sport* yang memanfaatkan *Kinect* dengan *integrated development environment XNA Game Studio* [2]. Bahasa yang digunakan dalam pembuatan *game* ini adalah bahasa *C#*. Dari tema *sport* yang digunakan adalah permainan *Dodgeball*.

2. DODGEBALL

Dodgeball adalah permainan olahraga yang dimainkan oleh 2 tim beranggotakan masing – masing 6 – 12 orang. 6 pemain aktif dari setiap tim bermain dalam sebuah set. Sebuah set diakhiri ketika semua pemain dalam tim tersebut keluar, dan poin diberikan kepada tim lawan. Seorang pemain dinyatakan keluar jika pemain tersebut terkena bola hidup yang lalu menyentuh objek yang tidak menjadi bagian aktif dari permainan, contoh: tanah, atap, dinding, maupun orang yang bukan pemain. Bola hidup adalah bola yang dilemparkan pemain sampai menyentuh benda mati maupun ditangkap oleh pemain aktif lawan.

Ukuran lapangan yang digunakan dalam permainan *dodgeball* adalah 18.29 meter x 9.14 meter, dan dibagi menjadi 2 bagian yang sama dan sebangun dan terdapat garis menyerang sebesar 3.05 meter dari garis tengah.



Gambar 1. Lapangan *Dodgeball*

Bola yang digunakan dalam permainan *dodgeball* adalah bola berdiameter 17 cm yang terbuat dari karet dengan berat antara 120 – 160 gram. Bola yang lebih berat dari 170 gram tidak digunakan dalam permainan *dodgeball* karena dapat membahayakan pemain. Pemain juga diharapkan menggunakan seragam dalam permainan dan juga menggunakan sepatu tenis ataupun sepatu olahraga untuk menghindari cedera.

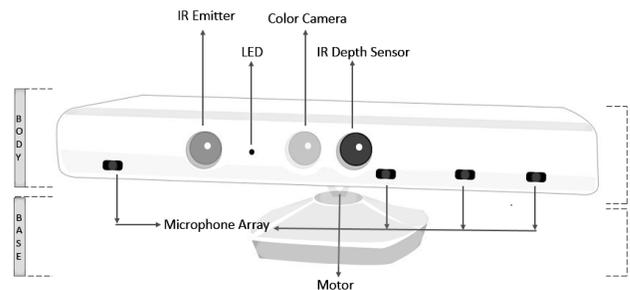
Lama permainan adalah 40 menit yang dibagi menjadi 2 bagian dan waktu istirahat 5 menit. Tujuan permainan adalah untuk mengeliminasi semua pemain lawan dalam waktu yang ditentukan. Tidak ada keputusan seri untuk permainan *dodgeball*. Permainan dimulai dengan 6 bola berada pada garis tengah. Pemain kemudian mengambil posisi pada garis ujung lapangan dan kemudian mengambil bola. Bola yang diambil dapat dilempar setelah bola berada dibelakang garis menyerang. Tim pertama yang berhasil mengeliminasi 6 pemain dari tim lawan mendapatkan 1 poin pada set yang dimainkan. Tim dengan poin yang lebih tinggi setelah 40 menit permainan menjadi pemenang.

3. KINECT

Kinect adalah *device* yang dibuat oleh *Microsoft* untuk *Xbox 360*, *Xbox One*, dan *Windows PC*. *Kinect* mendapatkan *input* dari *user* untuk berinteraksi dengan *game*. *Input* yang diterima berupa *natural user interface* yang terbagi menjadi *motion sensing* dan *speech recognition*. *Motion detection* dalam *Kinect* memanfaatkan konsep *skeleton tracking* yang mengenali dan mengikuti kerangka pemain. Setiap kerangka pemain menghasilkan 20 *joints* yang dianalisis fiturnya oleh *Kinect*. *Kinect* dapat mendeteksi hingga 6 kerangka pemain, namun *Kinect* hanya dapat melacak informasi lengkap sebanyak 2 kerangka. *Kinect* juga dapat mendeteksi *depth* atau jarak dari pemain maupun lingkungan pemain [7].

Kinect memiliki *color camera* yang mendeteksi warna *RGB*. *Kinect* bekerja dengan kecepatan 30 *frames per second (fps)* pada resolusi 640 x 480 *pixels* hingga resolusi maksimum 1280 x 960 *pixels* dengan kecepatan 12 *fps*. *Kinect* juga dapat mendeteksi kedalaman menggunakan *depth sensor* yang bekerja pada resolusi 320 x 240 *pixels* maupun 640 x 480 *pixels*. *Depth sensor* pada *Kinect* memanfaatkan *infrared emitter (IR emitter)* dan *IR depth sensor*, sehingga *Kinect* dapat mendeteksi pada jarak 800 mm hingga 4000 mm. Jangkauan pandang *Kinect* sejauh 43 derajat horizontal dan 57 derajat vertikal.

Kinect juga memiliki *microphone array* sejumlah 4 buah yang memungkinkan *Kinect* untuk mendapat *input* suara. 4 buah sensor dimanfaatkan agar *Kinect* mengetahui arah sumber suara dengan membandingkan waktu masing – masing sensor dalam mendapatkan sinyal suara. *Microphone array* pada *Kinect* juga mampu mengeliminasi suara gema dengan *Acoustic Echo Cancellation (AEC)* [1]. Pola suara yang diterima pada *microphone* dikurangi dengan pola suara yang dikeluarkan oleh *speaker*. *Kinect* juga melakukan *noise suppression* terhadap sinyal suara yang diterima untuk menghilangkan pola suara yang tidak berupa pembicaraan (*non-speech*). Penghilangan pola suara dilakukan agar sinyal suara yang mengandung *background noise* dapat dieliminasi, sehingga hasilnya adalah suara pembicaraan dari pemain. *Kinect* dapat melakukan *Automatic Gain Control (AGC)*, yakni proses untuk mempertahankan amplitudo dari suara pemain agar selalu konsisten, walaupun jarak pemain terhadap *Kinect* berubah – ubah [3].



Gambar 2. Komponen dalam *Kinect*

Fitur *skeletal tracking* pada *Kinect* berguna untuk melacak kerangka pemain yang berada pada jangkauan pandang *Kinect* dan juga untuk melacak setiap sendi (*joints*) pada kerangka pemain. *Kinect* memiliki kemampuan untuk mendeteksi hingga 6 kerangka pemain, namun hanya melacak 2 kerangka secara aktif, meliputi pelacakan sendi pada kerangka pemain. Satu kerangka pemain yang dilacak memiliki tiga macam kondisi (*state*), yakni:

- *NotTracked* yang menandakan bahwa objek tidak mewakili pemain yang sedang dilacak. Dalam kondisi *NotTracked*, *field Position* dan *Joint* memiliki nilai 0.
- *PositionOnly* menandakan bahwa kerangka pemain dideteksi, namun tidak dilacak secara aktif. *Field Position* pada kondisi *PositionOnly* tidak bernilai 0, namun *field Joint* memiliki nilai 0
- *Tracked*, pada kondisi ini, kerangka pemain dilacak secara aktif. Pada kondisi ini, *Field Position* dan *Joint* memiliki nilai bukan 0.

Joint atau sendi yang dilacak oleh *Kinect* berjumlah 20 buah untuk setiap kerangka. Bagian yang dilacak antara lain kepala, bahu kiri, bahu tengah, bahu kanan, siku kiri, siku kanan, pergelangan tangan kiri, pergelangan tangan kanan, tangan kiri, tangan kanan, tulang belakang, pinggul kiri, pinggul tengah, pinggul kanan, lutut kiri, lutut kanan, pergelangan kaki kiri, pergelangan kaki kanan, kaki kiri, dan kaki kanan. Setiap *Joint* yang dideteksi memiliki tiga macam *state*, antara lain:

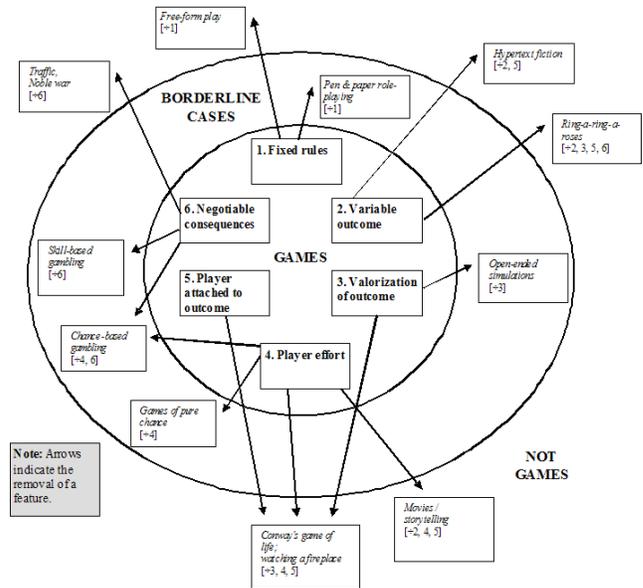
- *Inferred* yaitu kondisi dimana *Kinect* tidak dapat melihat *joint* pada *depth frame pixels*, namun *Kinect* telah menghitung perkiraan posisi *joint* tersebut.
- *NotTracked*, dimana posisi *joint* tidak dapat ditentukan, variabel *Position* memiliki nilai 0.
- *Tracked*, dimana *joint* dapat dideteksi dengan baik dan dilacak secara aktif oleh *Kinect*.

4. GAME

Overmars menjabarkan beberapa elemen yang dimiliki sebuah *game*, diantaranya adalah [5]:

- Adanya partisipasi aktif dari pemain, partisipasi dapat berupa pengontrolan ataupun pembuatan keputusan dari pemain dalam sebuah *game* yang mempengaruhi jalannya *game*.
- Adanya tujuan khusus yang harus dicapai oleh pemain, pemain dituntut untuk memenuhi tujuan yang ada dalam *game*, jika tujuan tersebut tidak dipenuhi, maka ada konsekuensi yang diterima pemain.
- Sebuah *game* bersifat dinamis, dimana *game* tersebut dapat dimainkan berulang kali dan memiliki beragam strategi untuk menang.

Juul menyimpulkan bahwa sebuah *game* adalah sebuah sistem formal yang bersifat *rule-based* yang memiliki hasil (*outcome*) yang bervariasi, dimana pemain mengeluarkan tenaga dan usaha untuk mempengaruhi jalannya permainan, dan pemain merasa terikat pada hasil akhirnya, serta konsekuensi dari permainan bersifat *optional*. Gambar 3 menjelaskan keenam elemen yang dimiliki sebuah *game* [4].



Gambar 3. Diagram Penentuan *Game* Berdasarkan 6 Elemen *Game*

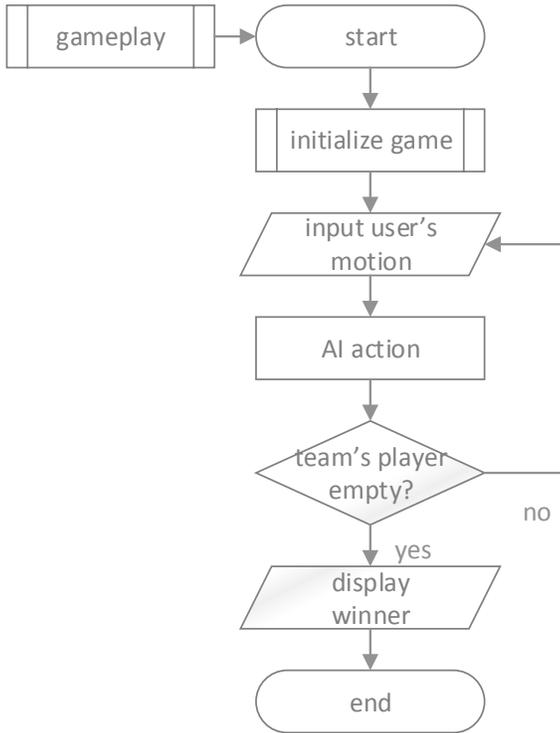
5. DESAIN SISTEM

Sistem kerja *game* dimulai dengan pengecekan ketersediaan alat pendeteksi berupa *Kinect*. Jika *Kinect* tidak ditemukan pada saat *game* dijalankan, maka secara otomatis *game* berhenti. Setelah *Kinect* terdeteksi, maka dijalankan inialisasi awal pada *game*. Inialisasi berupa proses *load resources* yang dibutuhkan untuk *graphic user interface* pada menu awal. *Game* menampilkan menu pilihan utama setelah inialisasi awal dijalankan. Pemain dapat memilih menu pilihan dengan menggerakkan tangan di depan *Kinect*. Setelah menu dipilih, maka *game* menjalankan sub-proses yang dipilih pemain.

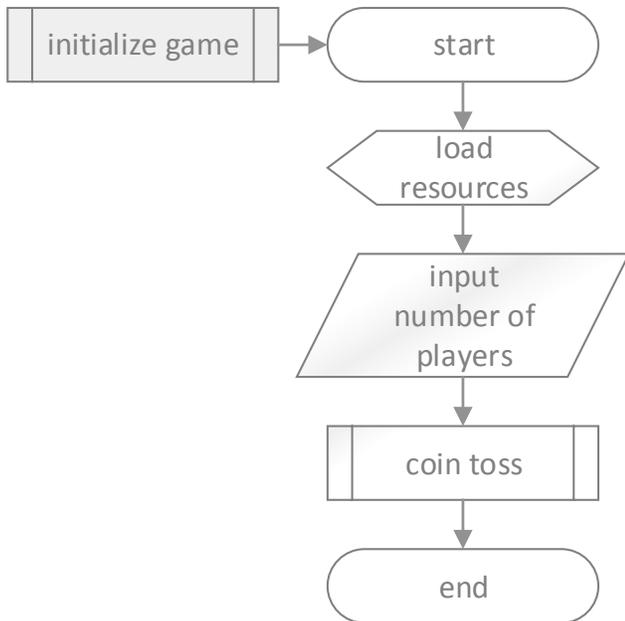
Pada sub-proses *Gameplay*, pemain dapat menjalankan permainan *dodgeball*. Permainan dimulai dengan inialisasi *game*. Inialisasi *game* berupa pemilihan jumlah pemain yang berpartisipasi dalam *game Dodgeball* dan proses pemberian bola kepada tim pemain. Sistem pemberian bola ditentukan dengan *coin toss* dimana pemain memilih bagian *head* atau *tail*. Sistem *coin toss* dilakukan sebanyak dua kali, untuk masing – masing bola. Setelah *coin toss* dilakukan, permainan dimulai tanpa batas waktu. Permainan bersifat *real time* sehingga pemain dapat melakukan gerakan apapun dan selalu dilacak oleh *Kinect*. Gerakan yang dibuat oleh pemain (termasuk saat pemain diam) dianggap sebagai *input* oleh *Kinect* dan digunakan untuk mengontrol *avatar* dalam *game*.

Artificial intelligence dalam *game* juga melakukan aksi selama permainan *dodgeball* berlangsung. Beberapa gerakan yang dapat dilakukan antara lain melempar (*throwing*), menangkap (*catching*), menghindari (*dodging*), dan mengambil bola (*taking*). Gerakan – gerakan yang dilakukan *artificial intelligence* juga dapat dilakukan oleh pemain dengan *motion* tertentu. Permainan terus berlangsung selama masing – masing tim memiliki pemain aktif dalam area permainan. Apabila salah satu tim tidak memiliki pemain, maka tim tersebut kalah dan tim lawan menang. Gambar

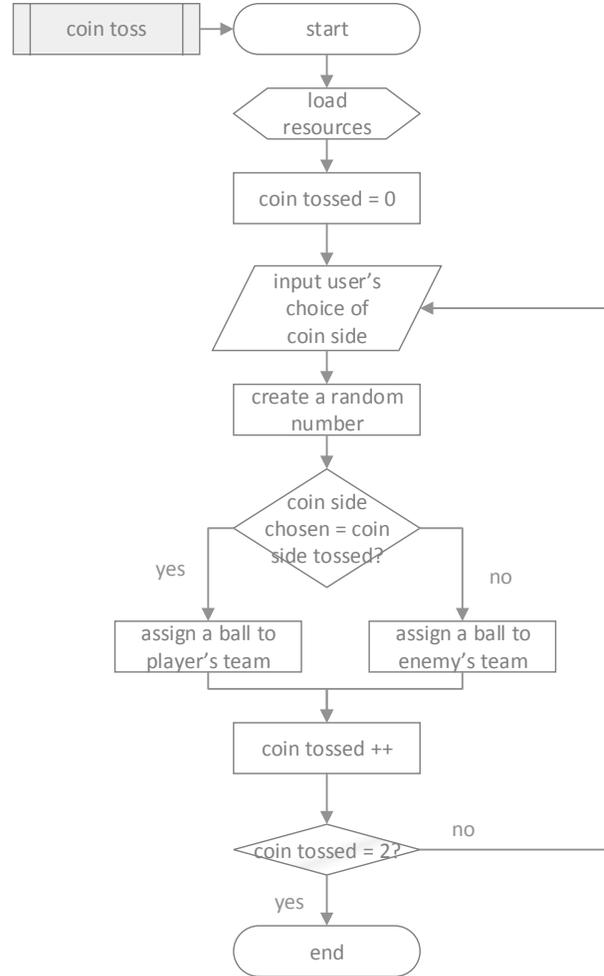
4 menunjukkan kinerja sub-proses *gameplay* pada *game Dodgeball*.



Gambar 4. Flowchart Sub-Proses *Gameplay*



Gambar 5. Flowchart Sub-Proses *Initialize Game*



Gambar 6. Flowchart Sub-Proses *Coin Toss*

Pemrograman *game Dodgeball* memanfaatkan konsep *object-oriented programming* yang menggunakan beberapa *class* di dalamnya. *Class* - *class* yang digunakan antara lain: *class Camera.cs*, *class Sprite.cs*, *class Model3D.cs*, *class Ball.cs*, *class Player.cs*, dan *class Field.cs*. *Class Camera.cs* memiliki fungsi untuk mengimplementasikan *fixed camera* yang menampilkan keseluruhan isi lapangan atau *field* dari *game Dodgeball*. Tampilan lapangan ditampilkan dengan ketinggian tertentu dari sudut pandang orang ketiga, tidak dari sudut pandang salah satu pemain (*first person*) agar orang yang bermain *game Dodgeball* dapat melihat seluruh kondisi lapangan secara bersamaan, dapat memperhatikan posisi dan gerakan lawan, dan juga kondisi teman satu tim. Selain itu, pemain juga dapat melihat dan memperkirakan arah datangnya bola yang dilempar oleh tim lawan.

Class Sprite.cs dan *class Model3D.cs* mengimplementasikan *resource manager* yang berfungsi untuk mengendalikan isi gambar, tekstur, model, dan *mesh* yang dibutuhkan untuk ditampilkan pada layar permainan. Tujuan dari implementasi *class Sprite.cs* dan *class Model3D.cs* adalah untuk mempercepat dan mempermudah pemrograman dan proses *debugging*. *Class Model3D.cs* juga dimodifikasi agar memiliki *collision checking*

method berupa implementasi *bounding sphere* pada model yang ada di dalam *class* tersebut.

Class Ball.cs merupakan *derived class* dari *class Model3D.cs*. *Class* ini mengimplementasikan semua *property* yang dimiliki bola *Dodgeball*. Atribut dalam *class* ini antara lain posisi, kecepatan, dan percepatan. Atribut – atribut dalam *class Ball.cs* digunakan untuk mengimplementasikan fisika yang terdapat dalam *game Dodgeball*. *Class Player.cs* juga merupakan *derived class* dari *class Model3D.cs*. Dalam *class* ini, terdapat *property* yang dibutuhkan untuk menggerakkan model 3D dari pemain. Animasi pergerakan yang dilakukan pemain maupun *artificial intelligence* diimplementasikan ke dalam *class Player.cs*.

Class Field.cs merupakan *class* yang mewadahi seluruh *class* yang menjadi elemen penting dalam permainan olahraga *Dodgeball*. Menggunakan konsep *composition*, *class Field.cs* memiliki 8 objek pemain yang direpresentasikan oleh *class Player.cs*, 2 objek bola yang direpresentasikan oleh *class Ball.cs*, dan 1 objek *fixed camera* yang direpresentasikan oleh *class Camera.cs*. Selain memiliki *property – property* bola, pemain, dan kamera, *class Field.cs* juga memiliki model lapangan yang direpresentasikan *class Model3D.cs*. Model lapangan mengimplementasikan *floor* dan *virtual wall* yang menjadi *boundary area* dari permainan *Dodgeball*. Secara teknis, pemain maupun bola tidak dapat melewati *boundary area* permainan. *Class Field.cs* juga menjadi penghubung dari *main code* kepada objek – objek permainan yang dibutuhkan.

6. PENGUJIAN DAN EVALUASI

Gambar 7 menunjukkan proses *dodging* yang dilakukan oleh pemain dalam *game Dodgeball*. Proses *dodging* dilakukan hanya dengan mengambil posisi dan pergerakan pemain melalui *device Kinect*.



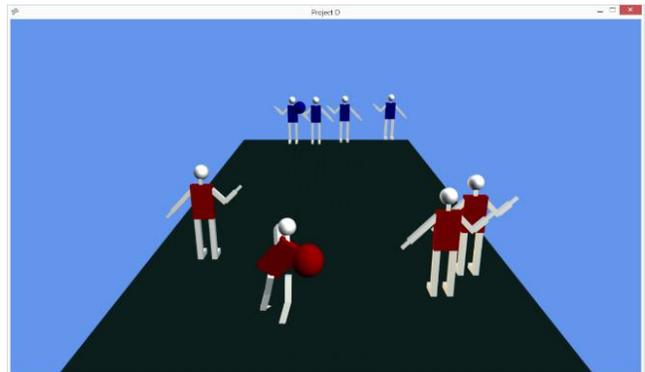
Gambar 7. Proses *Dodge* dalam *Game*

Dalam melakukan *dodging*, pemain dapat bergerak bebas ke samping kiri maupun kanan, ke depan maupun belakang, dan dengan berbagai pose. Pergerakan pemain bersifat natural dan tidak dibatasi *game controller*.



Gambar 8. Pergerakan Pemain Ke Depan

Gambar 9 dan 10 menunjukkan proses lain yang terjadi dalam *game Dodgeball*, yaitu pelemparan bola. *Action throwing* yang terjadi didasarkan pada pergerakan lengan kanan ke belakang untuk bersiap – siap melempar dan ke depan untuk melempar bola ke arah lawan.

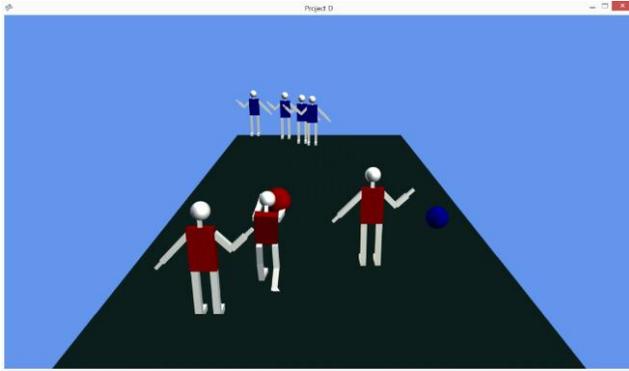


Gambar 9. Proses Persiapan Melempar Bola



Gambar 10. Proses Pelemparan Bola

Pemain juga dapat melakukan *action* lainnya, yaitu menangkap bola yang dilempar oleh tim lawan. Proses *catching* dilakukan dengan *gesture* memajukan kedua lengan ke depan, seolah – olah seperti menangkap bola yang melesat.



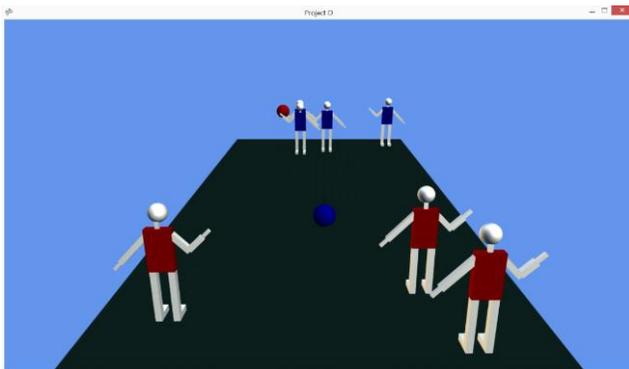
Gambar 11. Proses Penangkapan Bola

Pemain juga dapat mengambil bola mati (*dead ball*) yang sudah dilempar oleh tim lawan dan mengenai salah satu anggota tim maupun *virtual wall* yang menjadi *boundary area* lapangan dengan *action take ball*. Cara untuk melakukan *action* ini adalah dengan memosisikan pemain agar sama dengan posisi *dead ball*.



Gambar 12. Proses Pengambilan Bola

Pengujian juga dilakukan untuk *collision detection* yang digunakan dalam *game Dodgeball*. *Collision detection* berguna untuk menentukan apakah pemain dinyatakan keluar dari permainan akibat terkena bola lemparan lawan atau tidak.



Gambar 13. Pemain Terkena Lemparan Bola

Selain *collision detection*, pengujian dilakukan kepada proses tumbukan bola terhadap pemain maupun *virtual wall*. Dengan demikian, maka bola yang bertumbukan mengalami perubahan kecepatan ke arah berlawanan dari arah semula.

Evaluasi *virtual reality game Dodgeball* dilakukan berupa survey kepada 50 pemain yang telah menguji coba permainan *Dodgeball*. Pada evaluasi *game Dodgeball*, terdapat 8 kriteria yang dinilai dari skala 0 hingga 10. Aspek – aspek yang diuji antara lain:

- Tingkat kesenangan permainan (*game excitement level*)
- Tingkat kesulitan permainan (*game difficulty level*)
- Tingkat kemudahan kontrol permainan (*game-controlling ease level*)
- Tingkat ketertarikan permainan (*game attractiveness level*)
- Tingkat responsif permainan (*game responsiveness level*)
- Tingkat kerealistisan permainan (*game realism level*)
- Tingkat adiktif permainan (*game addiction level*)
- Tingkat kepuasan pemain (*player satisfaction level*)

Nilai 0 pada kriteria – kriteria diatas menunjukkan negasi atau kebalikan dari kriteria yang disebutkan, dan nilai 10 menunjukkan kriteria yang disebutkan terpenuhi secara menyeluruh. Pada survey, responden juga diminta saran, masukan, serta kritik untuk *game Dodgeball*. Berikut adalah hasil survey untuk masing – masing kriteria.

Tabel 1. Hasil Evaluasi *Game Dodgeball*

Kriteria Pengujian	Nilai yang Diperoleh
Tingkat kesenangan permainan	7.86
Tingkat kesulitan permainan	5.28
Tingkat kemudahan kontrol permainan	8.84
Tingkat ketertarikan permainan	7.34
Tingkat responsif permainan	9.82
Tingkat kerealistisan permainan	5.68
Tingkat adiktif permainan	7.56
Tingkat kepuasan pemain	8.82
Nilai keseluruhan (rata – rata)	7.65

7. KESIMPULAN

Berdasarkan hasil pengujian, dapat disimpulkan beberapa hal sebagai berikut:

- Pendeteksian pergerakan pemain melalui *device Kinect* sudah cukup baik dengan menunjukkan hasil yang relevan. *Skeleton* yang didapatkan *Kinect* sesuai dengan animasi pergerakan *character game* yang bermain dalam permainan *Dodgeball*.
- Nilai dari survey yang didapatkan *game Dodgeball* adalah 7.65 poin dari 10.00 poin. *Game Dodgeball* cukup layak disebut sebagai sebuah *game* karena memiliki elemen – elemen yang wajib dimiliki sebuah *game* yang baik.
- Pergerakan pemain yang terbatas pada *field of view Kinect* sudah diatasi dengan melakukan *scaling* pada pergeseran *character game* dan melakukan translasi agar posisi *character game* berada sesuai dengan posisi pemain pada dunia nyata.

Dibandingkan dengan *game fighting* yang digunakan sebagai referensi, *game Dodgeball* memiliki keunggulan pada segi sistem

permainan karena memiliki *projectile* berupa bola *dodgeball*, memiliki fisika permainan, dan memiliki *artificial intelligence*.

8. REFERENSI

- [1] Catuhe, D. 2012. *Programming with the Kinect for Windows Software Development Kit*. USA: Microsoft Press.
- [2] Drumm, L. 2012. *Microsoft XNA 4.0 Game Development Cookbook*. UK: Packt Publishing.
- [3] Jana, A. 2012. *Kinect for Windows SDK Programming Guide*. UK: Packt Publishing.
- [4] Juul, J. 2003. *The Game, the Player, the World: Looking for a Heart of Gameness*. Utrecht: Uthrect University.
- [5] Overmars, M. 2007. *Designing Good Games*. YoYo Games Ltd.
- [6] Semertzidis, A. 2012. *Game Development for Kinect*. UK. <<https://dl.dropboxusercontent.com/u/13602991/Thesis/Game%20Development%20For%20Kinect.pdf>>
- [7] Webb, J. and Ashley, J. 2012. *Beginning Kinect Programming with the Microsoft Kinect SDK*. USA: Apress