TUGAS AKHIR – TE 141599

AUGMENTED REALITY SIMULASI PERMAINAN BOLA BERBASIS SMART TERRAIN

Pondra Setiawan
NRP 2210100198

Dosen Pembimbing
Dr. Surya Sumpeno, ST., M.Sc.
Christyowidiasmoro, ST., MT.

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOVEMBER
Surabaya 2016

Halaman ini sengaja dikosongkan

FINAL PROJECT – TE 141599

AUGMENTED REALITY BALL GAMES SIMULATION BASED ON SMART TERRAIN

Pondra Setiawan
NRP 2210100198

Adviser
Dr. Surya Sumpeno, ST., M.Sc.
Christyowidiasmoro, ST., MT.

DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
Surabaya 2016

Halaman ini sengaja dikosongkan

AUGMENTED REALITY SIMULASI PERMAINAN BOLA BERBASIS *SMART TERRAIN*

TUGAS AKHIR

Diajukan guna Memenuhi Sebagian Persyaratan
Untuk Memeperoleh Gelar sarjana Teknik
Pada
Bidang Studi Teknik Komputer dan Telematika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Surya Sumpeno ST., M.Sc.,	Christyowidiasmoro, ST., MT.
NIP 196906131997021003	NIP 198301272009121004

SURABAYA
24 Januari 2016

ABSTRAK

Nama	:	Pondra Setiawan
Judul	:	*Augmented Reality* Simulasi Permainan Bola Berbasis *Smart Terrain*
Dosen Pembimbing	:	1. Dr. Surya Sumpeno, S.T, M.Sc. 2. Christyowidiasmoro, S.T, M.T

Augmented Reality merupakan salah satu teknologi yang populer dan banyak di kembangkan diberbagai bidang. Salah satunya ialah pada dunia game dimana dengan menggunakan sebuah marker tertentu dapat menampilkan objek virtual permainan diatas layar nyata secara real time. Perkembangan *augmented reality* ini berlanjut hingga era *smart terrain*, dimana marker hanya dibutuhkan saat inisialisasi awal saja. Dengan teknologi *smart terrain*, objek-objek yang berada pada lingkungan nyata dikenali secara dinamis dan objek tersebut akan digantikan dengan model virtual. Pada tugas akhir ini akan diajukan suatu permainan bola interaktif yang berbasis *smart terrain*, dimana pengguna dapat secara bebas mengatur area permainan itu sendiri. Metode yang digunakan pada sistem untuk mendeteksi objek permainan adalah dengan menggunakan teknologi *smart terrain* yang dikembangkan oleh vuforia. Dengan fitur rekognisi objek yang dimiliki oleh teknologi *smart terrain* akan membuat permainan yang dimainkan unik, berbeda-beda setiap kali dimainkan. Selain itu, dengan fitur *smart terrain* akan berbeda dengan game pada umumnya yang terbatas pada salah satu sisi sudut pandang permainan, tetapi juga pengguna dapat melihat sisi sudut pandang lain pada permainan sehingga seakan-akan pengguna berinteraksi secara langsung dengan permainan yang akan dimainkan. Dari pengujian yang dilakukan diperoleh hasil bahwa rekonstruksi objek nyata memiliki rasio perbandingan volume hingga 94.86% dari ukuran objek sesungguhnya.

Kata kunci: *Augmented Reality*, *Smart terrain*, Game

Halaman ini sengaja dikosongkan

ABSTRACT

Name	: Pondra Setiawan
Title	: *Augmented Reality Ball Games Simulation Based On Smart Terrain*
Advisors	: 1. Dr. Surya Sumpeno, S.T, M.Sc. 2. Christyowidiasmoro, S.T, M.T

Augmented Reality is one of the popular technology which being developed in various fields. One of the field is on gaming world where a marker can be used for visualizing virtual objects over real world surrounding on realtime manner. The development of augmented reality is continue to smart terrain era, a marker only used once for initiating the application. Based on smart terrain technology, real world objects can be recognized dynamically and replaced with virtual models. This final project is expected to provide a solution to apply an interactive ball games based on smart terrain method, where user can freely compose the game scene. The method used to recognizing real world object is smart terrain which is currently developed by Vuforia. With objects recognition based on smart terrain, we can make every game we play completely unique. Another advantage of smart terrain feature is user can view the games on all angles, different with any common games that can only be viewed at specific side. Hence, makes user seems to interact directly with the game that will be played. Based on the test result to reconstruct real world, objects that being reconstructed have volume percentage ratio up to 94.86% from real world dimension.

Keywords: Augmented Reality, Smart terrain, Game

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan Syukur kehadirat Tuhan YME atas segala limpahan berkah, serta rahmat-Nya, penulis dapat menyelesaikan penelitian ini dengan judul: ***Augmented Reality Simulasi Permainan Bola Berbasis Smart terrain***.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Jurusan Teknik Elektro ITS, Bidang Studi Teknik Komputer dan Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S-1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara yang telah memberikan dorongan spiritual dan material serta seluruh kerabat dan kolega penulis yang banyak membantu proses dalam menyelesaikan buku penelitian ini.
2. Bapak Dr., Eng., Ardyono P., S.T., M.Eng. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember
3. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Surya Sumpeno, ST., M.Sc. dan Bapak Christyowidiasmoro, ST., MT. atas bimbingan selama mengerjakan penelitian.
4. Bapak-ibu dosen pengajar Bidang Studi Teknik Komputer dan Telematika, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman angkatan e-50 serta teman-teman B201crew Laboratorium Bidang Studi Teknik Komputer dan Telematika.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, 22 Januari 2016

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

Penelitian ini dilatar belakangi oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1. Latar Belakang

Augmented reality telah menjadi fokus utama dari beberapa penelitian. Dalam konteks yang umum, *augmented reality* digunakan untuk memberikan pemahaman yang lebih baik terhadap keadaan lingkungan dengan menambahkan beberapa jenis informasi virtual atau skenario yang tampak nyata didalamnya. Bahkan, untuk pengembangan aplikasi yang kompleks dapat dikembangkan meskipun dengan sumber yang terbatas, termasuk didalamnya melibatkan komputasi geometri.

Perkembangan *augmented reality* ini berlanjut hingga era *smart terrain*, dimana marker hanya dibutuhkan saat inisialisasi awal saja. Dengan teknologi *smart terrain*, objek-objek yang berada pada lingkungan nyata dikenali secara dinamis dan objek tersebut akan digantikan dengan model virtual. Dengan konsep seperti itu, disusunlah suatu game interaktif permainan bola dengan menggunakan teknologi augmented reality *smart terrain*. Dengan *smart terrain*, objek yang berada di lingkungan nyata digunakan sebagai pemicu adanya pemain virtual didalam game, dimana pengguna dapat secara bebas mengatur area permainan itu sendiri. Dengan fitur rekognisi objek yang dimiliki oleh teknologi *smart terrain* akan membuat permainan yang dimainkan unik akan berbeda-beda setiap kali dimainkan. Selain itu, dengan fitur *smart terrain* akan berbeda dengan game pada umumnya yang terbatas pada salah satu sisi permainan, tetapi juga pengguna dapat melihat sudut sisi lain pada permainan sehingga seakan-akan pengguna berinteraksi secara langsung dengan permainan yang akan dimainkan.

Permainan berbasis *augmented reality smart terrain* ini akan sangat efisien apabila dikembangkan dalam piranti bergerak. Gambar yang diambil oleh kamera perangkat android, dideteksi lalu ditampilkan

pemain virtual diatasnya. Hal ini akan memudahkan pengguna untuk berinteraksi dengan permainan, seakan-akan game tersebut hadir di lingkungan nyata. Metode ini merupakan metode yang interaktif dan juga meningkatkan efisiensi karena dapat langsung diterapkan oleh pengguna android secara praktis.

1.2. Permasalahan

Saat ini penggunaan marker pada augmented reality hanya digunakan sebagai pemicu munculnya objek virtual. Seperti pada aplikasi *augmented reality* GIS (*Geographic Information System*), dimana objek virtual yang ditampilkan hanya menunjukkan informasi mengenai area tertentu, bukan untuk merekonstruksi bagaimana keadaan objek sebenarnya dilingkungan nyata. Selain itu peletakan objek penghalang dalam permainan bola selama ini diatur secara manual oleh para *developer*. Bagaimana apabila para *developer* ini memberikan kesempatan pada pengguna awam untuk mengatur posisi peletakan objek mereka sendiri secara realtime. Hal ini akan menambahkan tingkat ke interakifan game yang sedang dimainkan.

1.3. Tujuan

Tujuan penelitian ini adalah untuk mengembangkan aplikasi berbasis *smart terrain* yang mampu menampilkan permainan bola virtual yang interaktif pada perangkat android. Hasil yang diperoleh dapat dimanfaatkan untuk meningkatkan interaksi antara game dengan pengguna, dengan cara menggunakan objek yang ada di lingkungan nyata. Selain itu dapat juga memberikan alternatif kepada pengguna yang awam terhadap dunia *developing* suatu permainan agar bisa menentukan lokasi peletakan objek penghalang sendiri.

1.4. Batasan Masalah

Untuk memfokuskan permasalahan yang akan diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut diantaranya adalah:

1. Peletakan objek penghalang tidak dapat dipindahkan setelah melakukan proses rekonstruksi *smart terrain*.
2. Permainan akan berbasis pada prototipe permainan *pinball*.
3. Permainan akan menggunakan Unity3D sebagai *Game Engine*.

1.5. Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga lebih mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang hendak melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

BAB I PENDAHULUAN

Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, tujuan penelitian, metodologi penelitian dan sistematika laporan.

BAB II DASAR TEORI

Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada pengerjaan tugas akhir ini. Teori-teori ini digunakan sebagai dasar dalam pengerjaan, yaitu informasi terkait, teori mengenai model permainan yang digunakan, algoritma yang akan digunakan, teori mengenai Unity3D sebagai Game Engine, dan teori-teori penunjang lainya.

BAB III PERANCANGAN SISTEM DAN IMPLEMENTASI

Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Guna mendukung itu digunakanlah blok diagram agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk diimplentasikan pada pembuatan perangka.

BAB IV PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang pengujian yang dilakukan terhadap sistem dalam penelitian ini dan menganalisa sistem. Spesifikasi perangkat keras dan perangkat lunak yang digunakan juga disebutkan dalam bab ini. Tujuannya adalah sebagai variable kontrol dari pengujian yang dilakukan.

BAB V PENUTUP

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangkan lebih lanjut juga dituliskan pada bab ini.

1.6. Relevansi

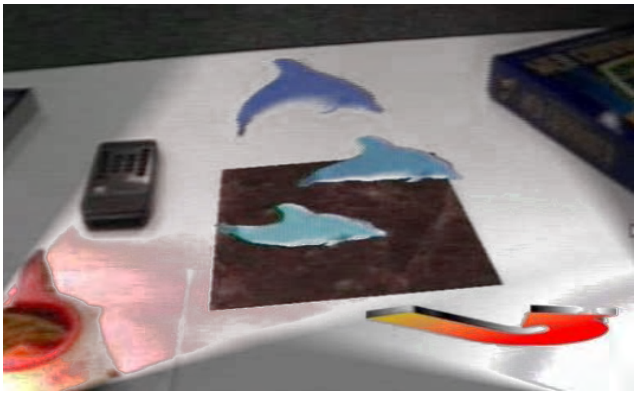
Penelitian ini memiliki relevansi dengan penelitian-penelitian tentang *augmented reality* yang telah dilakukan sebelumnya. Terutama penelitian pada perangkat android yang dikembangkan oleh Qualcomm Vuforia mengenai *smart terrain*. Pada tugas akhir ini diharapkan akan berkontribusi dalam hal penerapan game *augmented reality* interaktif pada piranti bergerak dimana pengguna dapat secara bebas menentukan area permainan yang diinginkan.

BAB 2
DASAR TEORI

Untuk mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini lebih terarah.

2.1. *Augmented Reality*

Augmented Reality merupakan suatu teknologi penggabungan antara objek nyata dengan objek maya dengan memproyeksikan dan menambahkan benda-benda tersebut secara real time. Dengan adanya AR, lingkungan nyata yang ada di sekitar akan dapat berinteraksi dalam bentuk virtual. Segala informasi tentang objek dan lingkungan dapat ditambahkan ke dalam sistem AR yang kemudian informasi-informasi tersebut akan ditampilkan ke lingkungan nyata secara real-time seperti pada Gambar 2.1



Gambar 2.1 Aplikasi *Augmented Reality* untuk menampilkan objek virtual lumba-lumba diatas lingkungan nyata.

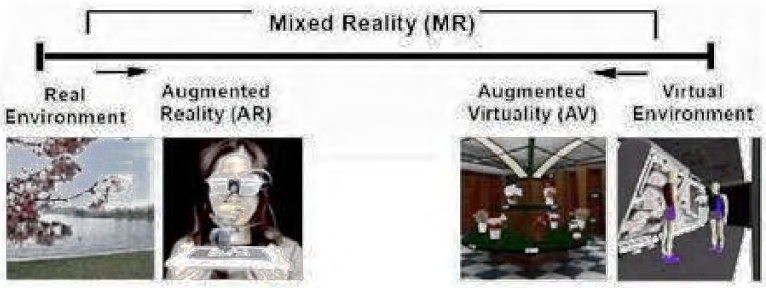
Ronald Azuma menyatakan bahwa Augmented Reality adalah sesuatu tentang menambahkan dunia nyata dengan informasi virtual dengan meningkatkan indra seseorang dan keterampilan manusia. Augmented Reality menggabungkan karakter virtual dengan dunia

nyata[1]. Terdapat beberapa karakteristik umum tentang Augmented Reality, yaitu:

1. Kombinasi dari lingkungan nyata dengan objek virtual.
2. Tampilan yang interaktif
3. Tampilan dalam bentuk 3D

Terdapat variasi lain dari teknologi *Augmented reality* (AR) yaitu Virtual Reality (VR). Berbeda dengan virtual reality dimana tampilan dunia nyata sepenuhnya digantikan dengan dunia maya, Augmented Reality memungkinkan pengguna untuk melihat dunia nyata, dengan objek maya yang dilapiskan diatasnya atau digabung dengan dunia nyata. Oleh karena itu, AR menambah realitas, bukan menggantinya. Idealnya, maka akan muncul ke pengguna bahwa benda virtual dan nyata tampil berdampingan di ruang yang sama.

Milgram's reality-virtuality continuum mendeskripsikan sebuah bagan yang memisahkan antara lingkungan nyata dan lingkungan virtual. Diantara kedua lingkungan itu terdapat *Augmented reality*, yang lebih dekat kepada lingkungan nyata dan *augmented virtuality* yang lebih dekat kepada lingkungan maya[2]. Diagram rangkaian kesatuan reality-virtuality Milgram ditunjukkan pada Gambar 2.2



Gambar 2.2 Perbedaan antara Augmented Reality dengan Virtual Reality[2]

2.2. Deteksi Fitur Natural

Dalam sistem Visi computer, deteksi fitur natural atau *Natural Feature Detection* merupakan pendekatan yang digunakan untuk mengekstraksi beberapa jenis fitur dan menyimpulkan isi dari suatu citra. Terdapat beberapa macam metode untuk mendeteksi fitur suatu citra diantaranya ialah SIFT, SURF, FERN dan lain lain. Terdapat beberapa macam metode untuk mendeteksi fitur suatu citra diantaranya ialah SIFT, SURF, FERN dan lain lain. Berikut ini adalah beberapa tahapan dalam mengekstraksi dan mendefinisikan suatu gambar dengan menggunakan metode SIFT yang telah di optimasi untuk penggunaannya pada perangkat bergerak [3].

2.2.1. *Feature Detection*

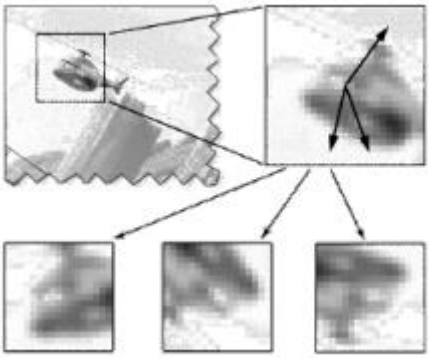
Feature detection atau deteksi fitur merupakan tahap untuk mengenali fitur-fitur yang terdapat dalam suatu citra. Algoritma deteksi fitur SIFT menggukan metode *Difference-of-Gaussian* (DoG) untuk menentukan tempat pencarian yang tidak hanya mendeteksi fitur, tetapi juga menghitung skala estimasi nya.

2.2.2. *Feature Tracking*

Setelah fitur terdeteksi, fitur selanjutnya dapat dikenali dengan cross correlation. Tracking fitur bukanlah proses yang sangat dibutuhkan kalkulasi yang besar, karena untuk mengenali suatu citra sifatnya independen antara frame satu dengan yang lainnya. Hal ini memiliki keuntungan yaitu kecepatan tracking akan meningkat karena setiap fitur yang dideteksi tidak harus dimatching dengan database.

2.2.3. *Descriptor Creation*

Descriptor dibuat berdasarkan orientasi fitur yang telah terdeteksi dan disimpan dalam bentuk koordinat vektor. Magnitude gradient serta orientasinya di perhitungkan lagi dan dibandingkan pusat dari fitur objek. Ekstraksi citra menggunakan metode SIFT dapat dilihat pada Gambar 2.3.



Gambar 2.3 Ekstraksi fitur menggunakan metode SIFT[4]

2.2.4. *Descriptor Matching*

Setelah semua fitur telah dideteksi dan disimpan, descriptor data yang dibuat dibandingkan dengan descriptor yang terdapat dalam database.

2.2.5. *Outlier Removal*

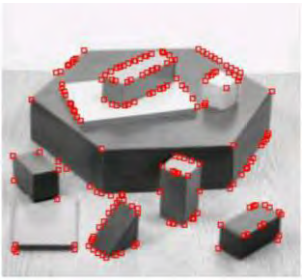
Meskipun SIFT dikenal sebagai deskriptor yang sangat kuat, masih menghasilkan outlier yang harus dihapus sebelum melakukan estimasi. Penghapusan outlier bekerja dalam tiga langkah. Langkah pertama menggunakan orientasi fitur. Semua orientasi fitur relatif diperbaiki terhadap sudut sudut rotasi berdasarkan orientasi fitur dalam database. Karena tracker terbatas pada target planar, target fitur harus memiliki orientasi yang sama. Diperkirakan orientasi terbaik untuk menyaring semua fitur yang tidak mendukung pelacakan. Karena fitur orientasi sudah tersedia, selain langkah ini cepat, langkah ini juga efisien dalam menghilangkan sebagian besar outlier. Langkah kedua menggunakan geometris sederhana. Semua fitur diurutkan berdasarkan angka kecocokannya, dan mulai dari fitur yang paling baik, lalu diuji semua fitur lain pada sisi yang sama antara kamera dan ruang objek. Langkah terakhir ialah menghilangkan outlier menggunakan matrix homografi dalam mode RANSAC yang memungkinkan kesalahan proyeksi ulang hingga 5 piksel [4].

2.2.6. *Target Data Acqusition*

SIFT adalah pendekatan berbasis model yang membutuhkan database fitur untuk dipersiapkan terlebih dahulu. Tracker ini saat ini terbatas pada target planar, oleh karena itu, sebuah citra ortografi dari target pelacakan saja sudah cukup. Akusisi data dilakukan dengan membuat suatu piramida gambar, dimana setiap level skalanya diperkecil dengan faktor sebesar 1.41 dari skala sebelumnya. Skala terbesar dan terkecil digunakan untuk mendefinisikan jangkauan fitur yang dapat dideteksi saat dijalankan.akan, pencocokan citra, pelacakan, mosaicing gambar, panorama stitching, pemodelan 3D dan pengenalan objek.

Pendekatan lain yang digunakan untuk mengekstraksi beberapa jenis fitur dan menyimpulkan isi dari suatu gambar pada umumnya menggunakan deteksi sudut/*corner detection*. *Corner detection* sering digunakan dalam deteksi gerakan, pencocokan citra, pelacakan, mosaicing gambar, panorama stitching, pemodelan 3D dan pengenalan obyek.

Terdapat berbagai metode yang digunakan untuk *Corner Detection* seperti moravec operator dan Harris/Plessey operator. Harris/Plessey Operator merupakan pengembangan lebih lanjut dari Moravec Operator, dimana Harris dan Stephens menggabungkan deteksi sudut dan tepi untuk mengatasi keterbatasan Moravec Operator[7].

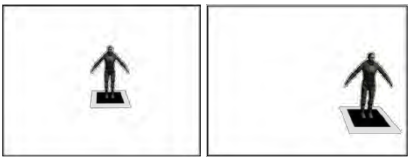


Gambar 2.4 Deteksi sudut[7]

2.3. *Fixed Point Augmented reality*

Menurut Ashari, Fixed point merupakan sebuah titik pada area lingkungan nyata yang tidak berubah-ubah nilainya. Pada augmented reality ini digunakan untuk menentukan posisi objek virtual pada lingkungan nyata[5].

Pada gambar Gambar 2.5 merupakan objek virtual pada augmented reality dengan marker. Posisi objek virtual dapat diletakkan dimana saja sesuai penanda yang digunakan. Pada gambar merupakan objek virtual dengan posisi yang semakin dekat dengan kamera, dan objek virtual mengalami proses scaling sehingga tampak lebih besar dari posisi sebelumnya.



a b

Gambar 2.5 Posisi Objek virtual dengan *marker* a) Posisi *marker* berada didepan kamera b) Posisi objek virtual lebih dekat dengan kamera[5]

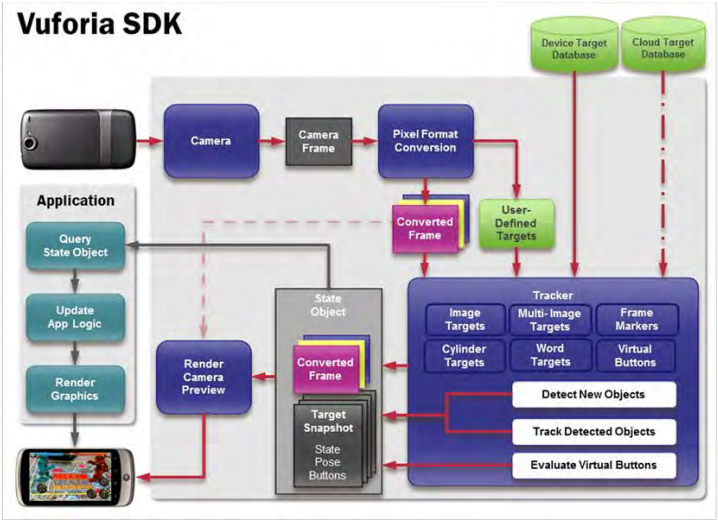
Posisi dari objek tersebut sudah ditentukan pada koordinat x, y dan z dengan nilai yang sudah ditentukan dan tidak bisa berubah-ubah dan relatif terhadap marker. Posisi tersebut terletak pada sebuah titik dimana sebagai acuan objek virtual untuk melakukan perpindahan sesuai skenario yang ditentukan.

2.4. Vuforia *Augmented Reality*

Vuforia adalah sebuah SDK (Software Development Kit) untuk membuat aplikasi berbasis augmented reality yang dikeluarkan oleh Qualcomm. Vuforia menyediakan teknologi visi komputer yang dengan sangat erat menyelaraskan grafis dari sebuah permukaan yang dicetak dengan objek 3D sederhana. Sebuah aplikasi AR berbasis Vuforia SDK dari komponen inti berikut vuforia[7]:

1. Kamera. Memastikan bahwa setiap frame ditangkap dan diteruskan secara efisien untuk dilacak.

2. Image Converter. Format piksel tunggal converter mengkonversi antara format kamera (misalnya YUV12) ke format yang sesuai untuk rendering OpenGL ES (misalnya RGB565) dan untuk pelacakan.
3. Pelacak. Pelacak berisi algoritma visi komputer yang mendeteksi dan melacak objek dunia nyata dalam bingkai kamera video
4. Video Background Renderer. Video Background renderer merender gambar kamera yang tersimpan dalam state objek.
5. Kode aplikasi. Pengembang aplikasi harus menginisialisasi semua komponen di atas dan melakukan tiga langkah kunci dalam kode aplikasi, yaitu:
6. Menanyakan pada state objek tentang target yang baru terdeteksi atau state terbaru dari elemen ini.
7. Memperbarui logika aplikasi dengan input data baru.
8. Merender tampilan grafis yang tertambah.
9. Target resource. Target resource dibuat menggunakan Sistem manajemen target online.



Gambar 2.6 Blok diagram Vuforia SDK[6]

Vuforia SDK akan melacak *trackable* yang merupakan kelas dasar yang mewakili semua objek yang berada di dunia nyata di enam derajat

kebebasan (6DoF). Alur proses yang terjadi pada pelacakan Vuforia dapat dilihat pada Gambar 2.6[6]. Ketika terlacak tiap *trackable* memiliki nama, ID, status dan informasi. Ada beberapa jenis trackable yang dimungkinkan untuk dideteksi oleh Vuforia, *yaitu Image target, Cylinder target, Multi target*.

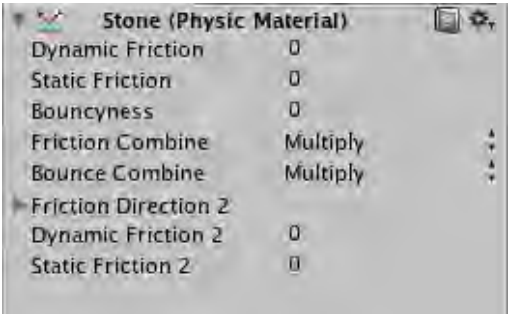
Smart terrain merupakan suatu teknik untuk membangun latar lingkungan nyata yang sesuai berdasarkan posisi dan orientasi benda-benda nyata. *Smart terrain* itu sendiri merupakan gabungan dari permukaan latar dan beberapa objek terdeteksi yang dibangun secara dinamis, seperi bangunan, menara, gedung, dan lain-lain. Vuforia sendiri sudah mengembangkan teknologi *smart terrain* berbasis Augmented reality. Salah satu contohnya ialah permainan untuk menggerakkan seekor pinguin di atas *smart terrain* yang dibentuk secara realtime pada **Gambar 2.7**.

Gambar 2.7 Aplikasi *Smart terrain* untuk menggerakkan penguin[6]

2.5. *Collider*

Komponen *Collider* untuk tujuan tumbukan fisik benda. *Collider*, umumnya tidak terlihat, tidak harus berbentuk sama persis seperti mesh objek. Pada umumnya, *collider* dengan bentuk sederhana dapat digunakan dan sering lebih efisien dalam gameplay. Bentuk *collider* paling sederhana disebut sebagai *collider* primitif. Dalam Unity3D,

collider sederhana ini adalah *Box Collider, Sphere Collider* dan *Capsul Collider*.



Gambar 2.8 Konfigurasi material fisik di Unity3D[8]

Dengan posisi dan ukuran yang baik, *collider coumpoud* yang merupakan gabungan dari beberapa *collider* dapat menyerupai bentuk objek sesungguhnya sekaligus dapat menghemat penggunaan prosesor. Pada kasus tertentu, penggunaan *collider* gabungan masih belum akurat. Sehingga pada Unity3D dapt digunakan Mesh *collider* untuk menggantikannya. *Collider* ini lebih membutuhkan banyak penggunaan prosesor sehingga hanya digunakan apabila dibutuhkan saja.

Pada saat *collider* mengalami interaksi, permukaan *collider* membutuhkkan material fisik untuk mensimulasikan properti material objek yang digunakan. Sebagai contoh, material fisik es akan membuat *collider* lebih licin, sedangkan material karet akan menghasilkan gesekan dan pantulan yang besar. Properti material fisik pada Unity3D dijelaskan pada table Tabel 2.1.

Tabel 2.1 Properti *Physic Material* Unity3D[8]

Properti	Fungsi
Dynamic friction	Gesekan yang digunakan pada saat objek mengalami pergerakan. Nilai umunnya berkisar antara 0-1. Nilai 0 akan membuat benda

Properti	Fungsi
	licin, sedangkan nilai 1 menyebabkan benda susah untuk digerakkan.
Static friction	Gesekan yang digunakan pada saat objek diam diatas permukaan. Nilai umunnya berkisar antara 0-1. Nilai 0 akan membuat benda licin, sedangkan nilai 1 menyebabkan benda susah untuk digerakkan.
Bounciness	Seberapa besar pantulan yang terjadi pada permukaan benda. Nilai 0 akan membuat benda tidak dipantulkan dan nilai 1 membuat benda dipantulkan tanpa kehilangan energy.
Friction Combine	Kombinasi antara gesekan dua buah *collider* benda
Bounce Combine	Seberapa besar kombinasi pantuln antara dua buah *collider* benda
Friction direction 2	Arah gesekan benda
Dynamic Friction 2	Jika arah gesekan di enable, gesekan dinamis akan digunakan dengan arah sesuai dengan friction direction 2
Static Friction 2	Jika arah gesekan di enable, gesekan static benda disesuaikan dengan friction direction 2

Material digunakan dan dihubungkan dengan mesh atau renderer partikel yang melekat pada gameObject. Material berhubungan dengan penyaji Mesh atau partikel yang melekat pada GameObject tersebut. Mesh memainkan bagian penting dalam mendefinisikan bagaimana objek ditampilkan. Mesh atau partikel Tidak dapat ditampilkan Tanpa material karena material meliputi referensi untuk Shader yang digunakan untuk membuat Mesh atau Partikel. Material digunakan untuk menempatkan Tekstur ke GameObjects.

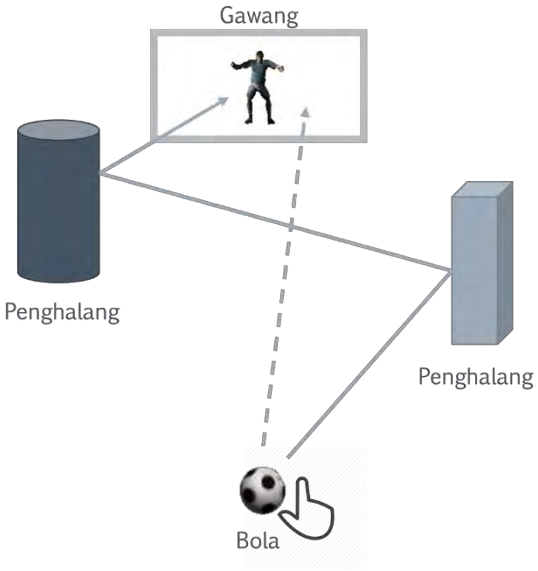
Unity3D mendukung pengembangan aplikasi Android. Sebelum dapat menjalankan aplikasi yang dibuat dengan Unity Android diperlukan adanya pengaturan lingkungan pengembang Android pada perangkat. Untuk itu pengembang perlu men-download dan menginstal SDK Android dan menambahkan perangkat fisik ke sistem. Unity Android memungkinkan pemanggilan fungsi kustom yang ditulis dalam C / C + + secara langsung dan Java secara tidak langsung dari script C #.

Halaman ini sengaja dikosongkan

BAB 3
DESAIN SISTEM DAN IMPLEMENTASI

3.1. Desain Sistem

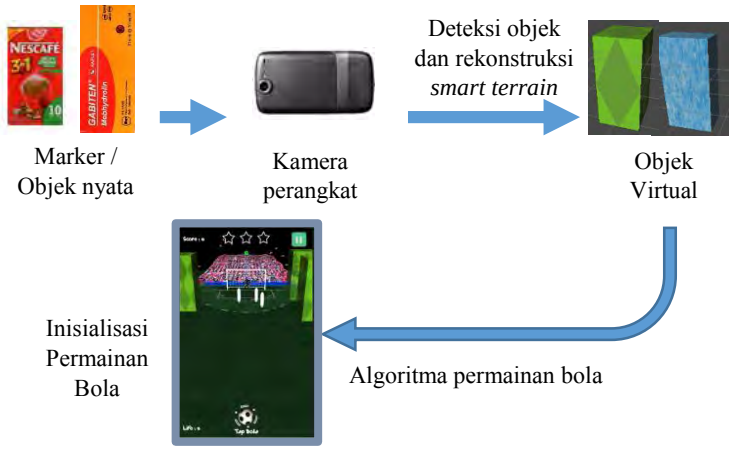
Dalam tugas akhir ini bertujuan untuk membuat aplikasi yang mampu untuk mendapatkan data tiga dimensi dari suatu objek nyata dengan menggunakan kamera. Dari data tiga dimensi yang didapatkan digunakan untuk merekonstruksi *smart terrain* dalam permainan. Kamera perangkat android merupakan sebuah alat visi yang digunakan untuk mengambil data tiap-tiap frame. Dari data frame yang diperoleh akan dikirimkan untuk diproses oleh sistem untuk dideteksi objek didalamnya. Tiap objek yang dideteksi oleh kamera akan digantikan dengan objek virtual yang serupa dengan jenis objek yang didefinisikan sebelumnya untuk membentuk *smart terrain*. Selanjutnya, *smart terrain* hasil rekonstruksi digunakan untuk implementasi permainan bola. Ilustrasi permainan yang akan dibuat digambarkan pada Gambar 3.1.



Gambar 3.1 Ilustrasi Permainan.

Berdasarkan ilustrasi permainan, posisi gawang ditentukan oleh sebuah marker. Sedangkan objek-objek penghalang ditentukan menggunakan teknologi *smart terrain*. Smart terrain digunakan untuk merekonstruksi keseluruhan objek dalam arena permainan. Setelah arena permainan telah selesai dibuat barulah pengguna dapat menggerakkan bola dengan cara memberikan input berupa sentuhan (*gesture*) pada tombol bola.

Pergerakan bola dapat diatur sesuai dengan selera pengguna. Bola dapat diarahkan memantul kepada objek penghalang lalu menuju gawang, ataupun langsung menuju kearah gawang yang dijaga oleh seorang penjaga gawang. Desain sistem secara keseluruhan dijelaskan pada Gambar 3.2.



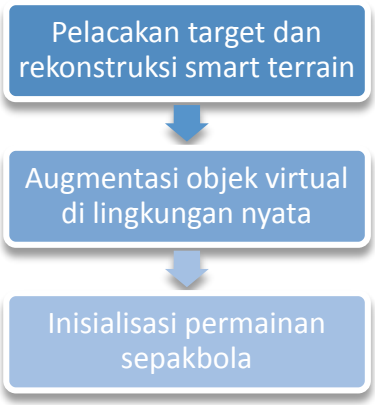
Gambar 3.2 Desain sistem secara umum

3.2. Alur Kerja

Di setiap frame dari citra yang didapatkan oleh kamera terdapat susunan fitur fitur yang merupakan informasi dari nilai citra itu sendiri. Proses deteksi fitur yang dilakukan bertujuan sebagai data masukan untuk pelacakan suatu penanda. Saat penanda telah terdeteksi, penanda ini dijadikan sebagai acuan lokasi *smart terrain* yang akan di rekonstruksi.

Rekonstruksi *smart terrain* didapat dengan cara menggerakkan kamera mendekati objek-objek nyata yang ada disekitarnya. Data-data hasil deteksi objek nyata akan disimpan dalam bentuk titik-titik point koordinat. Titik-titik point koordinat yang telah disimpan dikonversi kedalam bentuk data mesh untuk merender permukaan *smart terrain*.

Perancangan dilakukan dengan menggunakan pustaka Vuforia dari Qualcomm dan di desain pada IDE Unity3D dimana juga sekaligus sebagai prerender grafis untuk objek virtual dimensi tiga. Untuk keseluruhan proses, sistem ini dibagi menjadi tiga tahapan utama, yaitu: pelacakan target dan rekonstruksi *smart terrain*, augmentasi objek virtual dilingkungan nyata, dan Inisialisasi permainan bola. Gambaran alur kerja sistem dijelaskan dalam Gambar 3.3.

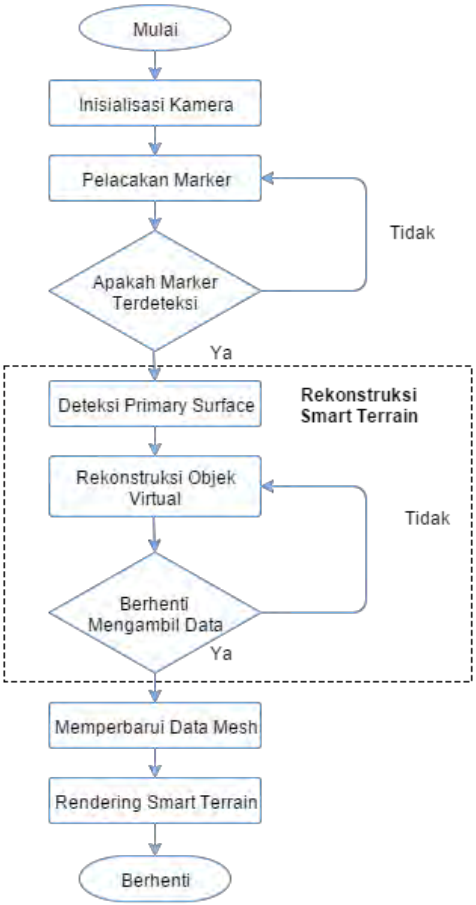


Gambar 3.3 Alur Kerja Sistem

3.3. Pelacakan Target dan Rekonstruksi *Smart terrain*

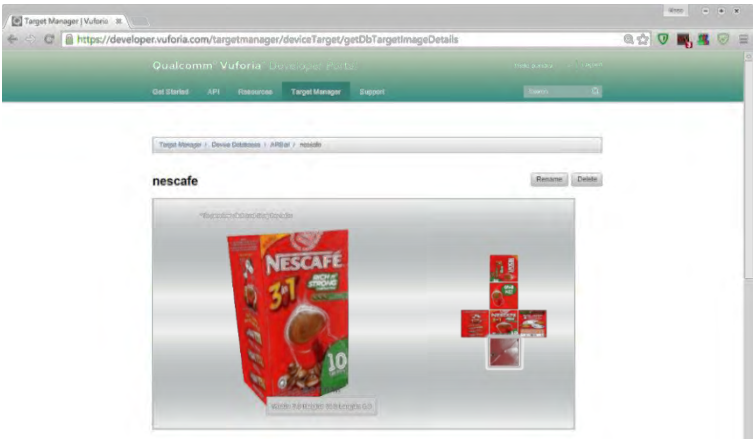
Sebelum memulai pelacakan dan rekonstruksi *smart terrain* seperti yang digambarkan dalam diagram alur sistem pada Gambar 3.4, tahapan pertama yang perlu dilakukan adalah menentukan marker inisialisasi. Marker inisialisasi ini berfungsi sebagai pemicu munculnya objek-objek pada *smart terrain*.

Pada tugas akhir ini marker inisialisasi yang digunakan berupa *multi target*. *Multi target* pada dasarnya merupakan beberapa *image target* yang diatur sehingga menjadi sebuah *trackable* yang terdiri dari beberapa target gambar tersebut. Pada proses pembuatan sebuah *multi target* Augmented Reality dibutuhkan enam buah *image target* yang merupakan enam sisi kotak yang akan dibentuk menjadi *multi target*.



Gambar 3.4 Pelacakan dan rekostruksi *smart terrain*

Untuk mengimplementasikan sebuah *multi target*, cara yang harus dilakukan adalah dengan menentukan *multi target* dengan parameter ukuran sisi-sisi *multi target* yaitu panjang, lebar, serta tingginya, yang disesuaikan dengan dimensi objek di dunia nyata. Setelah itu menentukan kualitas dan kuantitas fitur tiap sisi *multi target*, dan membuat file konfigurasi serta dataset dari *multi target* tersebut. Proses konfigurasi dan pembuatan dataset dilakukan secara online. Pada Gambar 3.5 dapat dilihat tampilan dari sistem manajemen target online untuk membuat sebuah *multi target*.



Gambar 3.5 Konfigurasi *multi target*

3.3.1. Inisialisasi Kamera Perangkat Android

Setelah marker sudah ditentukan, proses selanjutnya adalah inisialisasi kamera perangkat android, yaitu dengan menggunakan bantuan pustaka Vuforia untuk menangkap citra gambar.

```
CameraDevice.Instance.Init(CameraDevice.CameraDirection.C
    AMERA_BACK);
CameraDevice.Instance.SetFocusMode(CameraDevice.FocusMode
    .FOCUS_MODE_CONTINOUSAUTO);
CameraDevice.Instance.Start();
```

Kode 3.1 Inisialisasi kamera perangkat

Inisialisasi kamera dilakukan dengan fungsi start pada CameraDevice. Inisialisasi ini selain untuk melakukan reservasi pada memori juga untuk melakukan pengecekan pada kamera perangkat android. Jika kamera tidak terdeteksi maka akan terlempar exception dengan handler untuk mematikan program. Dengan pustaka Vuforia, penggunaan kamera tidak terbatas hanya pada penggunaan kamera belakang, melainkan juga kamera depan perangkat android. Saat penggunaan kamera tidak diatur, maka oleh Vuforia secara otomatis menggunakan kamera belakang perangkat.

Untuk mendapatkan hasil yang maksimal kamera diatur agar melakukan pelacakan gambar secara kontinyu dan focus. Agar tidak terjadi error dalam hal pelacakan marker. Inisialisasi arah kamera dan pencerminannya dapat dikonfigurasikan dengan mengatur ARCamera pada inspector Unity. Pada ARCamera ini juga dataset marker *multi target* yang telah ditentukan di*load* dan diaktifkan agar dapat dilakukan proses pelacakan.

3.3.2. Pelacakan Penanda *Multi Target*

```
TrackerManager.Instance.GetTracker<ImageTracker>().
    Start();
```

Kode 3.2 Inisialisasi tracker citra

Fungsi start di TrackerManager digunakan untuk memulai proses pelacakan marker *multi target* yang telah ditentukan. TrackerManager ini akan mendeteksi setiap fitur yang terdapat pada frame-frame citra yang ditangkap oleh kamera. Saat marker terdeteksi, TrackerManager ini akan mengirimkan handler berupa definisi marker yang ditemukan dan letak koordinatnya.

Gambar 3.6 adalah contoh objek virtual gawang yang ditampilkan saat *marker* terdeteksi. *Marker* yang telah terdeteksi, posisi koordinatnya akan digunakan untuk memulai pelacakan *smart terrain*. Pelacakan terrain ini meliputi permukaan latar *(surface)* dan objek *(props)* yang berada di dunia nyata. Hal ini dilakukan dengan cara menggerakkan posisi kamera disekitar marker dan objek nyata yang akan direkonstruksi.

Gambar 3.6 Objek virtual gawang ditampilkan saat marker terdeteksi

3.3.3. Rekonstruksi *Smart terrain*

Proses pelacakan *smart terrain* ini dilakukan berdasarkan beberapa tahapan. Proses pertama adalah kalibrasi posisi kamera antara letak kamera di dunia nyata terhadap *marker* inisialisasi. Proses kalibrasi dilakukan dengan cara mencari titik fitur dari tiap frame yang sama dengan fitur-fitur yang tersimpan di dalam *marker*. Saat *marker* terlacak, didapatkan informasi mengenai posisi dan rotasi kamera sebenarnya di dunia nyata. Selanjutnya, dilakukan proyeksi matrix terhadap objek-objek lain di sekitar marker.

```
SmartTerrainTracker SmartTerrain = TrackerManager.
    Instance.GetTracker<SmartTerrainTracker>();
SmartTerrain.Start ();
SmartTerrain.StartMeshUpdates ();
```

Kode 3.3 Pelacakan *smart terrain*

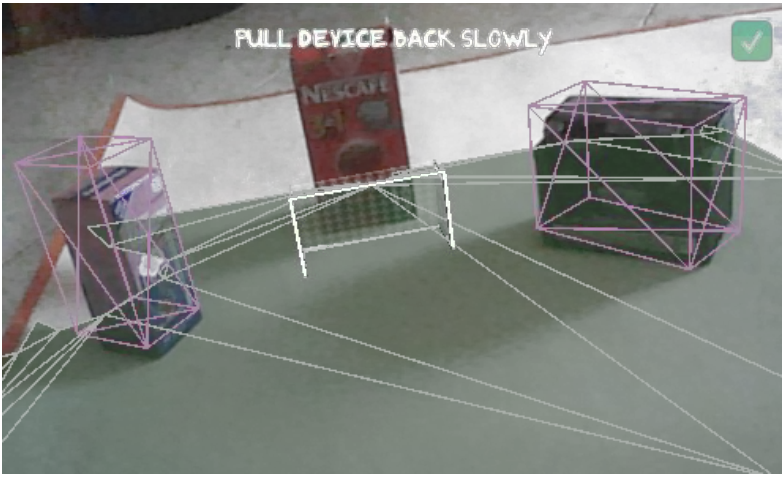
Objek-objek yang terdeteksi saat dilakukan proses pelacakan tracker SmartTerrain akan digunakan untuk memperbarui data mesh 3D untuk rekonstruksi terrain dan disimpan kedalam memory. Data mesh permukaan hasil rekonstruksi berupa latar planar, sedangkan objek-objek nyata diubah sesuai dengan model yang telah ditentukan sebelumnya.

Dalam Tugas Akhir ini, model prop objek yang digunakan adalah *cube*, *cylinder*, *capsule* dan 3d mesh. Dikarenakan fitur *smart terrain* dengan pustakan vuforia tidak dapat membedakan bentuk objek di dunia nyata, maka objek-objek nyata tersebut didefinisikan dengan cara menggunakan objek yang menyerupainya.

```
SmartTerrain.StopMeshUpdates ();
SmartTerrain.Stop ();
```

Kode 3.4 Menghentikan proses update rekonstruksi *smart terrain*

Saat pelacakan usai tracker SmartTerrain perlu dihentikan dengan menekan tombol cek, karena jika tidak dihentikan kemungkinan akan terjadi kesalahan rekonstruksi ataupun overload pada memory. Kode yang digunakan untuk menghentikan update data mesh ditunjukkan pada Kode 3.4. Meskipun update data mesh dihentikan proses tracking terrain tetap ada. Data mesh rekonstruksi terrain akan disimpan dalam memory dan akan digunakan sebagai titik acuan munculnya objek virtual permainan bola.



Gambar 3.7 Rekonstruksi data mesh terhadap objek

Pada Gambar 3.7 ditampilkan proses rekonstruksi *smart terrain* dari dua buah objek berupa balok. Pada gambar ditunjukkan bahwa latar rekonstruksi berwarna hijau dengan garis-garis *triangle mesh* disekitarnya. Sedangkan pada objek-objek nyata, dilingkupi oleh garis-garis triangle mesh berupa balok yang memiliki warna merah muda.

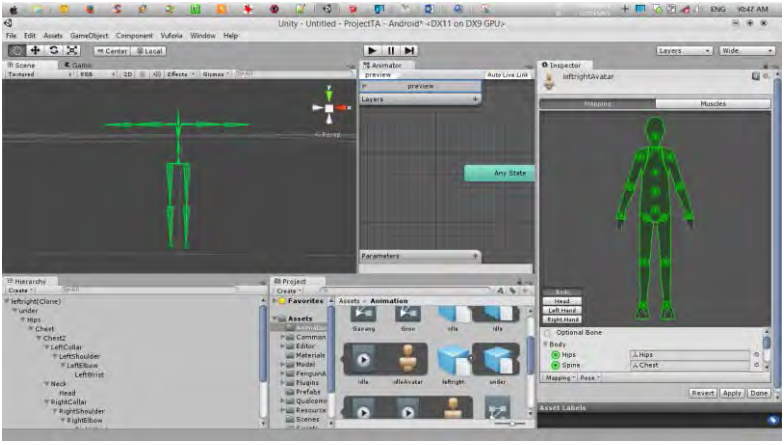
3.4. Augmentasi Objek Virtual Dilingkungan Nyata

Perancangan untuk objek Augmented Reality yang didesain pada perangkat lunak memiliki ukuran yang sebanding dengan objek sebenarnya pada dunia nyata. Ukuran dan lokasi harus ditentukan terlebih dahulu selama pembuatan objek virtual yang ditampilkan. Agar memiliki tingkat realitas yang tinggi, objek nyata yang ditampilkan terletak pada *scene* yang sama dan memperoleh pencahayaan yang sebanding dengan objek virtualnya.

Parameter ukuran antara objek nyata dengan objek virtual sangat penting diperhatikan agar semua informasi yang diambil maupun yang dikembalikan selama proses pengambilan citra berada di skala yang sama. Karena objek virtual yang digunakan dan ditambahkan pada dunia nyata harus memiliki tingkat realitas yang tinggi, sehingga seolah-olah tidak ada perbedaan dengan objek aslinya. Dimisalkan, apabila objek nyata A berada pada jarak 100 unit dari kamera, dengan proses skala 4 kali diperkecil. Objek nyata sebagai acuan untuk objek virtual memiliki ukuran setengah dari objek A, maka untuk ukuran objek virtual B harus memiliki perbandingan yang sama dengan aslinya, yaitu setengah dari objek nyata A. Dimana sumbu (x, y) sebagai acuan ukuran dari objek virtual dalam unit adegan dengan diukur dari sumbu horisontal dan vertikal dari objek nyata.

Pada implementasi *augmented reality smart terrain* ini menggunakan beberapa model yang diperlukan untuk permainan bola, seperti gawang, penjaga gawang, musuh dan bola beserta tekstur penonton untuk menambahkan tingkat realitas permainan. Objek-objek virtual ini, perlu ditentukan posisinya relative terlebih terhadap SmartTerrain yang telah direkonstruksi.

Model model virtual ini dirancang untuk melakukan beberapa gerakan dan perpindahan yang sudah didefinisikan sepanjang sumbu x, y, z. Salah satunya adalah model penjaga gawang, model ini harus memiliki beberapa animasi sesuai format Unity3D yang siap digunakan untuk melakukan beberapa gerakan. Sehingga perlu dilakukan konversi format dari motion capture data berupa bvh (BiovisionHierarchi) menjadi animasi yang dapat digunakan oleh Unity3D. Proses konversi ini dilakukan dengan bantuan tools bvhacker dan blender.

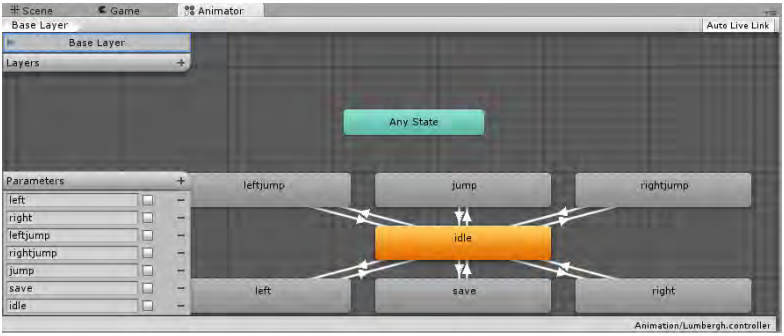


Gambar 3.8 Mapping dan rigging model

Sebelum di import ke Unity3D, file bvh terlebih dahulu dilakukan proses cleanup dengan cara membersihkan offset skeleton dan memotong frame yang tidak diperlukan. File animasi yang telah di import terlebih dahulu dilakukan proses *retargeting bones* animasi dengan model penjaga gawang yang akan di gunakan seperti pada Gambar 3.8.

Untuk animasi pergerakan perpindahan pada objek virtual menggunakan Lumbergh (Penjaga gawang). Pada animasi utama, yaitu animasi karakter dari objek vitual dimisalkan *idle* untuk proses awalnya dideskripsikan ke layer. Beberapa animasi utama yang digunakan pada objek virtual ini diantaranya adalah: *idle*, *save*, *jump*, *rightjump*, *right*, *leftjump*, *left*. Animasi yang dijalankan harus berjalan secara berurutan

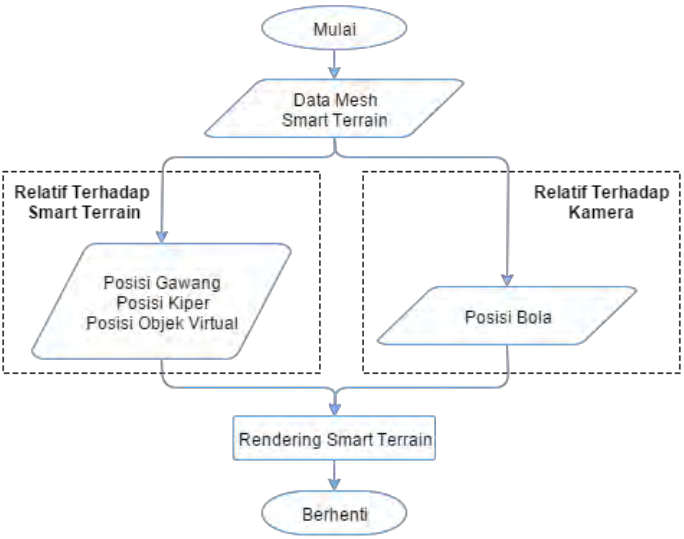
sesuai dengan state machine pada Gambar 3.9. Untuk pergerakan animasi penjaga gawang itu sendiri dapat dilihat pada Gambar 3.10. Dimana penjaga gawang dapat bergerak ke enam arah setelah melalui proses *idle*.



Gambar 3.9 State Machine model penjaga gawang

Selain model penjaga gawang, digunakan juga model lain seperti *cube*, *cylinder*, *capsule* dan *3d mesh* untuk mendeteksi terjadinya benturan pada bola. Tiap-tiap model tersebut memiliki besar nilai pantul tersendiri seperti pada material karet, es, kayu dan besi. Sehingga arah pantulan bola akan lebih bervariasi.

Gambar 3.10 Gerakan animasi penjaga gawang

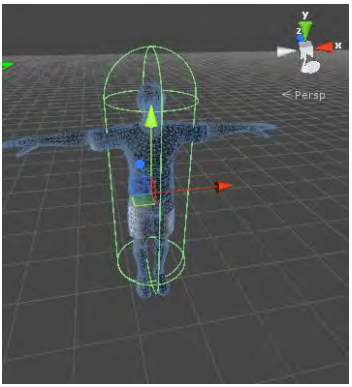


Gambar 3.11 Alur diagram menampilkan objek

Alur diagram untuk menampilkan objek virtual permainan pada aplikasi augmented reality dapat dilihat pada Gambar 3.11. Alur kerja untuk menampilkan objek permainan itu sendiri adalah sebagai berikut:

1. Mengambil data mesh yang didapatkan dari hasil rekonstruksi *smart terrain*.
2. Memberikan nilai posisi pada objek permainan sesuai dengan area *smart terrain*. Pada objek permainan disisipkan juga sebuah collider untuk mendeteksi adanya tumbukan yang terjadi pada bola.
3. Memasukkan animasi yang digunakan objek virtual. Animasi sudah dideskripsikan pada program dan bisa digunakan pada waktu yang sudah ditentukan
4. Mengolah animasi dari objek mainan virtual, dengan beberapa animasi yang digunakan supaya animasi objek virtual berjalan sesuai dengan posisi perpindahan yang sudah ditentukan
5. Model virtual di render diatas frame video
6. Hasil akhir ditampilkan kembali pada layar

Selain penempatan objek-objek virtual, ditentukan pula *collider* sebagai deteksi adanya tumbukan terhadap bola yang akan digerakkan. Penempatan *collider* pada penjaga gawang ditampilkan pada Gambar 3.12. *Collider* yang digunakan untuk mendeteksi pergerakan penjaga gawang menggunakan *collider* sederhana berupa *capsule collider*, dimana penggunaan *capsule collider* ini akan menghemat terjadinya komputasi terhadap tumbukan dibandingkan dengan menggunakan *mesh collider*, terutama apabila objek tersebut bergerak. Sehingga permainan tidak terlalu berat untuk dimainkan.



Gambar 3.12 Penempatan *collider* pada penjaga gawang

3.5. Rendering dan Inisialisasi Permainan Bola

Konsep inti dari permainan bola pada tugas akhir ini adalah dengan cara menggerakkan bola menuju ke gawang diatas latar *smart terrain*. Dimana pada *smart terrain* ini terdapat objek objek virtual yang disesuaikan dengan dunia nyata yang.berfungsi untuk membantu memantulkan bola ataupun menghalanginya. Setelah didapatkan data mesh *smart terrain*, dilakukan proses untuk mendapatkan input gerakan dari pengguna. Karena pada *augmented reality* posisi kamera akan berubah-ubah, bola virtual digunakan untuk mendeteksi input gerakan. Perlu dilakukan proses raycasting untuk mengetahui lokasi bola sesungguhnya pada latar *smart terrain* agar bola tidak melayang diatas latar. Kode yang digunakan untuk menentukan posisi bola dilatar *smart terrain* ditunjukkan pada Kode 3.5.

```
RaycastHit[] Hits;
Ray touchRay=Camera.main.ScreenPointToRay(Touch.
    position);
Hits=Physics.RaycastAll(touchRay);
Foreach(RaycastHit h in hits){
    String hitname=h.collider.gameObject.name;
    If(hitname==”Plane”){
      Vector3 touchStart=h.point;
    }
}
```

Kode 3.5 Inisial posisi transformasi bola terhadap latar *smart terrain*

Arah pergerakan bola itu sendiri ditentukan berdasarkan referensi titik awal dan titik tujuan yang disentuh oleh pengguna. Kode yang digunakan untuk menentukan input pergerakan bola ditunjukkan pada Kode 3.6. Input pergerakan bola didapatkan dengan cara membandingkan vektor titik akhir yang disentuh oleh pengguna dengan titik awal transformasi. Perbandingan ini digunakan untuk menentukan besarnya gaya yang diberikan kepada bola.

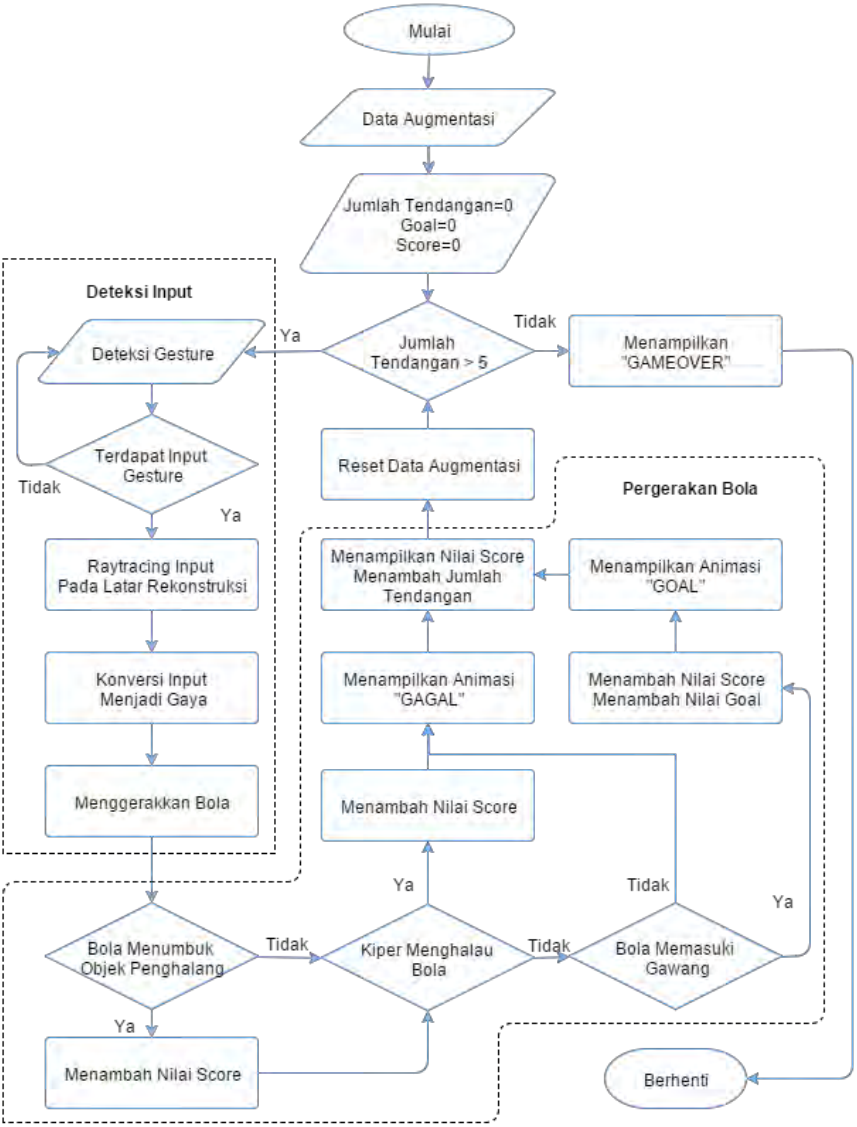
Bola bergerak sesuai dengan arah vektor gaya yang diberikan. Apabila bola tersebut mengalami proses tumbukan dengan objek virtual hasil rekonstruksi maka bola tersebut akan dipantulkan dengan besarnya unsur pantulan objek virtual itu dengan vektor normal.

```
RaycastHit h;
Ray touchRay = Camera.main.ScreenPointToRay(Touch.
    position);
if (Physics.Raycast (touchRay, out h)) {
    Vector3 touchEnd=h.point;
}
Vector3 force=touchEnd-touchStart;
rigidbody.AddForce(force*power);
```

Kode 3.6 Input besar gaya pada bola.

$$Gaya\ pada\ bola_{\overline{x,y,z}} = massa\ bola \times percepatan\ bola_{\overline{x,y,z}} \ldots (3.1)$$

$$Percepatan\ bola_{\overline{x,y,z}} = \left(Pawal_{\overline{x,y,z}} - Pakhir_{\overline{x,y,z}}\right) \times waktu^2 \ldots (3.2)$$



Gambar 3.13 Alur diagram permainan bola

Alur kerja diagram permainan bola digambarkan pada gambar Gambar 3.13 adalah sebagai berikut:

1. Input data augmentasi berupa posisi bola, gawang, objek penghalang, penjaga gawang dan animasinya beserta *collider* yang dimiliki objek tersebut.
2. Inisialisasi awal permainan, meliputi jumlah tendangan, nilai score dan goal.
3. Saat jumlah tendangan masih dalam jangkauan *threshold* dilakukan pendeteksian adanya input sentuhan dari pengguna. Dalam hal ini *threshold* yang digunakan adalah batasan jumlah tendangan yang diambil maksimal sebesar lima kali.
4. *Raytracing* input sentuhan dari pengguna terhadap latar rekosntruksi, dimana input sentuhan dari pengguna berupa titik awal dan titik akhir saat sentuhan dilepaskan.
5. Perhitungan gaya yang akan diberikan pada bola berdasarkan perbandingan titik akhir dan titik awal hasil dari *raytracing* input. Perhitungan gaya dijelaskan pada persamaan 3.2 yaitu dengan cara mengurangi vektor titik akhir sentuhan dengan titik awal. Nilai vektor tersebut kemudian akan dikalikan dengan massa yang dimiliki bola dan waktu tempuh kuadrat pergerakan bola.
6. Bola akan bergerak sesuai dengan gaya hasil perhitungan. Saat bola bergerak, bola akan mendeteksi adanya tumbukan terhadap objek penghalang virtual hasil rekonstruksi. Jika tidak terdapat tumbukan dilanjutkan ke tahap 8.
7. Apabila bola melewati *collider* objek-objek penghalang maka bola tersebut akan dipantulkan sesuai dengan material fisik yang dimiliki penghalang. Saat tumbukan ini terjadi, dilakukan juga proses penambahan nilai *score*.
8. Bola yang tetap bergerak akan mendeteksi adanya tumbukan dari pergerakan penjaga gawang. Dimana alur pergerakan penjaga gawang diatur secara acak di 6 titik pada gawang. Kondisi *idle* penjaga gawang merupakan animasi dimainkan dimana bola belum mendapatkan gaya, setelah bola dilepaskan maka akan dilakukan pemilihan acak pergerakan animasi penjaga gawang.

Adapun apabila penjaga gawang mampu menangkis menghalau bola, maka akan dilakukan penambahan nilai score.

9. Jika dari arah pergerakan pantulan bola berhasil memasuki gawang maka akan ditampilkan animasi goal dan dilakukan penambahan nilai *score* dan *goal*.
10. Dalam sekali siklus tendangan yang terjadi dibatasi oleh waktu selama 6 detik dimana jika bola tidak memasuki gawang akan ditampilkan animasi gagal. Perhitungan siklus ini dimulai saat bola mendapatkan gaya input.
11. Saat waktu 6 detik siklus usai, posisi dan orientasi bola serta penjaga gawang dikembalikan lagi ke posisi semula. Pada penjaga gawang juga diatur ulang animasi pergerakan idle. Selain itu terjadi penambahan berapa banyak jumlah tendangan yang telah dilakukan.
12. Ketika jumlah tendangan masih dalam batas jangkauan threshold maka akan diulangi lagi proses tahapan ke 3. Sebaliknya, saat jumlah tendangan melewati batas *threshold* yang telah diatur, permainan akan dihentikan dengan menampilkan gameover beserta hasil akhir yang diperoleh.

Halaman ini sengaja dikosongkan

BAB 4
PENGUJIAN DAN ANALISA

Pada Bab ini dibahas mengenai pengujian dari software dan hardware yang telah diimplementasikan untuk mengetahui apakah fungsi dari sistem yang direncanakan telah bekerja sesuai dengan rancangan.

4.1. Implementasi Perangkat Keras

Pengujian dilakukan dengan sistem operasi Windows 8 dengan menggunakan pustaka Vuforia. Integrated Development Environment (IDE) yang digunakan adalah Unity3D untuk merender dan menganalisa hasil pengujian. Untuk Komputer yang digunakan dalam memprogram dan merancang aplikasi *augmented reality* ini, memiliki spesifikasi sebagai berikut:

Tabel 4.1 Spesifikasi Komputer

Komponen	***Spesifikasi***
Sistem Operasi	*Windows 8 Profesional 32-bit*
Processor	*AMD C-50 @1 GHz (2CPUs)*
Memory	*2048MB RAM*
Versi DirectX	*DirectX 11*
Versi DxDiag	*6.03.9600.16384 32-bit Unicode*
Display Adapter Name	*AMD Radeon HD 6250 Graphics*
Total Display Memory	*1013MB*

Adapun spesifikasi perangkat android yang digunakan dalam pengujian augmented reality ini dijelaskan pada Tabel 4.2.

Tabel 4.2 Spesifikasi Perangkat Android

Komponen	***Spesifikasi***
Sistem Operasi	*Android Kitkat 4.4.4*
Model	*Sony Experia Tablet Z*
Processor	*Qualcomm MSM8974AB Snapdragon 801 Quadcore 2.3 GHz Krait 400 Adreno 330*
Memory	*3GB RAM*
Kamera	*Built in Sony Experia Tablet Z2*
Sensor gambar	*2MP Front Camera 8.1 MP Rear Cam*
Display	*1200 x 1920 pixels, 10.1 inches (224 ppi pixel density)*

4.2. Pengujian Sistem

Pada pengujian sistem, hal yang diamati pertama kali adalah apakah permainan ini telah bekerja sesuai dengan desain sistem yang diinginkan. Selain itu pengujian sistem dilakukan dengan melakukan simulasi terhadap *smart terrain* hasil rekonstruksi. Diantaranya ialah pengujian terhadap fungsi-fungsi yang terdapat dalam permainan. Selain itu dilakukan pengujian tentang deteksi dan rekonstruksi latar serta letak objek virtual dari berbagai sisi marker, deteksi berbagai jenis objek nyata, perbandingan ukuran hasil rekonstruksi dengan ukuran dunia nyata dan yang terakhir pengujian skenario permainan bola.

4.2.1. Pengujian Fungsionalitas Permainan

Pengujian ini dilakukan dengan mencoba seluruh proses dalam permainan dan mencoba fungsi-fungsi yang ada apakah telah berjalan dengan baik atau tidak. Hasil dari pengujian seluruh fungsi berjalan dengan sesuai seperti pada desain sistem. Hasil yang didapatkan dari pengujian kesesuaian sistem permainan ini terdapat pada Tabel 4.3. Dari tabel dapat dilihat bahwa keseluruhan sistem permainan telah dapat berjalan dengan baik.

Tabel 4.3 Hasil pengujian fungsi

No	Nama Jenis Fungsi	Fungsi dapat berjalan
1	Inisialisasi kamera perangkat	Ya
2	Pelacakan marker inisialisasi	Ya
3	Rekonstruksi *smart terrain*	Ya
4	Tombol penggerak bola	Ya
5	Tumbukan penghalang	Ya
6	Pergerakan penjaga gawang	Ya
7	Deteksi bola memasuki gawang	Ya
8	Penilaian *score* dan *goal*	Ya
9	Kondisi pause	Ya
10	Kondisi selesai	Ya

4.2.2. Pengujian Marker

Dari sistem yang telah dibuat dilakukan analisa terhadap *marker* untuk dapat menentukan rekonstruksi latar dari objek virtual yang ideal. Pengujian yang dilakukan ialah dengan cara menguji berbagai sisi *marker* inisialisasi yang digunakan serta pengujian kemampuan deteksi kamera terhadap marker dengan jarak tertentu.

4.2.2.1. Pengujian Berbagai Sisi Marker

Pengujian dilakukan untuk mengukur keberhasilan *multi target* menampilkan objek virtual dimana bila sisi *multi target* yang terlihat terbatas. Hal ini dilakukan dengan cara menguji kemampuan fitur pada salah satu sisi *multi target* untuk menampilkan objek virtual. Pengujian ini juga dilakukan untuk melihat pengaruh posisi objek apabila hanya ada

satu sisi *multi target* terlihat. Hasil pengambilan citra kamera disetiap sisi marker target ditampilkan pada Tabel 4.4.

Tabel 4.4 Pengujian tiap sisi *multi target*

Sisi	Gambar	Posisi Gawang
Atas		Pada sisi depan menghadap kearah depan kotak
Bawah		Objek tidak terlacak
Kanan		Pada sisi depan menghadap kearah depan kotak
Kiri		Pada sisi depan menghadap kearah depan kotak

Sisi	Gambar	Posisi Gawang
Depan		Pada sisi depan menghadap kearah depan kotak
Belakang		Objek tidak terlacak

Pada Tabel 4.4 terlihat bahwa *multi target* dapat ditampilkan walaupun yang terlacak oleh kamera hanya satu sisi saja. Hanya saja, objek tidak tampil pada bagian bawah dan belakang kotak karena fitur alami dari bagian bawah kotak tidak mencukupi untuk dilacak sebagai *trackable*. Hal ini terjadi karena bagian bawah kotak hanya berupa planar polos yang tidak memiliki tekstur sudut yang dibutuhkan untuk melacak objek.

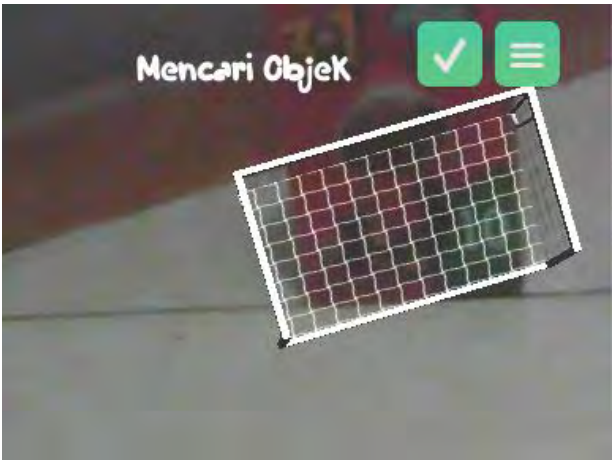
Pada tabel tersebut juga dapat dilihat setiap sisi *multi target* memunculkan objek tetap berada pada posisi tutup kotak dan menghadap ke arah depan kotak sesuai koordinat asli *multi target*. Hal ini terjadi karena pada *multi target* permainan, masing-masing *image target* pembentuk *multi target* dikonfigurasi dengan melakukan translasi dan rotasi pada *image target* sehingga tiap-tiap sisi *multi target* yang berada pada posisi berbeda akan tetap dianggap sejajar oleh sistem. Sehingga objek yang muncul akan sesuai dengan koordinat asal *multi target*.

Dari data pada Tabel 4.4 dapat disimpulkan bahwa penanda berupa *multi target* sudah ideal apabila digunakan untuk proses inisialisasi *smart terrain*. Dengan penanda berupa *multi target* permainan bola dapat dilakukan diberbagai sudut pandang, sehingga pengguna dapat secara bebas menentukan posisi bola yang akan dimainkan. Baik posisi bola

tersebut berada didepan *marker*, disamping maupun berada dibelakangnya. Berbeda halnya dengan penggunaan sebuah *image target*, dimana peletakan bola dalam permainan hanya dapat dilokasikan didepan *marker*.

4.2.2.2. Pengujian Jarak Antara Marker Dengan Kamera

Pengujian ini dilakukan untuk mengetahui posisi ideal marker bila dibandingkan dengan posisi kamera. Karena apabila marker tidak terdeteksi akan menyebabkan gagalnya inisialisasi *smart terrain* dan mengurangi nilai ke interaktifan ataupun bahkan kegagalan dalam inisialisasi permainan bola. Pengujian ini dilakukan pada sisi depan penanda, galat dalam pendeteksian *marker* ditunjukkan pada Gambar 4.1



Gambar 4.1 Posisi dan rotasi gawang saat jarak antara marker dengan kamera sebesar 50 cm

Pada Tabel 4.5 dapat diketahui bahwa marker dengan dimensi ukuran 13x7x6 cm^3 dan jarak lebih besar dari 50 cm dari kamera menyebabkan proses pelacakan menjadi tidak stabil. Sehingga untuk mengimplementasikan rekonstruksi *smart terrain* dengan ukuran *marker* tersebut idealnya berada pada jangkauan 10 cm hingga 50 cm dari kamera.

Tabel 4.5 Pengujian jarak kamera dengan marker

No	Jarak (cm)	Keterangan
1	10 cm	Marker terlacak
2	20 cm	Marker terlacak
3	30 cm	Marker terlacak
4	40 cm	Marker terlacak
5	50 cm	Marker terlacak tetapi posisi gawang tidak stabil
6	60 cm	Marker terlacak tetapi posisi gawang tidak stabil
7	70 cm	Marker tidak terlacak
8	80 cm	Marker tidak terlacak

4.2.3. Pengujian Rekonstruksi *Smart terrain*

Pengujian ini dilakukan untuk mengetahui apakah sistem mampu melakukan proses rekonstruksi yang sesuai dengan kondisi objek nyata. Baik berupa besarnya perbandingan ukuran antara objek nyata dengan hasil rekonstruksi dan lamanya waktu yang diperlukan untuk melakukan rekonstruksi objek dengan jumlah tertentu.

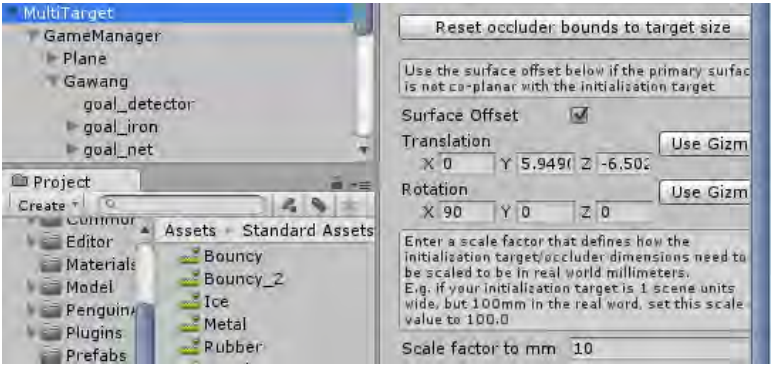
4.2.3.1. Pengujian Ukuran Hasil Rekonstruksi

Berdasarkan hasil perbandingan di Tabel 4.6, dapat dihitung rasio persentasi ukuran asli objek di dunia nyata dengan ukuran hasil rekonstruksi objek dengan *smart terrain*. Hal ini dilakukan dengan cara membandingkan volume antara objek nyata dengan hasil rekonstruksi. Keberhasilan rekonstruksi *smart terrain* yang baik hampir sepenuhnya bergantung pada kalibrasi kamera terhadap kondisi lingkungan nyata. Agar mendapatkan hasil yang maksimal kalibrasi dilakukan hingga objek nyata terlingkupi oleh data *mesh* objek virtual. Pada Tabel 4.6 didapatkan bahwa data *mesh* pada objek 1, 2 dan 3 hampir tepat melingkupi objek.

Sedangkan pada objek 4 terjadi kurangnya proses *training area*, hanya setengah dari objek nyata, sehingga rasio perhitungan perbandingan ukuran objek pun kurang maksimal. Akibatnya ukuran hasil rekonstruksi objek 4 tidak sesuai dengan yang diharapkan.

$$R_{x,y,z} = M_{x,y,z} \times \frac{VR_{x,y,z}}{VM_{x,y,z}} \quad \text{... (4.1)}$$

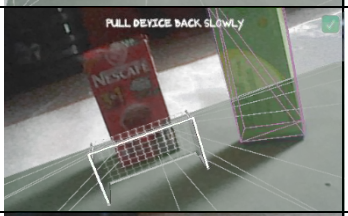
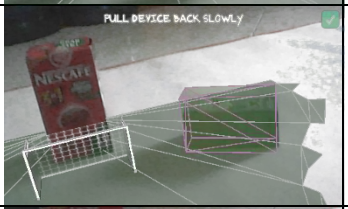
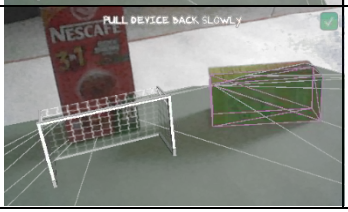
R = Ukuran rekonstruksi
VR = Ukuran virtual rekonstruksi
M = Ukuran marker
VM = Ukuran virtual marker



Gambar 4.2 Konfigurasi skala ukuran dimensi marker, satu unit pada Unity3D adalah sebesar 10mm didunia nyata.

Untuk menghitung besarnya ukuran rekonstruksi dijelaskan pada persamaan 4.1. Karena besar unit marker pada dimensi Unity3D dan ukuran sebenarnya diketahui, maka dapat dihitung ukuran hasil rekonstruksi dengan cara membandingkan ukuran marker dengan ukuran objek di dimensi permainan dengan ukuran marker. Perbandingan ukuran dimensi ini ditampilkan pada Gambar 4.2 dengan menggunakan fungsi *scale factor to mm.* Pada gambar tersebut digunakan nila sebesar 10, yang berarti bahwa setiap nilai satu unit yang dimiliki oleh objek pada dimensi virtual memiliki ukuran sebesar 10mm atau 1cm didunia nyata. Dengan perbandingan inilah dapat dihitung berapakah besarnya ukuran hasil rekonstruksi objek penghalang.

Tabel 4.6 Pengujian ukuran asli dengan hasil rekonstruksi

No	Gambar	Ukuran Asli (pxlxt)(cm)	Ukuran Rekonstruksi (pxlxt)(cm)
1		11x6.7x8.6	11x6.9x8.8
2		6.5x3.0x15	6.5x3.4x14.4
3		10.5x4.5x4.5	10.0x3.2x6.6
4		10.5x4.5x4.5	10.0x3.2x4.3

Perhitungan perbandingan ukuran hasil rekonstruksi ditunjukkan pada Tabel 4.7. Pada tabel tersebut rasio perbandingan volume yang didapatkan dari rekonstruksi *smart terrain* terbaik didapatkan pada objek 1 dan objek 2 yaitu sebesar 94.863% dan 91.2%. Pada objek ke 3, meskipun besar perbandingan volume diatas 99% atau mendekati jumlah yang sebenarnya, terdapat kesalahan dari perbandingan ukurannya dimana objek rekonstruksi memiliki selisih lebar sebesar kurang dari

1.3cm objek asli sedangkan ketinggian yang dimilikinya melebihi dari ukuran objek asli yaitu sebesar 2.1cm. Oleh sebab itu, volume ukuran objek belum dapat digunakan untuk membandingkan ukuran sebenarnya yang dimili oleh benda. Walaupun volume yang dimiliki menyerupai volume sebenarnya, tetapi justru *error* yang dimiliki lebih besar. Sedangkan pada objek ke 4, galat terjadi pada lebar objek yang berselisih sebesar 1.3cm, sehingga ukuran volume yang didapatkan juga tidak sesuai dengan volume sebenarnya. Perhitungan yang dilakukan untuk menghitung rasio perbandingan volume hasil rekonstruksi dengan volume objek nyata ditampilkan pada persamaan 4.2.

$$Rasio\ Volume = \left(V_{asli} - \left|\frac{V_{asli} - V_{rekonstruksi}}{V_{asli}}\right|\right) x\ 100\% \qquad \text{... (4.2)}$$

Tabel 4.7 Perbandingan rasio hasil rekonstruksi

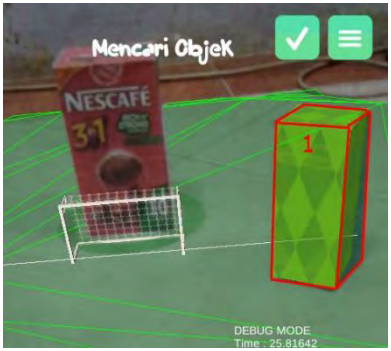
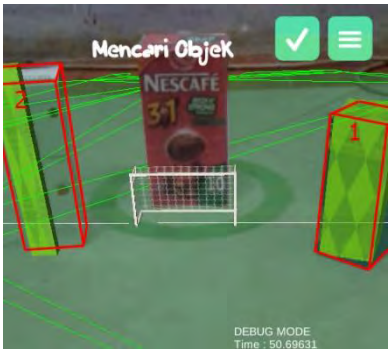
No	Ukuran Asli (pxlxt)(cm)	Ukuran Rekonstruksi (pxlxt)(cm)	Perbandingan Galat Ukuran (pxlxt)(cm)	Rasio volume (cm^3)
1	11x6.7x8.6	11x6.9x8.8	0x0.2x0.2	94.863%
2	6.5x3.0x15	6.5x3.4x14.4	0x0.4x0.6	91.200%
3	10.5x4.5x4.5	10.0x3.2x6.6	0.5x1.3x2.1	99.329%
4	10.5x4.5x4.5	10.0x3.2x4.3	0.5x1.3x0.2	64.714%

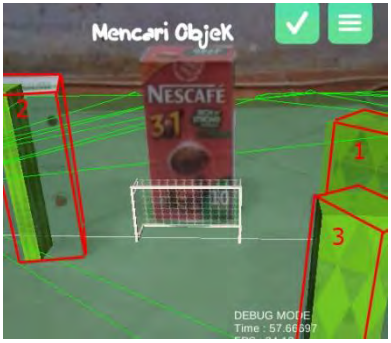
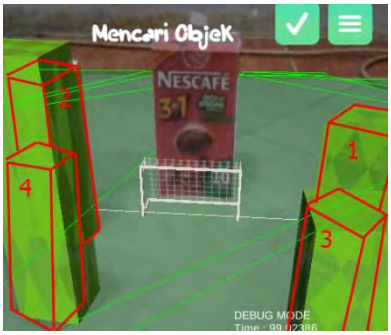
4.2.3.2. Pengujian Banyaknya Objek Dengan Lama Waktu Rekonstruksi

Pengujian yang selanjutnya dilakukan ialah menguji berapa lama waktu yang dibutuhkan untuk merekonstruksi objek-objek dilingkungan nyata. Dalam pengujian ini dilakukan proses rekonstruksi terhadap beberapa objek yang memiliki ukuran berbeda-beda. Waktu pengujian dihitung mulai dari saat marker inisialisasi telah terdeteksi hingga pengguna merasa cukup dilakukannya proses rekonstrusi. Hasil pengujian ini ditampilkan pada Tabel 4.8.

Pada Tabel 4.8 didapatkan bahwa untuk merekonstruksi ulang sebuah objek yang memiliki dimensi sebesar 4x4x10.5 cm^3 dibutuhkan waktu paling sedikit yaitu sebanyak 25.81 detik. Sedangkan waktu paling banyak sebesar 103.46 detik dibutuhkan untuk rekonstruksi 5 objek. Dengan jumlah objek penghalang sebanyak 5, proses rekonstruksi yang dilakukan membutuhkan waktu paling lama. Selain itu hasil rekonstruksi objek tidak sempurna. Dimana dua buah objek yang saling berhimpitan dianggap sebanyak satu objek.

Tabel 4.8 Waktu yang diperlukan untuk merekonstruksi objek nyata

No	Jumlah Objek Nyata	Waktu yang diperlukan (detik)	Hasil Rekonstruksi
1	1	25.81	
2	2	50.69	

No	Jumlah Objek Nyata	Waktu yang diperlukan (detik)	Hasil Rekonstruksi
3	3	57.66	
4	4	99.02	
5	5	103.46	

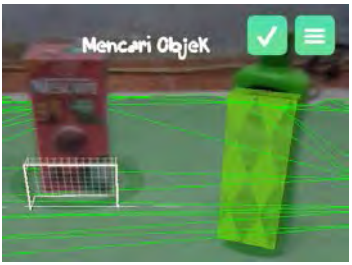
Pada Tabel 4.8 dapat dilihat bahwa objek pertama hampir menyerupai ukuran dan rotasi yang dimiliki benda. Sedangkan pada objek kedua hasil rekonstruksi masih belum sempurna, hanya merekonstruksi sebagian dari bentuk fisik objek. Hal ini kemungkinan besar disebabkan oleh kurangnya titik fitur yang dimiliki oleh objek kedua. Selain itu dapat juga karena objek memiliki warna yang hampir sama dengan latar rekonstruksi. Pada proses rekonstruksi tersebut juga membutuhkan area yang cukup luas sehingga marker inisialisasi awal beberapa kali tidak dapat terdeteksi. Hal ini ditandai dengan menghilangnya objek virtual gawang didepan marker.

4.2.3.3. Pengujian Rekonstruksi Bidang Tak Beraturan

Pengujian ini dilakukan untuk mengetahui apakah sistem mampu untuk mendeteksi dan reka ulang benda nyata. Pustaka Vuforia masih belum mampu untuk menampilkan bentuk fitur seutuhnya seperti yang dimiliki oleh benda tak beraturan. Pustaka ini hanya mampu untuk merekonstruksi suatu benda dengan cara objek virtual yang telah didefinisikan sebelumnya disesuaikan ukuran-ukuran yang dimiliki beserta posisi dan orientasinya didunia nyata. Dalam hal ini, objek virtual yang digunakan untuk proses rekonstruksi berupa balok. Hasil pengujian rekonstruksi bidang tak beraturan ditampilkan pada Tabel 4.9.

Pada Tabel 4.9 digunakan beberapa objek berbeda seperti botol dan gelas yang memiliki dimensi ukuran dan tekstur berbeda. pada tabel tersebut dapat dipahami bahwa teknologi smart terrain yang dikembangkan, mampu mengenali berbagai objek dengan bentuk yang tidak beraturan walaupun ukuran hasil rekonstruksinya belum sesuai. Namun, terdapat juga objek yang sukar untuk dikenali, seperti objek ketiga dan keempat. Objek ketiga pada tabel tersebut digunakan sebuah gelas yang memiliki nilai transparansi yang membuat objek tampak seakan-akan tidak terdapat dilator rekonstruksi sehingga sistem tidak mampu mengenalinya. Berbeda halnya dengan objek keempat dimana objek mampu dikenali namun karena memiliki tekstur yang hampir sama dengan latar hasil rekonstruksinya juga belum mencakup keseluruhan benda.

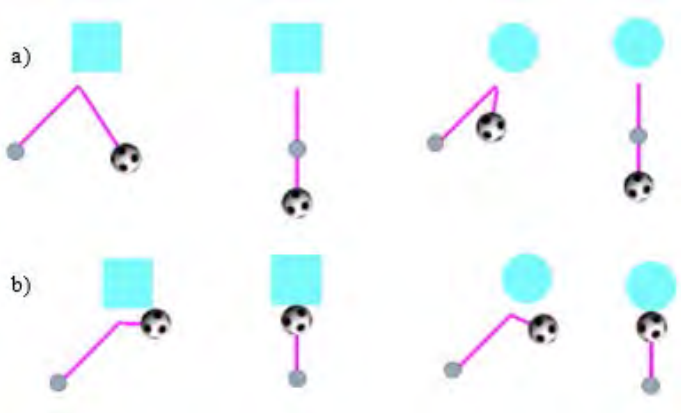
Tabel 4.9 Pengujian rekonstruksi benda dengan bidang tak beraturan

No	Hasil Rekonstuksi	Ukuran Asli (pxlxt)(cm)	Ukuran Rekonstruksi (pxlxt)(cm)
1		5x3x16	5.2x2.0x12.4
2		6x6x16	6.3x4.0x15.2
3		-	-

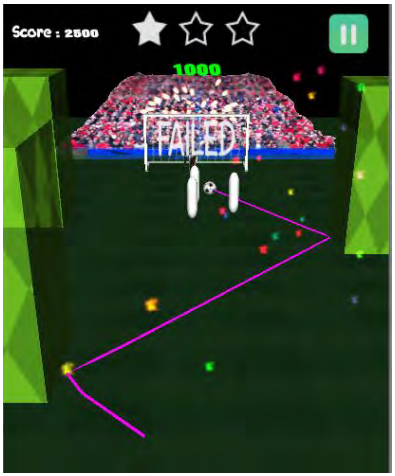
No	Hasil Rekonstuksi	Ukuran Asli (pxlxt)(cm)	Ukuran Rekonstruksi (pxlxt)(cm)
4		10x7x10	2.5x5.5x9.3
5		4x3.5x14	3.5x2.0x13.4

4.2.4. Pengujian *Collider* Dinamis

Berdasarkan hasil perbandingan antara objek penghalang bermateri fisik *bouncy* dan *ice*, didapatkan bahwa apabila benda berbentuk kotak dengan materi fisik berupa *bouncy* akan didapatkan pantulan yang sesuai dengan perbandingan vektor normal. Sedangkan pada benda berbentuk bulat pantulan dengan sudut tertentu tidak serupa perbandingannya dengan vektor normal. Sedangkan pada material *ice*, pada saat bola dipantulkan tegak lurus terhadap benda, maka bola tersebut akan menempel diam. Demikian juga dengan pantulan bola terhadap benda datar berupa kotak. Dengan sudut miring sebesar 45^{o}, bola hanya mengalami perubahan pantulan yang menghimpit bidang datar.



Gambar 4.3 Tampak atas a) Pengujian objek penghalang bermateri fisik *bouncy* b) Pengujian objek penghalang bermateri fisik *ice*



Gambar 4.4 Pengujian arah pantulan bola terhadap objek penghalang

Berdasarkan hukum kekekalan momentum dapat dihitung besarnya kecepatan yang terdapat pada bola saat sebelum menerima tumbukan dan setelah menerima tumbukan. Objek penghalang hasil rekonstruksi posisinya tetap tidak berubah sehingga memiliki kecepatan

sebesar 0m/s. Dengan massa bola dan massa penghalang diketahui dapat dihitung berapa besar kecepatan tumbukan yang terjadi.

$$m_a \times v_a + m_b \times v_b = m_a \times v_a' + m_b \times v_b' \quad \text{... (4.3)}$$

Pada Gambar 4.4 dapat dilihat bahwa bola yang mendapatkan input berupa gaya sebesar (-1.05, 0.0, 1.05) dengan sudut sebsar 315^o dipantulkan terhadap dua buah objek penghalang yang memiliki nilai lenting sempurna atau nilai *bounciness* sama dengan satu. Data pada Tabel 4.10 diperoleh dari perhitungan nilai kecepatan yang dimiliki bola pada saat bergerak dengan nilai lenting sempurna. Pada tabel tersebut, kecepatan sebelum dan sesudah terjadinya tumbukan bola terhadap objek penghalang memiliki nilai yang sama namun dengan arah gerakan berbeda.

Tabel 4.10 Kecepatan bola sebelum dan sesudah tumbukan dengan materi *bounciness* sebesar 1

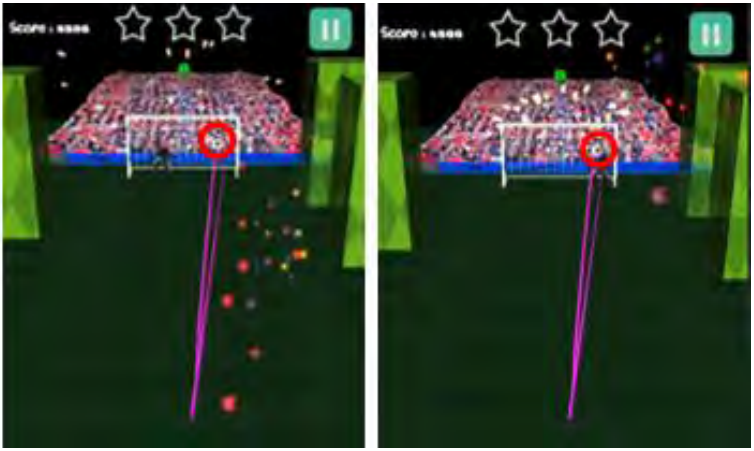
No	Gaya Awal (x,y,z)(N)	Kecepatan Bola Sebelum Tumbukan (x,y,z)(cm/s)	Kecepatan Bola Sesudah Tumbukan (x,y,z)(cm/s)
1	0.0, 0.0, 1.50	0.0, 0.0, 13.9	0.0, 0.0, -13.9
2	-1.05, 0.0, 1.05	6.2, 0.0, 7.1	-6.2, 0.0, 7.1
3	1.05, 0.0, 1.05	14.1, 0.0, 14.1	-14.1, 0.0, 14.1

Percobaan selanjutnya adalah dengan mengurangi nilai bounciness yang dimiliki bola, yaitu dengan menggunakan nilai sebesar 0.5. pada Tabel 4.11 diperoleh bahwa tidak hanya terjadi perubahan arah pergerakan bola melainkan juga besarnya nilai kecepatan yang dimiliki. Pada objek penghalang dengan nilai lenting 0.5 tersebut diperoleh kecepatan hasil tumbukan besarnya sama dengan setengah dari besar kecepatan bola sebelum tumbukan. Sehingga dari data tersebut dapat disimpulkan bahwa kecepatan bola sesuai dengan persamaan kekekalan momentum.

Tabel 4.11 Kecepatan bola sebelum dan sesudah tumbukan dengan materi *bounciness* sebesar 0.5

No	Gaya Awal (x,y,z)(N)	Kecepatan Bola Sebelum Tumbukan (x,y,z)(cm/s)	Kecepatan Bola Sesudah Tumbukan (x,y,z)(cm/s)
1	0.0, 0.0, 1.50	0.0, 0.0, 20.0	0.0, 0.0, -10.0
2	-105, 0.0, 1.05	-14.1, 0.0, 14.1	7.1, 0.0, 14.1
3	1.05, 0.0, 1.05	14.1, 0.0, 14.1	-7.1, 0.0, 14.1

4.2.5. Pengujian Pergerakan Arah Penjaga Gawang



a) b)

Gambar 4.5 Arah pergerakan penjaga gawang menghalau bola a) Bola tidak berhasil dihalau b) Bola dapat dihalau.

Pengujian dilakukan dengan cara menggerakkan bola dengan gaya yang tetap dibeberapa titik pada gawang sebanyak sepuluh kali dan dianalisa apakah pergerakan penjaga gawang mampu untuk menghalau datangnya bola yang mengarah pada gawang. Pergerakan bola diarahkan

menuju ke 6 titik pada gawang, yaitu pada posisi kiri bawah, tengah, kanan bawah, kiri atas, tengah atas dan kanan atas. Pada Gambar 4.5 ditampilkan bahwa bola diarahkan menuju ke sisi kanan atas gawang, digambar yang pertama penjaga gawang bergerak kearah sisi kiri gawang sehingga bola memasuki gawang, sedangkan pada gambar selanjutnya penjaga gawang bergerak kearah sisi kanan gawang sehingga bola berhasil dihalau.

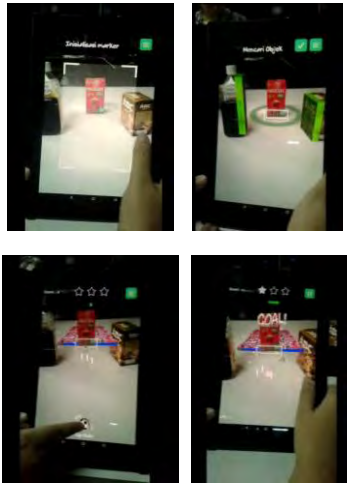
Hasil percobaan pengujian arah pergerakan penjaga gawang dalam menghalau bola ditunjukkan pada Tabel 4.12. Pada tabel tersebut dilakukan percobaan sebanyak sepuluh kali pada setiap titik. Peluang penjaga gawang dalam menghalau adanya bola berkisar antara satu hingga tiga kali. Peluang terbanyak bola memasuki gawang berada pada titik tengah atas gawang, dimana penjaga gawang hanya mampu menghalau satu tembakan dari sepuluh tembakan bola yang dilakukan. Sedangkan peluang terbaik penjaga gawang dalam menghalau bola berada pada titik kiri bawah dan kanan atas, dimana penjaga gawang mampu menghalau tiga kali tembakan dalam sepuluh kali percobaan.

Tabel 4.12 Pengujian pergerakan arah penjaga gawang untuk menangkap bola

No	Gaya konstan untuk menggerakkan bola (x,y,z)(N)	Arah pergerakan bola	Jumlah bola yang dapat dihalau
1	-0.115, 0, -1.50	Kiri bawah	3
2	0, 0, -1.50	Tengah	2
3	0.115, 0, -1.50	Kanan bawah	2
4	-0.115, 0.43, -1.50	Kiri atas	2
5	0, 0.43, -1.50	Tengah atas	1
6	0.115, 0.43, -1.50	Kanan atas	3

4.2.6. Pengujian Simulasi Permainan

Simulasi permainan dilakukan dengan cara mencoba memainkan permainan hingga selesai. Pada Gambar 4.6 ditampilkan keseluruhan proses simulasi permainan, dimulai dari proses inisialisasi marker, deteksi objek-objek penghalang kemudian deteksi adanya input berupa gesture terhadap tombol bola, lalu proses dimana animasi goal ditampilkan apabila bola memasuki gawang. Pengujian ini dilakukan untuk dibandingkan berapa banyak goal dan score maksimal yang dapat diperoleh dalam permainan. Dimana untuk penilaian score dijumlahkan dengan nilai 500 saat bola membentur objek penghalang dan nilai 1000 saat bola memasuki gawang. Pengujian ini dilakukan dengan menggunakan dua buah objek penghalang.



Gambar 4.6 Simulasi permainan bola

Pada Tabel 4.13 dari lima kali percobaan yang dilakukan dengan menggunakan gawang berukuran 9x4.5x3cm^3 diperoleh hasil bahwa jumlah goal yang didapatkan setiap kali siklus permainan berkisar antara satu hingga empat goal. Sedangkan jumlah nilai score yang dapat diperoleh oleh pengguna juga bervariasi mulai dari 500 hingga 6000.

Tabel 4.13 Pengujian simulasi permainan

No	Percobaan ke	Jumlah pantulan penghalang	Total Score	Goal
1	1	5	3500	1
2	2	3	5500	4
3	3	1	4000	3
4	4	6	5000	2
5	5	4	6000	4

Halaman ini sengaja dikosongkan

BAB 5
PENUTUP

5.1. Kesimpulan

Berdasarkan pada hasil perancangan, simulasi dan pengujian dari keseluruhan sistem dalam Tugas Akhir ini, maka diperoleh kesimpulan sebagai berikut:

1. Berdasarkan hasil pengujian dari penggunaan marker, kuantitas dan pola penyebaran fitur sudut dari multi target yang digunakan beserta ukuran dan jaraknya terhadap kamera merupakan penentu utama keberhasilan inisialisasi.
2. Ukuran objek hasil rekonstruksi tergantung pada banyaknya proses kalibrasi yang dilakukan terhadap objek penghalang. Dalam pengujian hasil perbandingan volume yang menyerupai aslinya memiliki rasio hingga 94.86% terhadap ukuran objek sebenarnya.
3. Banyaknya objek penghalang juga dapat mempengaruhi baiknya hasil rekonstruksi. Semakin banyak objek penghalang yang digunakan pada arena permainan semakin menurun perbandingan rasio rekonstruksinya. Dimana terdapat pula objek yang tidak berhasil direkonstruksi. Jumlah maksimal objek penghalang yang dapat dideteksi sebanyak 4 buah.
4. Waktu yang diperlukan untuk melakukan proses rekonstruksi smart terrain bervariasi. Dimana saat melakukan rekonstruksi terhadap sebuah objek penghalang waktu yang dibutuhkan hanya 25.81 detik. Sedangkan untuk rekonstruksi dengan jumlah objek penghalang sebanyak lima dibutuhkan waktu rekonstruksi sebesar 103.46 detik. Sehingga dapat disimpulkan bahwa semakin banyak objek penghalang yang akan dideteksi dan direkonstruksi, semakin banyak pula waktu yang dibutuhkan untuk proses rekonstruksinya.
5. Pada pengujian collider dinamis diperoleh hasil bahwa besarnya pantulan bola tergantung pada nilai *bounciness* yang dimiliki penghalang. Pada penggunaan objek penghalang yang memiliki lenting sempurna kecepatan bola sebelum dan sesudah memiliki besar yang sama namun arah pergerakan berbeda. Sedangkan pada objek penghalang dengan nilai bounciness 0.5 akan menghasilkan

besarnya kecepatan bola berkurang sebanyak setengah dari kecepatan bola sebelum mengalami tumbukan.

6. Pada pengujian penjaga gawang, banyakya bola yang mampu dihalau penjaga gawang berkisar antara satu hingga tiga kali tendangan dari sepuluh kali percobaan.
7. Dari pengujian simulasi permainan didapatkan hasil bahwa jumlah goal yang dapat diperoleh dari lima kali percobaan berkisar antara satu hingga 4 kali goal. Sedangkan nilai maksimal score berjumlah 6000.

5.2. Saran

Berdasarkan pengujian dan kesimpulan yang telah didapat, muncul beberapa kritik dan saran yang dapat diperhatikan untuk pengembangan aplikasi ini di masa yang akan datang. Beberapa kritik dan saran tersebut diantaranya:

1. Library Vuforia yang digunakan untuk merekonstruksi *Smart terrain* masih belum sempurna. Dimana objek-objek nyata yang dideteksi hanya terbatas diubah ke bentuk model tertentu, tidak mengikuti bentuk asli objek nyata.
2. Library Vuforia juga tersedia untuk platform peranti bergerak lain nya. Disarankan untuk dapat mengimplementasikan pengembang an aplikasi Augmented Reality ini pada peranti bergerak lain pada masa yang akan datang.

Apabila penelitian ini akan dikembangkan lebih lanjut pada masa yang akan datang, disarankan untuk menggunakan sumber daya yang sekurang-kurangnya sama atau lebih besar dari yang digunakan pada penelitian ini

DAFTAR PUSTAKA

[1] **A. Ronald T,** "*A survey of augmented reality*," pp. 355–385, August 1997.

[2] **P. Milgram and F. Kishino,** "A taxonomy of mixed reality visual displays," vol. E77-D, No.12, December 1994.

[3] **D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg,** "Pose tracking from natural features on mobile phones," 2008.

[4] **D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg,** "Real-time detection and tracking for augmented reality on mobile phones," vol. 16, NO. 3, 2010.

[5] **A. Riska Wahyu,** "Fixed point augmented reality menggunakan kinect," ITS, Surabaya, 2013.

[6] **Qualcomm,** "Qualcomm vuforia developer portal." http://developer.vuforia.com. Online; Diakses pada 12-01-2016

[7] **Y. Rizki,** "Markerless augmented reality pada perangkat android," ITS, Surabaya, 2012.

[8] **Unity,** "Unity Documentation | Unity Manual - Physics." http://docs.unity3d.com. Online; Diakses pada 12-01-2016

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS

Pondra Setiawan lahir di Gresik pada tanggal 07 Juli 1991. Ia menyelesaikan pendidikan SD di SDN 472 Surabaya pada tahun 2003, kemudian melanjutkan pendidikan di SMPN 28 Surabaya dan menyelesaikan pendidikan SMP pada tahun 2006, kemudian pendidikan SMA dilanjutkan di SMAN 13 Surabaya yang diselesaikan pada tahun 2009. Setelah lulus SMA, ia memilih untuk melanjutkan pendidikan di Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan jurusan Teknik Elektro dan mengambil konsentrasi bidang studi Teknik Komputer dan Telematika. Hal ini tidak lepas dari ketertarikannya dalam bidang teknologi.

Halaman ini sengaja dikosongkan