

***HAND GESTURE RECOGNITION PADA APLIKASI  
PENGENALAN BANGUN RUANG DENGAN LEAP  
MOTION CONTROLLER***

**SKRIPSI**



**Disusun Oleh :**

**David Prasetyo Budi Arianto**

**123112706450005**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS NASIONAL  
JAKARTA  
2015**

## LEMBAR PERSETUJUAN TUGAS AKHIR

Nama Mahasiswa : David Prasetyo Budi Arianto  
Nomor Pokok Mahasiswa : 123112706450005  
Program Studi : Teknik Informatika  
Program Kekhususan : Teknologi Multimedia  
Judul Tugas Akhir : *Hand Gesture Recognition Pada Aplikasi Pengenalan Bangun Ruang dengan Leap Motion Controller*

Telah disetujui untuk diseminarkan di depan Sidang Tugas Akhir

Pembimbing I,

Pembimbing II,

( Septi Andryana, S. Kom., MMSI )

NID: 0103010799

( Fauziah, S. Kom., MMSI )

NID. 0104090784

Mengetahui

Ketua Program Studi Teknik Informatika

(Septi Andryana, S. Kom., MMSI)

NID: 0103010799

## ABSTRAK

Dalam mempelajari bangun ruang seringkali pelajar Sekolah Dasar mengalami kesulitan pemahaman terhadap materi yang diberikan. Maka diperlukan suatu alat peraga interaktif dan menarik dalam bentuk aplikasi multimedia Pengenalan Bangun Ruang supaya dapat meningkatkan minat belajar dan pemahaman siswa-siswi dalam mempelajari materi bangun ruang. Penulis menggunakan teknologi *Leap Motion Controller* untuk mengimplementasikan *Hand Gesture Recognition* yaitu pengembangan dari *Hand Motion Tracking*. Dengan demikian, aplikasi dapat dikontrol dengan menggunakan gerakan tangan tanpa *keyboard* ataupun *mouse*. Penulis menemukan bahwa variasi dan persentasi ketepatan gestur pada aplikasi-aplikasi sebelumnya belum optimal. Dengan menggunakan metode *Multimedia Development Life Cycle* (MDLC), penelitian ini bertujuan menambah variasi jenis gestur yang digunakan serta meningkatkan persentasi ketepatan dalam mengenal gestur. Hasil dari penelitian adalah aplikasi dengan fitur 10 jenis definisi gestur, yaitu: *Tap, Swipe, Clap, Double Outward Swipe, Circular, Fly Control, Pinch, Release, Push* dan *Pull*. Ketepatan penerjemahan gestur mencapai 91.6% pada *frame rate* 30 fps dan eksekusi CPU sebesar 45,53 ms.

Kata kunci : Aplikasi Multimedia, *Gesture Recognition*, *Leap Motion*, *Hand Motion Tracking*

## ABSTRACT

*In studying geometry, kids in elementary school often have a trouble to understand the provided subjects. It would require an innovative and interesting props that take advantage of information technology advances in order to increase children's interest in learning and understanding geometry. One thing to do is implementing Hand Gesture Recognition into a multimedia educational application. By utilizing Leap Motion technology, the tracking has to be Markerless or without a marker. The application can be controlled using hand gestures without a keyboard or mouse. The author found some variations and the percentage of gesture accuracy on the previous applications are not optimal enough. By using Multimedia Development Life Cycle (MDLC) as a method, the accuracy has been increased. As a result of this project, the application developed into 10 types of gestures : Tap, Swipe, Clap, Double Outward Swipe, Circular, Fly Control, Pinch, Release, Push Pull. Accuracy of gesture interpretation has obtained 91.6% on 30 fps frame rate and CPU execution on 45,53 ms.*

*Keyword : Multimedia Application, Hand Gesture Recognition, Leap Motion, Hand Motion Tracking*

## KATA PENGANTAR

Puji syukur penulis ucapkan kehadirat Tuhan karena hanya atas berkat dan anugerah yang dilimpahkan-Nya penulis dapat menyelesaikan skripsi ini pada waktunya. Skripsi ini merupakan kegiatan penelitian sebagai salah satu syarat untuk memperoleh gelar Strata Satu (S-1) di Universitas Nasional Jurusan Teknik Informatika. Maka untuk memenuhi persyaratan tersebut, penulis mencoba untuk menerapkan ilmu yang telah penulis dapat di bangku kuliah ke dalam bentuk karya tulis yang berjudul **“HAND GESTURE RECOGNITION PADA APLIKASI PENGENALAN BANGUN RUANG DENGAN LEAP MOTION CONTROLLER”**.

Oleh karena itu, dalam penulisan skripsi ini, penulis berharap adanya kritik dan saran dari semua pihak yang nantinya dipergunakan untuk menyempurnakannya.

Selama penyusunan karya tulis , penulis telah banyak memperoleh bantuan, bimbingan dan dorongan dari berbagai pihak. Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada:

1. Bapak Dr. Ucuk Darusalam S.T., MT. selaku Dekan Fakultas Teknologi Komunikasi dan Informatika,
2. Ibu Septi Andryana S.Kom, MMSI selaku pembimbing I dan ibu Fauziah S.Kom, MMSI selaku pembimbing II,
3. Bapak Dr. Iskandar Fitri S.T., MT. selaku Dosen Metodologi Penelitian yang telah memberikan ilmunya yang sangat bermanfaat bagi penulis,
4. Dosen penguji dan seluruh Staff Pengajar / Dosen Fakultas Teknologi Komunikasi dan Informatika, yang telah banyak memberikan ilmunya yang sangat bermanfaat bagi penulis,

5. Orang tua, dan adikku tercinta atas dukungannya baik moril maupun materil serta perhatiannya yang menjadi motivasi sangat berarti bagi penulis,
6. Rekan-rekan semester akhir dan sesama Asisten Laboratorium yang turut memberikan dukungan,
7. Para personil *Dhalang Production, Asrika Films*, komunitas IT dan komunitas hobi yang secara langsung maupun tidak turut memberikan inspirasi,
8. Rekan-rekan alumni Fakultas Teknologi Komunikasi dan Informatika yang telah berbagi pengalaman,
9. Dan rekan-rekanku yang tidak dapat disebutkan satu persatu di lingkungan kampus Universitas Nasional maupun di luar kampus, yang telah banyak membantu penulis dalam menyelesaikan skripsi ini.

Akhir kata, penulis mengucapkan terima kasih dan berharap agar skripsi ini dapat bermanfaat bagi penulis pada khususnya dan pembaca pada umumnya.

Jakarta, Januari 2016

David Prasetyo

## DAFTAR ISI

### HALAMAN JUDUL

HALAMAN PERNYATAAN ORISINALITAS .....	ii
HALAMAN PERSETUJUAN SIDANG AKHIR .....	ii
HALAMAN PENGESAHAN .....	ii
HALAMAN PERSETUJUAN TUGAS AKHIR .....	ii
ABSTRAK .....	iii
ABSTRACT .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xv

<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
--------------------------------	----------

1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	3
1.3 Rumusan Masalah .....	3
1.4 Batasan Masalah .....	4
1.5 Tujuan Penelitian .....	4
1.6 Manfaat Penelitian .....	5
1.7 Metode Penelitian .....	6
1.7.1 Metodologi Pengumpulan Data .....	6
1.7.2 Metodologi Pengembangan Aplikasi .....	6
1.8 Sistematika Penulisan .....	8

<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>10</b>
--------------------------------------	-----------

2.1	Hand Motion Tracking.....	10
2.2	Struktur Tangan Manusia.....	13
2.3	<i>Hand Gesture Recognition</i> .....	14
2.4	<i>Leap Motion Cotroller</i> .....	15
2.5	Konsep Bangun Ruang.....	17
2.6	Literatur Jurnal .....	19
2.6.1	Literatur Jurnal 1 .....	19
2.6.2	Literatur Jurnal 2 .....	20
2.6.3	Literatur Jurnal 3 .....	21
2.6.4	Literatur Jurnal 4 .....	22
2.6.5	Literatur Jurnal 5 .....	23
2.6.6	Literatur Jurnal 6 .....	23
2.6.7	Literatur Jurnal 7 .....	24
2.6.8	Literatur Jurnal 8 .....	25
2.6.9	Literatur Jurnal 9 .....	26
2.6.10	Literatur Jurnal 10.....	27
2.7	Literatur Aplikasi .....	28
2.7.1	Literatur Aplikasi 1 .....	28
2.7.2	Literatur Aplikasi 2 .....	29
2.7.3	Literatur Aplikasi 3 .....	30
2.7.4	Literatur Aplikasi 4 .....	31
2.7.5	Literatur Aplikasi 5 .....	31
2.8	Tabel Jurnal Pembanding.....	33
2.9	Tabel Aplikasi Pembanding .....	37
<b>BAB III</b>	<b>PERANCANGAN APLIKASI.....</b>	39
3.1	Kerangka Penelitian .....	39

3.2	Tahap Perancangan .....	41
3.2.1	Perancangan Aplikasi .....	41
3.2.2	Rancangan <i>Hand Gesture Recognition</i> .....	43
3.3	Spesifikasi Hardware .....	44
3.3.1	PC.....	44
3.3.2	<i>Leap Motion Controller</i> .....	45
3.4	Spesifikasi Software.....	47
3.4.1	Leap Motion Core Services (SDK).....	47
3.4.2	Unity 3D Versi 5.1.1f1 .....	49
3.4.3	Blender 2.68a.....	52
3.4.4	GIMP 2.8 .....	52
3.5	Tahap Pre-Production .....	53
3.5.1	Perancangan Flowchart Aplikasi .....	53
3.5.2	Perancangan Storyboard Aplikasi.....	54
3.6	Tahap Produksi .....	57
3.6.1	Instalasi Leap Motion SDK .....	57
3.6.2	Membuat Project Baru .....	59
3.6.3	<i>Import Leap Motion Package</i> .....	61
3.6.4	Set Scene pada Unity 5 .....	62
3.6.5	Perancangan Sprite .....	65
3.6.6	Pemodelan Objek 3D dan Animasi.....	67
3.6.7	Proses Import Model ke Unity.....	72
3.6.8	Perancangan Scene Mini Game .....	73
3.6.9	Implementasi Gesture Recognition.....	75
3.6.10	Hasil Kompilasi Versi Alpha dan Beta.....	80
<b>BAB IV</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b> .....	84
4.1	Implementasi .....	84

4.1.1	Tahap Paska Produksi .....	84
4.1.2	Antarmuka Versi <i>Golden Master</i> .....	86
4.2	Pengujian.....	91
4.2.1	Pengujian <i>Leap Motion Controller</i> .....	91
4.2.2	Pengujian Aplikasi Pengenalan Bangun Ruang .....	99
<b>BAB V</b>	<b>PENUTUP</b> .. .. .. ..	<b>85</b>
4.1	Kesimpulan .....	60
4.1	Saran.....	60
<b>DAFTAR PUSTAKA .....</b>		<b>86</b>

## DAFTAR GAMBAR

Gambar 2.1	Diagram penerapan hand motion tracking .....	10
Gambar 2.2	Tracking ujung jari berdasarkan koordinat X, Y, Z .....	11
Gambar 2.3	Pola gestur menghasilkan bentuk tertutup dan terbuka .....	12
Gambar 2.4	Model Persendian Tangan dan Tipe Gerakannya .....	14
Gambar 2.5	Citra yang Terdeteksi Leap Motion .....	16
Gambar 2.6	Sistem koordinat Leap Motion Controller .....	16
Gambar 2.7	Control Gesture pada game Honey Bee Dance .....	19
Gambar 2.8	Navigasi Drag and Stop pada ruang virtual 3D .....	21
Gambar 2.9	Pengenalan Gestur dengan Leap Motion dan Kinect .....	22
Gambar 2.10	<i>Tracking Record</i> pada aplikasi .....	24
Gambar 2.11	Hidden Markov Model Sederhana .....	25
Gambar 2.12	Grafik Ketepatan Leap Motion dalam mengenal Gestur ....	26
Gambar 2.13	Eksplorasi ruangan secara virtual .....	27
Gambar 2.14	Aplikasi 3D Geometry .....	28
Gambar 2.15	Aplikasi GeoCrazy .....	29
Gambar 2.16	Aplikasi Form and Function 3D.....	30
Gambar 2.17	Aplikasi Zoology .....	31
Gambar 2.18	Aplikasi Molecules .....	32
Gambar 3.1	Kerangka Penelitian .....	39
Gambar 3.2	Flowchart Tahap Perancangan .....	41
Gambar 3.3	<i>Hand Gesture Recognition</i> dengan <i>Leap Motion</i> .....	43
Gambar 3.4	PC yang digunakan dalam pembuatan aplikasi .....	44
Gambar 3.5	Perangkat <i>Leap Motion Controller</i> .....	45

Gambar 3.6 Ilustrasi komponen <i>Leap Motion Controller</i> .....	45
Gambar 3.7 Sketsa struktur perangkat <i>Leap Motion</i> .....	46
Gambar 3.8 Leap Motion Control Panel .....	47
Gambar 3.9 Leap Motion Calibration Tool .....	48
Gambar 3.10 Leap Motion Diagnostic Visualizer .....	49
Gambar 3.11 Workspace Unity 5 .....	50
Gambar 3.12 Flowchart aplikasi .....	53
Gambar 3.13 Halaman Download Leap Motion SDK .....	57
Gambar 3.14 Program SDK yang telah diinstall .....	58
Gambar 3.15 Status <i>tracking</i> pada posisi <i>off</i> .....	58
Gambar 3.16 Status <i>tracking</i> pada posisi <i>on</i> .....	58
Gambar 3.17 Jendela awal New Project .....	59
Gambar 3.18 Penentuan lokasi penyimpanan project .....	60
Gambar 3.19 Pengaturan rasio layar aplikasi .....	60
Gambar 3.20 Proses Import Custom Package .....	61
Gambar 3.21 Import Leap Motion Asset Package .....	62
Gambar 3.22 Prefab Hand Controller .....	62
Gambar 3.23 Variabel-varibel <i>Hand Controller</i> .....	63
Gambar 3.24 Objek disusun sesuai Storyboard.....	63
Gambar 3.25 Susunan Button pada Main Menu .....	64
Gambar 3.26 Susunan Desain Robot dan Kota .....	64
Gambar 3.27 Pengaturan Image pada GIMP.....	65
Gambar 3.28 Susunan Layer pada GIMP.....	66
Gambar 3.29 Perancangan Sprite pada GIMP .....	66
Gambar 3.30 Mengubah Tipe Image.....	67
Gambar 3.31 Penyusunan Sprite pada Scene .....	67

Gambar 3.32 Menambahkan Objek Plane .....	68
Gambar 3.33 Tahap Duplikasi Objek .....	68
Gambar 3.34 Desain Jaring-jaring Kubus .....	69
Gambar 3.35 Penambahan Texture .....	69
Gambar 3.36 Proses pewarnaan Bangun Ruang .....	70
Gambar 3.37 Objek bangun Ruang Bertekstur .....	70
Gambar 3.38 Proses Penambahan Keyframe .....	71
Gambar 3.39 Animasi Bangun Ruang.....	71
Gambar 3.40 Proses Import Blender Model ke Unity .....	72
Gambar 3.41 Animator pada Unity .....	72
Gambar 3.42 Set Arena Mini Game .....	73
Gambar 3.43 Penambahan Objek Keranjang .....	74
Gambar 3.44 Penambahan Objek 3D Button.....	74
Gambar 3.45 Pinch Gesture .....	78
Gambar 3.46 Tampilan Menu Utama .....	80
Gambar 3.47 Scene Mini Game 1 .....	81
Gambar 3.48 Tampilan Menu Bangun Ruang (1).....	81
Gambar 3.49 Tampilan Materi Bangun Ruang (2).....	82
Gambar 3.50 Tampilan Latihan Soal .....	82
Gambar 3.51 Scene Mini Game 2 .....	83
Gambar 4.1 Pengaturan Installforge untuk Aplikasi .....	84
Gambar 4.2 Penambahan <i>Source Folder</i> Aplikasi .....	85
Gambar 4.3 Tampilan <i>Splash Screen</i> .....	86
Gambar 4.4 Tampilan Menu Utama.....	86
Gambar 4.5 Tampilan Bantuan .....	87
Gambar 4.6 Tampilan Tentang.....	87

Gambar 4.7	Tampilan Mini Game 1 .....	88
Gambar 4.8	Tampilan Menu Bangun Ruang.....	88
Gambar 4.9	Tampilan Scene Pengenalan Bangun Ruang .....	89
Gambar 4.10	Fitur <i>Zoom-In</i> Bangun Ruang.....	90
Gambar 4.11	Scene Mini Game 2 .....	90
Gambar 4.12	Kondisi A (tanpa cahaya) .....	91
Gambar 4.13	Kondisi B (ada cahaya) .....	92
Gambar 4.14	Pengujian dengan Leap Motion Visualizer .....	92
Gambar 4.15	<i>Leap Motion Interaction Box</i> .....	101
Gambar 4.16	Grafik Evaluasi Tingkat Ketepatan 10 Gestur.....	105
Gambar 4.17	<i>Unity 5 Profiler</i> .....	107

## DAFTAR TABEL

Tabel 2.1 Perbandingan Algoritma <i>Gesture Recognition</i> .....	15
Tabel 2.2 Bangun Ruang Sisi Datar .....	17
Tabel 2.3 Bangun Ruang Sisi Lengkung .....	18
Tabel 2.4 Hasil Perbandingan Jurnal Pembanding 1 sampai 5 .....	33
Tabel 2.5 Hasil Perbandingan Jurnal Pembanding 6 sampai 10 .....	35
Tabel 2.6 Hasil Perbandingan Aplikasi Pembanding.....	37
Tabel 3.1 Spesifikasi Hardware PC .....	44
Tabel 4.1 Persentase Ketepatan <i>Tracking</i> pada Sumbu Y .....	94
Tabel 4.2 Persentase Ketepatan <i>Tracking</i> pada Sumbu Z.....	95
Tabel 4.3 Persentase Ketepatan <i>Tracking</i> pada Sumbu X .....	96
Tabel 4.4 Ketepatan Gestur Statis.....	97
Tabel 4.5 Pengujian Interface pada Scene Menu Utama .....	99
Tabel 4.6 Pengujian Interface pada Scene Mini Game 1 .....	99
Tabel 4.7 Pengujian Interface pada Scene Pengenalan Bangun Ruang ...	99
Tabel 4.8 Pengujian Interface pada Scene Mini Game 2 .....	100
Tabel 4.9 Perubahan Variabel Tiap Versi .....	101
Tabel 4.10 Ketepatan Gestur Dinamis pada Sudut 60° .....	102
Tabel 4.11 Ketepatan Gestur Dinamis pada Sudut 45° .....	103
Tabel 4.12 Ketepatan Gestur Dinamis pada Sudut 30° .....	103
Tabel 4.13 Ketepatan Gestur Dinamis pada Sudut 15° .....	104
Tabel 4.14 Ketepatan Gestur Dinamis pada Sudut 0° .....	104
Tabel 4.15 Tabel Evaluasi Tingkat Ketepatan 10 Gestur .....	106
Tabel 4.16 Kinerja Setiap Gestur .....	108
Tabel 4.17 Pengujian Suara.....	109

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Pembelajaran bangun ruang bagi pelajar sekolah dasar tentu membutuhkan suatu media visual. Namun dalam kegiatan belajar mengajar bangun ruang, memvisualisasikan bangun ruang secara 3 dimensi tidak cukup hanya dengan buku pelajaran. Selain itu, kegiatan belajar mengajar tersebut akan berhasil jika PAKEM terpenuhi. PAKEM adalah singkatan dari Pembelajaran Aktif, Kreatif, Efektif dan Menyenangkan <sup>[1]</sup>. Untuk itu diperlukan suatu media kreatif dan menyenangkan yang dapat memberi pengetahuan kepada peserta didik bagaimana merepresentasikan bentuk bangun ruang. Ditambah dengan melibatkan teknologi informasi yang kini tidak asing bagi semua usia.

Setelah kemunculan Kinect, dirilisnya *Leap Motion Controller* pada tahun 2013 menambah variasi dari implementasi dan pengembangan *Natural User Interface (NUI) Device* yang mendukung *real-time motion tracking* dalam bidang sains komputer. Potensi penggunaan Leap Motion dapat mencakup aspek yang sangat luas seperti *personal PC controller*, presentasi interaktif, virtualisasi di bidang pendidikan dan kesehatan, industri kreatif seperti game dan musik, hingga simulasi dan pemodelan sistem bidang lainnya. *Leap Motion Controller* adalah perangkat menggunakan teknik *Hand Motion Tracking* yaitu *tracking* jari tangan berbasis data kedalaman. Teknik ini dikembangkan menjadi *Hand Gesture Recognition* yang digunakan sebagai alternatif interaksi manusia dan komputer. Berdasarkan tren teknologi tersebut, penelitian ini bertujuan untuk merancang dan menghasilkan produk aplikasi multimedia interaktif

sebagai media pembelajaran bangun ruang bagi peserta didik sekolah dasar dengan memanfaatkan teknologi *Leap Motion*.

Sebagai acuan penelitian, penulis mengambil literatur dari beberapa jurnal dan aplikasi pembanding yang menggunakan Leap Motion. Dari jurnal yang diambil, penulis menemukan beberapa kekurangan seperti pada jurnal milik Ok-Hue Cho dengan judul *Development of Serious Game for Kids using Leap Motion based on Honey Bee Dance* <sup>[8]</sup>. Terdapat beberapa kekurangan yaitu interface yang masih dalam bentuk dua dimensi dan tidak adanya *Virtual Hand*. Terlihat juga belum optimalnya penggunaan *gesture recognition* (pengenalan gestur) yang diterapkan pada aplikasi. Selain itu, Bruno Fanini dalam Jurnal berjudul *A 3D Interface to Explore and Manipulate multi-scale Virtual Scenesusing the Leap Motion Controller* <sup>[9]</sup> menjelaskan bahwa *Leap Motion* dapat digunakan untuk melakukan eksplorasi pada ruang 3D virtual. Namun kekurangannya adalah aplikasi tidak disertai dengan interface yang mumpuni seperti tombol navigasi. Dari segi ketepatan aplikasi dalam mengenal gestur tangan, penulis mengambil referensi dari Pedro Miguel dengan jurnal berjudul *Analysis and Evaluation of Gesture Recognition using LeapMotion* <sup>[16]</sup> dengan metode sintaks dapat menghasilkan ketepatan rata-rata mencapai 87.3 %. Metode inilah yang menghasilkan persentase ketepatan tertinggi.

Dari 5 aplikasi pembanding yang diambil, penulis melihat bahwa yang paling mendekati konsep penulis adalah aplikasi bernama *3D Geometry* yang diproduksi oleh Quwei Lab <sup>[18]</sup>. Berdasarkan aplikasi pembanding tersebut penulis kembali menemukan kekurangan, yaitu interaksi user yang masih menggunakan kursor. Tidak ada *Virtual Hand* untuk membuat user berinteraksi secara langsung dengan objek bangun ruang. Tidak adanya virtual hands ini juga menyulitkan user dalam mengetahui posisi tangan secara virtual pada sumbu acuan x,y,z. Selain itu, untuk mengimplementasikan *pinch gesture*, penulis akan

menambahkan fitur *mini game* di dalam aplikasi dengan menampilkan bentuk bangun ruang dalam kehidupan sehari-hari.

Berdasarkan latar belakang di atas, maka penulis mengangkat topik penelitian dengan judul *Hand Gesture Recognition* pada Aplikasi Pengenalan Bangun Ruang dengan *Leap Motion Controller*.

## 1.2 Identifikasi Masalah

Berdasarkan judul penelitian yang dibuat dan dengan melihat jurnal serta aplikasi pembanding, maka ditemukan beberapa permasalahan yang diidentifikasi kedalam beberapa poin yaitu ;

1. Perlunya aplikasi multimedia interaktif untuk mendukung pembelajaran bangun ruang.
2. Fitur *hand gesture recognition* yang ditemukan pada aplikasi sebelumnya masih sedikit variasinya dengan ketepatan pengenalan gestur rata-rata dibawah 90%.
3. Fitur dan *interface* pada aplikasi maupun pada jurnal pembanding masih kurang lengkap.

## 1.3 Rumusan Masalah

1. Mengacu pada situs resmi Leap Motion yaitu <http://leapmotion.com>, masih belum banyak aplikasi Leap Motion yang bersifat edukatif terutama yang ditujukan untuk mendukung pembelajaran pengenalan bangun ruang.
2. Belum optimalnya variasi tipe dan tingkat ketepatan *gesture recognition* (pengenalan gestur) yang diterapkan pada aplikasi pembanding.

3. Selain belum adanya menu navigasi dan fitur zoom, juga belum adanya *Virtual Hand* yang diterapkan pada aplikasi sebelumnya. *Virtual Hand* dibutuhkan untuk mengetahui posisi tangan yang dijabarkan dalam titik-titik koordinat *fingertips* (ujung jari) pada bidang kartesian 3 dimensi.

#### **1.4 Batasan Masalah**

Penulisan dibatasi ruang lingkupnya supaya tidak menyimpang dari tujuan penelitian. Adapun yang menjadi batasan masalah adalah sebagai berikut ;

1. Menggunakan *Leap Motion* untuk mengimplementasikan teknik *Hand Motion Tracking* dengan fitur *hand gesture recognition* pada aplikasi pengenalan bangun ruang.
2. Aplikasi dibuat dengan menggunakan Unity 5, Blender versi 2.68a, GIMP versi 2.8 dan berjalan pada platform desktop dengan sistem operasi Windows minimal versi Windows 7.
3. Aplikasi pengenalan bangun ruang memuat konten materi yang ditujukan untuk pelajar Sekolah Dasar meliputi kubus, balok, prisma, limas, kerucut, tabung dan bola.

#### **1.5 Tujuan Penelitian**

Tujuan yang ingin dicapai pada pembuatan tugas akhir dengan judul Teknik Hand Motion Tracking pada Aplikasi Pengenalan Bangun Ruang dengan Leap Motion Controller adalah ;

1. Menghasilkan aplikasi 3D interaktif pengenalan bangun ruang dengan memanfaatkan teknologi Leap Motion yang ditujukan untuk pelajar sekolah dasar.
2. Meningkatkan efisiensi dan akurasi pengenalan gestur (*gesture recognition*) dengan bidang acuan (x,y), (x,z) dan (y,z) pada framerate 30 FPS serta eksekusi CPU dibawah 60ms dan GPU dibawah 10ms. Aplikasi diharapkan dapat mengenal gestur dengan persentasi ketepatan diatas 90%.
3. Menggunakan, menambahkan dan mengembangkan fitur secara interface maupun jenis-jenis gestur pada penelitian-penelitian sebelumnya yaitu *dynamic hand gesture* untuk diimplementasikan kedalam aplikasi. Aplikasi diharapkan dapat mengenal 10 jenis gestur. Fitur *hand gesture recognition* inilah yang menjadi alternatif interaksi manusia dan komputer yang *mouse-less* dan *keyboard-less*.

## 1.6 Manfaat Penelitian

1. Membantu kegiatan pembelajaran dalam memperkenalkan bangun ruang khususnya untuk pengajar maupun pelajar Sekolah Dasar.
2. Ikut serta dalam pengembangan aplikasi Leap Motion dengan teknik *hand motion tracking*.
3. Memaksimalkan fasilitas hardware Leap Motion untuk mendeteksi gerakan tangan (*hand motion tracking*) dan pengenalan pola gerak (*gesture recognition*).

## 1.7 Metode Penelitian

### 1.7.1 Metodologi Pengumpulan Data

#### 1. Studi Pustaka

Mencari dan mempelajari referensi melalui jurnal, buku cetak, *e-book*, artikel dan sumber internet yang terkait dengan *hand motion tracking* dan *Leap Motion Controller*.

### 1.7.2 Metodologi Pengembangan Aplikasi

Metode pengembangan aplikasi yang digunakan dalam penelitian ini adalah Metode Pengembangan Siklus Multimedia atau sering disebut MDLC (*Multimedia Development Life Cycle*). MDLC terdiri dari 3 fase atau tahapan yaitu :

#### 1. Tahap Pre-Production

Pada tahap ini penulis;

- a. Menggali ide dan konsep aplikasi multimedia yang akan dibuat ,
- b. Mencari aplikasi pembanding ,
- c. Merencanakan *engine* dan *software* pendukung yang akan digunakan serta menentukan *target platform* ,
- d. Pembuatan diagram alur (*flowchart*) aplikasi dan *storyboard*.

#### 2. Tahap Production

Pada tahap ini penulis;

- a. Mencari atau membuat objek gambar 2D maupun objek 3D yang diperlukan untuk dimasukkan ke dalam aplikasi ,
- b. Mendesain dan menyusun *3D world scene* ,
- c. Proses import model, implementasi *coding*, *hand motion tracking*, dan *hand gesture recognition* dengan metode

- sintaks, serta fitur-fitur Leap Motion ke dalam scene atau aplikasi ,
- d. Melakukan revisi desain ,
  - e. Membangun Alpha version, yaitu versi produk yang seluruh fungsinya telah dibangun, diimplementasikan dan diintegrasikan ,
  - f. Membangun Beta version, yaitu versi produk yang seluruh fungsinya telah dibangun, diimplementasikan dan diintegrasikan namun belum diuji secara keseluruhan.
  - g. Melakukan evaluasi terhadap performa aplikasi dan tingkat ketepatan pengenalan gestur tangan.

### **3. Tahap Post-Production**

Pada tahap ini penulis;

- a. Proses pengujian Beta version ,
- b. Proses revisi aplikasi berdasarkan hasil pengujian versi Beta ,
- c. Melakukan rilis Golden Master (*final product*) dari produk multimedia dan menyimpan semua material yang dipakai dalam proses produksi ,
- d. Produk dengan tahap Golden Master inilah yang akan dirilis melalui media yang memungkinkan ,
- e. Melakukan *maintenance* pada produk untuk mengevaluasi produk serta menjaga kualitas dan fungsi produk multimedia.

## 1.8 Sistematika Penulisan

### BAB I PENDAHULUAN

Bab ini berisi penjelasan mengenai latar belakang penulisan, identifikasi masalah, perumusan masalah, ruang batasan masalah, ruang lingkup, tujuan dan manfaat dari penulisan, metodologi penelitian yang digunakan, serta sistematika penulisan.

### BAB II TINJAUAN PUSTAKA

Studi pustaka dilakukan melalui literatur konsep yang menjabarkan tentang konsep teknik dan algoritma *motion tracking*, analisa 10 jurnal pembanding dan 5 aplikasi pembanding, serta dilengkapi berbagai teori umum yang mendasar tentang Leap Motion dan *Hand Motion Tracking* serta perangkat lunak yang mendukung dalam kaitannya dengan aplikasi yang akan dibuat.

### BAB III PERANCANGAN APLIKASI

Berisi pembahasan tentang rancangan aplikasi mulai dari tahap *pre-production* yaitu perancangan *flowchart*, *Storyboard* aplikasi. Lalu tahap *production* yang meliputi desain *sprite*, pemodelan 3D, penerapan 10 jenis *gesture control*, penulisan kode program, *bug fixing* dan proses ekspor kedalam format *executable*.

### BAB IV IMPLEMENTASI DAN PENGUJIAN

Berisi tahap *pasca-production* yaitu pembahasan tentang langkah-langkah instalasi, penggunaan, hingga pengujian aplikasi sehingga dihasilkan tingkat akurasi pengenalan gestur yang tinggi.

## **BAB V PENUTUP**

Bab ini berisi kesimpulan-kesimpulan yang didapat selama melakukan penelitian tugas akhir, seperti pembuatan program, dan penulisan laporan, serta saran-saran agar aplikasi ini dapat lebih baik lagi.

## **Daftar Pustaka**

Berisi daftar pustaka dan acuan dalam penulisan "*Hand Gesture Recognition* pada Aplikasi Pengenalan Bangun Ruang Dengan *Leap Motion Controller*"

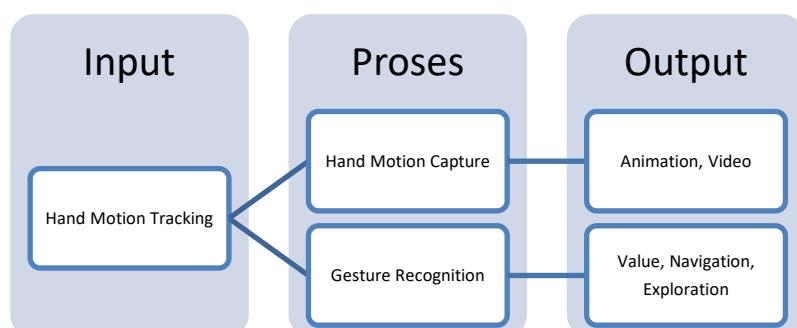
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Hand Motion Tracking

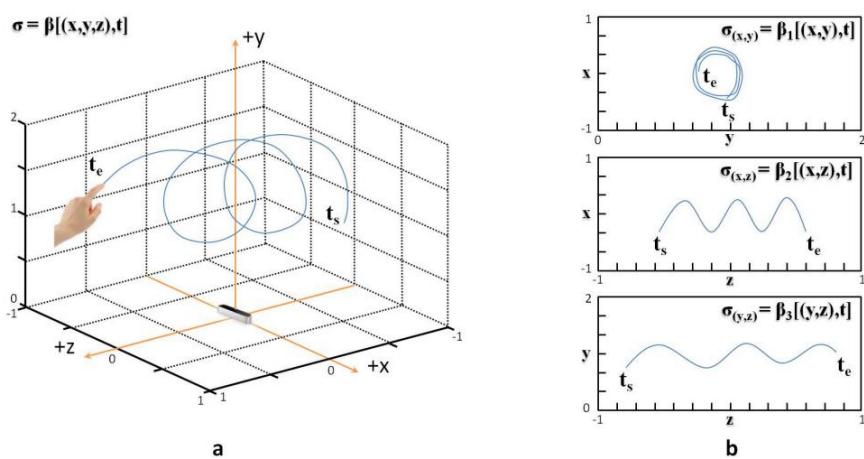
*Motion Tracking* adalah istilah yang digunakan untuk mendeskripsikan perekaman gerakan dan penerjemahan gerakan tersebut menjadi model digital. *Motion tracking* disimulasikan sebagai alat analisis photogrammetric dalam penelitian biomechanics pada tahun 1970-an dan 1980-an, serta meluas ke ranah edukasi, latihan, olahraga, dan baru saja ke ranah animasi komputer untuk televisi, sinema, dan video games [2].

Istilah *Hand Motion Tracking* merupakan salah satu istilah untuk salah satu implementasi teknik tracking and motion, dimana proses tracking dilakukan pada objek tangan manusia, *Hand Motion Tracking* dapat di implementasikan dalam banyak hal dan bidang, seperti dapat digunakan untuk mengenal bahasa isyarat, dan dapat dipakai menjadi teknik interaksi antara manusia dengan komputer dan lain sebagainya. Secara garis besar, teknik hand motion tracking meliputi hand motion capture dan gesture recognition. Sehingga pada aplikasinya dapat menghasilkan produk multimedia baik itu animasi maupun video. Ataupun menghasilkan produk aplikasi dengan kontrol gestur yang dapat dikembangkan menjadi fitur eksplorasi dan navigasi.



Gambar 2.1 Diagram penerapan *hand motion tracking*

Dalam literatur milik Danilo Avola, dkk dalam jurnal yang berjudul Markerless Hand Gesture Interface Based on LEAP Motion Controller [3] , membagi definisi layer hand motion tracking kedalam 3 layer yaitu Data Pre-Processing Layer (DPP-L), Feature Extraction and Recognition Layer (FER-L) serta Definition and Storage Layer (DS-L). DPP-L dibagi ke dalam 2 sub modul yaitu 4D Data Structur Manager (4D DSM) dan 4D Data Structure Processing(4D DSP).



**Gambar 2.2** Tracking ujung jari berdasarkan koordinat X, Y, Z

Gambar 2.2 merupakan contoh *tracking gesture* menggunakan tangan dengan ujung jari sebagai point. Diketahui bahwa (a) merupakan representasi pada ruang 3D (x,y,z) dengan (t) sebagai waktu. Sedangkan (b) adalah proyeksi pada refrensi datar 2D (x,y) , (x,z) dan (y,z) dengan acuan terhadap waktu (t).

Pada bidang 3 dimensi, alur tracking tersebut dapat dirumuskan sebagai berikut :

$$\sigma = \beta[(x, y, z), t] \quad (2.1)$$

dimana  $\beta$  merepresentasikan fungsi dari koordinant 3D spasial (x,y,z) yang diperoleh dari data posisi ujung jari tangan.

Sedangkan pada bidang 2 dimensi diperoleh 3 alur bergantung koordinat referensi dengan rumus :

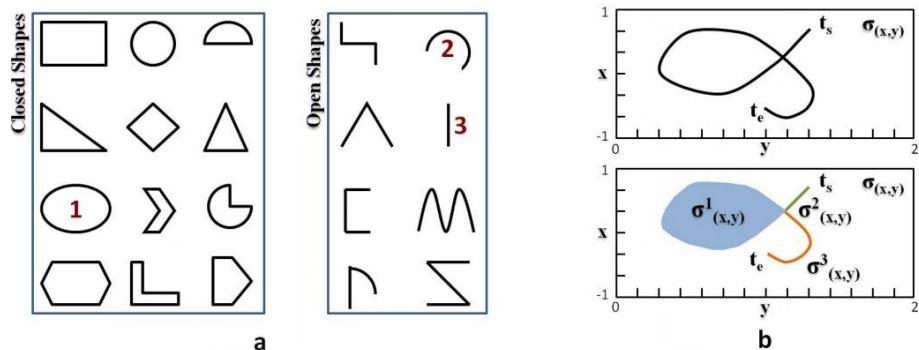
$$\sigma(x,y) = \beta_1[(x, y), t] \quad (2.2)$$

$$\sigma(x,z) = \beta_2[(x, z), t] \quad (2.3)$$

$$\sigma(y,z) = \beta_3[(y, z), t] \quad (2.4)$$

dimana  $\beta_1$ ,  $\beta_2$  dan  $\beta_3$  merepresentasikan fungsi dari koordinat spasial (bidang  $(x,y)$ ,  $(x,z)$  dan  $(y,z)$ )

Pada *Feature Extraction and Recognition Layer* menggunakan algoritma pengenalan pola gerak dengan menyimpan definisi pola yang telah direkam sebelumnya ke dalam library atau database (DS-L). Algoritma dengan melakukan ekstraksi ke pola 2 dimensi yaitu mengacu pada sumbu  $(x, y)$ .



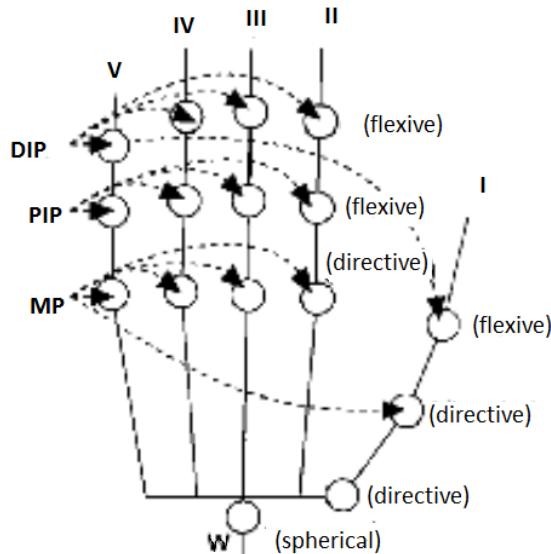
**Gambar 2.3** Pola gestur menghasilkan bentuk tertutup dan terbuka

Gambar 2.3 merupakan pola tertutup dan terbuka (a), dan ilustrasi dari algoritma dari pola tertutup yang tidak sempurna (b).

## 2.2 Struktur Tangan Manusia

Affan, pada jurnal Tracking Gerak Tangan Berbasis Pyramidal Lucas-Kanade [4] menjelaskan bahwa tangan manusia adalah sebuah struktur mekanik kompleks yang terdiri atas beberapa segmen tulang, ligamen-ligamen yang menghubungkan antar segmen tulang secara leluasa, otot-otot yang berperan sebagai motor gerak, tendon yang berperan untuk menghubungkan otot dengan tulang, dan kulit serta saraf-saraf halus yang menyelubungi otot dan tulang. Tulang-tulang saling terhubung pada pesendian dan tidak berubah ukurannya. Otot-otot menghasilkan tenaga penggerak dan menggerakkan sendi-sendi. Berdasarkan tipe gerakan dan rotasi yang mungkin dilakukan, sendi-sendi pada tangan manusia dapat diklasifikasikan sebagai *flexion*, *twist*, *directive*, atau *spherical*.

Contoh sendi bertipe *flexion* (engsel) dengan 1 DOF adalah lutut dan siku, sedangkan contoh sendi bertipe *twist* dengan 1 DOF adalah sendi pronation dari lengan bawah. Gerakan direktif dari sendi dengan 2 DOF menghasilkan gerakan *flexion* dengan arah lebih dari 2. Sendi *spherical* (Sendi Peluru), sebagaimana pada sendi bahu, memiliki 3 DOF dan dapat melakukan gerakan *directive* dan *twist* secara simultan. Gambar 1 mengilustrasikan tipe-tipe sendi pada tangan manusia. Setiap jari (II-V) memiliki empat DOF (dua pada MP, satu pada PIP, dan satu DIP). Sedangkan ibu jari (I) memiliki lima DOF (dua pada MP, dua pada PIP, dan satu pada DIP). Sendi pada pergelangan tangan dilibatkan karena tangan dipandang sebagai bagian terpisah dari lengan bawah. Berdasar pada klasifikasi gerak sendi tersebut, secara keseluruhan tangan manusia memiliki 27 DOF, termasuk 6 DOF untuk memindah posisi dan mengubah orientasi tangan secara keseluruhan.



**Gambar 2.4** Model Persendian Tangan dan Tipe Gerakannya

### 2.3 Hand Gesture Recognition

*Gesture recognition* (Pengenalan Gestur) adalah pola gerak natural manusia yang dikonversi menjadi bentuk digital supaya dikenal oleh komputer. *Gesture recognition* umumnya digunakan sebagai media input alternatif selain keyboard atau mouse. Istilah gesture recognition ini dapat dipersempit menjadi *hand gesture recognition*, yaitu pengenalan gerak khusus bangian tangan.

*Hand gesture recognition* diklasifikasikan kedalam tipe statis dan dinamis. Gestur statis adalah gestur yang cenderung diam, berbasis template atau pola berdasarkan vektor jari yang sudah ditentukan sebelumnya. Sedangkan gestur dinamis adalah gestur bergerak yang membutuhkan metode khusus yang dapat menerjemahkan gerakan.

Secara umum terdapat 2 metode algoritma pengenalan pola gerak (Gesture Recognition Algorithm) yaitu Metode Definition and Storage Layer (DS-L) dan Metode Sintaks. Metode DS-L adalah metode dengan melakukan perekaman gestur dan menyimpan pola dalam database, lalu

dipanggil satu persatu untuk dicocokan dengan gestur yang diinput. Sedangkan metode sintaks adalah metode penerjemahan gestur secara real-time menggunakan logika atau sintaks yang sudah didefinisikan. Hasil evaluasi terhadap 2 metode tersebut dapat disimpulkan pada tabel berikut.

**Tabel 2.1** Perbandingan Algoritma *Gesture Recognition*

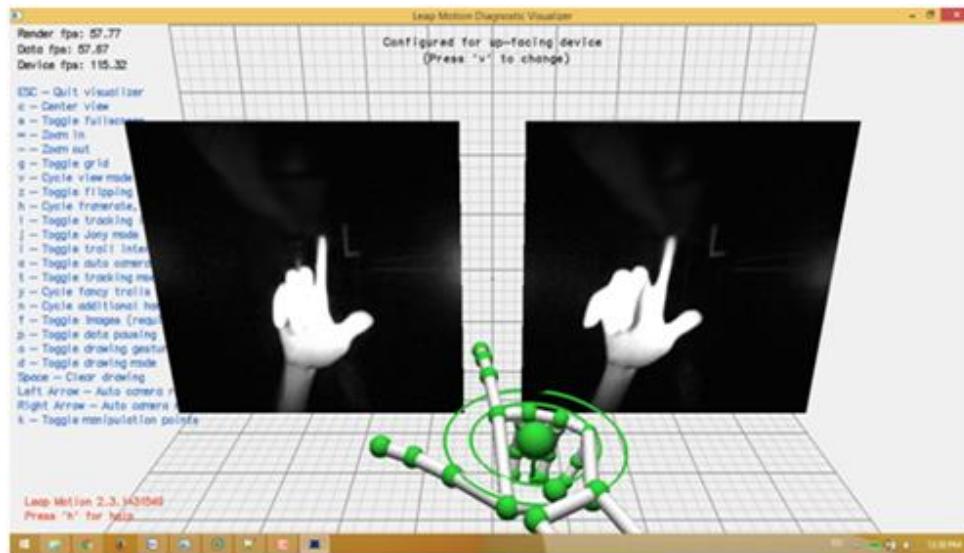
Pembanding	Metode DS-L	Metode Sintaks
Jumlah Variasi Gestur	Lebih banyak	Terbatas
Waktu Eksekusi	Terdapat <i>delay</i> yang lebih lama	Mendekati <i>Real-Time</i>
Penggunaan Memori	Lebih banyak	Lebih Sedikit
Teknik yang digunakan	Menggunakan teknik Gesture Recording artinya lebih mudah dalam pendefinisian gestur. Makin banyak gestur yang direkam maka performa semakin rendah.	Pendefinisian tiap gestur memiliki sintaks berbeda namun menghasilkan performa tinggi.
Tipe Gestur	Lebih cocok untuk gestur statis	Gestur statis maupun dinamis dapat difasilitasi dengan baik
Popularitas metode	Jarang dipakai	Lebih banyak dipakai

## 2.4 Leap Motion Controller

Leap Motion adalah sebuah perangkat keras yang dapat mendeteksi pergerakan jari-jari tangan dan memungkinkan user untuk berinteraksi dengan komputer dengan cara menggerakkan jari atau melambaikan tangan tanpa harus menyentuh atau menggunakan mouse. Leap Motion dirilis untuk dijual kepada umum pada bulan Juli tahun 2013 oleh Leap Motion, Inc.

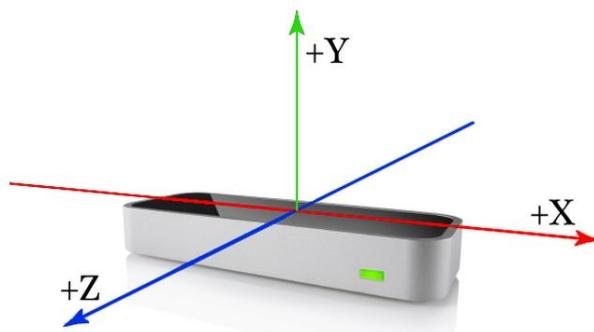
Leap Motion Controller menggunakan pencitraan infrared (IR) untuk menentukan posisi dari *predefined-objects* pada ruang terbatas

secara real-time. Secara teknis algoritma yang digunakan sejauh ini masih dirahasiakan terkait dengan hak paten Leap Motion, Inc.



**Gambar 2.5** Citra yang Terdeteksi Leap Motion

Leap Motion menggunakan sistem koordinat Kartesian tangan-kanan untuk menentukan posisi pada jangkauan sensor. Jarak efektif penggunaan Leap Motion adalah 25-600 milimeter di atas sensor dengan sudut jangkauan sekitar  $150^{\circ}$  [5].



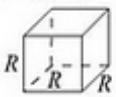
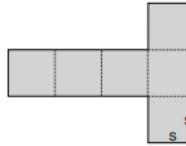
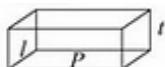
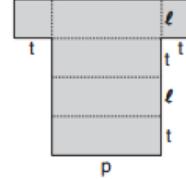
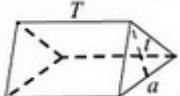
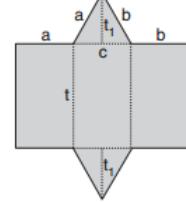
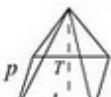
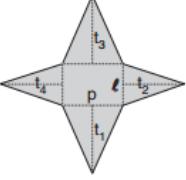
**Gambar 2.6** Sistem koordinat *Leap Motion Controller*

Sumber : <http://www.leapmotion.com>

## 2.5 Konsep Bangun Ruang

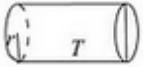
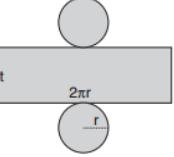
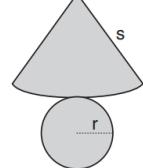
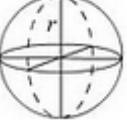
Pada buku oleh Nur Laila Indah Sari berjudul Asyiknya Belajar Bangun Ruang Sisi Datar yang diterbitkan oleh PT Balai Pustaka [6]. Bangun ruang sisi datar adalah bangun ruang yang memiliki sisi berbentuk datar (bukan sisi lengkung). Bangun ruang sisi datar yang dimaksud meliputi kubus, balok, prisma, dan limas.

**Tabel 2.2** Bangun Ruang Sisi Datar

Nama Bangun	Definisi	Luas dan Volume	Jaring-jaring
Kubus 	Kubus adalah bangun ruang yang semua sisinya berbentuk persegi dan memiliki rusuk-rusuk yang sama panjang.	$L = 6R^2$ $V = R^3$	
Balok 	Balok adalah bangun ruang yang memiliki tiga pasang sisi berhadapan yang sama bentuk dan ukurannya dimana setiap sisinya berbentuk persegi panjang.	$L = 2(pt + pl + pt)$ $V = p l t$	
Prisma Segitiga 	Prisma adalah bangun ruang yang memiliki alas dan atap yang sama bentuk dan ukurannya. Semua sisi bagian samping sebuah prisma berbentuk persegi panjang.	$V = \text{Luas alas} \times T$ Luas alas bergantung pada bentuk alas	
Limas Segiempat 	Limas adalah bangun ruang yang terdiri dari bidang alas dan bidang sisi tegak yang berbentuk segitiga	$\frac{1}{3} \times \text{Luas alas} \times T$	

Selain bangun ruang sisi datar yang sudah tertera pada tabel di atas keempat penulis mengambil juga literatur bangun ruang sisi lengkung dalam buku berjudul Gemar Matematika 6 yang diterbitkan oleh Pusat Perbukuan Departemen Pendidikan Nasional tahun 2008 [7] .

**Tabel 2.3** Bangun Ruang Sisi Lengkung

Nama Bangun	Definisi	Luas dan Volume	Jaring-jaring
Tabung (Silinder) 	Tabung merupakan prisma tegak yang alasnya berbentuk lingkaran.	$L = 2 \pi r T$ $V = \pi r^2 T$	
Kerucut 	Kerucut tersusun dari dua bangun datar, yaitu lingkaran sebagai alas dan selimut yang berupa bidang lengkung (juring lingkaran).	$L = \pi r s + 4 \pi r^2$ dimana s adalah apotema $V = 1/3 \pi r^2 T$	
Bola 	Bola merupakan bangun ruang sisi lengkung yang dibatasi oleh satu bidang lengkung. Bola dapat dibentuk dari bangun setengah lingkaran yang diputar sejauh 360° pada garis tengahnya.	$L = 4 \pi r^2$ $V = 4/3 \pi r^3$	Untuk bangun ruang bola tidak dapat dibuat jaring-jaringnya karena bentuk lengkung 3 dimensi pada bola tidak dapat diubah ke dalam bentuk datar atau bentuk 2 dimensi secara sempurna.

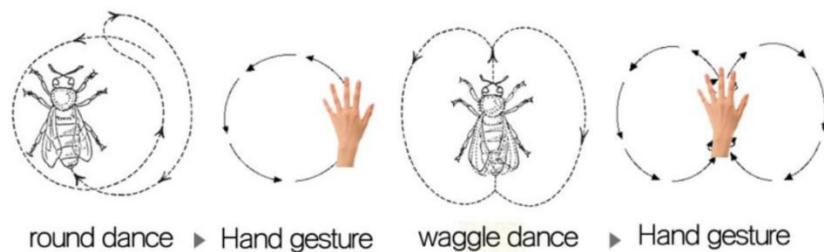
## 2.6 LITERATUR JURNAL

Penulis mengambil 10 literatur jurnal sebagai pembanding. Jurnal yang diambil adalah jurnal yang menggunakan Leap Motion Controller sebagai device untuk mendukung pelacakan gerak tangan hand motion tracking dengan metode markerless. Selain menganalisa presentasi ketepatan aplikasi dalam mengenal gestur, penulis juga menganalisa fitur yang ada pada jurnal untuk dijadikan acuan aplikasi yang akan dibuat.

### 2.6.1 Literatur Jurnal 1

Ok-Hue Cho menggunakan Leap Motion controller dalam jurnal berjudul *Development of Serious Game for Kids using Leap Motion based on Honey Bee Dance*<sup>[8]</sup>. Permainan edukatif ini dirancang untuk menumbuhkan pemahaman ada anak-anak tentang tingkah laku lebah pekerja bagaimana mereka mencari makanan. Dengan menggunakan Leap Motion sebagai alat input dan kontrol maka pemain akan lebih tertarik dengan game ini.

Menurut Janet H. Murray pada jurnal ini, 'tenggelam' dalam suatu permainan adalah suatu pengalaman yang sangat menyenangkan bagi user sehingga user merasa hidup di dalam dunia fantasi. Seorang pemain yang merasa 'tenggelam' dalam game akan terus tertarik untuk melanjutkan game tersebut. Aspek tersebut merupakan salah satu kunci sukses sebuah game.



Gambar 2.7 Control Gesture pada game *Honey Bee Dance*

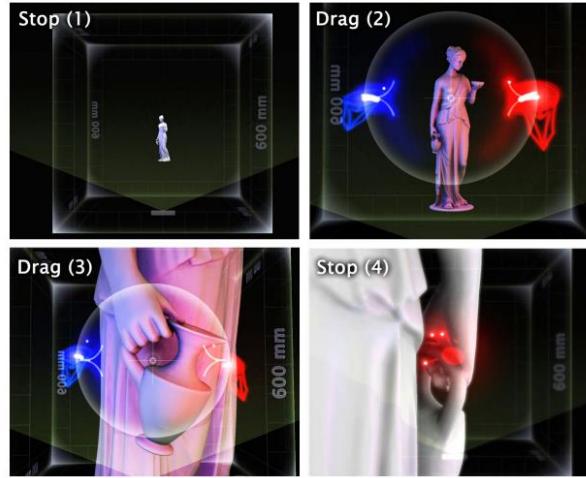
Gambar 2.7 menunjukkan bahwa game yang dirancang menggunakan *hand gesture* untuk menggerakkan objek lebah. Dalam game tersebut terdapat 2 tipe gerakan yaitu *round dance* dan *waggle dance*.

### 2.6.2 Literatur Jurnal 2

Bruno Fanini dalam Jurnal berjudul *A 3D Interface to Explore and Manipulate multi-scale Virtual Scenesusing the Leap Motion Controller*<sup>[9]</sup>. Melakukan eksplorasi objek secara virtual adalah topik yang kompleks dan membutuhkan perhatian khusus. Jika menggunakan standar peripheral ada beberapa model navigasi yang umum diterapkan, misalnya pan - rotate - zoom menggunakan mouse. Namun mouse masih memiliki keterbatasan karena hanya dapat melakukan beberapa aksi seperti click dan drag.

Dengan adanya Leap Motion, penulis jurnal berusaha mengembangkan metode untuk melakukan eksplorasi dalam area virtual 3D, terutama dengan jangkauan yang lebih luas.

Framework yang digunakan pada literatur jurnal kedua ini adalah OpenSceneGraph (OSG). OSG merupakan open source 3D rendering middleware dan salah satu scene graph API yang terbaik. Framework ini umum digunakan untuk mengolah objek virtual 3D secara real-time sekaligus menganalisis performa, portabilitas dan skalabilitas menyediakan beberapa set fitur dengan fungsionalitas yang sangat lengkap.



**Gambar 2.8** navigasi *Drag and Stop* pada ruang virtual 3D

Gambar 2.8 memperlihatkan navigasi *Drag and Stop* pada ruang 3D memanfaatkan Leap Motion. Jurnal tersebut menjelaskan bahwa Leap Motion controller dapat digunakan sebagai alat untuk melakukan eksplorasi (navigasi) dan manipulasi lebih leluasa terhadap objek 3D virtual.

### 2.6.3 Literatur Jurnal 3

Jurnal berjudul *Hand Gesture Recognition With Leap Motion And Kinect Devices* milik Giulio Marin dan Fabio Dominio<sup>[10]</sup>, mencoba megabungkan kemampuan Kinect dan Leap Motion untuk meningkatkan akurasi pengenalan gestur tangan. Namun penulis hanya akan mengambil referensi tingkat ketepatan pengenalan gestur yang diperoleh oleh Leap Motion. Pada jurnal ini didefinisikan 10 gestur statis dalam database (DS-L). Diketahui bahwa fitur gestur yang terdapat pada jurnal diperoleh akurasi sebesar 80,86%.



**Gambar 2.9** Pengenalan Gestur dengan Leap Motion dan Kinect

#### 2.6.4 Literatur Jurnal 4

Ayung Candra pada penelitian berjudul Hand Motion Tracking Berbasis Gesture Understanding Menggunakan Leap Gesture Untuk Galeri Batik Virtual 3 Dimensi [11]. Tujuan dari penelitian tersebut diantaranya adalah Merancang Sistem virtual yang Interaktif sebagai pengganti Keyboard dan Mouse dalam menjalankan Galeri batik Virtual 3 Dimensi. Perancangan Desain Virtual untuk galeri dibuat dalam tiga ruang yang terdiri atas ruang satu sebagai ruang interaksi menu, ruang dua sebagai ruang pemilihan motif dan ruang tiga sebagai ruang untuk galeri karya.

Metode yang digunakan untuk visualisasi Galeri adalah Normal Mapping dengan memanfaatkan Game Engine Unity3D. Untuk aplikasi dan pengontrol virtual, peneliti mengembangkan hand motion tracking menggunakan Perangkat Leap gesture dalam Leap Motion controller.

Penelitian ini dibagi menjadi 3 tahap, diantaranya, Proses perancangan desain awal (modelling 3D) menggunakan teknik Normal mapps, Proses penentuan skenario gesture dalam bentuk FSM (Finite-state machine) pada setiap ruang dan desain gesture yang memanfaatkan hand motion tracking dalam perangkat Leap Motion controller.

Hasil dari penelitian ini, didapatkan respon untuk Hand Motion Tracking dalam Galeri Batik 3 Dimensi yang dikelompokkan menjadi 4 konsep hand gesture understanding diantaranya adalah *Tap Gesture, Flyhand, Trusther* dan *Hold*.

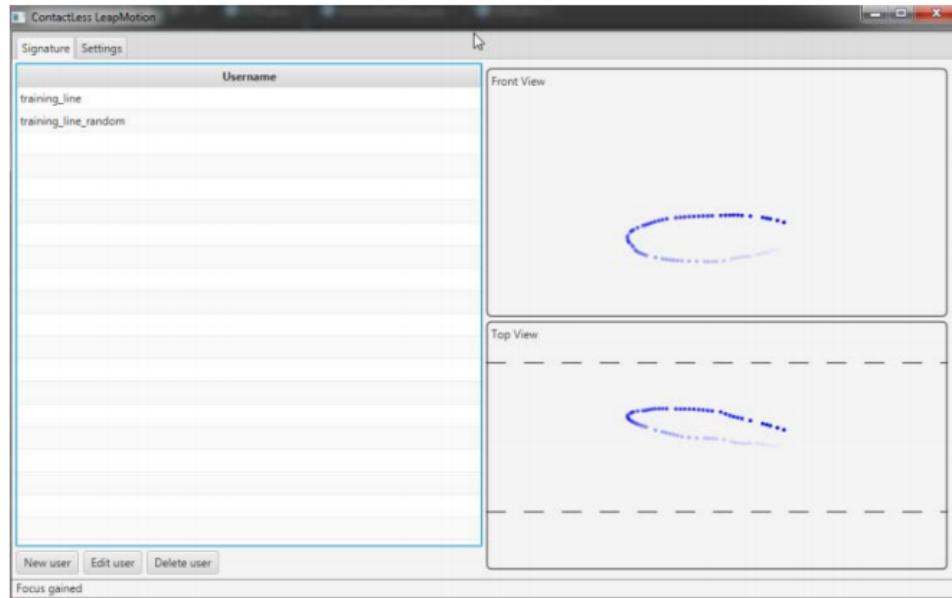
### 2.6.5 Literatur Jurnal 5

Alexandros Preventis, Kostas Stravoskoufos dari Technical University of Crete pada jurnal berjudul *Personalized Motion Sensor Driven Gesture Recognition in the FIWARE Cloud Platform* [12]. Jurnal ini berkonsep pada cloud application (FIWARE) yang dapat diintegrasikan dengan beberapa mocap device seperti Leap Motion, Microsoft Kinect dan Asus Xtion PRO. Hal ini memungkinkan pengembangan aplikasi secara cloud dan menyimpan gesture library pada cloud.

Aplikasi dikembangkan dengan menggunakan sumber data JSON (Javascript) yang memungkinkan user melakukan HTTP requests yaitu GET, POST dan DELETE. Variasi gestur yang terdapat dalam aplikasi adalah *Hand Number, Finger Number, Hand Distance, Hand Direction, Palm Normal, Palm Position, Sphere Center, Spere Radius, Pitch, Roll* dan *Yaw*.

### 2.6.6 Literatur Jurnal 6

Pada Jurnal Alexandre Burry yang berjudul *Contactless Sensor LeapMotion* [13] didapatkan bahwa ketepatan *Leap Motion* dalam membaca gestur dikatakan dapat mencapai 80% melalui algoritma Dynamic Time Warping (DTW) dan Support Vector Machines (SVM). Ketika gerakan dibuat di atas sensor LeapMotion , pengguna diinformasikan dengan hasil proses pengenalan .Gestur selanjutnya akan diterjemahkan, apakah merupakan gestur yang terdaftar pada sistem atau tidak.

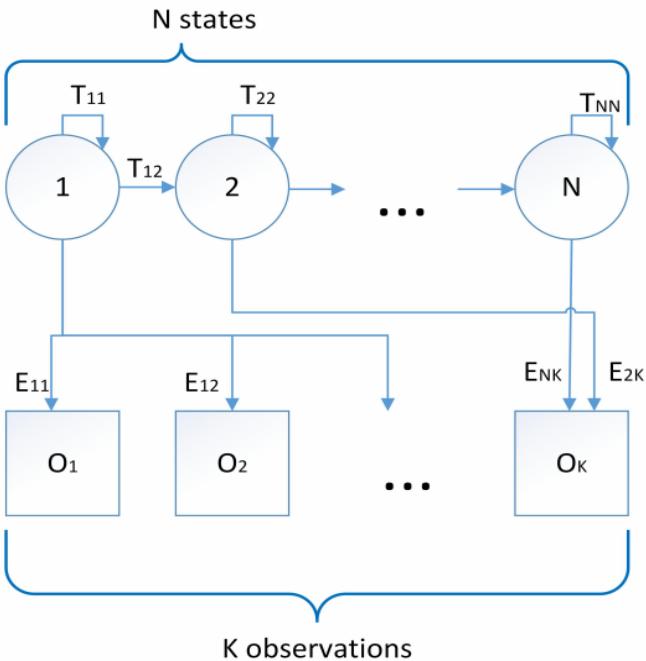


**Gambar 2.10** *Tracking Record* pada aplikasi

### 2.6.7 Literatur Jurnal 7

Michal dan kawan-kawan dalam jurnal *Gesture Recognition Library for Leap Motion Controller* [14] mendefinisikan 6 buah gestur dinamis yaitu "123", door, circle, scissors, gun, moving the object. Dengan mengambil 120 sampel per gesture, dilakukan pengujian dengan Hidden Markov Model (HMM).

Penelitian ini menguji akurasi ketepatan gesture dengan menambah variasi jumlah state (K) pada HMM. Dalam percobaannya diambil 4 nilai yang berbeda. Dari pengujian tersebut didapatkan persentase ketepatan dari setiap variasi nilai K. Nilai K yang melebihi 10 ternyata tidak meningkatkan akurasi terlalu signifikan dibandingkan yang dibawah 10.



**Gambar 2.11** Hidden Markov Model Sederhana

### 2.6.8 Literatur Jurnal 8

Ayung Candra dengan jurnal berjudul *3-Dimensional Virtual Classroom for Journey* [15]. Penelitian ini dikembangkan untuk menghadirkan teknik dan visualisasi desain kelas secara 3 dimensi dengan mengelompokkan beberapa perintah kepada Agen untuk menggantikan keberadaan makhluk hidup lain dalam mengontrol virtual classroom tersebut. Rancangan Agen dalam penelitian ini berupa manusia lain yang diambil dari assetstore Unity3D dikombinasikan dengan desain interaksi pengenalan gesture sebagai salah satu alat pengontrol jalanya aplikasi dalam virtual classroom, dimana desain interaksi memanfaatkan alat bantu Leapmotion controller yang terdiri atas Key Tap dan tap gesture, swipe gesture, hold gesture dan circle gesture.

### 2.6.9 Literatur Jurnal 9

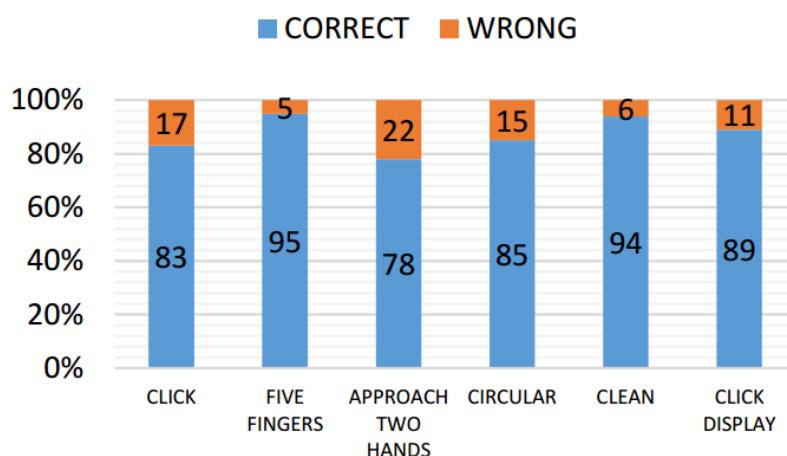
Pedro Miguel dengan jurnal berjudul *Analysis and Evaluation of Gesture Recognition using LeapMotion*<sup>[16]</sup>. Jurnal ini membantu penulis dalam mengetahui seberapa tepat Leap Motion dalam mengenal gestur tangan. Pada pengujinya, ditentukan 6 jenis gestur yaitu: *click*, *five fingers*, *approach hands*, *circular*, *clean* dan *click display*.

Gestur swipe dapat dikenal dengan gerakan tangan secara vertikal maupun horizontal menggunakan jari telunjuk maupun telapak tangan.

$$H = |D_x| > |D_y| \quad (2.5)$$

Definisi pergerakan sirkular adalah gerakan jari memutar dengan radius minimal 5mm dengan minimum lengkungan  $1.5\pi$  radian. Jika sudut kurang dari  $90^\circ$  maka didefinisikan sebagai *clockwise* (searah jarum jam) sedangkan kebalikannya adalah *counter-clockwise*.

$$\frac{D.C}{\|D\| \|C\|} \leq \frac{\pi}{2} \quad (2.6)$$



**Gambar 2.12** Grafik ketepatan Leap Motion dalam mengenal gestur

Dalam jurnal ini disimpulkan bahwa ketepatan *Leap Motion Controller* dalam mengenal gestur rata-rata dapat mencapai 86.1 %.

### 2.6.10 Literatur Jurnal 10

Louis Sousa, dkk. pada jurnal berjudul *A 3D Gesture Recognition Interface for Energy Monitoring and Control Applications* [17]. Interface didesain untuk menaggapi 6 tipe swipe gesture yang berbeda. 3 gestur adalah kebalikan dari 3 lainnya.



**Gambar 2.13** Eksplorasi Ruangan Secara Virtual

- (1) "Pilih atau batal pilih" adalah dari swipe dari atas ke bawah atau kebalikannya yang artinya mengacu pada sumbu y.
- (2) "Pilih lantai yang berbeda" adalah gerakan tangan dari depan ke belakang atau kebalikannya yang artinya mengacu pada sumbu z.
- (3) "Pilih objek berikutnya atau sebelumnya" didefinisikan sebagai gerakan swipe ke kanan atau kekiri yang artinya mengacu pada sumbu x.

## 2.4 LITERATUR APLIKASI

Penulis mengambil 10 literatur aplikasi sebagai pembanding. Aplikasi yang menjadi referensi adalah aplikasi yang sudah didistribusikan secara umum dan dapat didownload di situs <http://www.leapmotion.com>. Sama seperti literatur jurnal, pada literatur aplikasi penulis menganalisa *interface*, variasi pengenalan gestur, dan fitur lainnya.

### 2.7.1 LiteraturAplikasi 1

Nama Aplikasi : 3D Geometry

Developer : Quwei Lab

Versi : 1.0

Platform : Windows



**Gambar 2.13** Aplikasi 3D Geometry

Sumber : <http://leapmotion.com>

Keterangan :

Aplikasi 3D Geometry ini [18] adalah pengembangan dari versi yang tidak menggunakan Leap Motion sebagai alat kontrolnya. Penulis melihat

aplikasi tersebut cukup lengkap dan menarik secara desain interface. User dapat mempelajari bangun ruang, informasi bangun ruang, rumus menghitung luas dan volume. Namun masih ada beberapa kekurangan yaitu variasi gestur yang masih minim serta tidak adanya contoh bangun ruang dalam kehidupan sehari-hari dan juga tidak memiliki fitur *virtual hands*.

### 2.7.2 Literatur Aplikasi 2

Nama Aplikasi : GeoCrazy

Developer : GrowingSoft

Versi : 1.0

Platform : Windows



**Gambar 2.15** Aplikasi GeoCrazy  
Sumber : <http://leapmotion.com>

Keterangan :

GeoCrazy [19] adalah aplikasi edukatif untuk pengenalan materi-materi geografi dan sejarah dunia. Dalam aplikasi tersebut sudah menggunakan *gesture recognition*. Yaitu fungsi click digantikan oleh gerakan *grab* (menggenggam). Selain itu walaupun sudah terdapat fitur

tombol navigasi yang cukup lengkap namun karena belum terdapat *virtual hands* seleksi pada menu navigasi masih menggunakan kursor.

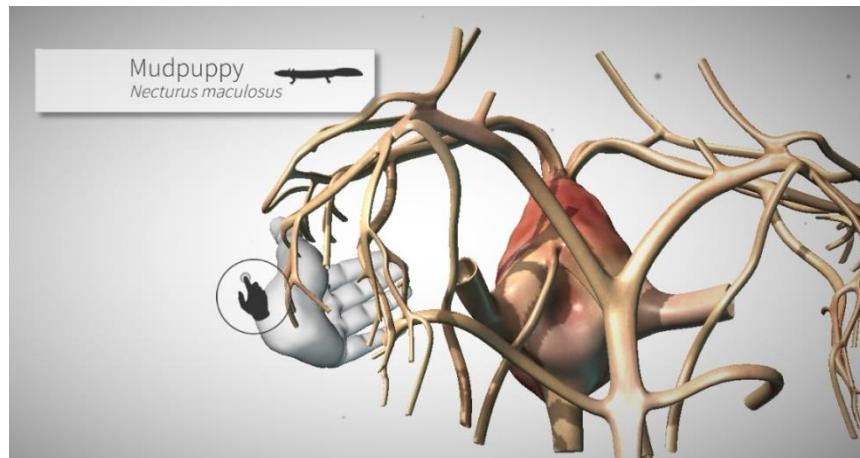
### 2.7.3 Literatur Aplikasi 3

Nama Aplikasi : Form and Function 3D

Developer : Brendan Polley

Versi : 1.0

Platform : Windows



**Gambar 2.16** Aplikasi Form and Function 3D

Keterangan :

Form and Function 3D [20] adalah aplikasi edukatif untuk mempelajari anatomi makhluk hidup. Versi gratis hanya menyajikan anatomi jantung pada hewan. Pada aplikasi tersebut terlihat sudah memiliki fitur *virtual hands* untuk melakukan eksplorasi di ruang 3D virtual seperti gestur *swipe action*.

#### 2.7.4 Literatur Aplikasi 4

Nama Aplikasi : Cyber Science - Motion: Zoology

Developer : Sunset Lake Software

Versi : 1.0

Platform : Windows, OS X



**Gambar 2.17** Aplikasi Zoology  
Keterangan :

Aplikasi Zoology ini [21] belum menggunakan *gesture recognition* karena masih mengandalkan tombol spasi untuk melakukan seleksi ke bagian tubuh hewan yang akan dieksplorasi. Hal tersebut menandakan bahwa pemanfaatan Leap Motion pada aplikasi belum maksimal. Selain itu, Zoology juga belum memiliki tombol navigasi serta tidak adanya *virtual hands*.

#### 2.7.5 Literatur Aplikasi 5

Nama Aplikasi : Molecules

Developer : Sunset Lake Software

Versi : 1.0

Platform : OS X



**Gambar 2.18** Aplikasi Molecules  
Sumber : <http://leapmotion.com>

Keterangan :

Molecules [22] adalah aplikasi visualisasi 3D bentuk molekul. User dapat melakukan gerakan memutar dan grab (menggenggam), artinya sudah dilengkapi dengan fitur *gesture recognition*. Aplikasi ini saat ini mendukung file struktur molekul dalam format file PDB , SDF , dan XYZ . File-file ini dapat didownload dari sejumlah sumber , termasuk RCSB Protein Data Bank dan NCBI PubChem .

## 2.6 Tabel Jurnal Pembanding

**Tabel 2.5** Hasil Perbandingan Jurnal Pembanding 1 sampai 5

Pembanding	Jurnal	Jurnal 2	Jurnal 3	Jurnal 4	Jurnal 5
Perangkat Hand Motion Tracking	Leap Motion	Leap Motion	Leap Motion , Kinect	Leap Motion	Leap Motion
Metode Hand Motion Tracking	Markerless	Markerless	Markerless	Markerless	Markerless
Metode Gesture Recognition	Tidak Diketahui	Sintaks	DS-L	Tidak Diketahui	DS-L
<i>Virtual hands</i>	-	✓	✓	-	-
Gesture Recognition	✓	✓	-	✓	✓
- Key Tap / Click	-	-	-	✓	-
- Swipe	-	✓	-	✓	-
- Circular	✓	✓	-	-	✓
- Clap	-	-	-	-	-
- Double Outward Swipe	-	-	-	-	-
- Fly Control	-	-	-	-	-
- Push	-	-	-	-	-
- Pinch	-	-	-	-	-
- Release	-	-	-	-	-
- Pull	-	-	-	-	✓
Eksplorasi Ruang 3D	-	✓	-	-	-
- Zoom In	-	✓	-	-	-
- Zoom Out	-	✓	-	-	-
- Rotate	-	✓	-	-	-
Manipulasi Objek 3D	-	✓	-	-	-
- Translate	-	✓	-	-	-
- Rotate	-	✓	-	-	-
- Scale	-	✓	-	-	-
Ketepatan Mengenal Gestur	-	-	80,86	-	71.1 %

Keterangan :

Jurnal 1 : *Development of Serious Game for Kids using Leap Motion based on Honey Bee Dance*

- Jurnal 2 : *A 3D Interface to Explore and Manipulate multi-scale Virtual Scenes using the Leap Motion Controller*
- Jurnal 3 : *Hand Gesture Recognition With Leap Motion And Kinect Devices*
- Jurnal 4 : *Hand Motion Tracking berbasis Gesture Understanding Menggunakan Leap Gesture Untuk Galeri Batik Virtual 3 Dimensi*
- Jurnal 5 : *Personalized Motion Sensor Driven Gesture Recognition in the FIWARE Cloud Platform*

Dari literatur jurnal pertama hingga kelima, seluruhnya menggunakan Leap Motion device dengan metode markerless. Artinya dengan Leap Motion, user tidak perlu lagi menggunakan marker untuk mendekripsi posisi dan pergerakan tangan manusia. Pada jurnal satu, interface masih disajikan dalam bentuk 2D. Jurnal kedua hingga keempat adalah aplikasi berbasis 3D. Sementara pada jurnal kelima berbentuk *internet application (cloud)*.

Fitur *gesture recognition* (pengenalan pola gerak) terdapat pada jurnal satu sampai lima dengan jurnal ketiga yang mempunyai hitungan ketepatan sebesar 80,86 dan jurnal kelima sebesar 71,1%. Kedua jurnal menggunakan metode DS-L (*Definition and Storage Layer*). Fitur ini penting sebagai alternatif navigasi pengganti tombol keyboard. Seperti tombol enter bisa digantikan dengan gerakan mengepal, atau tombol navigasi arah kanan dan kiri dapat diganti dengan *swipe gesture*.

**Tabel 2.5** Hasil Perbandingan Jurnal Pembanding 6 sampai 10

Pembanding	Jurnal 6	Jurnal 7	Jurnal 8	Jurnal 9	Jurnal 10	Usulan
Perangkat Hand Motion Tracking	Leap Motion					
Metode Hand Motion Tracking	Markerless	Markerless	Markerless	Markerless	Markerless	Markerless
Metode Gesture Recognition	DS-L	DS-L	Sintaks	Sintaks	Sintaks	Sintaks
<i>Virtual hands</i>	-	-	-	-	-	✓
Gesture Recognition	✓	✓	✓	✓	✓	✓
- Key Tap / Click	-	-	✓	✓	-	✓
- Swipe	-	✓	✓	✓	✓	✓
- Circular	✓	✓	✓	✓	-	✓
- Clap	-	-	-	-	-	✓
- Double Inward Swipe	-	-	-	-	-	✓
- Fly Control	-	-	-	-	-	✓
- Pinch	-	-	-	-	-	✓
- Release	-	-	-	-	-	✓
- Push	-	-	-	-	✓	✓
- Pull	-	-	-	-	✓	✓
Eksplorasi Ruang 3D	-	-	-	-	✓	✓
- Zoom In	-	-	-	-	✓	✓
- Zoom Out	-	-	-	-	✓	✓
- Rotate	-	-	-	-	✓	✓
Manipulasi Objek 3D	-	-	✓	-	-	✓
- Translate	-	-	✓	-	-	✓
- Rotate	-	-	✓	-	-	✓
- Scale	-	-	-	-	-	✓
Ketepatan Mengenal Gestur	80%	77.2%	-	86.1 %		> 90%

Keterangan :

Jurnal 6 : *Contactless Sensor LeapMotion*

Jurnal 7 : *Gesture Recognition Library for Leap Motion Controller*

- Jurnal 8 : *3-Dimensional Virtualclassroom for Journey Based on Hand Gesture Understanding using Leap Motion Controller*
- Jurnal 9 : *Analysis and Evaluation of Gesture Recognition using LeapMotion*
- Jurnal 10 : *A 3D Gesture Recognition Interface for Energy Monitoring and Control Applications*

Dari segi ketepatan mengenal gestur, jurnal keenam memiliki ketepatan Pada jurnal keenam sampai sepuluh, tidak ada fitur ataupun kemampuan aplikasi pada jurnal dalam melakukan eksplorasi ruang 3D maupun dalam memanipulasi objek 3D.

Persentasi ingkat ketepatan pengenalan gestur pada jurnal keenam adalah 80%. Sedangkan pada jurnal ketujuh mencapai 77,2% dan *gesture recognition* pada jurnal kesembilan dengan metode sintaks mencapai 86,1%. Metode inilah yang menghasilkan tingkat ketepatan tertinggi.

Namun kelemahan dari hampir seluruh jurnal yang dijadikan literatur adalah tidak adanya fitur *virtual hands*. Tampilan *virtual hands* hanya disajikan pada jurnal kesembilan. Namun jurnal kesembilan tidak menggunakan perangkat Leap Motion dan masih menerapkan metode *marker-based*. *Virtual hands* adalah fitur Leap Motion yang menjadi antarmuka utama. Dengan *virtual hands*, user dapat berinteraksi secara lebih aktif didalam dunia virtual yang telah diprogram sebelumnya.

## 2.7 Tabel Aplikasi Pembanding

Agar lebih memahami kelebihan maupun kekurangan yang terdapat pada aplikasi-aplikasi tersebut maka disajikan tabel pembanding sebagai berikut:

**Tabel 2.6 Hasil Perbandingan Aplikasi Pembanding**

Pembanding	Aplikasi 1	Aplikasi 2	Aplikasi 3	Aplikasi 4	Aplikasi 5	Usulan
Perangkat Hand Motion Tracking	Leap Motion					
Metode Hand Motion Tracking	Markerless	Markerless	Markerless	Markerless	Markerless	Markerless
<i>Virtual hands</i>	-	-	✓	-	-	✓
Gesture Recognition	✓	✓	✓	-	-	✓
- Key Tap / Click	✓	-	✓	-	-	✓
- Swipe	✓	-	✓	-	-	✓
- Circular	✓	-	-	-	-	✓
- Clap	-	-	-	-	-	✓
- Double Outward Swipe	-	-	-	-	-	✓
- Fly Control	-	-	-	-	-	✓
- Pinch	-	-	-	-	-	✓
- Release	-	-	-	-	-	✓
- Push	-	-	-	-	-	✓
- Pull	-	-	-	-	-	✓
Fitur Navigasi	✓	✓	-	-	✓	✓
Eksplorasi Ruang 3D	-	✓	-	✓	✓	✓
- Zoom In	-	✓	-	✓	✓	✓
- Zoom Out	-	✓	-	✓	✓	✓
- Rotate	-	✓	-	✓	✓	✓
Manipulasi Objek 3D	✓	✓	✓	✓	✓	✓
- Translate	-	-	✓	✓	-	✓
- Rotate	✓	-	✓	✓	-	✓
- Scale	-	-	-	-	-	✓
Platform	Windows	Windows	Windows	Windows,	OSX	Windows

Keterangan :

Aplikasi 1 : 3D Geometry

Aplikasi 2 : GeoCrazy

Aplikasi 3 : Form and Function 3D

Aplikasi 4 : Cyber Science - Motion: Zoology

Aplikasi 5 : Molecules

Setelah dilakukan pengamatan terhadap kelima literatur aplikasi, diketahui bahwa seluruhnya menggunakan perangkat Leap Motion dengan metode *markerless*. Seluruh aplikasi hanya tersedia dalam versi desktop dengan sistem operasi windows maupun OSX. Untuk fitur *virtual hands*, hanya literatur aplikasi kedua yang sudah menggunakan *virtual hands*. Kemampuan aplikasi dalam memanipulasi objek 3D terdapat pada seluruh aplikasi pembanding. Tidak terdapat fitur navigasi pada aplikasi keempat dan keempat. Selain itu diamati pula kemampuan aplikasi dalam melakukan eksplorasi ruang 3D yaitu kemampuan dalam mengaplikasikan fitur *zoom-in*, *zoom-out*, *camera orientation* dan sebagainya. Fitur tersebut terdapat pada aplikasi ketiga, keempat dan kelima.

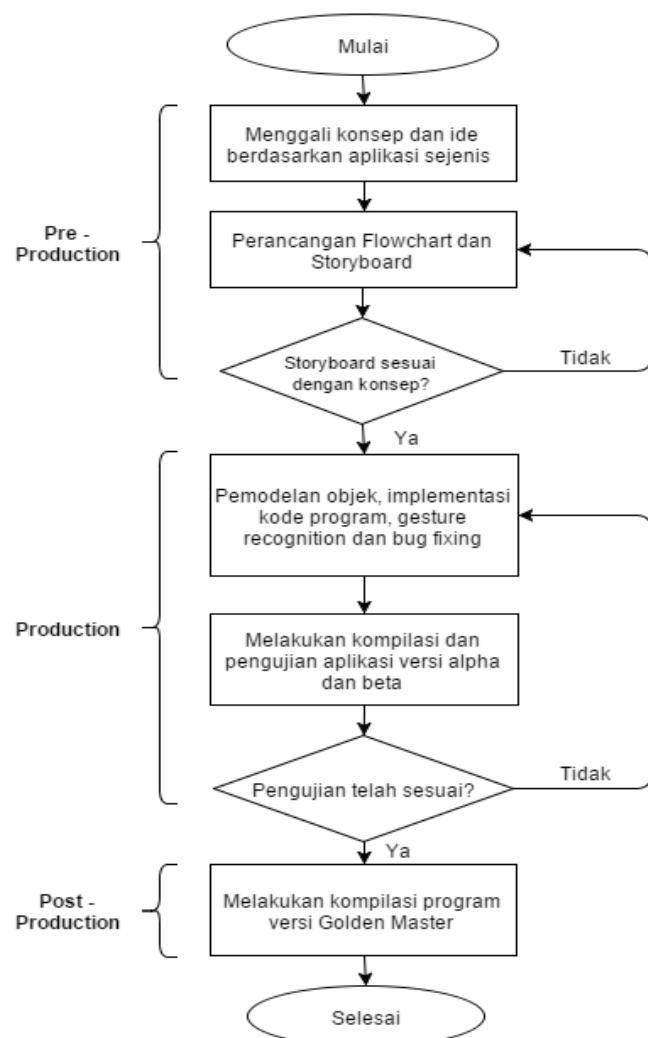
Pada aplikasi ketiga dan keempat tidak terlihat adanya menu navigasi. Lalu pada aplikasi keempat diketahui pula bahwa tidak ada fitur pengenal gestur (*gesture recognition*). Sedangkan *virtual hands* hanya terdapat pada aplikasi ketiga.

## BAB III

### PERANCANGAN APLIKASI

#### 3.1 Kerangka Penelitian

Pada kerangka penenelitian penulis menjabarkan perancangan aplikasi yang akan dibuat berdasarkan pada metode pengembangan aplikasi multimedia yaitu *Multimedia Development Life Cycle* (MDLC).



Gambar 3.1 Kerangka Penelitian

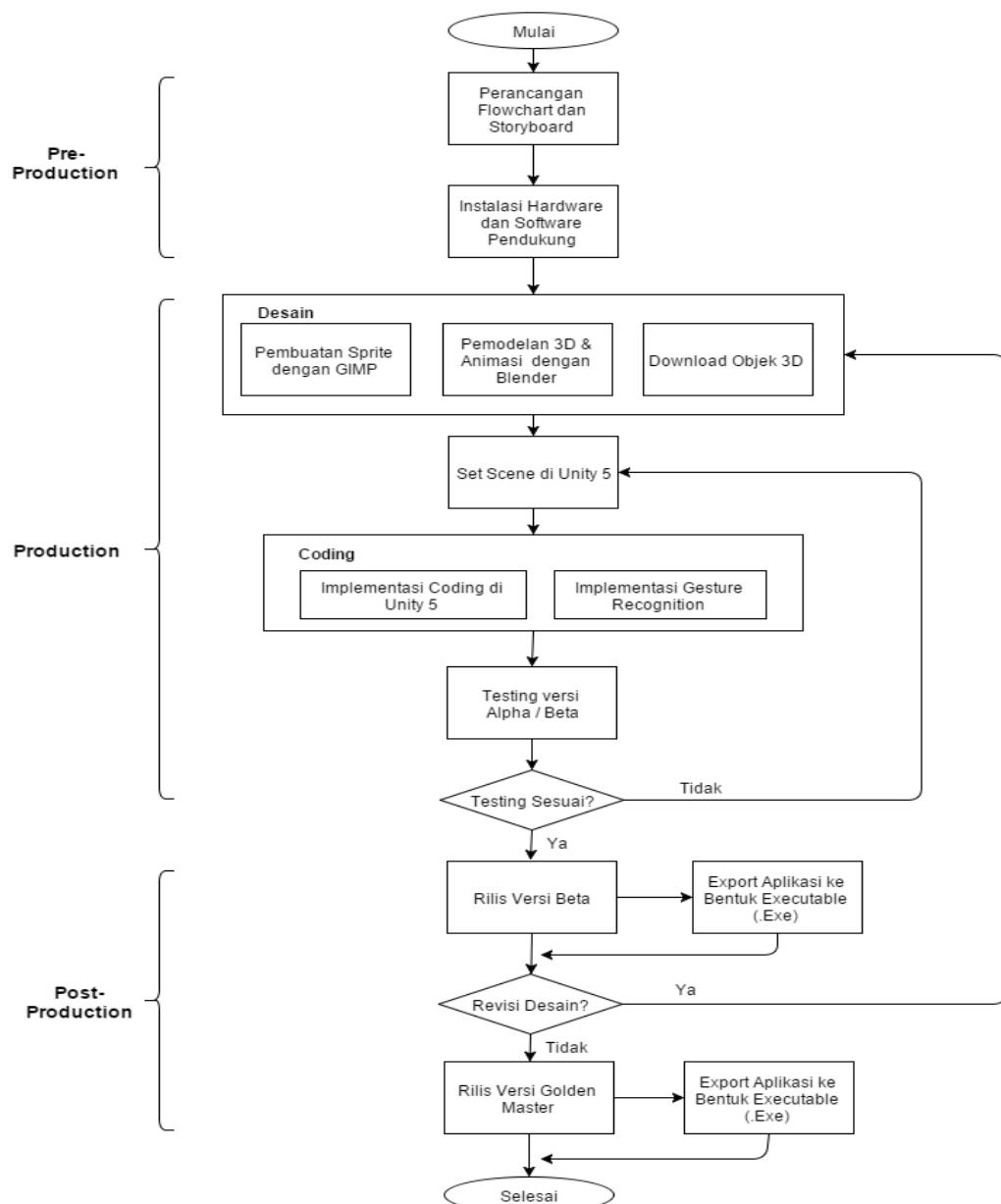
Dari flowchart pada Gambar 3.1 dapat dijelaskan tahap-tahap perancangan sebagai berikut:

1. Kerangka penelitian diawali dengan tahap pra produksi. Konsep dan ide digali dengan mengacu pada aplikasi-aplikasi pembanding yang sudah ada.
2. Perancangan flowchart dan storyboard aplikasi diperlukan untuk menyampaikan konsep aplikasi multimedia secara jelas, terstruktur dan lebih akurat.
3. Setelah perancangan storyboard telah sesuai dengan desain konsep, maka selanjutnya penulis melakukan implementasi kode, graphic design, 3D modelling dan animasi untuk mendukung aplikasi multimedia.
4. Pengujian pertama dilakukan dengan merilis atau mengkompilasi program versi alpha. Kompilasi program versi alpha lalu dilanjutkan versi beta. Pengujian aplikasi versi alpha dan beta disesuaikan dengan standar konsep yang telah ditentukan sebelumnya.
5. Setelah pengujian versi alpha dan beta sesuai, dilakukan revisi desain akhir untuk menuju versi golden master yaitu versi final yang siap untuk digunakan.

### 3.2 Tahap Perancangan

#### 3.2.1 Perancangan Aplikasi

Pada tahap ini penulis menjelaskan perancangan aplikasi yang akan dibuat juga berdasarkan metode MDLC. Tahap ini merupakan penjelasan teknis dari kerangka penelitian yang digambarkan pada flowchart berikut :



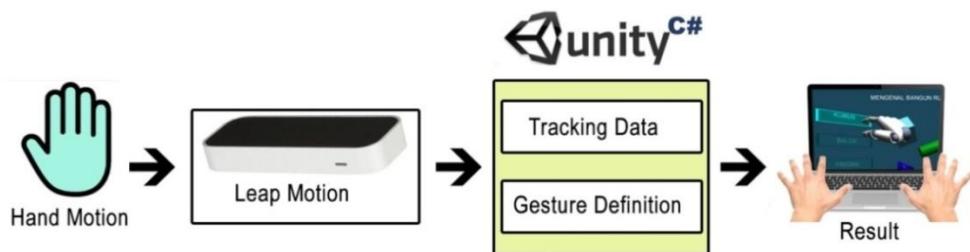
**Gambar 3.2** Flowchart Tahap Perancangan

Dari flowchart pada gambar 3.2 dapat diuraikan sebagai berikut :

1. Perancangan Storyboard adalah tahap pembuatan gambar sketsa perencanaan untuk menunjukkan secara visual bagaimana tampilan dan detail properti setiap level atau stage.
2. Instalasi Hardware dan *Software* pendukung yaitu pemasangan *Leap Motion* dan penginstalan SDK serta *software* lain pendukung pembuatan aplikasi.
3. Desain Grafis adalah proses pembuatan image 2D dengan menggunakan *software* GIMP untuk keperluan perancangan *button*, *sprite* dan *texture*.
4. Pada tahap ini dilakukan perancangan 3D model untuk bangun ruang dan objek pendukung lain serta animasi yang dibuat dengan Blender.
5. Set Scene adalah tahap perancangan yang dimulai dengan mendesain interface pada scene Unity. Interface didesain dengan menyiapkan objek-objek game yang diletakan pada folder Assets.
6. Implementasi kode adalah penerapan algoritma aplikasi kedalam bahasa pemrograman C#. Debugging & Bug Fixing pada tahap ini adalah melakukan pengecekan efisiensi kode serta perbaikan jika diperlukan.
7. *Hand Gesture Recognition* adalah metode pengenalan gestur atau gerakan tangan yang diperoleh dari data *hand motion tracking* lalu dikonversi menjadi perintah-perintah tertentu untuk menggantikan input mouse maupun keyboard pada komputer
8. Revisi Desain dilakukan jika desain belum sesuai dengan storyboard serta mengutamakan aspek interface yang user friendly.

### 3.2.2 Hand Gesture Recognition dengan *Leap Motion*

*Hand Gesture Recognition* dengan menggunakan *Leap Motion* memiliki beberapa keuntungan diantaranya adalah tidak perlu menggunakan marker . Dengan demikian data *hand motion tracking* lebih mudah diolah dan menghasilkan tingkat efisiensi dan ketepatan yang tinggi.



**Gambar 3.3** *Hand Motion Tracking* dengan *Leap Motion*

Keterangan :

1. *Leap Motion* dipasang ke PC melalui kabel USB. Jika lampu indikator berwarna hijau, artinya perangkat telah aktif sehingga *tracking data* dapat dilakukan.
2. Citra tangan dan gerakannya (*hand motion*) akan direkam oleh *Leap Motion Device* sehingga menghasilkan *tracking data* berupa posisi tangan pada sumbu acuan x,y,z. Karena *motion tracking* bersifat *realtime*, maka gerakan *Virtual Hands* yang ada pada aplikasi akan mengikuti gerakan tangan.
3. Unity dengan bahasa pemrograman C# akan menyimpan kode perintah dan logika untuk mengenal gerakan tangan (*Gesture Definition*). Kumpulan logika inilah yang dinamakan dengan metode sintaks.
4. Hasilnya, user dapat mengontrol objek, maupun menu navigasi di dalam aplikasi yang dapat digunakan untuk mempelajari bangun

ruang. Kontrol yang dihasilkan sesuai definisi gestur yang diterjemahkan.

### 3.3 Spesifikasi Hardware

Dalam pembangunan dan implementasi aplikasi diperlukan perangkat keras sebagai berikut :

#### 3.3.1 PC

Dalam pembuatan aplikasi multimedia berbasis 3D, diperlukan PC dengan spesifikasi yang mumpuni untuk menangani *motion tracking* secara *real-time* dengan performa yang baik.



**Gambar 3.4** PC yang digunakan dalam pembuatan aplikasi

**Tabel 3.1** Spesifikasi Hardware PC

<b>Notebook Series</b>	Asus A43SM-VX034D
<b>Processor</b>	Intel Core i5-2450M
<b>Graphic Card</b>	NVIDIA GeForce GT 630M, 2 GB DDR3
<b>RAM</b>	8 GB, DDR3
<b>Operating System</b>	Windows 8.1 Enterprise 64-bit
<b>Layar</b>	14" HD Color Shine (LED) , res ( 1366 x 768 )
<b>Drive Optik</b>	Super-Multi DVD
<b>Hard Disk Drive</b>	500 GB ( 5400 rpm )
<b>Audio</b>	Altec Lansing Speaker

### 3.3.2 Leap Motion Controller

#### A. Hardware



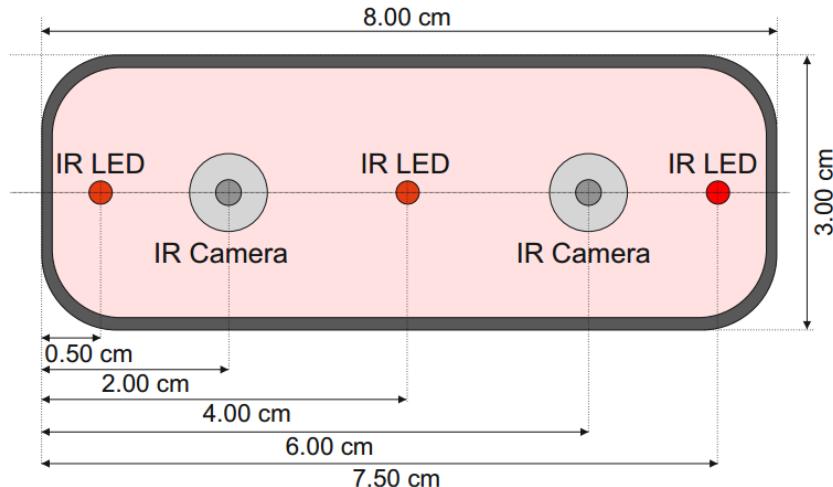
**Gambar 3.5** Perangkat *Leap Motion Controller*



**Gambar 3.6** Ilustrasi Komponen *Leap Motion Controller*

Sumber: <http://leapmotion.com>

## B. Spesifikasi



**Gambar 3.7** Sketsa struktur perangkat *Leap Motion*

Sumber: <http://leapmotion.com>

Dimensi (P x L x T) : 7,5 x 3,0 x 1,1 cm

Koneksi Peripheral : USB 3.0

Lampu LED : 3 Buah

Sensor : 2 Kamera Infra Red Monokromatik

*Leap Motion* berukuran relatif kecil, mudah untuk dibawa-bawa dan tidak memerlukan tempat yang luas. Hanya saja untuk membuat posisi tubuh nyaman dalam menggunakannya, sebaiknya perangkat diletakan di tempat yang datar dan sejajar dengan keyboard komputer. Sedangkan fitur kamera inframerah membuat *Leap Motion Controller* dapat melakukan deteksi terhadap citra tangan maupun benda-benda lain walaupun dalam keadaan tidak ada cahaya.

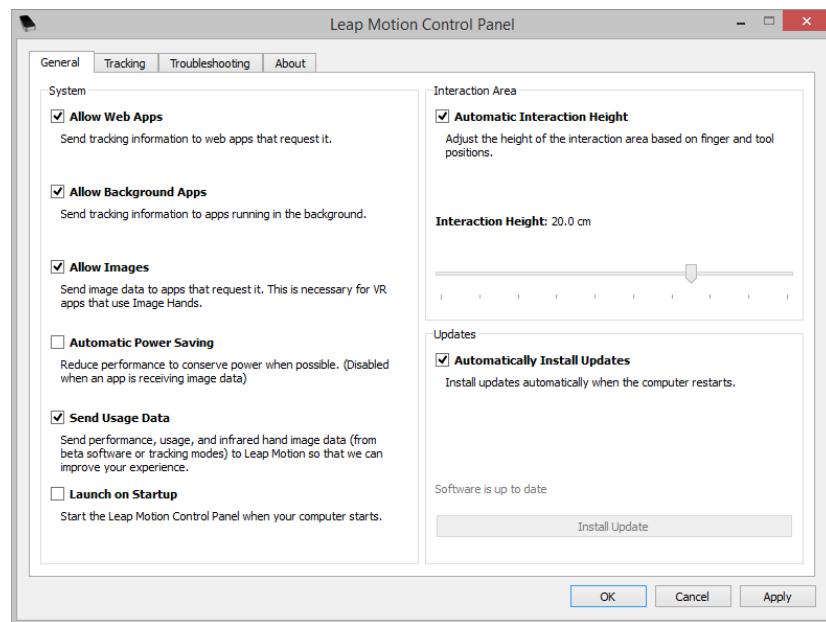
### 3.4 Spesifikasi *Software*

*Software* yang digunakan dalam pembuatan aplikasi mencakup SDK dari *Leap Motion*, Unity 5 sebagai game engine, GIMP sebagai image editor, dan Blender untuk pemodelan objek 3D dan animasi.

#### 3.4.1 Leap Motion Core Services (SDK)

*Leap Motion* Core Services adalah paket *software* yang termasuk dalam *Leap Motion* SDK meliputi :

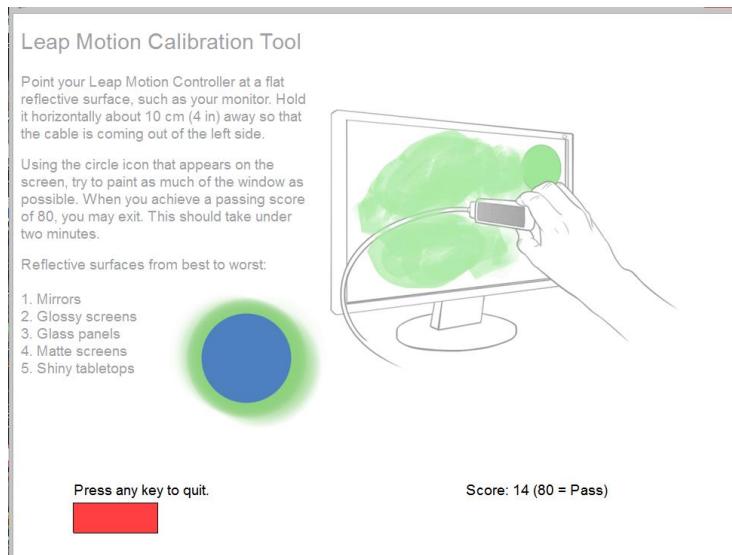
##### A. LeapControlPanel.exe



Gambar 3.8 *Leap Motion Control Panel*

*Leap Motion Control Panel* adalah modul aplikasi pengaturan sensor *Leap Motion*. User dapat melakukan pengaturan *feature permission*, area interaksi, *tracking* dan *troubleshooting*.

## B. Recalibrate.exe

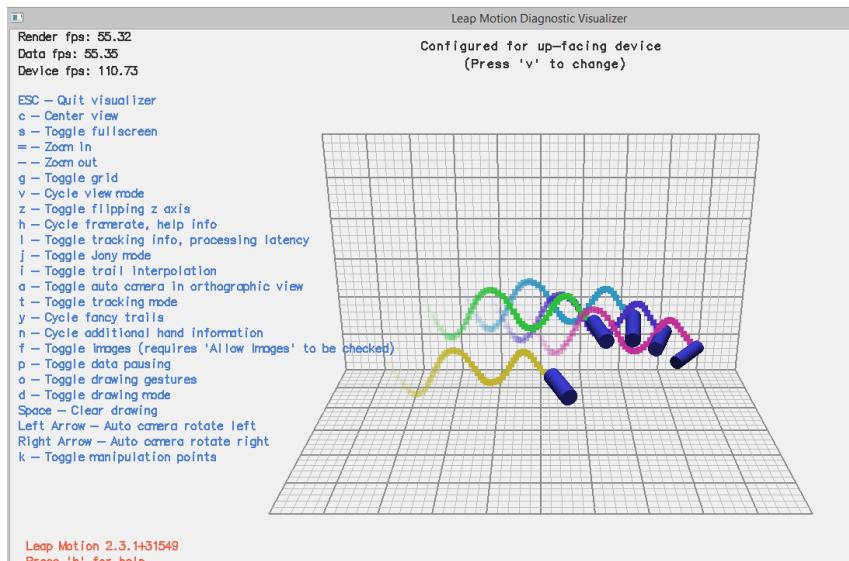


**Gambar 3.9** *Leap Motion Calibration Tool*

*Leap Motion* Calibration Tool digunakan untuk melakukan kalibrasi terhadap sensor supaya sensor meninjau ulang kepekaan dan sensitifitasnya. Tools ini berguna jika user merasa perangkat sensor tidak menginterpretasikan gerakan tangan dengan baik.

Untuk melakukan proses recalibrate , user dapat membuka lalu memposisikan *Leap Motion* sejajar menghadap layar komputer. Setelah itu, lakukan gerakan seperti “mewarnai” keseluruhan layar hingga *score* mencapai angka 80.

### C. VisualizerApp.exe



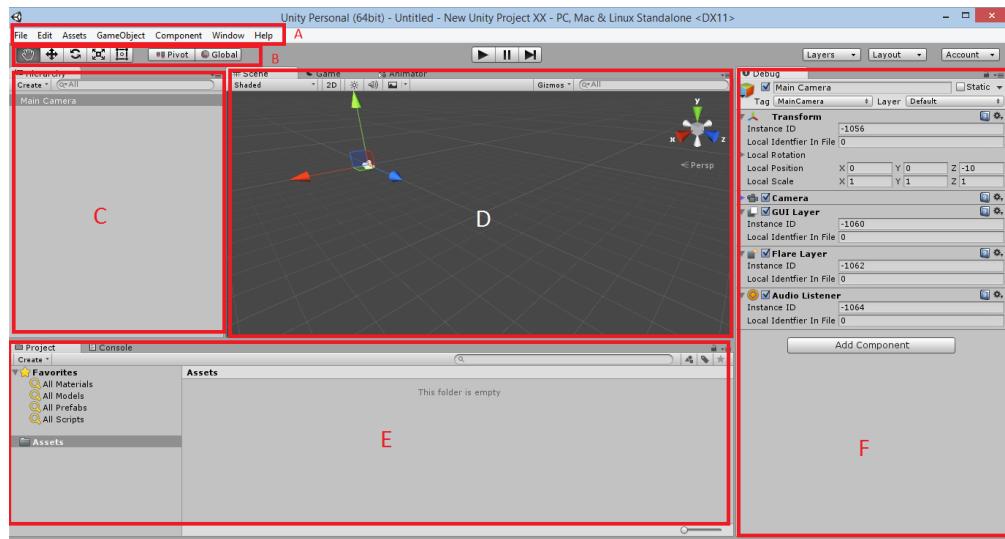
**Gambar 3.10** *Leap Motion Diagnostic Visualizer*

*Leap Motion* Diagnostic Visualizer adalah aplikasi yang disiapkan oleh vendor untuk para pengembang guna memberikan informasi tentang proses hand motion tracking yang dikerjakan oleh perangkat *Leap Motion*. Dengan aplikasi ini, pengembang dapat mengetahui data-data yang dihasilkan dari proses tracking seperti *render fps*, *data fps*, *device fps*, *skeleton*, koordinat tangan secara *real time*, output kamera infrared dan informasi lain yang sudah disajikan cukup menarik secara visual.

#### 3.4.2 Unity 3D Versi 5.1.1f1

Unity adalah game engine lintas platform yang berisi paket-paket tools yang dapat digunakan sebagai aplikasi untuk membuat game atau aplikasi multimedia lain baik dalam bentuk 2D maupun 3D. Salah satu fitur Unity untuk mempermudah programmer dalam membuat aplikasi / game adalah Component. Unity telah menyediakan paket-paket umum non-script maupun script untuk mengembangkan game 2D maupun 3D.

## A. Workspace Unity 5



**Gambar 3.11** Workspace Unity 5

### A. Menu Bar :

Berisi menu utama pada Unity. Terdiri dari File, Edit, Assets, GameObject, Component, Window dan Help.

### B. Tool Bar

1. Zoom Tool = Memperdekat/jauh penampakan
2. Move Tool = Memindahkan/menggeser objek 3D
3. Rotate Tool = Memutar objek 3D
4. Scale Tool = Memperbesar/memperkecil ukuran objek 3D

### C. Hierarchy View:

Susunan objek-objek pada game yaitu objek parent dan child

### D. Game Window

Set *world game* atau tampilan game.

### E. Project View

Kumpulan objek-objek yang dapat dipakai dalam game. Mulai dari script, model 3D, sprite, texture, prefab dan sebagainya. Umumnya secara default ada pada folder Assets.

### F. Debug / Inspector

Pemperlihatkan proses debugging game secara dinamis dan nilai (value) variabel yang ada pada game.

## B. Kelebihan dan Kekurangan Unity 5

Unity 5 memiliki beberapa kelebihan dibandingkan game engine lainnya diantaranya adalah ;

- Unity 3D dapat digunakan untuk melakukan development terhadap aplikasi sederhana hingga yang kompleksitasnya tinggi.
- Unity 3D mendukung bahasa pemrograman C# yang sudah mengacu pada pola pemrograman berorientasi objek.
- Manajemen file yang baik terlihat pada sistem file pada unity yang dapat dikelompokan kedalam folder dan sub folder.
- Memiliki asset store dengan developer yang cukup banyak aktif melalui forum online.
- Fitur ekspor ke banyak platform OS seperti Windows, android, linux dan dapat pula dimainkan lewat web browser dengan tambahan plugin unity web player.
- Konversi kode menjadi GUI (Graphic User Interface)

Namun ada beberapa kekurangan yaitu:

- Beberapa assets seperti Shader, texture, model hanya tersedia jika membayar lisensi Unity Pro.
- Unity 3D tidak dapat melakukan modelling atau desain objek. Proses modelling atau desain dilakukan di *software* lain seperti blender atau 3DS MAX.

### 3.4.3 Blender 2.68a

Walaupun blender bersifat open source, namun *software* ini memiliki beberapa keunggulan yang tidak kalah dari *software* berbayar lainnya. *Software* ini dapat diunduh dan digunakan secara gratis pada website resminya di [www.blender.org](http://www.blender.org). Blender selalu mengeluarkan versi terbarunya dalam jangka waktu yang berdekatan. Setiap versi baru yang dikeluarkan, selalu ada tools baru yang ditambahkan beserta bug-bug yang sudah diperbaiki. Perkembangan Blender sangat pesat karena banyak developer yang terlibat dalam pengembangan Blender. Para developer ini tergabung dalam suatu komunitas terbuka yang anggotanya bebas dan tersebar di seluruh dunia.

Keunggulan lain dari Blender adalah kesederhanaannya. Kesederhanaan *software* ini membuat Blender tidak memerlukan spesifikasi komputer yang tinggi untuk sekedar menjalankannya. File instalasinya yang kecil membuat blender juga dapat diinstall dan dijalankan secara portable. Sekitar 40 - 60 Mb sebelum diinstall dan 150 Mb setelah di install. Ukuran file yang kecil ini jauh melampaui *software* sejenis yang file instalasinya membutuhkan ruang hardisk sekitar 2 hingga 3 Gb.

### 3.4.4 GIMP 2.8

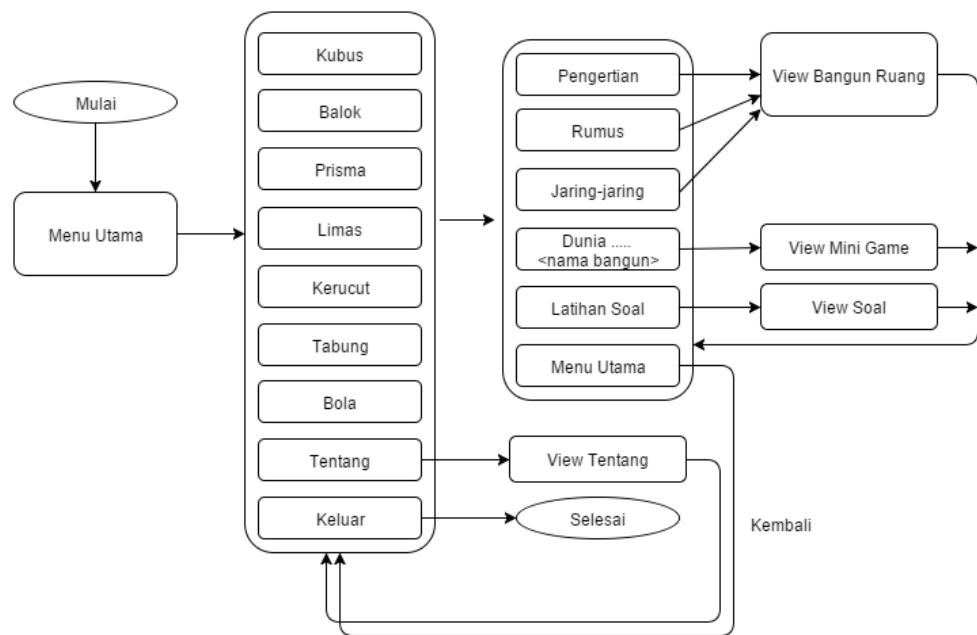
Gimp adalah sebuah aplikasi pengolah gambar raster atau bitmap dan program manipulasi gambar yang bisa berjalan di semua platform sistem operasi. Gimp merupakan singkatan dari GNU Manipulation Program, aplikasi ini bersifat Free Open Source *Software*, sehingga kita dapat mendapatkan aplikasi ini secara gratis tanpa harus membayar lisensi dari si pembuat program. Gimp sangat cocok digunakan untuk *image manipulation tasks*, termasuk *photo retouching*, *image composition*, dan *image construction*<sup>[23]</sup>.

### 3.5 Tahap Pre-Production

Tahap Pre-Production adalah tahap awal pada siklus pengembangan aplikasi multimedia berbasis metode MDLC. Pada tahap ini, dilakukan penggambaran konsep aplikasi yang telah digali sebelumnya melalui flowchart dan storyboard.

#### 3.5.1 Perancangan Flowchart Aplikasi

Flowchart aplikasi menggambarkan alur program Pengenalan Bangun Ruang, berupa menu-menu dan *view (scene)*.

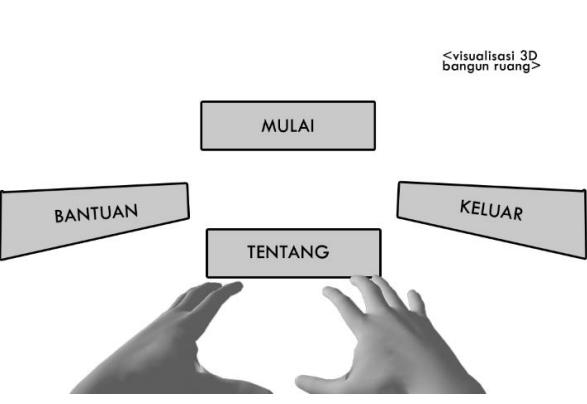
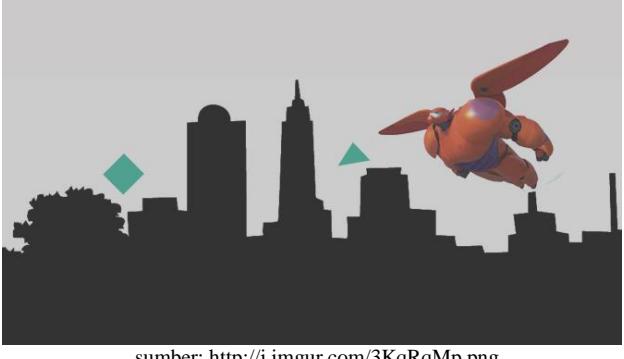


**Gambar 3.12** Flowchart Aplikasi

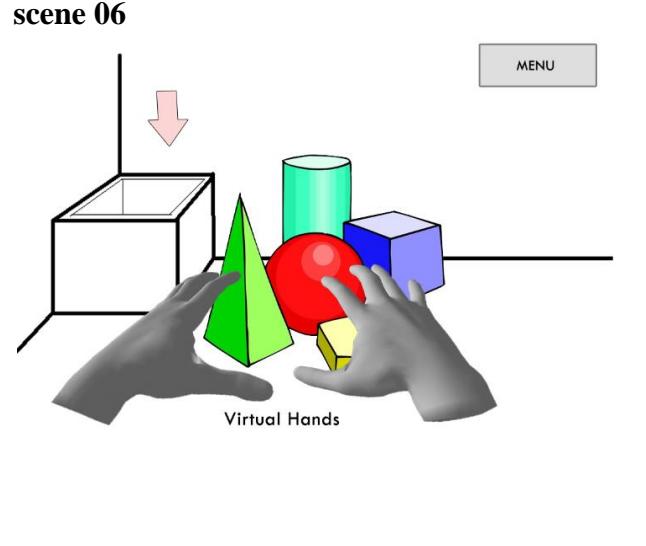
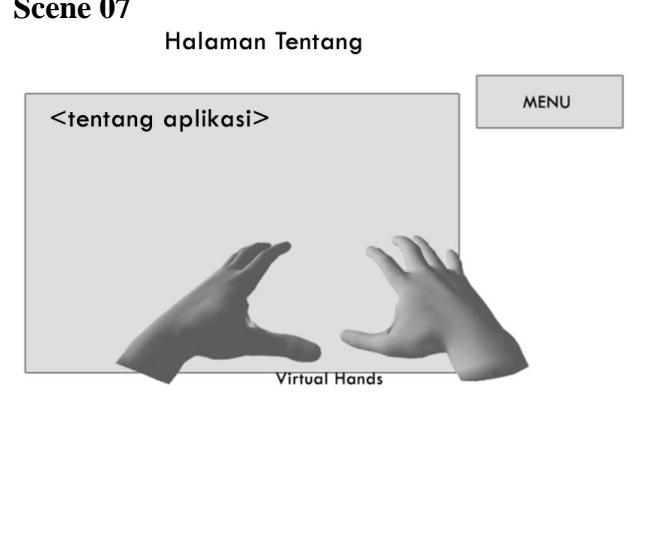
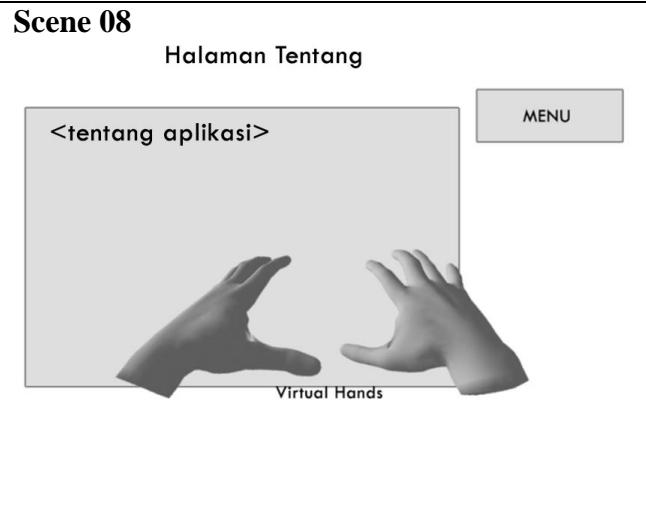
Pada scene menu utama terdapat beberapa menu pilihan tipe bangun ruang yang disajikan meliputi kubus, balok, prisma, limas, kerucut, tabung dan bola. Sedangkan pada menu bangun ruang, user dapat memilih sajian materi berupa pengertian, rumus, jaring-jaring dan latihan soal.

### 3.5.2 Perancangan Storyboard Aplikasi

*Storyboard* aplikasi pengenalan bangun ruang terdiri dari 6 *scene* yang menjadi antarmuka utama. Setiap scene mengandung beberapa objek yang mendukung navigasi seperti *virtual button* dan juga *virtual hands*.

Scene	Keterangan
<b>scene 01</b>  <p style="text-align: center;">&lt;visualisasi 3D bangun ruang&gt;</p> <p style="text-align: center;">MULAI</p> <p style="text-align: left;">BANTUAN</p> <p style="text-align: right;">TENTANG</p> <p style="text-align: right;">KELUAR</p>	<b>MENU UTAMA</b> Animasi : Bangun ruang berputar Gestur : Key Tap Suara : audiobg.ogg User dapat memilih menu untuk memilih menu dengan menyentuh button melalui <i>virtual hand</i> .
<b>scene 02</b>  <p>sumber: <a href="http://i.imgur.com/3KqRqMp.png">http://i.imgur.com/3KqRqMp.png</a></p>	<b>MINI GAME 1</b> Animasi : Robot Terbang Gesture : Fly Control Suara : JetSound.mp3 User ditugaskan untuk melakukan eksplorasi di dalam sebuah kota, dimana user harus mencari stasiun pengenalan bangun ruang. Setiap stasiun, memiliki sebuah komputer canggih yang menjelaskan tentang sebuah bangun ruang.

<p><b>scene 03</b></p> <p>&lt;NAMA BANGUN RUANG&gt;</p>	<p><b>STASIUN BANGUN RUANG</b></p> <p>Animasi : Bangun ruang berputar</p> <p>6 buah 3D button</p> <p>Gestur : <i>Key Tap, Circle</i></p> <p>User dapat memilih menu dengan menyentuh button melalui virtual hand.</p>
<p><b>Scene 04</b></p> <p>&lt;NAMA BANGUN RUANG&gt;</p>	<p><b>PENGENALAN BANGUN RUANG</b></p> <p>Animasi : Bangun ruang terbuka, Bangun ruang tertutup, Bangun ruang berputar</p> <p>Gestur : <i>Key Tap, Swipe (left/right), Push, Pull, Close Fist</i></p> <p>Fitur : <i>Zoom out, Zoom in, Rotate, Translate</i></p> <p>User dapat melakukan kontrol terhadap bagun ruang dengan menggunakan gestur</p>
<p><b>scene 05</b></p> <p>LATIHAN SOAL</p>	<p><b>LATIHAN SOAL</b></p> <p>3 buah Sprite Soal 1 buah 3D Button</p> <p>Gestur : <i>Key Tap, Swipe</i></p> <p>Menunjukkan soal-soal yang berhubungan dengan bangun ruang terkait.</p>

<p><b>scene 06</b></p> 	<p><b>MINI GAME 2</b></p> <p>1 Buah 3D button</p> <p>Gesture : Pinch, Release, Throw</p> <p>Fitur : Physics</p> <p>Pemain memasukan bangun ruang yang benar ke dalam kotak. Jika benar maka akan mendapat score.</p>
<p><b>Scene 07</b></p> <p>Halaman Tentang</p> 	<p><b>TENTANG (CREDIT)</b></p> <p>Menampilkan Informasi tentang aplikasi.</p> <p>1 buah 3D Button 1 buah sprite Tentang</p> <p>Gestur : <i>Key Tap</i></p> <p>User dapat melihat informasi aplikasi dan dapat kembali ke menu utama.</p>
<p><b>Scene 08</b></p> <p>Halaman Tentang</p> 	<p><b>BANTUAN</b></p> <p>Menampilkan bantuan penggunaan aplikasi.</p> <p>1 buah 3D Button 1 buah sprite Tentang</p> <p>Gestur : <i>Key Tap</i></p> <p>User dapat melihat info bantuan dan dapat kembali ke menu utama.</p>

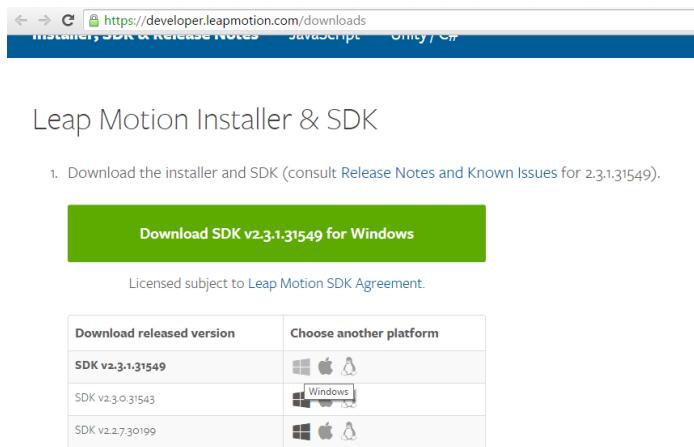
### 3.6 Tahap Produksi

#### 3.6.1 Instalasi *Leap Motion Software Development Kit (SDK)*

*Leap Motion Software Development Kit (SDK)* diperlukan bagi pengembang aplikasi *Leap Motion* untuk mengetahui kinerja *Leap Motion* serta pustaka-pustaka (*libraries*) kode yang diperlukan. Pada penulisan ini penulis menggunakan SDK Versi 2.3.1+31549. SDK juga dapat digunakan oleh pengembang untuk menganalisa sensor *Leap Motion* dengan berbagai tools dan viewer yang ada di dalamnya.

Untuk mendownload SDK terbaru, telebih dahulu login pada situs resmi *Leap Motion* untuk mendownload SDK. Jika belum memiliki ID, maka diperlukan registrasi yang dapat dilakukan secara gratis.

Jika sudah berhasil login dengan akun yang terdaftar sebelumnya, maka SDK dapat didownload di halaman <https://developer.leapmotion.com/downloads>



2. Go to the [getting started](#) page for next steps.

**Gambar 3.13** Halaman Download *Leap Motion* SDK

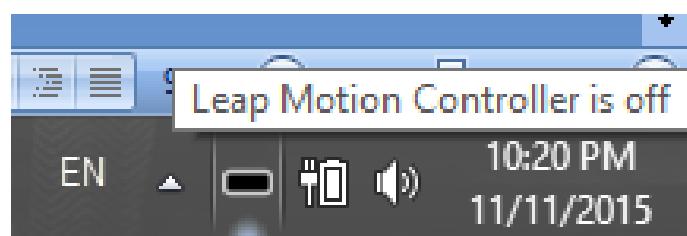
Terdapat 3 versi platform OS yang tersedia yaitu Windows, Mac, dan Linux. Pada penelitian ini penulis menggunakan versi windows.

Setelah diinstal, akan ada dua buah shortcut program pada desktop yaitu *Leap Motion App Home* dan *Leap Motion Control Panel*.



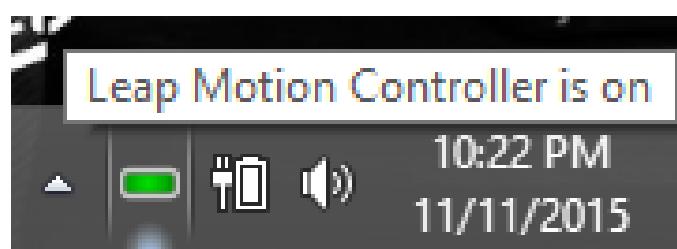
**Gambar 3.14** Program SDK yang telah diinstall

Untuk memulai perancangan aplikasi, perlu diketahui terlebih dahulu apakah *controller device* berjalan dengan baik atau tidak. Jalankan *Leap Motion Control Panel* maka ikon *Leap Motion* akan muncul di pojok kanan taskbar. Jika status pada ikon belum menyala hijau artinya fungsi tracking pada *Leap Motion* belum berjalan.



**Gambar 3.15** Status *tracking* pada posisi *off*

Untuk menjalankan fungsi tracking maka klik kanan pada icon, lalu pilih *resume tracking*.

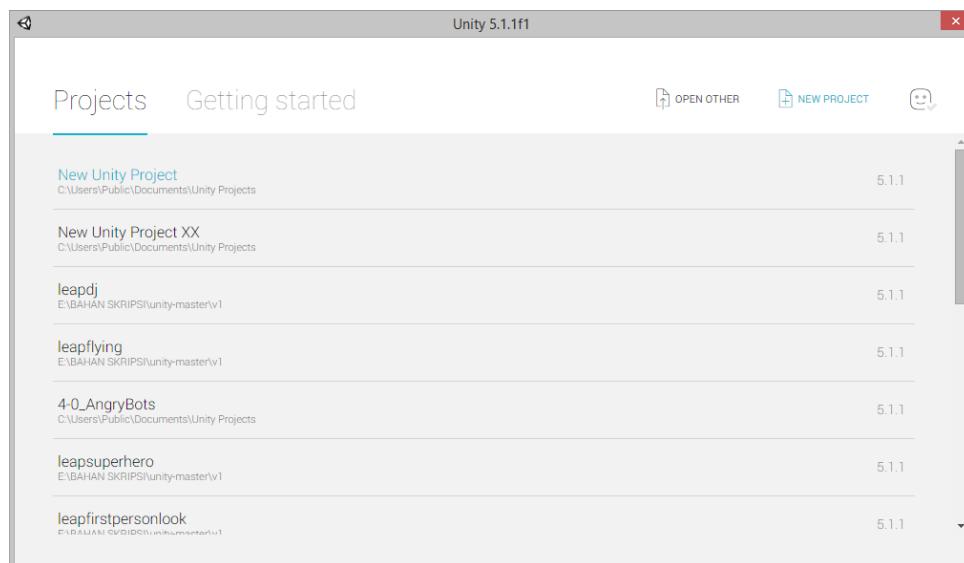


**Gambar 3.16** Status *tracking* pada posisi *on*

Buka Visualizer (VisualizerApp.exe) dengan klik kanan pada icon, lalu pilih Visualizer. Maka, jika *Leap Motion* berhasil terkoneksi dengan baik, tracking gerakan tangan secara real-time akan terlihat pada layar.

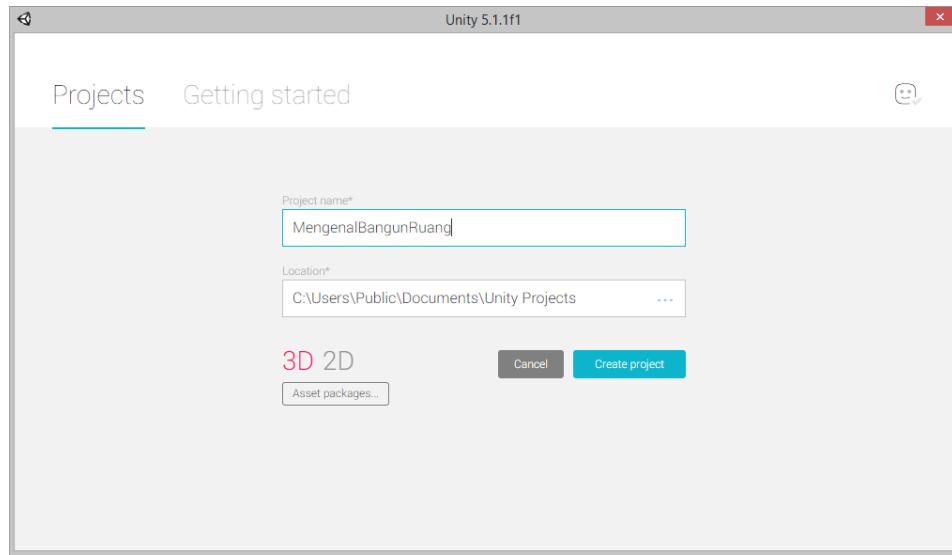
### 3.6.2 Membuat Project Baru

Setelah aplikasi game engine Unity dibuka, maka akan muncul jendela berisi project-project yang terakhir dibuka. Selain itu terdapat menu Open Other untuk membuka project lainnya dan menu New Project untuk membuat project baru. Pilih menu New Project.



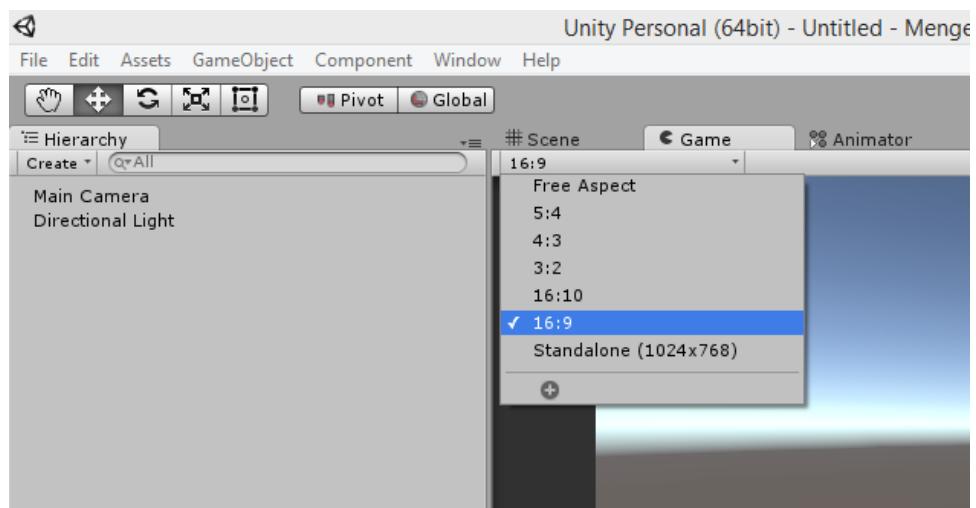
**Gambar 3.17** Jendela Awal New Project

Setelah itu, project diberi nama, dan dipastikan bahwa asset package bertipe 3D. Lokasi penyimpanan project juga ditentukan. Dalam hal ini penulis menyimpan project Unity pada folder path C:\Users\Public\Documents\Unity Projects .



**Gambar 3.18** Penentuan Lokasi Penyimpanan Project

Maka secara default, Unity akan membuatkan sebuah scene baru yang dapat langsung dipergunakan untuk merancang aplikasi. Scene berarti satu tampilan layar yang berisi sekumpulan objek game yang aktif. Scene dapat digunakan untuk membuat menu, level, dan lain-lain. Untuk mendapatkan resolusi yang baik pada layar bertipe widescreen, tentukan rasio layar pada game dengan pengaturan 16:9.

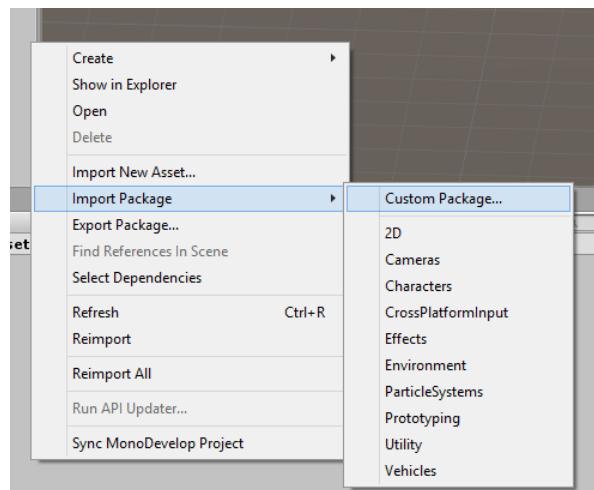


**Gambar 3.19** Pengaturan Rasio Layar Aplikasi

### 3.6.3 Import *Leap Motion Package*

Paket Unity yang diperlukan dalam perancangan aplikasi Pengenalan Bangun Ruang dengan *Leap Motion Controller* adalah paket resmi dari *Leap Motion* yaitu LeapMotionCoreAsset versi 2.312.

Untuk menerapkan Custom Package, klik kanan pada project view lalu Import Package → Custom Package seperti yang tertera pada gambar berikut ;



Gambar 3.20 Proses Import Custom Package

Pada pilihan pada saat import package, paket OVR tidak perlu dicentang karena OVR adalah plugin untuk perangkat Oculus Virtual Reality.

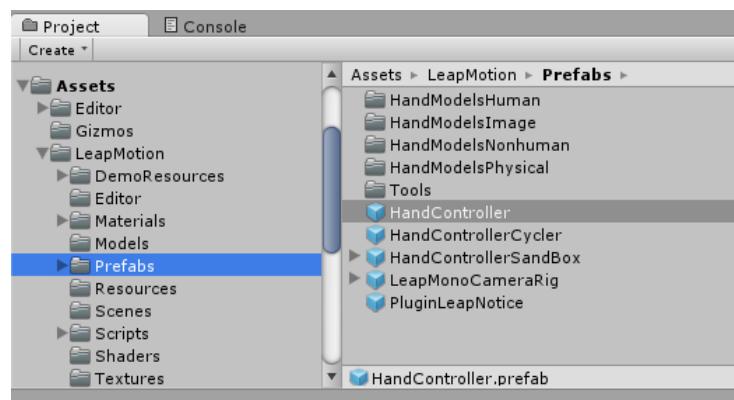


**Gambar 3.21** Import *Leap Motion* Asset Package

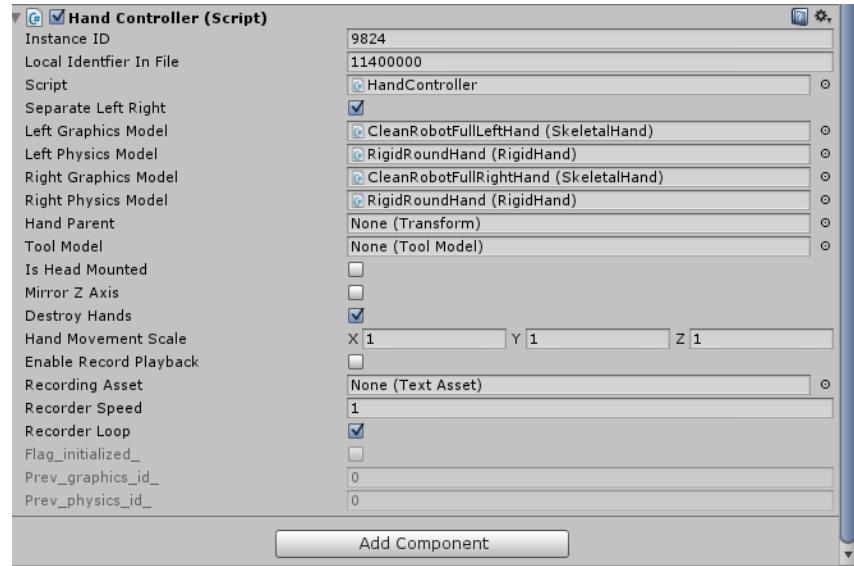
### 3.6.4 Set Scene pada Unity 5

Setelah package dari *Leap Motion* diimport, objek-objek pada scene disusun berdasarkan spesifikasi desain yang telah dirancang pada storyboard.

Komponen utama adalah *virtual hands* yang merupakan UI controller berbentuk tangan manusia. Untuk mengimplementasikan virtual hands, paket *Leap Motion* telah menyediakan prefab *HandController*.

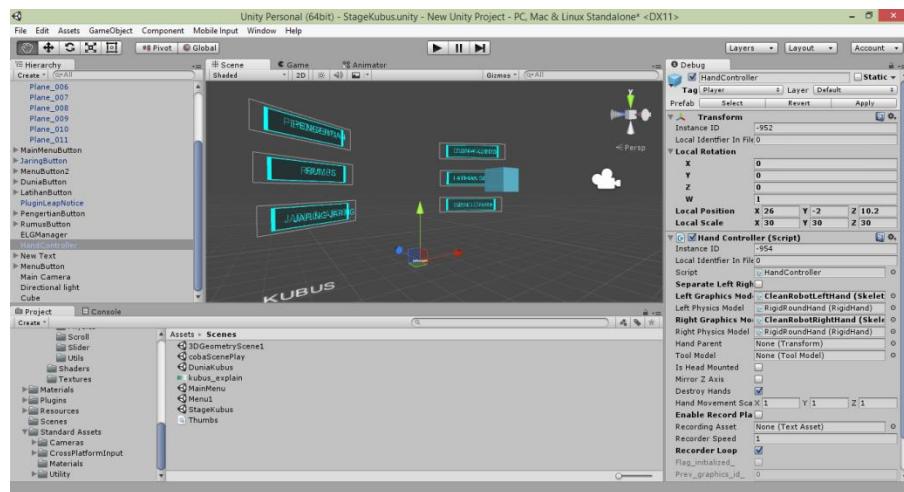


**Gambar 3.22** Prefab Hand Controller



**Gambar 3.23 Variabel-varibel Hand Controller**

Untuk menambahkan virtual button, drag kedalam workspace prefab virtual button yang terletak pada folder path Assets\LeapMotion\Widgets\Prefab dengan nama prefab DemoToggleButton. Setelah itu, button diatur pada posisi di atas handController dan disusun berdasarkan rancangan.



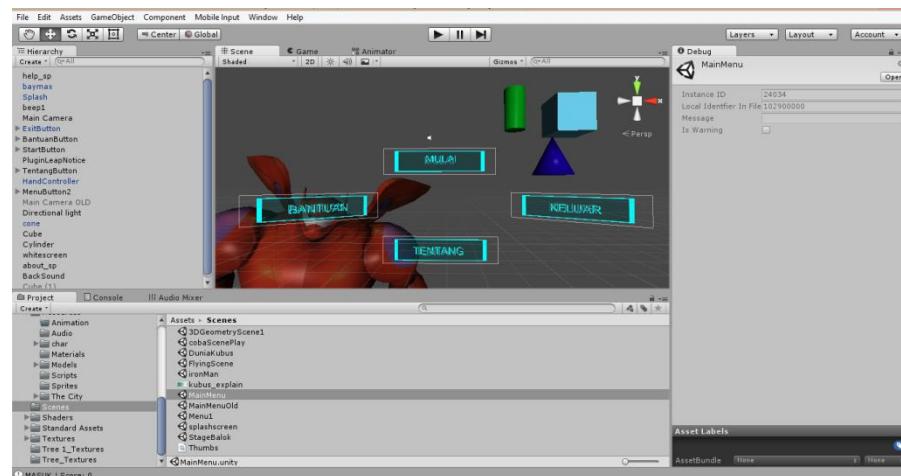
**Gambar 3.24 Objek disusun sesuai Storyboard**

Struktur objek pada sebuah virtual button.

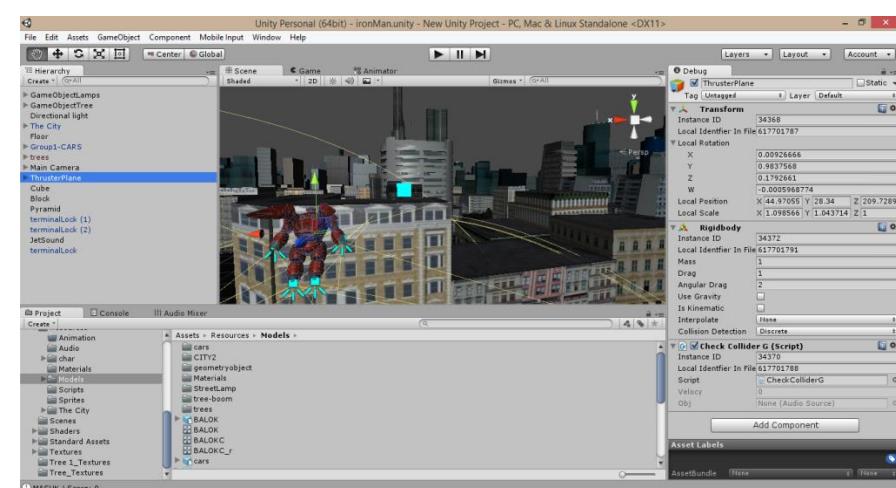
```

<Objek>Button
-- Button
---- Canvas
----- OnGraphics
----- Button-Square-Text-ON
----- OffGraphics
----- Button-Square-Text-OFF
---- BotGraphics
----- Button_Panel_Bottom
----- Button_Plane_Bottom

```



**Gambar 3.25** Susunan Button pada Main Menu

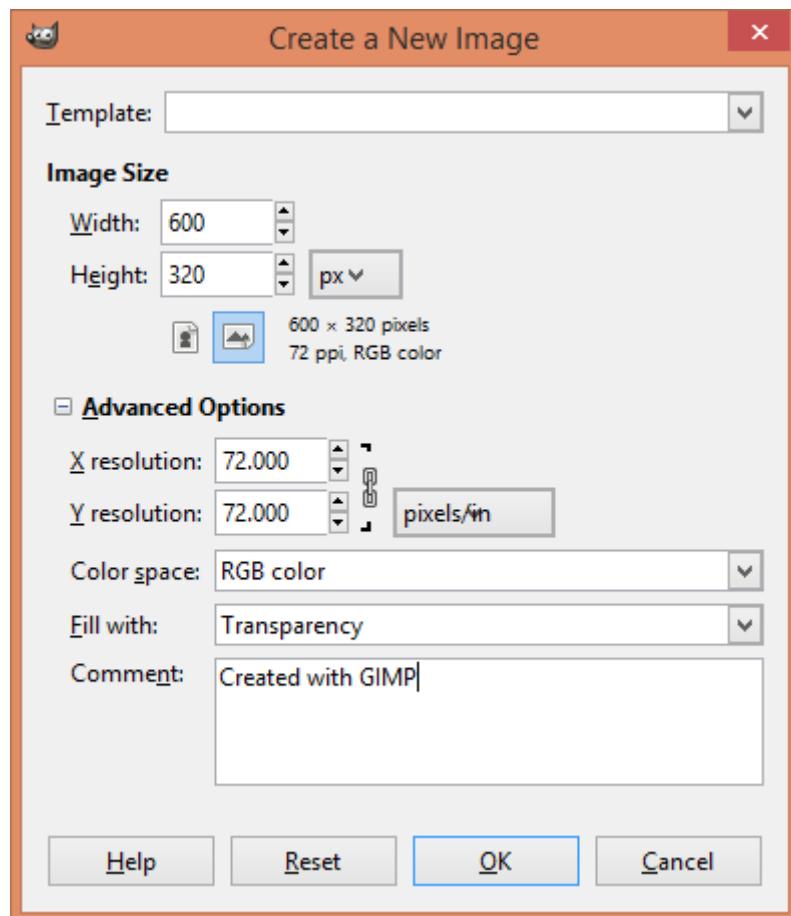


**Gambar 3.26** Susunan Desain Robot dan Kota

### 3.5.5 Perancangan Sprite

Sprite adalah objek 2D atau sebuah image yang dapat dimanipulasi kedalam ruang 3D. Pada tugas akhir ini, penulis merancang sprite dengan menggunakan GIMP versi 2.8. Sprite yang dirancang adalah image untuk kebutuhan penyajian materi bangun ruang.

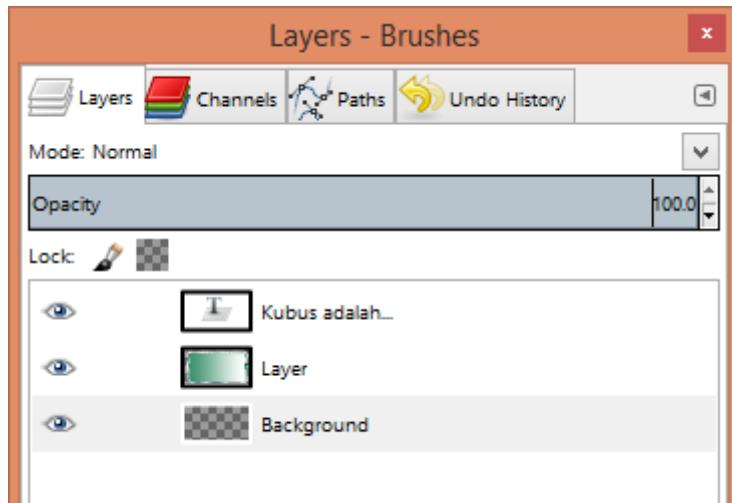
Setelah GIMP dibuka, klik menu File → New. Pengaturan awal project dibuat seperti yang ada pada gambar berikut ;



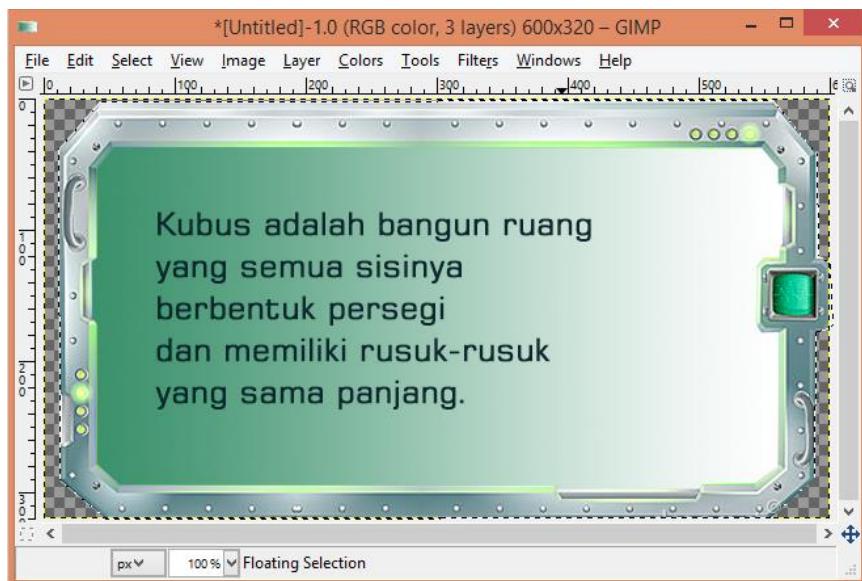
**Gambar 3.27** Pengaturan Image pada GIMP

Layer 2 ditambahkan dengan cara menekan tombol Ctrl+Shift+n. Pada Layer 2 inilah background disisipkan. Sedangkan untuk

menambahkan text, pilih toolbox Text lalu ketikan materi yang akan disajikan. Maka akan dihasilkan susunan layer seperti berikut ;



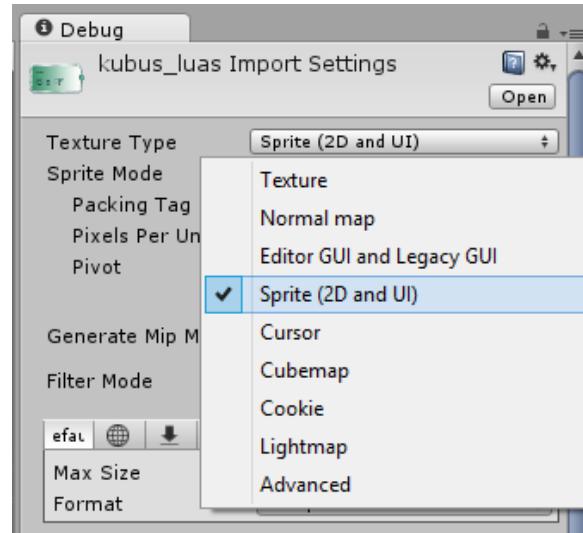
**Gambar 3.28** Susunan Layer pada GIMP



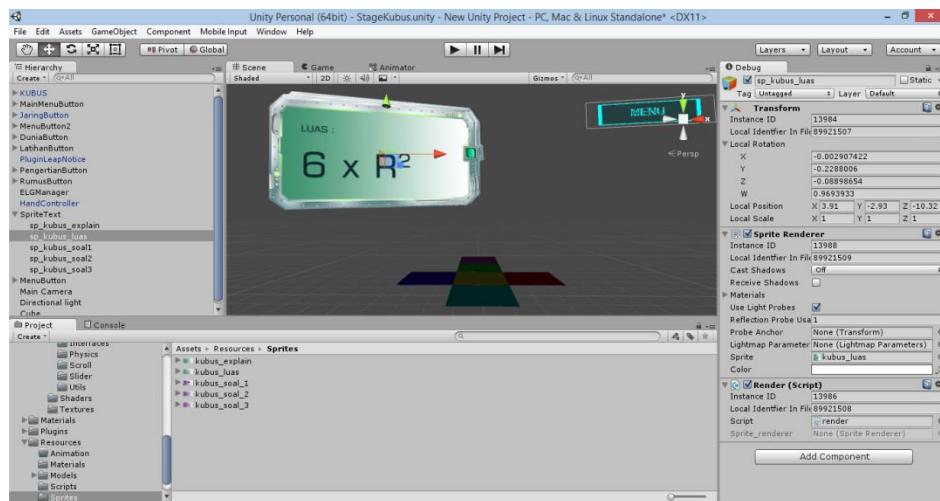
**Gambar 3.29** Perancangan Sprite pada GIMP

Setelah selesai, hasil disimpan dengan format .png di dalam folder Asset/Resource(Sprite).

Jika sebuah image ingin dijadikan sprite, maka Unity perlu mengetahui tipe sebuah image. Untuk mengubah tipe image menjadi sprite maka pada panel debug, Texture Type diubah menjadi Sprite (2D and UI)



**Gambar 3.30** Mengubah Tipe Image

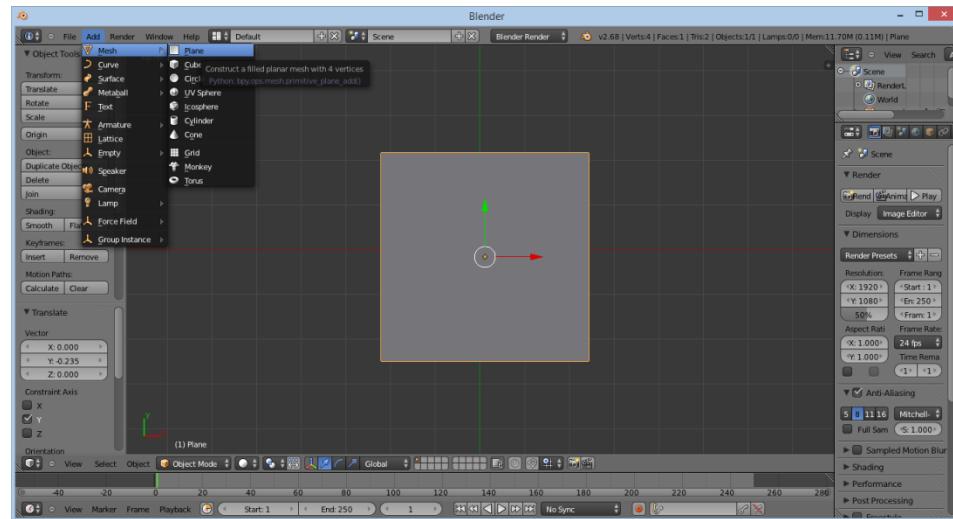


**Gambar 3.31** Penyusunan Sprite pada Scene

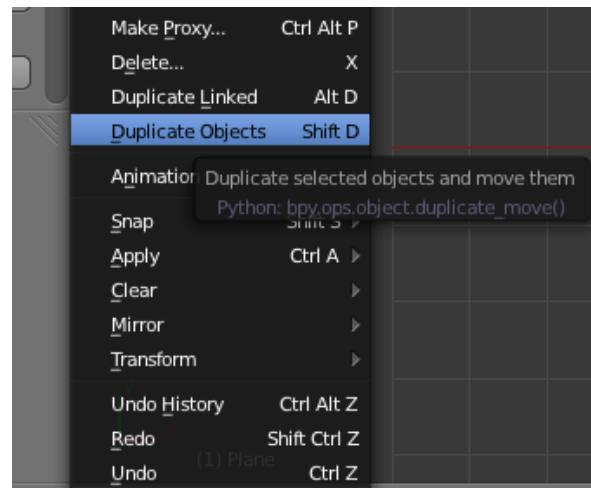
### 3.5.6 Pemodelan Objek 3D dan Animasi

Pemodelan dan animasi objek bangun ruang dibuat dengan menggunakan blender 2.68a. Kelebihan Blender adalah dapat langsung diintegrasikan dengan Unity 5 tanpa harus melakukan ekspor ke format 3D tertentu. Dengan kata lain, Unity 5 dapat melakukan import format default blender yaitu format *.blend*.

Buka blender. Untuk menambahkan bangun datar, pilih menu add → Mesh→ Plane.

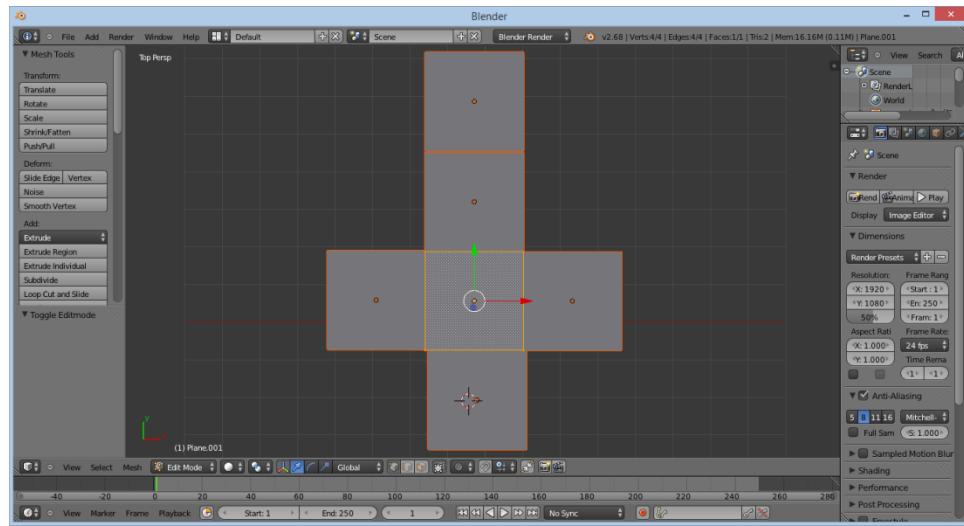


Gambar 3.32 Menambahkan objek Plane



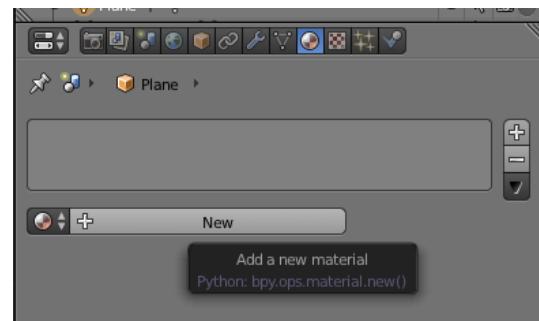
Gambar 3.33 Tahap Duplikasi objek

Objek persegi (plane) diduplikat dan dibentuk hingga membentuk jaring-jaring bangun tertentu. Dalam hal ini jaring-jaring kubus.



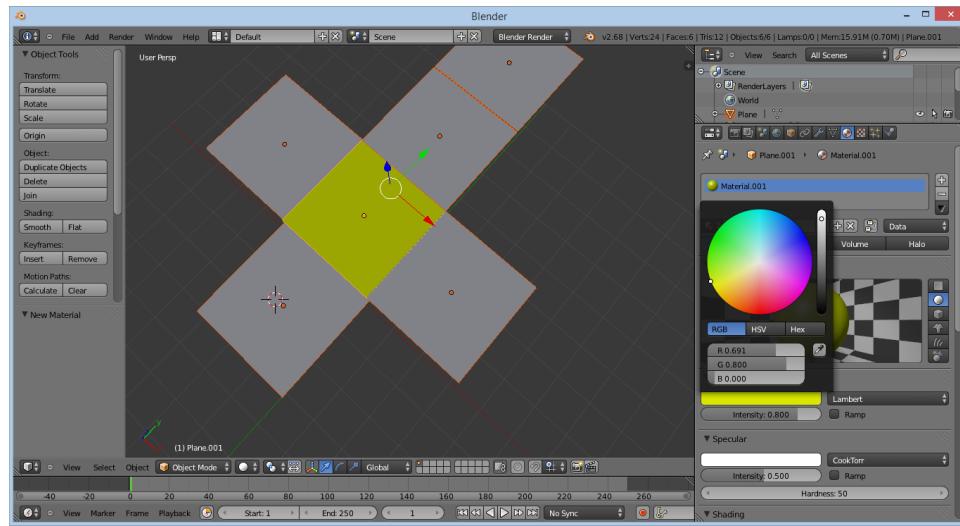
**Gambar 3.34** Desain Jaring-jaring Kubus

Setelah jaring-jaring bangun ruang terbentuk, maka dilakukan penambahan material untuk memberikan warna pada setiap bidang mesh. Pada Material, pilih tombol New.



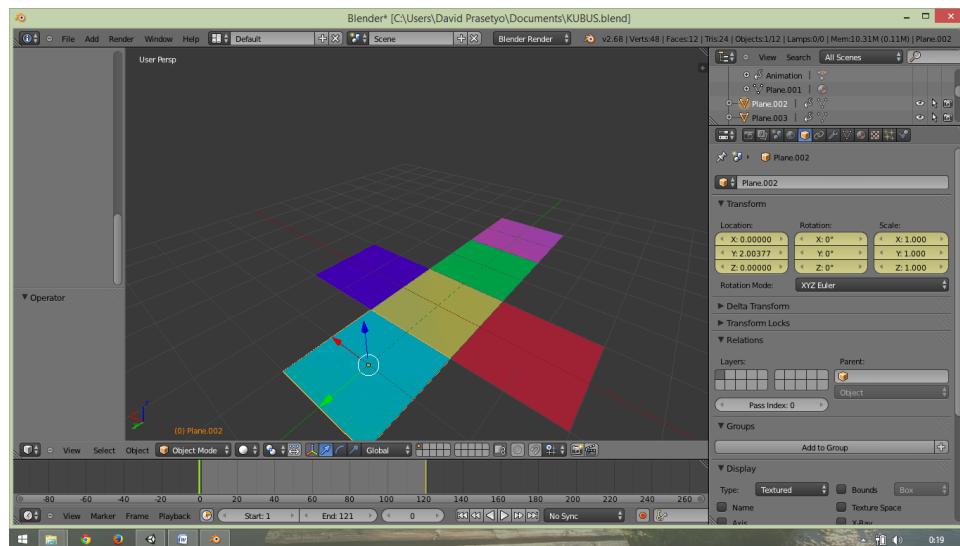
**Gambar 3.35** Penambahan Texture

Pada bagian ini , dapat dientukan warna material yang diinginkan melalui color picker. Setiap surface dapat diberikan warna atau material texture yang berbeda-beda.



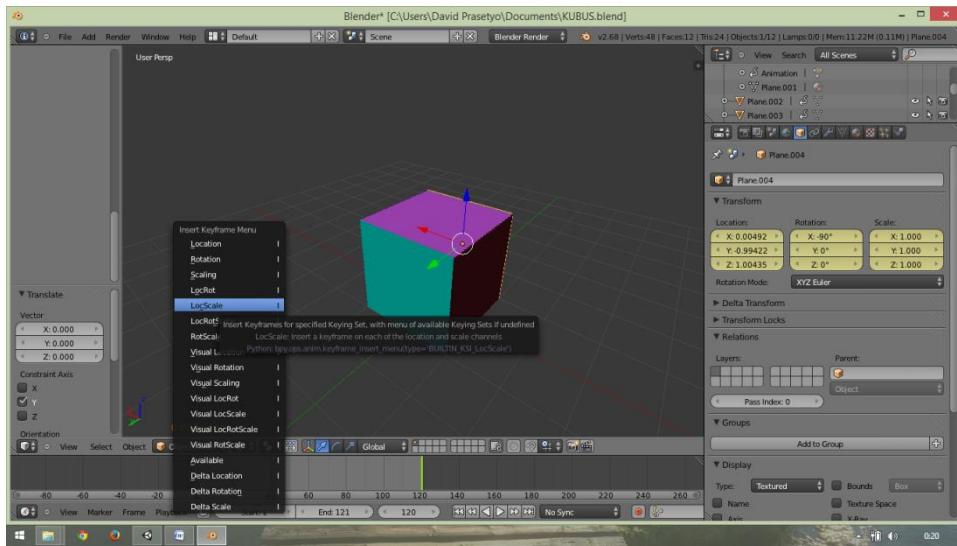
**Gambar 3.36** Proses Pewarnaan Texture

Setelah semua bidang diwarnai maka hasilnya seperti gambar berikut



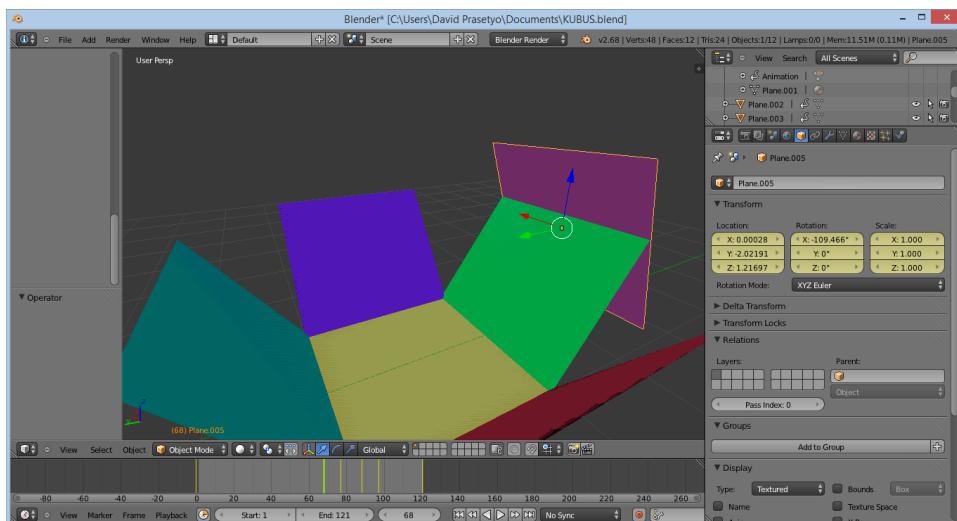
**Gambar 3.37** Objek bangun Ruang Bertekstur

Pada proses penganimasian objek, penulis melakukan penentuan titik-titik frame untuk meletakan keyframe. Pada kasus berikut ditentukan banyak frame dalam satu durasi animasi adalah 120 frame.



Gambar 3.38 Proses penambahan keyframe

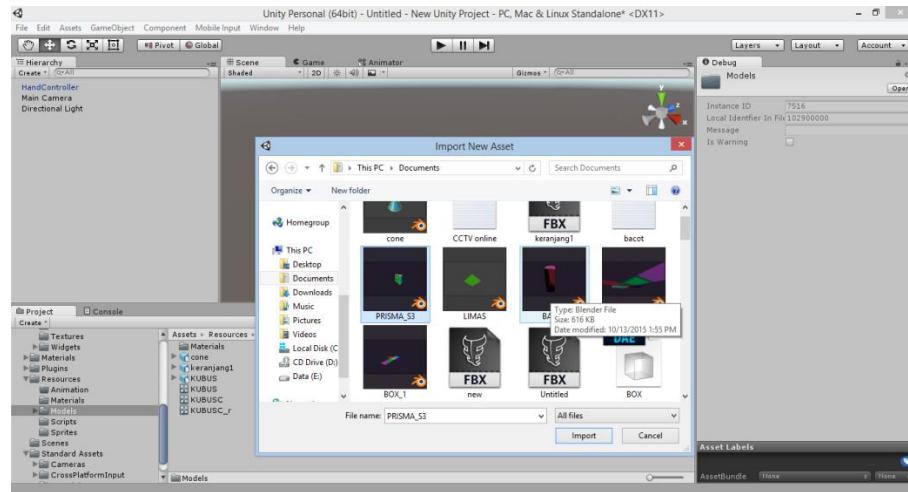
Pada frame ke-x posisi direkam dalam insert keyframe. Mode keyframe adalah locrotscale (location, rotation, scale).



Gambar 3.39 Animasi Bangun Ruang

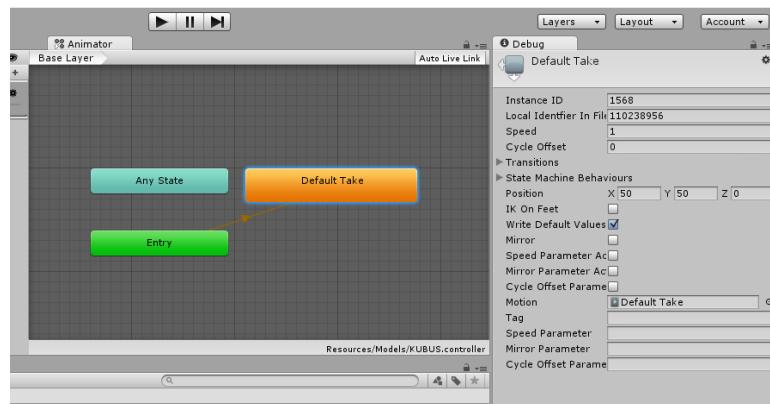
### 3.5.7 Proses Import Model ke Unity

Pada folder path ..../Asset/Resource/Models import, klik kanan lalu Import New Asset. Pilih model yang telah dibuat pada blender dengan format *.blend* untuk diimport kedalam workspace Unity.



**Gambar 3.40** Proses Import Blender Model ke Unity

Setelah animasi diekstrak, pada tab animator State Default Take nilai speed diisi 1 untuk menghasilkan gerakan animasi aslinya yaitu gerak menutup dari jaring-jaring menjadi bangun ruang. Setelah itu animasi diduplikat dan diberi nilai speed -1 untuk menghasilkan gerakan terbalik yaitu bangun ruang menjadi jaring-jaring.

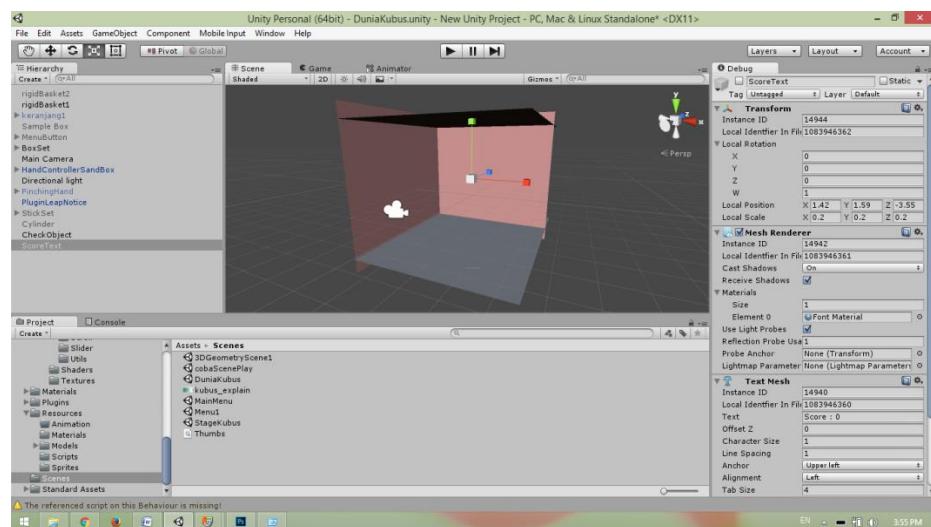


**Gambar 3.41** Animator pada Unity

### 3.5.8 Perancangan Scene Mini Game

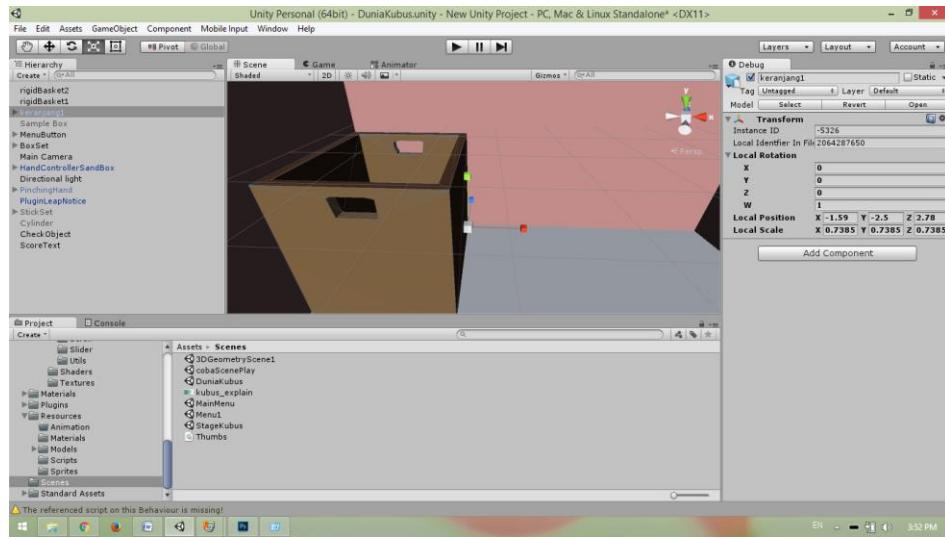
Scene baru dibuat dengan cara pilih File → New Scene. Untuk mencegah scene tidak tersimpan, maka dapat langsung dilakukan penyimpanan scene dengan memilih menu File → Save Scene As. Simpan pada path folder ../Assets/Resource/Scene.

Arena permainan dibuat terlebih dahulu untuk membatasi pemain dan objek-objek supaya tidak keluar dari arena. Yang dibuat dari 6 buah objek persegi (plane) dengan penambahan beberapa *component* yaitu Mesh collider, Mesh Renderer dan Box Collider. Objek Hand Controller juga diimport untuk menampilkan fitur virtual hand.



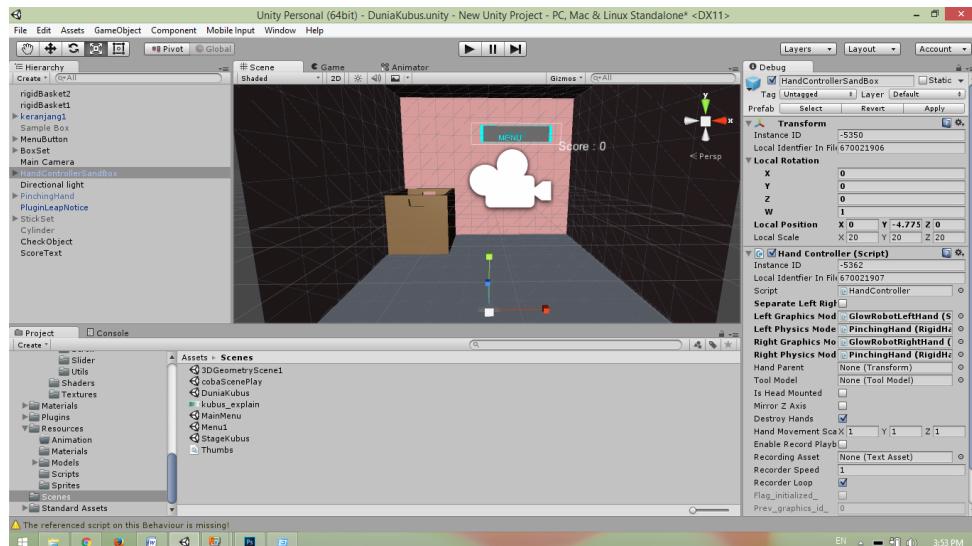
Gambar 3.42 Set Arena Mini Game

Setelah arena siap, penulis menambahkan objek kardus atau keranjang untuk meletakan objek bangunan ruang yang benar. Objek untuk karakter robot dan desain kota juga didownload. Objek didapatkan secara gratis dan bebas lisensi dengan mendownloadnya di <http://3dwarehouse.sketchup.com>



Gambar 3.43 Penambahan Objek Keranjang

Untuk membuat GUI seperti menu dapat menggunakan GUIText dan GUITexture. GUITexture adalah objek 2D yang menampilkan sebuah texture 2D pada layar. Texture ini dapat digunakan untuk menampilkan logo, button, virtual joystick, dan sebagainya.



Gambar 3.44 Penambahan Objek 3D Button

Langkah selanjutnya adalah mengimport objek-objek 3D lain yang berhubungan dengan bangun ruang yang di sekitar kita kedalam scene. Seperti objek dadu, rubik, cone, bola, kayu dan lain-lain. Nantinya ditambahkan script *grabable objet* supaya objek dapat dipilih dan diangkat oleh *virtual hands*. Tambahkan juga property rigidbody pada setiap objek supaya fisik objek dapat terdeteksi.

### 3.5.9 Implementasi Gesture Recognition

Pengenalan gestur tangan (*hand gesture recognition*) pada aplikasi dibuat menjadi 10 variasi tipe gestur yaitu; *Tap*, *Swipe*, *Clap*, *Double Outward Swipe*, *Circular*, *Close Fist*, *Pinch*, *Release*, *Push* dan *Pull*. Tipe-tipe gestur tersebut dijabarkan sebagai berikut ;

#### 1. Swipe Gesture

Seperti dijabarkan oleh Pedro Miguel dalam jurnal berjudul *Analysis and Evaluation of Gesture Recognition using LeapMotion* [16] , Gestur swipe dapat dikenal dengan mendefinisikan gerakan tangan secara vertikal maupun horisontal menggunakan jari telunjuk maupun telapak tangan.

$$H = |D_x| > |D_y| \quad (3.1)$$

maka implementasi kodennya adalah sebagai berikut:

```
if(gestype == "gTYPESWIPE") {
    switch(type){
        case 1 :
            if(gesture.Position.x < 0){
                spinLeft = true ;
                spinRight = false ;
            }
            if(gesture.Position.x > 0){
                spinLeft = false ;
                spinRight = true ;
            }
            break;
    }
}
```

## 2. Key / Button Tap Gesture

```
private void CheckTrigger()
{
    float scale = transform.lossyScale.z;
    m_localTriggerDistance = triggerDistance / scale;
    m_localCushionThickness = Mathf.Clamp(cushionThickness / scale, 0.0f,
                                          0.001f);
    if (m_isPressed == false)
    {
        if (transform.localPosition.z > m_localTriggerDistance)
        {
            m_isPressed = true;
            buttonPressed();
        }
    }
    else if (m_isPressed == true)
    {
        if (transform.localPosition.z < (m_localTriggerDistance - m_localCushionThickness))
        {
            m_isPressed = false;
            buttonReleased();
        }
    }
}
```

## 3. Clap Gesture

```
if(clapGestureRegistered) {
    if(mFrame.Hands.Count == 2) {
        bool leftHandSwipeIn = (!PalmIsHorizontal(mFrame.Hands.Leftmost)) && mFrame.
            Hands.Leftmost.PalmVelocity.x > EasyLeapGesture.MinClapVelocity;
        bool rightHandSwipeIn = (!PalmIsHorizontal(mFrame.Hands.Rightmost)) && mFrame.
            Hands.Rightmost.PalmVelocity.x < -EasyLeapGesture.MinClapVelocity;
        if(leftHandSwipeIn && rightHandSwipeIn) {
            if(mFrame.Hands[0].StabilizedPalmPosition.DistanceTo(mFrame.Hands[1].
                StabilizedPalmPosition) < EasyLeapGesture.MaxPalmClapDistance)
                ClapRecognised(EasyLeapGestureState.STATESTART);
            } else ClapRecognised(EasyLeapGestureState.STATEINVALID);
    }
}
```

## 4. Double Outward Swipe Gesture

```
if(doubleOutwardsSwipeGestureRegistered && leftHandSwipeOut &&
    rightHandSwipeOut) {
    if(mFrame.Hands[0].StabilizedPalmPosition.DistanceTo(mFrame.Hands[1].
        StabilizedPalmPosition) > EasyLeapGesture.MaxPalmDistance)
        DoubleOutwardsSwipeRecognised(EasyLeapGestureState.STATESTART);
} else DoubleOutwardsSwipeRecognised(EasyLeapGestureState.STATEINVALID);
```

## 5. Push Gesture

```
if(pushGestureRegistered) {
    if(mFrame.Hands[0].PalmVelocity.y < -EasyLeapGesture.MinPushPullVelocity)
        PushGestureRecognised(EasyLeapGestureState.STATESTART);
    else PushGestureRecognised(EasyLeapGestureState.STATEINVALID);
}
```

## 6. Pull Gesture

```

if(pullGestureRegistered) {
    if(mFrame.Hands[0].PalmVelocity.y > EasyLeapGesture.MinPushPullVelocity)
        PullGestureRecognised(EasyLeapGestureState.STATESTART);
    else PullGestureRecognised(EasyLeapGestureState.STATEINVALID);
}

private void PullGestureRecognised(EasyLeapGestureState state) {
    if(state == EasyLeapGestureState.STATEINVALID || Time.time <
        pullRestTime + EasyLeapGesture.PullRecoveryTime) {
        gestureList.Remove(-(int)EasyLeapGestureType.PULL);
        return;
    }
    pullRestTime = Time.time;
    RecordNewGesture(-(int)EasyLeapGestureType.PULL,EasyLeapGestureState.
        STATESTART,EasyLeapGestureState.STATEUPDATE,EasyLeapGestureType.
        PULL,-1,mFrame.Hands[0].StabilizedPalmPosition);

}

```

## 7. Circular Gesture

```

String clockwiseness;
if (circle.Pointable.Direction.AngleTo (circle.Normal) <= Math.PI / 2) {
    clockwiseness = "clockwise";
} else {
    clockwiseness = "counterclockwise";
}

```

## 8. Fly Control

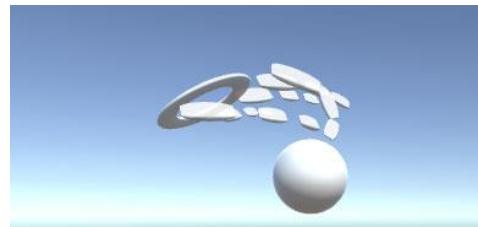
```

void FixedUpdate () {
    Frame frame = m_leapController.Frame();
    if (frame.Hands.Count >= 2) {
        Hand leftHand = GetLeftMostHand(frame);
        Hand rightHand = GetRightMostHand(frame);
        Vector3 avgPalmForward = (frame.Hands[0].Direction.ToUnity() + frame.Hands[1].Direction.ToUnity(
            )) * 0.5f;
        Vector3 handDiff = leftHand.PalmPosition.ToUnityScaled() - rightHand.PalmPosition.ToUnityScaled(
            );
        Vector3 newRot = transform.parent.localRotation.eulerAngles;
        newRot.z = -handDiff.y * 20.0f;
        newRot.y += handDiff.z * 3.0f - newRot.z * 0.03f * transform.parent.GetComponent<Rigidbody>().
            velocity.magnitude;
        newRot.x = -(avgPalmForward.y - 0.1f) * 100.0f;
        if(forceMult < -3){
            forceMult = -4.0f;
            StartCoroutine (MyCoroutine ());
        }else{
            forceMult = 10.0f;
        }
        if (frame.Fingers.Count < 3) {
            forceMult = -3.0f;
        }
        transform.parent.localRotation = Quaternion.Slerp(transform.parent.localRotation, Quaternion.
            Euler(newRot, 0.1f));
        transform.parent.GetComponent<Rigidbody>().velocity = transform.parent.forward * forceMult;
    }
}

```

## 9. Pinch Gesture :

Hansaem Lee, Junseok Park, pada jurnal berjudul *Hand Gesture Recognition in Multi-space of 2D/3D* [24].



**Gambar 3.45** Pinch Gesture

Sumber: *Leap Motion Documentation*

Metode pengenalan gestur pinch pada bidang 2D. Definisi pinch dan release atau gerakan mencubit (mencapit) dan melepaskannya secara 2 Dimensi dijabarkan dengan algoritma sebagai berikut:

- (1) titik sentuhan harus berada di sekeliling objek interaktif.
- (2) jarak setiap ujung jari yang menjepit objek harus lebih kecil daripada nilai threshold. Nilai threshold adalah jarak yang ditentukan dari lebar objek interaktif.

Data jari membutuhkan koordinat dari setiap jari yang pertamakali menjangkau bidang sentuh ( $\text{finger\_1}_{x\text{-in}}$ ,  $\text{finger\_1}_{y\text{-in}}$ ), ( $\text{finger\_2}_{x\text{-in}}$ ,  $\text{finger\_2}_{y\text{-in}}$ ), ..., ( $\text{finger\_n}_{x\text{-in}}$ ,  $\text{finger\_n}_{y\text{-in}}$ ),.. maka nilai maksimum dan minimum jarak jari dijabarkan sebagai berikut:

$$\begin{aligned} \max_x &= \max(\text{finger\_1}_{xin}, \text{finger\_1}_{xout}, \dots, \text{finger\_n}_{xin}, \text{finger\_n}_{xout}) \\ \min_x &= \min(\text{finger\_1}_{xin}, \text{finger\_1}_{xout}, \dots, \text{finger\_n}_{xin}, \text{finger\_n}_{xout}) \\ \max_y &= \max(\text{finger\_1}_{yin}, \text{finger\_1}_{yout}, \dots, \text{finger\_n}_{yin}, \text{finger\_n}_{yout}) \\ \min_y &= \min(\text{finger\_1}_{yin}, \text{finger\_1}_{yout}, \dots, \text{finger\_n}_{yin}, \text{finger\_n}_{yout}) \end{aligned} \quad (3.2)$$

pada saat itu, saat titik tengah objek mereferensi ke titik interaksi / *interaction point* ( $\text{Object}_x$ ,  $\text{Object}_y$ )

$$\min_x \leq \text{Object}_x \leq \max_x \quad (3.3)$$

$$\min_y \leq \text{Object}_y \leq \max_y$$

Saat formula (3.3) terpenuhi, maka kondisi (1) juga terpenuhi. Untuk memenuhi kondisi (2), nilai pergerakan jarak jari haruslah melebihi nilai treshold. Untuk menghitung nilai dari jarak dinamis maka digunakan metode Euclidean :

$$distance_{finger\_n} = \sqrt{finger\_n_{xin} + finger\_n_{xout}} \quad (3.4)$$

Sehingga gestur 'pinch' dinyatakan berhasil jika jumlah dari jarak jari lebih kecil atau sama dengan nilai threshold :

$$distance_{finger\_1} + \dots + distance_{finger\_n} \leq Threshold \quad (3.5)$$

maka implementasi kodennya adalah sebagai berikut:

```
protected PinchState GetNewPinchState() {
    HandModel hand_model = GetComponent<HandModel>();
    Hand leap_hand = hand_model.GetLeapHand();
    Vector leap_thumb_tip = leap_hand.Fingers[0].TipPosition;
    float closest_distance = Mathf.Infinity;
    for (int i = 1; i < HandModel.NUM_FINGERS; ++i) {
        Finger finger = leap_hand.Fingers[i];

        for (int j = 0; j < FingerModel.NUM_BONES; ++j) {
            Vector leap_joint_position = finger.Bone((Bone.BoneType)j).NextJoint;

            float thumb_tip_distance = leap_joint_position.DistanceTo(leap_thumb_tip);
            closest_distance = Mathf.Min(closest_distance, thumb_tip_distance);
        }
    }
    float proximal_length = leap_hand.Fingers[0].Bone(Bone.BoneType.TYPE_PROXIMAL).Length;
    float trigger_distance = proximal_length * grabTriggerDistance;
    float release_distance = proximal_length * releaseTriggerDistance;
    if (closest_distance <= trigger_distance)
        return PinchState.kPinched;
    if (closest_distance <= release_distance && pinch_state_ != PinchState.kReleased &&
        !ObjectReleaseBreak(current_pinch_position_))
        return PinchState.kReleasing;
    return PinchState.kReleased;
}
```

## 10. Release Gesture

Sedangkan untuk gestur release (melepaskan) merupakan kebalikan operator kondisi dari gestur mencapit ;

$$\sum(RDistance) \geq Threshold_{distance} \quad (3.6)$$

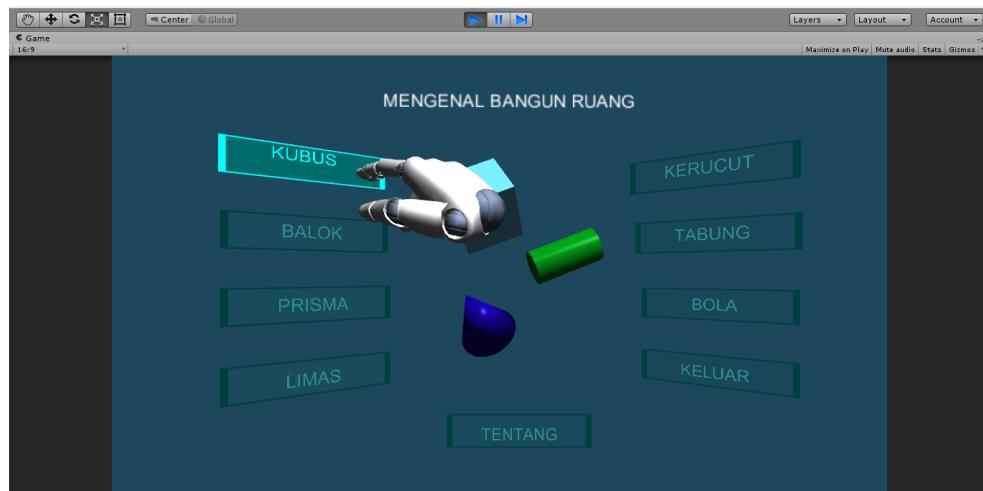
$$\sum(angle) \geq Threshold_{angle}$$

maka implementasi kodennya adalah sebagai berikut ;

```
if (closest_distance <= release_distance && pinch_state_ != PinchState.kReleased &&
    !ObjectReleaseBreak(current_pinch_position_))
    return PinchState.kReleasing;
```

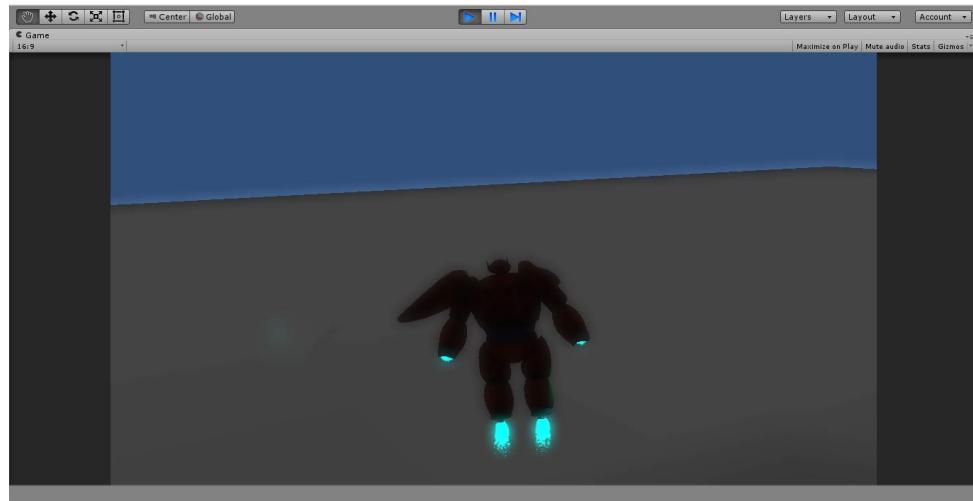
### 3.5.9 Hasil Kompilasi Versi Alpha & Beta

Pada hasil rilis versi alpha , prototip aplikasi telah berjalan baik. Interface aplikasi telah didesain sesuai dengan storyboard. Pada menu utama user dapat memilih menu bangun ruang.



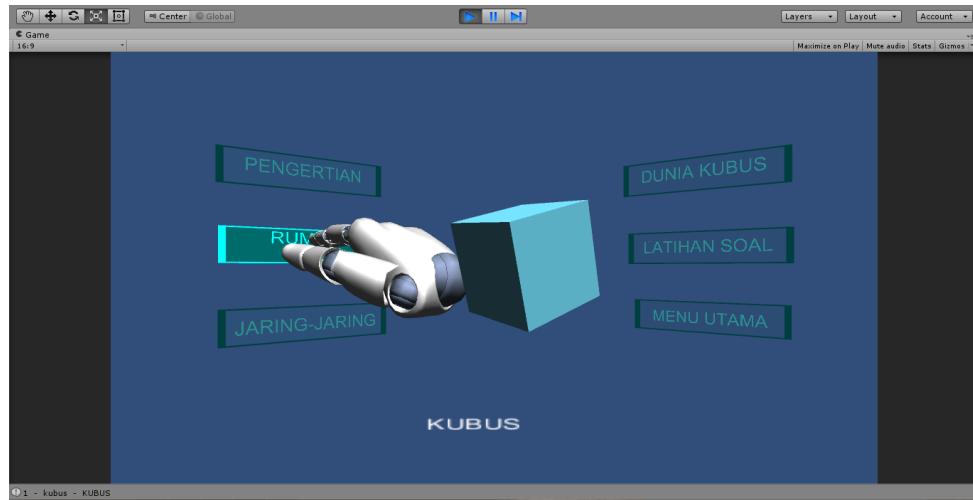
**Gambar 3.46** Tampilan Menu Utama

Di dalam scene mini game 1 , karakter robot diberikan kemampuan untuk terbang dan dikendalikan dengan gestur *fly control* . Nantinya karakter ini akan mampu untuk menyusuri kota untuk mencari titik-titik adatu stasiun pengenalan bangun ruang



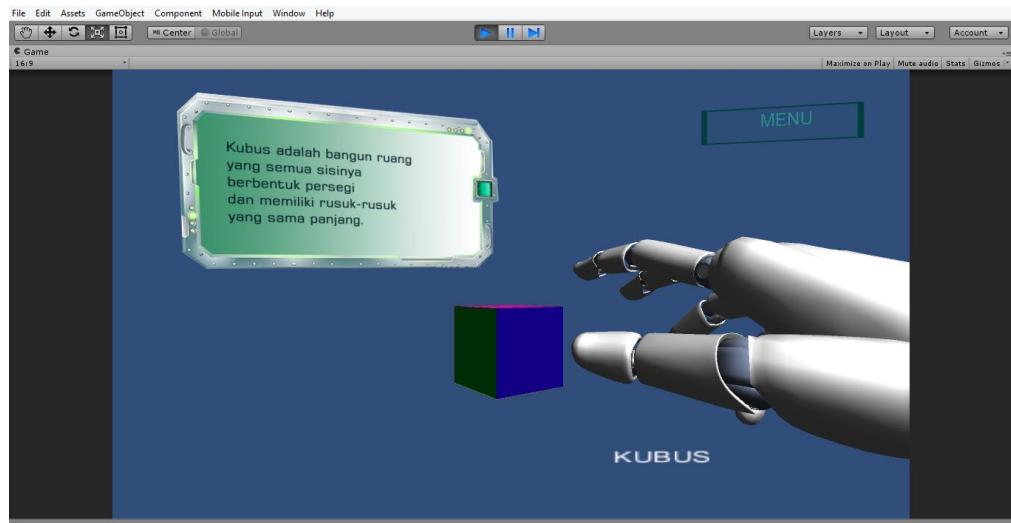
**Gambar 4.47** Scene Mini Game 1

Sedangkan pada menu bangun ruang, user dapat memilih materi yang akan disajikan dengan menggunakan *key tap gesture*.



**Gambar 3.48** Tampilan Menu Bangun Ruang (1)

Implementasi gestur *swipe*, *clap double outward swipe*, sudah berjalan dengan baik. Animasi bangun ruang terbuka atau tertutup telah diterapkan.



**Gambar 3.49** Tampilan Materi Bangun Ruang (2)



**Gambar 3.50** Tampilan Latihan Soal

Scene mini game pada gambar berikut sudah memasuki versi beta, user memilih bangun ruang yang sesuai dengan materi yang disajikan. Tugas user adalah memasukkannya ke dalam kotak. Jika benar, maka akan mendapat nilai atau *score*.



**Gambar 3.51** Scene Mini Game 2

## **BAB IV**

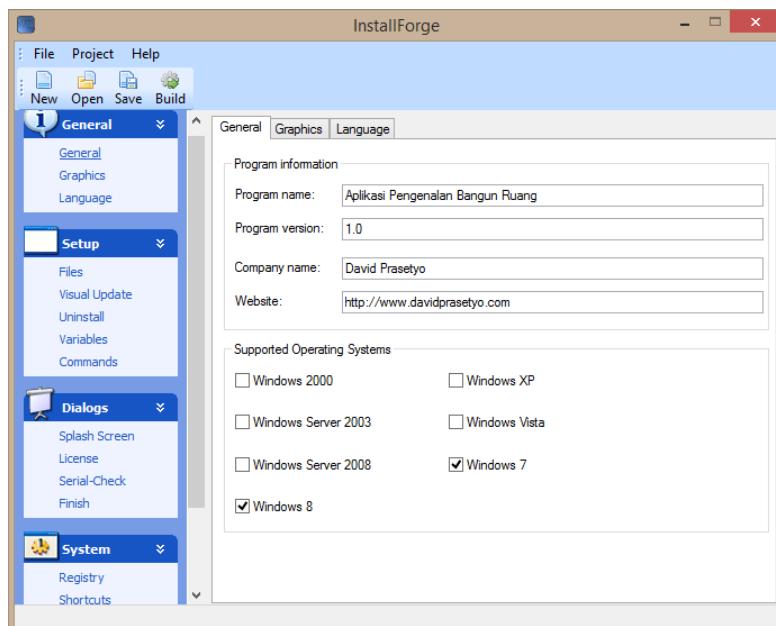
### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1 Implementasi**

Tahap implementasi sistem merupakan tahap perwujudan konsep menjadi aplikasi atau perangkat lunak supaya siap untuk dioperasikan oleh user.

##### **4.1.1 Tahap Paska Produksi**

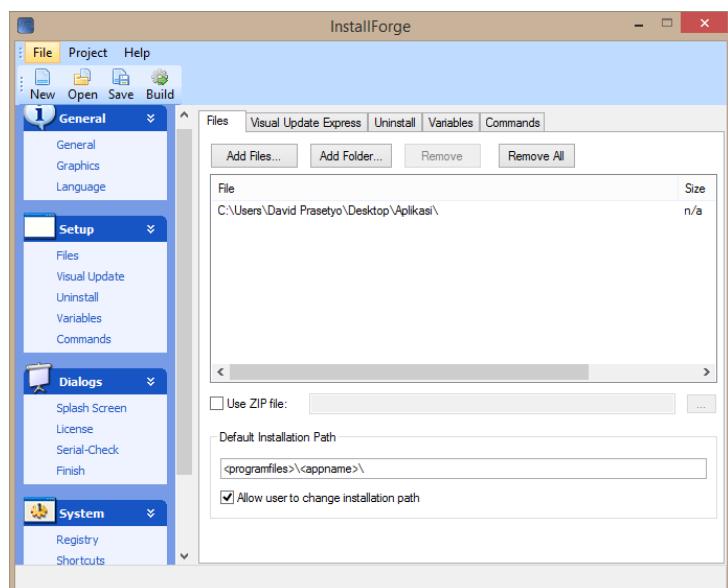
Tahap Pasca Produksi yaitu proses rilis versi *Golden Master* adalah tahap akhir aplikasi setelah melewati pengujian dan evaluasi pada versi alpha dan beta. Versi *Golden Master* dikompilasi (*compile*) dan dipaketkan kedalam sebuah aplikasi *installer* supaya memudahkan proses instalasi dan distribusi aplikasi. *Installer* dibuat menggunakan software InstallForge.



**Gambar 4.1** Pengaturan Installforge untuk Aplikasi

Aplikasi diberikan informasi sesuai dengan nama program dan versinya. Sedangkan untuk kompatibilitas aplikasi dibatasi hanya dapat berjalan di Windows 7 dan Windows 8.

Setelah itu file-file distribusi aplikasi disiapkan dalam satu folder untuk dipaketkan kedalam *installer* seperti yang diperlihatkan pada gambar di bawah ini.



**Gambar 4.2** Penambahan *Source Folder* Aplikasi

Setelah dilakukan proses *build* maka akan dihasilkan installer berekstensi *executable* (.exe). Melalui *installer* ini aplikasi dapat diinstall pada komputer dengan spesifikasi yang telah ditentukan.

#### 4.1.2 Antarmuka Versi Golden Master

Aplikasi Pengenalan Bangun Ruang versi *Golden Master* merupakan aplikasi multimedia edukatif dan interaktif yang ditujukan untuk pelajar sekolah dasar. Maka desain aplikasi yang dibuat harus terkesan menarik dan lebih berwarna.



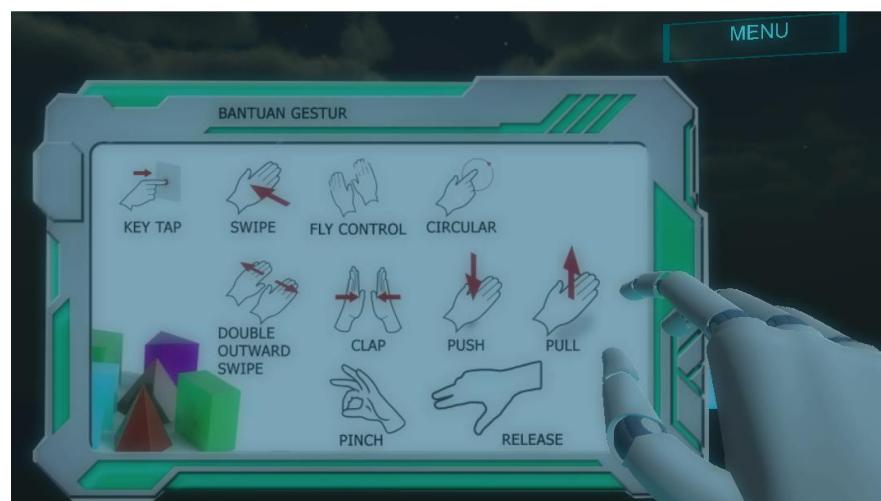
Gambar 4.3 Tampilan *Splash Screen*

Saat aplikasi dijalankan, yang pertama kali muncul adalah intro atau awalan yang berbentuk splash screen selama 2 detik. Setelahnya, pengguna akan diarahkan ke menu utama.



Gambar 4.4 Tampilan Menu Utama

Pengguna dapat memulai dengan menyentuh tombol "MULAI" dengan virtual hands. Selain itu, pada menu utama juga dilengkapi dengan tombol "BANTUAN" untuk mengetahui penggunaan aplikasi dan tombol "TENTANG" untuk menampilkan informasi lebih lanjut.



Gambar 4.5 Tampilan Bantuan

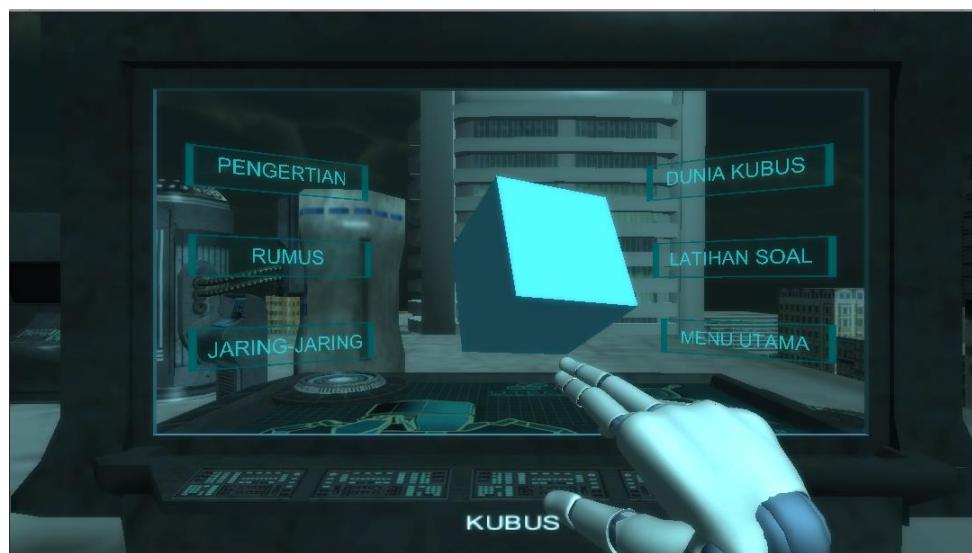


Gambar 4.6 Tampilan Tentang



Gambar 4.7 Tampilan Mini Game 1

Setelah menyentuh tombol "MULAI" maka user melalui karakter robot harus melakukan eksplorasi di dalam sebuah kota untuk mencari stasiun-stasiun pengenalan bangun ruang. Jika robot menyentuh bangun ruang yang bercahaya terang , maka user akan masuk ke dalam scene pengenalan bangun ruang.



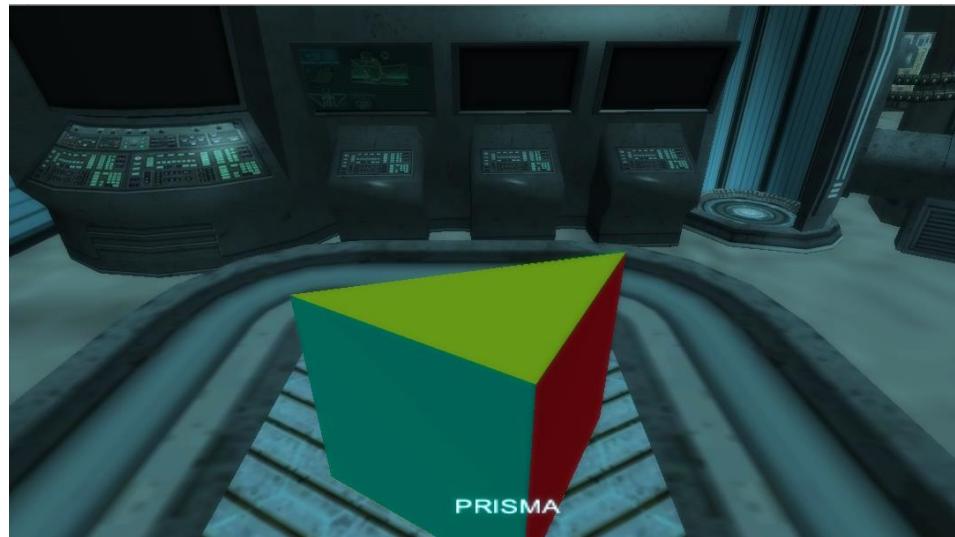
Gambar 4.8 Tampilan Menu Bangun Ruang

Di dalam scene pengenalan bangun ruang ini, pengguna akan disajikan materi-materi dari sebuah bangun ruang. Mulai dari pengertian, rumus, jaring-jaring dan latihan soal.



**Gambar 4.9** Tampilan Scene Pengenalan Bangun Ruang

Pada view yang berbeda, disajikan bentuk bangun ruang 3D beserta penjelasannya. User juga dapat melakukan beberapa gestur tangan dinamis. Gestur swipe untuk memutar bangun ruang, gestur . Sedangkan dengan melakukan gestur push dan pull, maka fitur zoom-in dan zoom-out akan aktif.



Gambar 4.10 Fitur *Zoom-In* Bangun Ruang



Gambar 4.11 Scene Mini Game 2

Pada tahap ini scene Mini Game 2 telah melalui revisi desain untuk menghasilkan tampilan yang lebih menarik. Fitur yang dimiliki pada scene tersebut masih sama seperti versi beta, yaitu gestur pinch dan release. Serta penambahan score jika memilih bangun yang benar.

## 4.2 Pengujian

Pengujian dilakukan melalui beberapa tahap dan jenis pengujian. Penulis membagi pengujian menjadi 2 objek pengujian yaitu pengujian alat atau *Leap Motion Controller* dan Pengujian Aplikasi. Pengujian *Leap Motion Controller* meliputi pengujian yang melibatkan beberapa faktor dan parameter ukuran antara lain cahaya, area interaksi (x,y,z), dan interpretasi gestur statis.

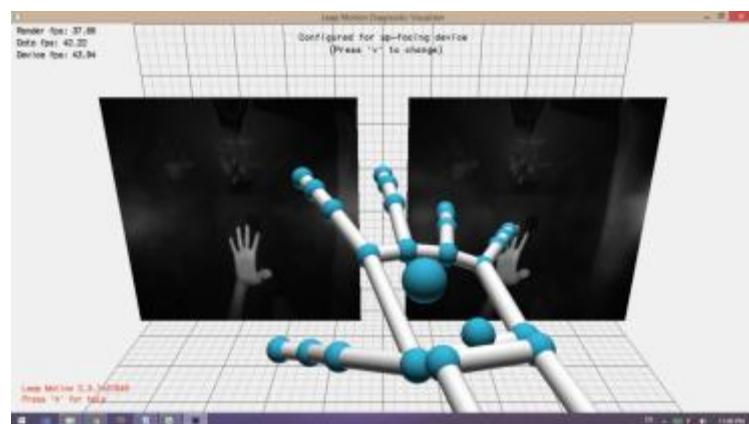
Sedangkan pengujian aplikasi Pengenalan Bangun Ruang adalah tahap perhitungan persentase ketepatan dalam mengenal 10 definisi gestur tangan dinamis dengan parameter derajat interaksi. Pada tahap pengujian tersebut juga diamati kesesuaian *interface* dengan *storyboard* dan performa yang difokuskan pada waktu eksekusi CPU, besar *FPS* (*Frame Per Second*) dan penggunaan memori.

### 4.2.1 Pengujian Leap Motion Controller

#### a. Pengujian Cahaya

Penulis melakukan pengamatan dengan dua kondisi berbeda yaitu pada kondisi ruangan terang (ada cahaya) baik sumber cahaya matahari maupun lampu.

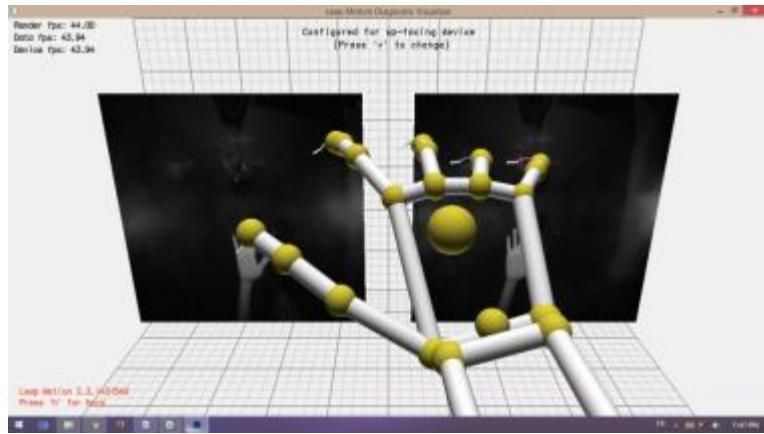
##### a. Kondisi tidak ada cahaya



**Gambar 4.12** Kondisi A (tanpa cahaya)

Hasil pengujian dengan *Leap Motion Visualizer* dan juga pengujian pada aplikasi yang dibuat menunjukkan bahwa citra tangan dapat terbaca.

b. Kondisi ada cahaya (ruangan)

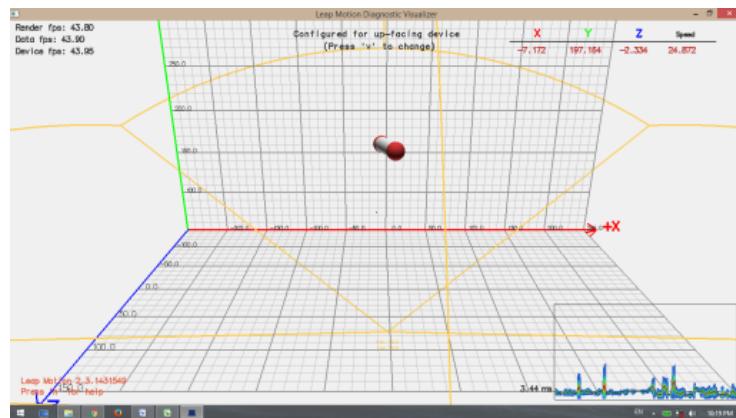


**Gambar 4.13** Kondisi B (ada cahaya)

Hasil pengujian dengan *Leap Motion Visualizer* dan juga pengujian pada aplikasi yang dibuat menunjukkan bahwa citra tangan dapat terbaca.

Melalui pengujian cahaya ini, didapatkan hasil yang sama. Leap Motion Contoller menggunakan kamera infrared sebagai sensor penangkap citra. Oleh karena itu kondisi cahaya tidak berpengaruh pada kemampuan tracking.

b. Pengujian Area Interaksi



**Gambar 4.14** Pengujian dengan Leap Motion Visualizer

Pengukuran dilakukan dengan menggunakan *software Leap Motion Visualizer* yang merupakan software dari *Leap Motion* untuk para pengembang aplikasi. Dengan *software* tersebut, pengembang dapat melakukan analisa secara *real-time*.

Skala interpretasi *Leap Motion* dengan jarak nyata adalah 10:1 jika diukur dalam satuan centimeter (cm), atau 1:1 jika diukur dalam milimeter (mm). Namun demikian, skala tersebut tidak absolut karena berbagai faktor. Maka penulis mengukur persentase kesalahan tiap jarak terhadap sumbu-sumbu acuan di dalam jangkauan area interaksi (*interaction box*) , dengan rumus berikut ;

$$x = \frac{||i| - q|}{q} \cdot 100\% \quad (4.1)$$

Keterangan :

$q$  = Jarak nyata (mm),  $q \neq 0$

$i$  = Jarak interpretasi

Setelah itu, rata-rata kesalahan dihitung dengan rumus persentase sebagai berikut ;

$$R = \frac{\sum x}{n} \quad (4.2)$$

Keterangan :

$R$  = Rata-rata persentase kesalahan (%)

$\sum x$  = Total persentase kesalahan yang terbaca *Leap Motion* (%)

*n* = Jumlah sampel data valid yang terbaca Leap Motion

a. Terhadap Sumbu Y

**Tabel 4.1** Persentase Ketepatan *Tracking* pada Sumbu Y

Sampel	Jarak Nyata (cm)	Jarak Interpretasi (sumbu Y)	Render FPS	Kesalahan (%)
1	1	-	-	-
2	2	23	43	15
3	3	37	44	23,3
4	4	40	44	0
5	5	46	44	8
6	10	101	42	1
7	15	152	44	1,3
8	20	194	44	3
9	25	240	44	4
10	30	312	43	0.6
11	35	355	43	1,4
12	40	430	44	7,5
13	45	474	44	5,3
14	50	560	42	12
15	55	-	-	-
16	60	-	-	-
17	70	-	-	-

Maka nilai R terhadap sumbu y adalah ;

$$R_y = \frac{\Sigma x}{n} = \frac{85,9 \%}{13} = 6,60 \%$$

b. Terhadap Sumbu Z

**Tabel 4.2** Persentase Ketepatan *Tracking* pada Sumbu Z

Sampel	Jarak Nyata (cm)	Jarak Interpretasi (sumbu Y)	Render FPS	Kesalahan (%)
1	30	-	-	-
2	25	-260	42	4
3	10	-104	42	4
4	5	-55	42	10
5	2	-22	44	10
6	1	-11	43	10
7	1	10	43	0
8	2	24	43	20
9	3	35	44	16,6
10	4	40	44	0
11	5	46	44	8
12	10	101	42	1
13	15	152	42	1,3
14	18	184	44	2,2
15	25	252	43	0,8
16	30	336	43	12

Maka nilai R terhadap sumbu z adalah ;

$$R_z = \frac{\Sigma x}{n} = \frac{100,0 \%}{15} = 6,66 \%$$

c. Terhadap Sumbu X

**Tabel 4.3** Persentase Ketepatan *Tracking* pada Sumbu X

Sampel	Jarak Nyata (cm)	Jarak Interpretasi (sumbu Y)	Render FPS	Kesalahan (%)
1	25	-270	42	8
2	10	-112	43	12
3	5	-55	44	10
4	2	-20	42	0
5	1	-11	42	10
6	1	11	43	10
7	2	22	43	10
8	3	32	44	6,6
9	4	40	43	0
10	5	52	43	4
11	10	101	42	1
12	15	152	43	1,3
13	20	230	44	15
14	25	265	43	6
15	30	320	43	6,6
16	40	-	-	-

Maka nilai R terhadap sumbu x adalah ;

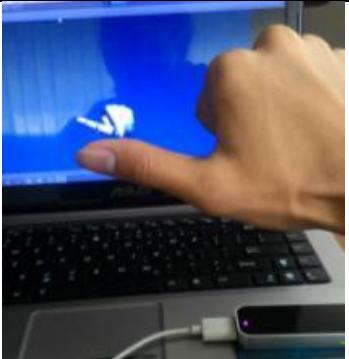
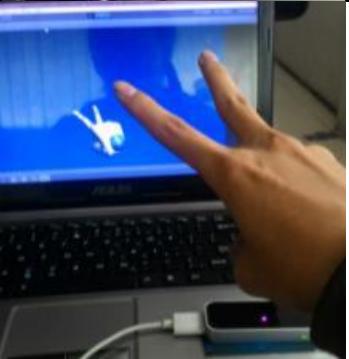
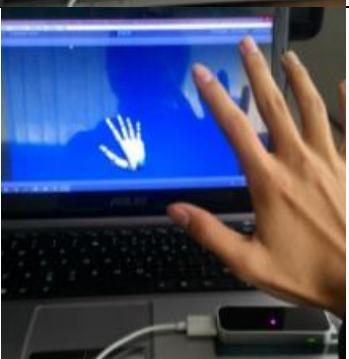
$$R_x = \frac{\Sigma x}{n} = \frac{92,6\%}{15} = 6,61 \%$$

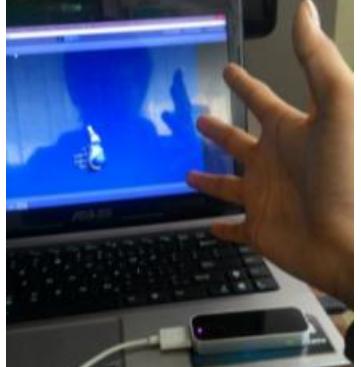
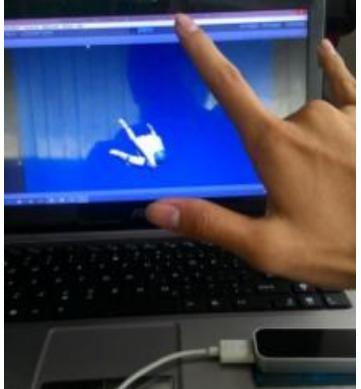
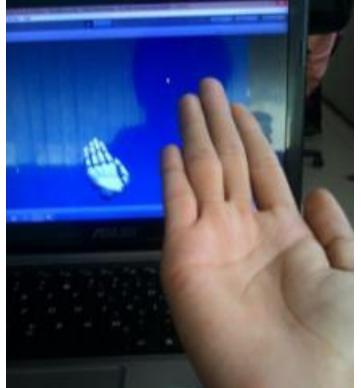
Pada pengujian ketepatan tracking diperoleh rata-rata nilai kesalahan interpretasi dari ketiga sumbu terhadap jarak nyata sebesar 6,6% . Yang artinya *Leap Motion* memiliki ketepatan *tracking* mencapai 94,4 %.

### c. Pengujian Gestur Statis

Pengujian terhadap gestur statis diperlukan untuk mengetahui bagaimana *Leap Motion* secara *default* dapat menginterpretasikan bentuk tangan user pada dunia nyata dengan *virtual hands*. Pengujian dilakukan dengan melakukan beberapa buah gestur dalam posisi-posisi sederhana

**Tabel 4.4** Ketepatan Gestur Statis

Posisi	Gambar	Terbaca?
1		Ya
2		Ya
3		Ya

4		Ya
5		Ya
6		Ya
	Persentase Ketepatan	100%

## 4.2.2 Pengujian Aplikasi Pengenalan Bangun Ruang

### a. Pengujian Antarmuka

Pengujian antarmuka (*interface*) adalah pemeriksaan kesesuaian aplikasi versi Golden Master terhadap storyboard yang telah dirancang.

**Tabel 4.5** Pengujian Interface pada Scene Menu Utama

Event	Objek	Hasil
Gestur Key Tap	Button Keluar	Keluar dari aplikasi
Gestur Key Tap	Button Tentang	Kamera diarahkan ke posisi sprite Tentang
Gestur Key Tap	Button Menu	Kamera diarahkan kembali ke menu utama
Gestur Key Tap	Button Bantuan	Kamera diarahkan ke posisi sprite Bantuan
Gestur Key Tap	Button Mulai	Berpindah ke scene Mini Game 1

**Tabel 4.6** Pengujian Interface pada Scene Mini Game 1

Event	Objek	Hasil
Gestur Fly Control	Aktor	User mengendalikan aktor yaitu sebuah objek robot yang dapat terbang dengan menggunakan kedua tangan user.
Collision	Aktor dan objek bangun ruang	Berpindah ke scene Stasiun Pengenalan Bangun ruang
Collision	Aktor dan bangunan	Aktor akan memantul

**Tabel 4.7** Pengujian Interface pada Scene Pengenalan Bangun Ruang

Event	Objek	Hasil
Gestur Key Tap	Button Pengertian	Sprite pengertian ditampilkan
Gestur Key Tap	Button Rumus	Sprite rumus ditampilkan
Gestur Key Tap	Button Jaring-jaring	Sprite jaring-jaring ditampilkan
Gestur Key Tap	Button Dunia X	Berpindah ke Scene Mini Game 2 yaitu Dunia Bangun Ruang X

		(sesuai dengan topik yang dibahas)
Gestur Swipe	Objek bangun ruang	Animasi bangun ruang berputar
Gestur Double Outward Swipe	Objek bangun ruang	Animasi bangun ruang terbuka (jaring-jaring)
Gestur Clap	Objek bangun ruang	Animasi bangun ruang menutup
Gesur Tap	Button Latihan	Sprite latihan soal ditampilkan
Gestur Circle	Sprite latihan soal	Ganti soal

**Tabel 4.8** Pengujian Interface pada Scene Mini Game 2

Event	Objek	Hasil
Gestur Pinch	Virtual Hands dan objek bangun ruang	Bangun ruang akan terangkat
Gestur Release	Virtual Hands dan objek bangun ruang	Bangun ruang akan terjatuh
Gestur Key Tap	BackButton	Kembali ke scene Pengenalan Bangun Ruang
Collision	CheckObject dan Bangun Ruang	Score bertambah jika bangun yang dimasukkan benar

### b. Pengujian *Dynamic Hand Gesture*

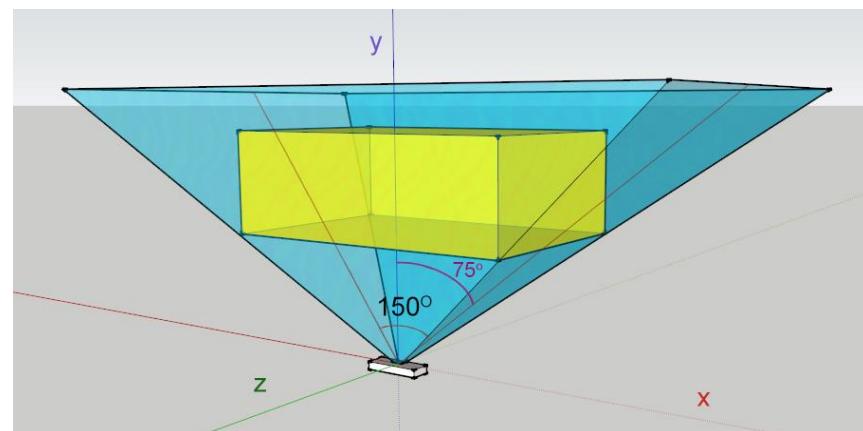
Pengujian *Dynamic Hand Gesture* (gestur tangan dinamis) dilakukan setelah melakukan pengujian (tahap final) untuk mengetahui nilai-nilai ideal dari setiap variabel parameter yang ditentukan dengan satuan tipe data float (f). Sesuai dengan metode siklus pengembangan aplikasi multimedia (MDLC), variabel parameter pada tiap gestur direvisi pada tiap tahapan versi untuk menghasilkan penerjemahan gestur yang lebih baik. Maka, pada tahap final (*Golden Master*) sudah didapatkan nilai

akhir yang merupakan nilai tiap variabel yang paling ideal dalam merepresentasikan gestur dan outputnya.

**Tabel 4.9** Perubahan Variabel Tiap Versi

Gesture	Variabel Parameter	Range	Versi Alpha	Versi Beta	Versi Akhir
Key Tap	triggerDistance	0-1f	0.01f	0.025f	0.025f
Circle	MinRadius	50-100f	50f	60f	70f
Swipe	MinSwipeVelocity	100-300f	100f	200f	200f
Clap	ClapRecoveryTime	0.1-0.5f	0.1f	0.15f	0.15f
	MaxPalmClapDistance	50-100f	50f	70f	90f
Double Outward Swipe	DoubleOutwardsRecoveryTime	0.1-0.5f	0.1f	0.2f	0.2f
Pull	PullRecoveryTime	0.1-0.5f	0.1f	0.2f	0.2f
Push	PushRecoveryTime	0.1-0.5f	0.1f	0.2f	0.3f
Pinch	pinchTriggerDistance	0.5-10f	0.5f	0.7f	0.7f
	pinchObjectDistance	1.0-5.0f	1.0f	1.5f	2.0f
Release	releaseTriggerDistance	1.0-5.0f	1.0f	1.2f	1.2f
Fly Control	forceMult	5-20f	5.0f	7.0f	10.0f

Dari 10 *Hand Gesture* yang didefinisikan, tiap-tiap gestur diuji dan dicatat dengan jumlah berapa kali gestur dikenal, dan berapa kali tidak dikenal. Untuk itu setiap gestur diambil 20 sampel melalui perlakuan 20 kali percobaan untuk tiap sudut evaluasi yang berbeda-beda. Pengujian dilakukan didalam area interaksi Leap Motion.



**Gambar 4.15** Leap Motion Interaction Box

Mengacu pada situs resmi Leapmotion, leapmotion.com, *interaction box* (gambar 4.15) atau jangkauan Leap Motion dalam melakukan tracking data adalah sekitar  $150^\circ$ . Artinya *Leap Motion* memiliki jangkauan  $75^\circ$  untuk setiap sumbu pengacunya jika diukur derajatnya dari sumbu +Y. Namun pada sudut  $75^\circ$ , *Leap Motion* sudah tidak dapat membaca data *tracking* dengan baik, maka penulis menentukan sudut ideal maksimum penggunaan untuk pengujian ini adalah  $60^\circ$ . Maka penulis mengambil 5 kelompok sampel berdasarkan sudut kemiringan antara sumbu +y yaitu  $60^\circ$ ,  $45^\circ$ ,  $30^\circ$ ,  $15^\circ$  dan  $0^\circ$ .

Hasil berikut ini diuji dengan status *high performance mode* dengan komputer prosesor Core i5, RAM 8GB dan VGA 2GB. *Leap Motion* dihubungkan dengan USB 3.0 pada sistem operasi Windows 8.1. Spesifikasi tersebut diperlukan untuk mendapatkan hasil yang maksimal.

**Tabel 4.10** Ketepatan Gestur Dinamis pada Sudut  $60^\circ$

No	Jenis Gestur	Dikenal	Tidak Dikenal	Benar (%)	Salah (%)
1	- Key Tap	16	4	80	20
2	- Swipe	17	3	85	15
3	- Circular	17	3	85	15
4	- Clap	16	4	80	20
5	- Double Outward Swipe	18	2	90	10
6	- Fly Control	15	5	75	25
7	- Pinch	17	3	85	15
8	- Release	20	0	100	0
9	- Push	16	4	80	20
10	- Pull	17	3	85	15
Total		169	31	84.5	15.5

Presentase Benar **84.5 %**

**Tabel 4.11** Ketepatan Gestur Dinamis pada Sudut 45°

No	Jenis Gestur	Dikenal	Tidak Dikenal	Benar (%)	Salah (%)
1	- Key Tap	20	0	100	0
2	- Swipe	18	2	90	10
3	- Circular	17	3	85	15
4	- Clap	17	3	85	15
	- Double Outward Swipe				
5	Swipe	20	0	100	0
6	- Fly Control	17	3	85	15
7	- Pinch	18	2	90	10
8	- Release	20	0	100	0
9	- Push	18	2	90	10
10	- Pull	19	1	95	5
Total		184	16	92	8

Presentase Benar **92 %****Tabel 4.12** Ketepatan Gestur Dinamis pada Sudut 30°

No	Jenis Gestur	Dikenal	Tidak Dikenal	Benar (%)	Salah (%)
1	- Key Tap	20	0	100	0
2	- Swipe	20	0	100	0
3	- Circular	17	3	85	15
4	- Clap	20	0	100	0
	- Double Outward Swipe				
5	Swipe	18	2	90	10
6	- Fly Control	16	4	80	20
7	- Pinch	19	1	95	5
8	- Release	20	0	100	0
9	- Push	17	3	85	15
10	- Pull	17	3	85	15
Total		184	16	92	8

Presentase Benar **92 %**

**Tabel 4.13** Ketepatan Gestur Dinamis pada Sudut 15°

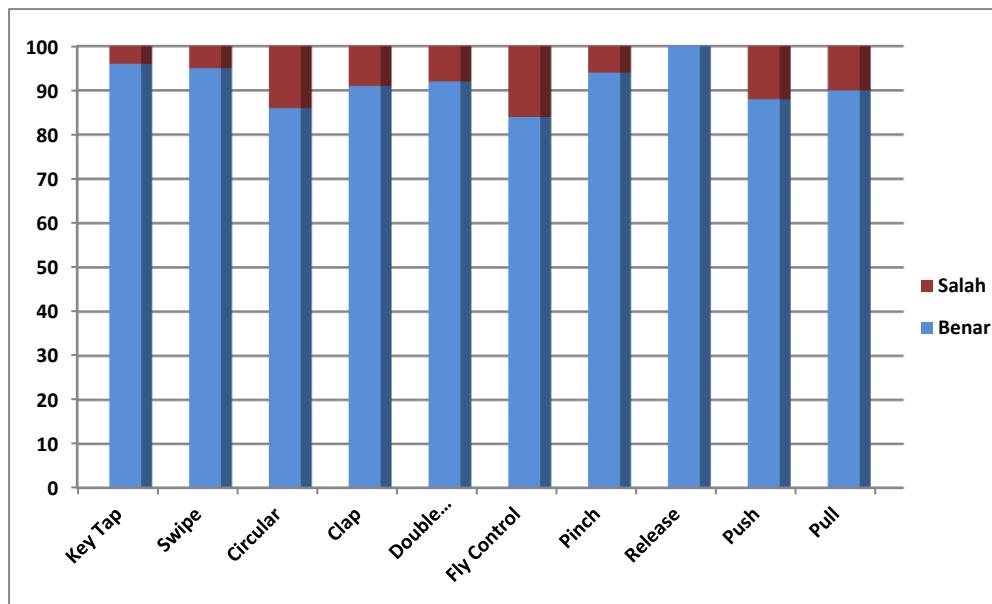
No	Jenis Gestur	Dikenal	Tidak Dikenal	Benar (%)	Salah (%)
1	- Key Tap	20	0	100	0
2	- Swipe	20	0	100	0
3	- Circular	18	2	90	10
4	- Clap	19	1	95	5
5	- Double Outward Swipe	18	2	90	10
6	- Fly Control	17	3	85	15
7	- Pinch	20	0	100	0
8	- Release	20	0	100	0
9	- Push	19	1	95	5
10	- Pull	18	2	90	10
Total		189	11	94.5	5.5

Presentase Benar **94.5 %****Tabel 4.14** Ketepatan Gestur Dinamis pada Sudut 0°

No	Jenis Gestur	Dikenal	Tidak Dikenal	Benar (%)	Salah (%)
1	- Key Tap	20	0	100	0
2	- Swipe	20	0	100	0
3	- Circular	17	3	85	15
4	- Clap	19	1	95	5
5	- Double Outward Swipe	18	2	90	10
6	- Fly Control	19	1	95	5
7	- Pinch	20	0	100	0
8	- Release	20	0	100	0
9	- Push	18	2	90	10
10	- Pull	19	1	95	5
Total		190	10	95	5

Presentase Benar **95 %**

Jika hasil kelima nilai sudut (derajat) parameter tersebut digabungkan, perlakuan untuk menguji tiap gestur seluruhnya adalah sejumlah 100 kali. Maka hasil evaluasi seluruh percobaan diperoleh grafik sebagai berikut;



**Gambar 4.16** Grafik Evaluasi Tingkat Ketepatan 10 Gestur

Dengan detail angka yang tersaji pada tabel 4.15.

**Tabel 4.15** Tabel Evaluasi Tingkat Ketepatan 10 Gestur

No	Jenis Gestur	Benar	Salah
1	Key Tap	96	4
2	Swipe	95	5
3	Circular	86	14
4	Clap	91	9
5	Double Outward Swipe	92	8
6	Fly Control	84	16
7	Pinch	94	6
8	Release	100	0
9	Push	88	12
10	Pull	90	10
	$\sum y$	916	84
	Rata-rata	<b>91.6</b>	<b>8.4</b>

Dengan demikian maka rata-rata tingkat ketepatan ( $\bar{x}$ ) <sup>[28]</sup> dalam mengenal gestur tangan dinamis adalah ;

$$\bar{x} = \frac{91,6}{100} 100\% = \mathbf{91,6\%} \quad (4.3)$$

Persentasi ketepatan tertinggi ada pada gestur *Release* yaitu sebesar 100% , sedangkan persentasi ketepatan terendah adalah gestur *Fly Control* yaitu 84%.

Sedangkan untuk nilai standar deviasi (S) <sup>[29]</sup> diukur dengan persamaan berikut ;

$$S = \sqrt{\frac{\sum y - \frac{(\sum y)^2}{n}}{n-1}} \quad (4.4)$$

dimana  $\sum y$  adalah jumlah pembacaan salah yang dicatat dan  $n$  adalah jumlah variasi gestur. Sebelumnya sudah dihitung terlebih dulu nilai  $\sum y^2 = 918$ .

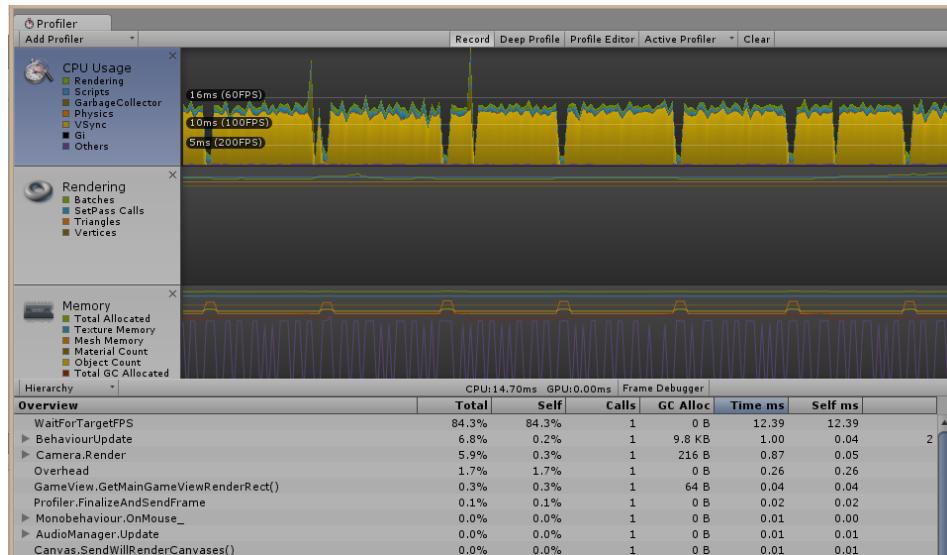
$$S = \sqrt{\frac{918 - \frac{84^2}{10}}{10-1}} = \sqrt{\frac{918 - \frac{7056}{10}}{9}}$$

$$S = \sqrt{\frac{212,4}{9}} = \sqrt{23,6} = \mathbf{4,85}$$

Standar deviasi yang dihasilkan adalah 4,85 yang artinya simpangan nilai pembacaan gestur dalam data yang didapat cukup kecil.

### c. Pengujian Kinerja

Dalam tahap pengujian kinerja atau performa, penulis menggunakan *tester tools* atau modul *monitoring* dan pengujian yang sudah termasuk satu paket dalam aplikasi Unity 5. Modul ini bernama *profiler*.



**Gambar 4.17 Unity 5 Profiler**

Pada pengujian kinerja aplikasi, dilakukan pengamatan dalam penggunaan *resource* tiap-tiap gestur. Standar pembulatan Frame per Second (fps) bagi aplikasi 3D pada umumnya dibagi menjadi 3 satuan standar fps yaitu 15 fps, 30 fps dan 60 fps.

**Tabel 4.16** Kinerja Setiap Gestur

No	Jenis Gestur	Penggunaan Memori (Mb)	FPS (fps)	Eksekusi Kode (ms)	Total Eksekusi CPU (ms)
1	Key Tap	6.6	15	3.7	82.9
2	Swipe	7.3	30	13.3	45.8
3	Circular	8.1	15	12.9	69.8
4	Clap	8.2	30	11.8	50.34
5	Double Outward Swipe	6.3	15	3.9	66.32
6	Fly Control	7.3	60	6.1	30.37
7	Pinch	7	60	1.2	26.53
8	Release	7.9	60	1.2	16.71
9	Push	7.1	60	3.6	26.14
10	Pull	7.5	30	3.2	40.44
	<b>Rata-rata</b>	<b>7.33</b>	<b>37.5</b>	<b>6.09</b>	<b>45.535</b>

Keterangan :

Rata-rata penggunaan memori per gestur = 7,3 Mb

Rata-rata eksekusi kode per gestur = 6,09 ms

Rata-rata FPS (*Frame per Second*) = 37,5 fps

(Mendekati 30 FPS)

Rata-rata total eksekusi CPU (*milisecond*) = 45,53 ms

Pengujian kinerja diatas menunjukkan bahwa aplikasi menggunakan memori untuk setiap gestur sebanyak 7,3 Mb (Megabyte) dengan rata-rata eksekusi kode setiap gestur sebesar 6,09 ms. Selain itu FPS pada aplikasi secara keseluruhan juga cukup ideal karena mencapai 30 FPS.

#### d. Pengujian Suara

Pengujian audio dilakukan dengan mengamati suara atau *audio* yang digunakan dalam perancangan aplikasi. Dari hasil pengamatan, diperoleh data pengujian *audio* pada tabel berikut;

**Tabel 4.17** Pengujian Suara

Nama File	Ukuran Original	Kompresi	Frekuensi	Jumlah Channel	Bit rate
audiobg.ogg	654 Kb	6,7 %	44100 Hz	2	128 kbps
audiobg2.ogg	230 Kb	8,3 %	44101 Hz	2	128 kbps
beep1.mp3	15 Kb	31,8 %	44102 Hz	2	128 kbps
beep2.mp3	73 Kb	19,5 %	44103 Hz	2	128 kbps
jetsound.mp3	38 Kb	39,9 %	44104 Hz	2	64 kbps

Tabel 4.17 menunjukkan bahwa penggunaan file audio rata-rata berfrekuensi 44100 Hz dengan Bit rate 128 kbps. Unity 5 juga melakukan kompresi terhadap file suara yang digunakan pada aplikasi untuk menghemat penggunaan memori saat file suara dieksekusi.

## **BAB V**

### **PENUTUP**

Berdasarkan apa yang telah penulis lakukan pada penelitian ini, dapat ditarik kesimpulan dan saran yang nantinya dapat dipergunakan untuk pengembangan lebih lanjut pada aplikasi multimedia dengan teknologi *Leap Motion Controller*.

#### **5.1 Kesimpulan**

Kesimpulan yang didapat dari penelitian ini adalah sebagai berikut;

- a. Aplikasi Pengenalan Bangun Ruang telah mencapai tahapan versi *Golden Master*, yaitu versi akhir yang ditujukan untuk pelajar sekolah dasar.
- b. Leap Motion Controller adalah sebuah alat pendekripsi gerak tangan yang mempunyai tingkat kesalahan *tracking* yang relatif kecil yaitu 6,6 %.
- c. Dalam aplikasi Pengenalan Bangun Ruang ini terdapat fitur antarmuka berupa *virtual hands*, *virtual button*, serta 10 jenis definisi gestur tangan dinamis yaitu *Tap*, *Swipe*, *Clap*, *Double Outward Swipe*, *Circular*, *Fly Control*, *Pinch*, *Release*, *Push* dan *Pull* dengan sudut penggunaan ideal dibawah 45°.
- d. Dengan menggunakan metode sintaks untuk implementasi *hand gesture recognition* serta pengembangan aplikasi *Multimedia Development Life Cycle* (MDLC), aplikasi dapat mengenal 10 jenis definisi gestur dinamis dengan presentasi ketepatan mencapai 91,6% pada *frame rate* 30 fps dan eksekusi CPU sebesar 45,53 ms.

## 5.2 Saran

Saran dari penulis berdasarkan penelitian ini adalah sebagai berikut;

- a. Leap Motion Controller nantinya dapat dikombinasikan dengan beberapa teknologi seperti Augmented Reality atau Virtual Reality, atau dapat juga dipadukan dengan alat lain seperti Oculus Rift dan Kinect.
- b. Teknologi *Leap Motion* diharapkan dapat dikembangkan supaya diterapkan ke dalam *mobile device* atau *smartphone*
- c. Aplikasi Pengenalan Bangun Ruang dapat dikembangkan menjadi *serious-game* karena banyak skenario dan fitur yang bisa ditambahkan supaya lebih menarik.

## DAFTAR PUSTAKA

- [1] Anonim. (2010). Pembelajaran PAKEM. [Online]. Available : <https://veronikacloset.files.wordpress.com/2010/06/pakem.pdf>
- [2] Motion Capture (2012) [Online]. Available : [http://wonoboyo-desa-kelurahan-rektor.us/id1/learning-713/motion-capture\\_102277\\_wonoboyo-desa-kelurahan-rektor.html](http://wonoboyo-desa-kelurahan.rektor.us/id1/learning-713/motion-capture_102277_wonoboyo-desa-kelurahan-rektor.html)
- [3] A. Danilo dan A. Petracca, “Markerless HandGesture Interface Based on LEAP Motion Controller,” Dep. of Computer Science. Sapienza University, 2013.
- [4] M. Affan dan Hariadi, “Tracking Gerak Tangan Berbasis Pyramidal Lucas-Kanade“. Jurusan Teknik Elektro, FTI-ITS Sukolilo, Surabaya
- [5] Leap Motion Team. (2013). *API Overview* [Online]. Available : [https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Overview.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html)
- [6] L. Nur, “Asyiknya Belajar Bangun Ruang Sisi Datar,” Jakarta: Balai Pustaka, 2012 .
- [7] Y.D Sumanto dan K. Heny, *Gemar Matematika 6* , Jakarta : Pusat Perbukuan Departemen Pendidikan Nasional, 2008
- [8] C. Ok-Hue, “Development of Serious Game for Kids using Leap Motion based on Honey Bee Dance,” Advanced Science and Technology Letters, Vol. 46, pp.260-264 SESRC, 2014.
- [9] F. Bruno, “A 3D Interface to Explore and Manipulate multi-scale Virtual Scenes,” The 7th International Conference on Advances in Computer-Human Interactions ACHI, CNR ITABC, 2014.

- [10] M. Giulio , D. Fabio, “*Hand Gesture Recognition With Leap Motion And Kinect Devices,*” IEEE International Conference, pp. 1565 - 1569, 2014.
- [11] C. Ayung, “Hand Motion Tracking Berbasis Gesture Understanding Menggunakan Leap Gesture Untuk Galeri Batik Virtual 3 Dimensi,” Institut Teknologi Surabaya, 2014.
- [12] P. Alexandros dan S. Kostas, “Personalized Motion Sensor Driven Gesture Recognition in the FIWARE Cloud Platform,” Technical University of Crete (TUC), Proc. IEEE, pp. 19-26, Juni 2015.
- [13] B. Alexandre, “Contactless Sensor LeapMotion,” University of Calgary, CH-1700 Fribourg , Juli 2014.
- [14] N. Michal, P. Olgierd, “Gesture Recognition Library for Leap Motion Controller,” Poznan University of Technology Faculty of Computing Institute of Computing Science, 2014.
- [15] C. Ayung, “3-Dimensional Virtual Classroom for Journey Based on Hand Gesture Understanding using Leap Motion Controller,” Prosiding Konvensi Nasional Asosiasi Pendidikan Teknologi dan Kejuruan (APTEKINDO) ke 7 FPTK UPI, Bandung, November 2014.
- [16] M. Pedro, “Analysis and Evaluation of Gesture Recognition using LeapMotion,” Proc. of the 10<sup>th</sup> Doctoral Symposium in Informatics Engineering - DSIE, pp.83-94, 2015.
- [17] S. Louis, “A 3D Gesture Recognition Interface for Energy Monitoring and Control Applications,” ISE - University of the Algarve, 2015.
- [18] Quwei Lab. (2014). *3D Geometry* [Online]. Available : <https://apps.leapmotion.com/apps/3d-geometry>
- [19] GrowingSoft. (2014). *GeoCrazy* [Online]. Available : <https://apps.leapmotion.com/apps/geocrazy>

- [20] Brendan Polley. (2014). *Form and Function 3D* [Online]. Available : <https://apps.leapmotion.com/apps/form-and-function-3d>
- [21] CyberScience. (2014). *Cyber Science - Motion: Zoology* [Online]. Available : <https://apps.leapmotion.com/apps/cyber-science-motion-zoology>
- [22] Sunset Lake Software. (2014). *Molecules* [Online]. Available : <https://apps.leapmotion.com/apps/molecules>
- [23] Tim Air Putih, *Panduan Penggunaan Aplikasi FOSS The GIMP*, Jakarta : Yayasan Air Putih, 2010.
- [24] L. Hansaem, P. Junseok , “Hand Gesture Recognition in Multi-space of 2D/3D,” IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.6, June 2015
- [25] Meli . (2012) . *System Development Life Cycle*. [Online]. Available : <http://blog.stikom.edu/meli/files/2012/02/SDLC.pdf>
- [26] S. Mischa. *Leap Motion Development Essentials* . Birmingham : PACKT Publishing, 2013.
- [27] S. Blackman . *Unity for Absolute Beginners* . New York : Apress , 2014.
- [28] Rory. (2013). *Rata-rata Hitung (Mean)* [Online]. Available : <http://www.rumusstatistik.com/2013/07/rata-rata-mean-atau-rataan.html>
- [29] Rory. (2013). *Varian Dan Standar Deviasi (Simpangan Baku)* [Online]. Available : <http://rumusstatistik.com/2013/07/varian-dan-standar-deviasi-simpangan.html>