



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI TERAPI PASCA STROKE UNTUK LATIHAN PERGERAKAN JARI TANGAN DENGAN MENGGUNAKAN LEAP MOTION CONTROLLER

Franky Setiawan Daldiri
NRP 5113100056

Dosen Pembimbing
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Dr. Darlis Herumurti, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI TERAPI PASCA STROKE UNTUK LATIHAN PERGERAKAN JARI TANGAN DENGAN MENGGUNAKAN LEAP MOTION CONTROLLER

Franky Setiawan Daldiri
NRP 5113100056

Dosen Pembimbing
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Dr. Darlis Herumurti, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - K1141502

DESAIN A POST-STROKE THERAPY APPLICATION FOR FINGER MOVEMENT EXERCISES USING LEAP MOTION CONTROLLER

Franky Setiawan Daldiri
NRP 5113100056

Advisor
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Dr. Darlis Herumurti, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**Rancang Bangun Aplikasi Terapi Pascastroke Untuk
Latihan Pergerakan Jari Tangan Dengan Menggunakan
Leap Motion Controller**

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Interaksi Grafis dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

Franky Setiawan Daldiri

NRP : 5113100056

Disetujui oleh Dosen Pembimbing Tugas Akhir :

WIJAYANTI NURUL KHOTIMAH, S.Kom., M.Kom.
NIP: 198603122012122004 (Pembimbing 1)

Dr. DARLIS HERUMURTI, S.Kom., M.Kom.
NIP: 197712172003121001 (Pembimbing 2)



**SURABAYA
JULI 2017**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI TERAPI PASCASTROKE UNTUK LATIHAN PERGERAKAN JARI TANGAN DENGAN MENGUNAKAN LEAP MOTION CONTROLLER

Nama Mahasiswa : Franky Setiawan Daldiri
NRP : 5113100056
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Dosen Pembimbing 2 : Dr. Darlis Herumurti, S.Kom., M.Kom.

ABSTRAK

Teknologi pendeteksi gerak tangan dan alat saat ini berhasil dikembangkan dengan baik pada perangkat lunak bernama Leap Motion Controller. Alat ini berfungsi sebagai alternatif kontrol input dalam interaksi antara manusia dengan komputer tanpa menggunakan sentuhan.

Ide yang digunakan dalam tugas akhir ini akan dibangun sebuah aplikasi terapi stroke yang dipadukan dengan permainan simulasi seorang astronot pada luar angkasa. Aplikasi ini dibangun menggunakan Unity dengan Leap Motion SDK. Dengan tugas akhir ini diharapkan pengguna dapat melakukan terapi secara berkesinambungan dengan tidak bosan karena terapi pascastroke merupakan masa penting untuk memulihkan otot motorik pasien sebagai pengguna aplikasi.

Aplikasi telah diuji oleh seorang pasien pascastroke dengan data yang diperoleh yaitu 8 menit 46 detik. Data ini akan digunakan sebagai acuan perkembangan pasien, oleh karena itu aplikasi ini diharapkan dapat membantu pasien pascastroke walaupun perkembangan dari pasien tersebut belum dapat ditentukan karena masih membutuhkan banyak data yang diperoleh dari pasien.

Kata kunci: Leap Motion, Pascastroke, Unity.

[Halaman ini sengaja dikosongkan]

Desain And Implementation Of Therapy Applications For Exercise Movement Training Using Leap Motion Controller

Student Name : Franky Setiawan Daldiri
Student ID : 5113100056
Major : Department Of Informatics FTIf-ITS
Advisor 1 : Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Advisor 2 : Dr. Darlis Herumurti, S.Kom., M.Kom.

ABSTRACT

Hand gesture detection technology and tools are now well developed in software called Leap Motion Controller. This tool itself serves as an alternative in input control in the interaction between humans and computers without using a touch.

The idea used in this final project will be an application of stroke therapy combined with the simulation game of an astronaut in outer space. This app is built using Unity with Leap Motion SDK. The goal is expected users can do therapy on an ongoing basis with no boredom because post stroke therapy is an important period to restore the patient's motor muscles as a user.

Applications have been tested by a patient with a post-stroke with data obtained that is 8 minutes 46 seconds, this data will be used as a reference of patient development therefore this application is expected to help patients pascastroke although the development of the patient can not be determined because it still needs a lot of data obtained Of the patients.

Keywords: Leap Motion, Post Stroke, Unity.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji bagi Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul :

“Rancang Bangun Aplikasi Terapi Pasca Stroke Untuk Latihan Pergerakan Jari Tangan Dengan Menggunakan Leap Motion Controller”

Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa
2. Papa, Mama, Kakak dan keluarga yang selalu memberikan dukungan, baik moral maupun material, secara penuh untuk menyelesaikan Tugas Akhir ini.
3. Ibu Wijayanti Nurul Khotimah selaku dosen pembimbing pertama yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir ini. Tanpa bimbingan rutin setiap minggu mungkin Tugas Akhir ini akan semakin sulit selesai.
4. Bapak Darlis Herumurti selaku dosen pembimbing kedua yang telah bersedia meluangkan waktu untuk memberikan petunjuk serta solusi dari masalah yang ada selama proses pengerjaan Tugas Akhir ini.
5. Bapak/Ibu dosen, staf dan karyawan Jurusan Teknik Informatika ITS yang telah banyak memberikan dukungan, ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
6. Ahmad Rizky Haqiqi, Ayu Kardina Sukmawati, Budi Pangestu, Devira Wiena Pramintya, Eko Putro Fitrianto,

Gian Sebastian Anjasmara, Hariyanto dan Lophita Y Napitupulu yang selalu siap sedia untuk membantu memecahkan masalah dan menghibur disaat penulis mengalami kebosanan maupun membutuhkan pertolongan.

7. Cindy Angela, Gani Ariyanto, Harun Victory, Hongki Cahyo, Meryl Natasha Lesmono dan Romario Wijaya yang bagaikan malaikat selalu memberikan canda tawa serta kebahagiaan.
8. Abdurrachman Rafif Sayudha, Fajar Ade Putra, Ken Genesius Meta Basundara dan Tikva Mooy yang selalu menyediakan waktu untuk berkumpul dan bercerita membahas kesulitan dalam mengerjakan Tugas Akhir ini.
9. Teman-teman penghuni laboratorium Interaksi, Grafika dan Seni yang selalu memberikan solusi dari masalah yang ada pada Tugas Akhir ini.
10. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan Tugas Akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, 5 Juni 2017
Penulis

Franky Setiawan Daldiri

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
DAFTAR KODE SUMBER	xxvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan.....	3
1.3. Batasan Permasalahan	3
1.4. Tujuan.....	4
1.5. Manfaat.....	4
1.6. Metodologi	4
1.6.1. Penyusunan proposal Tugas Akhir	4
1.6.2. Studi literatur	4
1.6.3. Analisis dan Perancangan Sistem.....	5
1.6.4. Implementasi	5
1.6.5. Pengujian dan Evaluasi	5
1.6.6. Penyusunan Buku Tugas Akhir	6
1.7. Sistematika Penulisan.....	6
BAB II DASAR TEORI.....	9
2.1. Rancang Bangun Perangkat Lunak	9
2.2. Stroke	9
2.3. Terapi Fisik Anggota Gerak Tangan Pascastroke	10
2.4. Fisioterapi.....	12
2.5. Unity 3D.....	13
2.6. Leap Motion Controller (LMC).....	15
2.7. SDK Leap Motion Developer.....	17
2.7.1. Pointables	18
2.7.2. InteractionBox	19

2.8.	Blender	19
2.9.	Teknologi Informasi untuk Terapi Rehabilitasi Penderita Pascastroke	20
2.9.1.	<i>Kinect</i>	20
2.9.2.	<i>Augmented Reality</i>	21
2.9.3.	Konsol <i>Wii</i>	21
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		23
3.1.	Analisis Perangkat Lunak.....	23
3.1.1.	Deskripsi Umum Perangkat Lunak.....	23
3.1.2.	Spesifikasi Kebutuhan Fungsional Perangkat Lunak	24
3.2.	Perancangan Perangkat Lunak.....	25
3.2.1.	Model Kasus Penggunaan	25
3.2.2.	Definisi Kasus Penggunaan	26
3.2.2.1.	Melihat Penjelasan Aplikasi	26
3.2.2.2.	Melakukan Pelatihan	27
3.2.2.3.	Melihat Hasil Latihan	29
3.2.3.	Definisi Aktor	30
3.2.4.	Perancangan Model 3D	30
3.2.4.1.	Perancangan Model 3D Tombol On	31
3.2.4.2.	Perancangan Model 3D Tuas.....	32
3.2.4.3.	Perancangan Model 3D Tuas Geser	33
3.2.4.4.	Perancangan Model 3D Tuas Putar	33
3.2.4.5.	Perancangan Model 3D Bintang.....	34
3.2.4.6.	Perancangan Model 3D Kotak Sebagai Penampung Asteroid	35
3.2.5.	Arsitektur Umum Sistem	35
3.2.6.	Rancangan Antarmuka Aplikasi.....	36
3.2.6.1.	Rancangan Antarmuka Menu Utama.....	37
3.2.6.2.	Rancangan Antarmuka Pelajari Selanjutnya.....	38
3.2.6.3.	Rancangan Antarmuka Menu Tahap 1	38
3.2.6.4.	Rancangan Antarmuka Menu Tahap 2	39
3.2.6.5.	Rancangan Antarmuka Menu Tahap 3	40
3.2.6.6.	Rancangan Antarmuka Menu Tahap 4	40
3.2.6.7.	Rancangan Antarmuka Tahap 1	41

3.2.6.8.	Rancangan Antarmuka Tahap 2	44
3.2.6.9.	Rancangan Antarmuka Tahap 3	45
3.2.6.10.	Rancangan Antarmuka Tahap 4 Level 1	45
3.2.6.11.	Rancangan Antarmuka Tahap 4 Level 2	46
3.2.6.12.	Rancangan Antarmuka Tahap 4 Level 3	47
3.2.7.	Perancangan Proses Aplikasi.....	48
3.2.7.1.	Rancangan Proses Penentuan Level Pelatihan	48
3.2.7.2.	Rancangan Proses Sistem Tiap Tahap.....	50
3.2.7.3.	Rancangan Proses Pemilihan Objek Latihan	51
3.2.7.4.	Rancangan Proses Hasil Pelatihan.....	51
3.2.7.5.	Rancangan Proses Melihat Penjelasan Tiap Tahap	52
BAB IV IMPLEMENTASI.....		53
4.1.	Lingkungan Implementasi	53
4.2.	Implementasi Model 3D	53
4.2.1.	Implementasi Model 3D Tombol On	54
4.2.2.	Implementasi Model 3D Tuas	54
4.2.3.	Implementasi Model 3D Tuas Geser	55
4.2.4.	Implementasi Model 3D Tuas Putar	56
4.2.5.	Implementasi Model 3D Bintang	56
4.2.6.	Implementasi Model 3D Kotak Sebagai Penampung Asteroid	57
4.3.	Implementasi Antar muka	57
4.3.1.	Impelementasi Antarmuka Menu Utama.....	57
4.3.2.	Impelementasi Antarmuka Pelajari Selanjutnya..	58
4.3.3.	Impelementasi Antarmuka Menu Tahap 1	59
4.3.4.	Implementasi Antarmuka Menu Tahap 2	59
4.3.5.	Impelementasi Antarmuka Menu Tahap 3	60
4.3.6.	Impelementasi Antarmuka Menu Tahap 4	61
4.3.7.	Impelementasi Antarmuka Tahap 1 Level 1.....	61
4.3.8.	Impelementasi Antarmuka Tahap 1 Level 2.....	62
4.3.9.	Impelementasi Antarmuka Tahap 1 Level 3.....	63
4.3.10.	Impelementasi Antarmuka Tahap 1 Level 4.....	63

4.3.11.	Impelementasi Antarmuka Tahap 1 Level 5.....	64
4.3.12.	Impelementasi Antarmuka Tahap 2.....	65
4.3.13.	Impelementasi Antarmuka Tahap 3.....	65
4.3.14.	Impelementasi Antarmuka Tahap 4 Level 1.....	66
4.3.15.	Impelementasi Antarmuka Tahap 4 Level 2.....	67
4.3.16.	Impelementasi Antarmuka Tahap 4 Level 3.....	67
4.4.	Implementasi Proses.....	68
4.4.1.	Implementasi Controller Tahap 1.....	68
4.4.2.	Implementasi Tahap 1 Level 1.....	74
4.4.3.	Implementasi Tahap 1 Level 2.....	75
4.4.3.1.	Mendeteksi Collider Tangan Pengguna Dengan Tombol.....	75
4.4.3.2.	Trigger Pergantian Tampilan Monitor.....	76
4.4.4.	Implementasi Tahap 1 Level 3.....	76
4.4.5.	Implementasi Tahap 1 Level 4.....	77
4.4.6.	Implementasi Tahap 1 Level 5.....	79
4.4.6.1.	Mendeteksi Besar Sudut Pergelangan Tangan Pengguna.....	79
4.4.6.2.	Check Besar Sudut Putar Pergelangan Tangan Pengguna.....	81
4.4.6.3.	Trigger Lampu Mesin Berdasarkan Besaran Sudut Putar Pergelangan Tangan Pengguna.....	81
4.4.7.	Implementasi Tahap 2.....	83
4.4.8.	Implementasi Tahap 3.....	87
4.4.9.	Implementasi Instantiate Line Tahap 4.....	90
4.4.10.	Implementasi Tahap 4 Level 1.....	94
4.4.11.	Implementasi Tahap 4 Level 2.....	98
4.4.12.	Implementasi Tahap 4 Level 3.....	103
4.5.	Implementasi Integrasi Leap Motion dengan Unity ..	107
BAB V PENGUJIAN DAN EVALUASI.....		109
5.1.	Lingkungan Pengujian.....	109
5.2.	Pengujian Aplikasi.....	109
5.2.1.	Skenario Pengujian Fungsionalitas.....	110
5.2.2.	Hasil Uji Coba Fungsionalitas.....	110
5.2.2.1.	Uji Coba Melihat Penjelasan Aplikasi.....	111

5.2.2.2.	Uji Coba Latihan Terapi.....	112
5.2.2.3.	Uji Coba Melihat Hasil Terapi	114
5.3.	Pengujian Pengguna	115
5.3.1.	Kuisisioner Pengguna Dalam Pengujian Aplikasi	118
5.3.2.	Hasil Pengujian Pengguna.....	120
5.3.3.	Kritik dan Saran Pengguna.....	121
5.4.	Konsultasi Dengan Ahli Fisioterapi	121
5.5.	Evaluasi Pengujian	122
BAB VI KESIMPULAN DAN SARAN.....		125
6.1.	Kesimpulan.....	125
6.2.	Saran.....	125
DAFTAR PUSTAKA		127
LAMPIRAN		131
BIODATA PENULIS		135

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Unity3D	14
Gambar 2.2 Perkembangan Leap Motion Controller	15
Gambar 2.3 Leap Motion Controller	16
Gambar 2.4 Area Jangkauan Inframerah dalam Dua Dimensi	16
Gambar 2.5 Area Jangkauan Inframerah dalam Tiga Dimensi ...	17
Gambar 2.6 Representasi Finger	18
Gambar 2.7 Representasi Tool	18
Gambar 2.8 Representasi Wilayah dari InteractionBox	19
Gambar 2.9 Tampilan Antarmuka Blender	20
Gambar 3.1 Diagram Kasus Penggunaan	26
Gambar 3.2 Perancangan Model 3D Tombol On	32
Gambar 3.3 Perancangan Model 3D Tuas 3D	32
Gambar 3.4 Perancangan Model 3D Tuas Geser 3D	33
Gambar 3.5 Perancangan Model 3D Tuas Putar 3D	34
Gambar 3.6 Perancangan Model 3D Bintang	34
Gambar 3.7 Perancangan Model 3D Box Penampung Asteroid ..	35
Gambar 3.8 Rancangan Sederhana Arsitektur Aplikasi	36
Gambar 3.9 Rancangan Antarmuka Menu Utama	37
Gambar 3.10 Rancangan Antarmuka Tutorial Terapi	38
Gambar 3.11 Rancangan Antarmuka Menu Tahap 1	39
Gambar 3.12 Rancangan Antarmuka Menu Tahap 2	39
Gambar 3.13 Rancangan Antarmuka Menu Tahap 3	40
Gambar 3.14 Rancangan Antarmuka Menu Tahap 4	41
Gambar 3.15 Rancangan Antarmuka Tahap 1 Level 1	42
Gambar 3.16 Rancangan Antarmuka Tahap 1 Level 2	42
Gambar 3.17 Rancangan Antarmuka Tahap 1 Level 3	42
Gambar 3.18 Rancangan Antarmuka Tahap 1 Level 4	43
Gambar 3.19 Rancangan Antarmuka Tahap 1 Level 5	43
Gambar 3.20 Rancangan Antarmuka Tahap 2	44
Gambar 3.21 Rancangan Antarmuka Tahap 3	45
Gambar 3.22 Rancangan Antarmuka Tahap 4 Level 1	46

Gambar 3.23 Rancangan Antarmuka Tahap 4 Level 2.....	47
Gambar 3.24 Rancangan Antarmuka Tahap 4 Level 3.....	48
Gambar 4.1 Implementasi Model 3D Tombol On.....	54
Gambar 4.2 Implementasi Model 3D Tuas	55
Gambar 4.3 Implementasi Model 3D Tuas Geser	55
Gambar 4.4 Implementasi Model 3D Tuas Putar	56
Gambar 4.5 Implementasi Model 3D Bintang.....	56
Gambar 4.6 Implementasi Model 3D Kotak Penampung Asteroid	57
Gambar 4.7 Implementasi Antarmuka Menu Utama.....	58
Gambar 4.8 Implementasi Antarmuka Pelajari Selanjutnya.....	58
Gambar 4.9 Implementasi Antarmuka Menu Tahap 1	59
Gambar 4.10 Implementasi Antarmuka Menu Tahap 2	60
Gambar 4.11 Implementasi Antarmuka Menu Tahap 3	60
Gambar 4.12 Implementasi Antarmuka Menu Tahap 4	61
Gambar 4.13 Implementasi Antarmuka Tahap 1 Level 1.....	62
Gambar 4.14 Implementasi Antarmuka Tahap 1 Level 2.....	62
Gambar 4.15 Implementasi Antarmuka Tahap 1 Level 3.....	63
Gambar 4.16 Implementasi Antarmuka Tahap 1 Level 4.....	64
Gambar 4.17 Implementasi Antarmuka Tahap 1 Level 5.....	64
Gambar 4.18 Implementasi Antarmuka Tahap 2.....	65
Gambar 4.19 Implementasi Antarmuka Tahap 3.....	66
Gambar 4.20 Implementasi Antarmuka Tahap 4 Level 1.....	66
Gambar 4.21 Implementasi Antarmuka Tahap 4 Level 2.....	67
Gambar 4.22 Implementasi Antarmuka Tahap 4 Level 3.....	68
Gambar 4.23 Hasil Proses Impor <i>Package Leap Motion</i>	108
Gambar 4.24 Implementasi Input Area Leap Motion.....	108
Gambar 5.1 Uji Coba Saat Melihat Penjelasan Aplikasi.....	112
Gambar 5.2 Uji Coba Aplikasi Saat Tahap 1 Level 1	113
Gambar 5.3 Uji Coba Aplikasi Saat Tahap 3	113
Gambar 5.4 Uji Coba Aplikasi Saat Tahap 4 Level 1	114
Gambar 5.5 Uji Coba Saat Pengguna Dalam Tahap 1.....	115
Gambar 5.6 Pengujian Pengguna Tahap 2.....	116
Gambar 5.7 Pengujian Pengguna Tahap 3.....	117
Gambar 5.8 Pengujian Pengguna Tahap 4 Level 2.....	117

Gambar 5.9 Pengujian Pengguna Tahap 4 Level 3	118
Gambar A.0.1 Flowchart Rancangan Proses Aplikasi	131
Gambar A.0.2 Kuesioner Responden Lembar 1	132
Gambar A.0.3 Kuesioner Responden Lembar.....	133

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Rentang Gerak.....	13
Tabel 3.1 Daftar Spesifikasi Kebutuhan Fungsional Perangkat Lunak.....	24
Tabel 3.2 Daftar Kasus Penggunaan	25
Tabel 3.3 Spesifikasi Kasus Penggunaan Melihat Penjelasan Aplikasi	27
Tabel 3.4 Spesifikasi Kasus Penggunaan Melakukan Pelatihan	28
Tabel 3.5 Spesifikasi Kasus Penggunaan Melihat Hasil Latihan	29
Tabel 3.6 Deskripsi Pengguna.....	30
Tabel 3.7 Daftar Objek Dari Unity Asset Store	31
Tabel 3.8 Spesifikasi Atribut Antarmuka Menu Utama.....	37
Tabel 3.9 Spesifikasi Atribut Antarmuka Tutorial Terapi.....	38
Tabel 3.10 Spesifikasi Atribut Antarmuka Menu Tahap 1.....	39
Tabel 3.11 Spesifikasi Atribut Antarmuka Menu Tahap 2.....	40
Tabel 3.12 Spesifikasi Atribut Antarmuka Menu Tahap 3.....	40
Tabel 3.13 Spesifikasi Atribut Antarmuka Menu Tahap 4.....	41
Tabel 3.14 Spesifikasi Atribut Antarmuka Tahap 1.....	43
Tabel 3.15 Spesifikasi Atribut Antarmuka Tahap 2.....	44
Tabel 3.16 Spesifikasi Atribut Antarmuka Tahap 3.....	45
Tabel 3.17 Spesifikasi Atribut Antarmuka Tahap 4 Level 1.....	46
Tabel 3.18 Spesifikasi Atribut Antarmuka Tahap 4 Level 2.....	47
Tabel 3.19 Spesifikasi Atribut Antarmuka Tahap 4 Level 3.....	48
Tabel 4.1 Spesifikasi Lingkungan Implementasi	53
Tabel 5.1 Lingkungan Pengujian Sistem.....	109
Tabel 5.2. Hasil Uji Coba Melihat Penjelasan Aplikasi.....	111
Tabel 5.3 Hasil Uji Coba Latihan Terapi	112
Tabel 5.4 Uji Coba Melihat Hasil Terapi	114
Tabel 5.5 Rentang Nilai.....	119
Tabel 5.6 Kuesioner Pengguna.....	119
Tabel 5.7 Hasil Pengujian Pengguna Berumur 60.....	120

Tabel 5.8 Hasil Keseluruhan Pengujian Pengguna Dalam Persenentase	121
Tabel 5.9 Kritik dan Saran Pengguna.....	121

DAFTAR KODE SUMBER

Kode Sumber 4.1 Controller Tahap 1	73
Kode Sumber 4.2 Mendeteksi Collider Tangan Pengguna Dengan Monitor.....	74
Kode Sumber 4.3 Mendeteksi Collider Tangan Pengguna Dengan Tombol	76
Kode Sumber 4.4 Trigger Pergantian Tampilan Monitor.....	76
Kode Sumber 4.5 Mendeteksi Collider Tangan Pengguna Dengan Tuas	77
Kode Sumber 4.6 Mendeteksi Tuas Geser Dalam Posisi Penuh .	79
Kode Sumber 4.7 Mendeteksi Besar Sudut Pergelangan Tangan Pengguna	80
Kode Sumber 4.8 Check Besar Sudut Putar Pergelangan Tangan Pengguna	81
Kode Sumber 4.9 Trigger Lampu Mesin Berdasarkan Besar Sudut Putar Pergelangan Tangan Pengguna.....	83
Kode Sumber 4.10 Collider Tangan Pengguna Dengan Bintang Sebagai Objek Benda	87
Kode Sumber 4.11 Deteksi Objek Memasuki Kotak Dan Animasi	90
Kode Sumber 4.12 Animasi Kotak.....	90
Kode Sumber 4.13 Instantiate Line Tahap 4.....	94
Kode Sumber 4.14 Pengecekan Collider Dua Benda Terhadap Instantiate Line Pengguna	97
Kode Sumber 4.15 Deteksi Collider Pada Sphere Pertama.....	97
Kode Sumber 4.16 Deteksi Collider Pada Sphere Kedua	98
Kode Sumber 4.17 Pengecekan Collider Tiga Benda Terhadap Instantiate Line Pengguna	102
Kode Sumber 4.18 Deteksi Collider Pada Sphere Ketiga	102
Kode Sumber 4.19 Pengecekan Collider Empat Benda Terhadap Instantiate Line Pengguna	106
Kode Sumber 4.20 Deteksi Collider Pada Sphere Keempat	107

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan, batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Perkembangan teknologi komputer khususnya di bidang Interaksi Manusia dan Komputer (IMK) merupakan perkembangan teknologi yang menjadi pusat perhatian sejak munculnya perangkat baru yang memanfaatkan *motion capture* untuk menunjang interaksi dengan komputer (seperti *Nintendo Wii Remote*, *Kinect* dan *Leap Motion*) [1]. Penggunaan antarmuka dengan memanfaatkan gerakan dapat meningkatkan keuntungan dan kemudahan, terlebih lagi interaksi merupakan bagian penting dalam aplikasi seperti seni, bantuan medis, dan simulasi. *Leap Motion* yang nantinya akan digunakan dalam tugas akhir ini merupakan alat penerima sensor gerakan jarak dekat. Alat ini digunakan untuk menangkap input dan kontrol terhadap komputer tanpa menyentuh sama sekali.

Stroke atau *cerebrovascular accident* (CVA), adalah hilangnya fungsi-fungsi otak dengan cepat, karena gangguan suplai darah ke otak. Hal ini dapat terjadi karena iskemia (berkurangnya aliran darah) dikarenakan oleh penyumbatan (*thrombosis*, *arterial embolism*), atau adanya pendarahan (*hemorrhage*) [2]. Stroke dapat berdampak pada kehidupan penderitanya seperti kesulitan untuk berbicara, menggerakkan anggota tubuh, bahkan depresi yang berat. Kerusakan akibat stroke bisa meluas dan berlangsung lama, sebelum pulih seutuhnya penderita perlu melakukan terapi secara simultan dan dalam waktu yang panjang. Namun sebagian besar dari mereka tidak melakukan terapi secara terus menerus dikarenakan biaya terapi yang relatif mahal.

Saat ini perkembangan teknologi sangatlah pesat, bahkan teknologi sudah dapat menyokong berbagai aspek kehidupan seperti *healthcare* yang bergerak pada bidang kesehatan. Berbagai macam inovasi pun telah di realitaskan mulai dari deteksi dini gejala stroke, sampai terapi pascastroke. Aplikasi terapi pascastroke terdahulu yaitu menggunakan teknologi *augmented reality*, konsol *wii* dan *kinect*. Kekurangan pada aplikasi terdahulu dengan menggunakan teknologi *augmented reality* yaitu tangan yang terkadang hilang bila marker tertutupi dan kurang halus dalam pergerakannya, sedangkan pada konsol *wii* pasien harus membawa *remote wii* pada genggamannya dimana hal tersebut membuat pasien kelelahan pada tangan untuk memegang *remote wii* tersebut dan teknologi yang menggunakan *kinect* yang merupakan aplikasi terapi pascastroke yang sangat inovatif akan tetapi pada bagian tertentu seperti gerakan tangan yang kurang macam gerak serta kurang detail. Berawal dari hal-hal tersebut maka aplikasi dengan Leap Motion Controller (LMC) memiliki kelebihan berupa gerakan yang lebih halus serta dapat fokus pada terapi gerak tangan dengan bermacam-macam gerakan yang membantu perkembangan otot tangan pasien pascastroke.

Dalam aplikasi yang akan dibuat, pengguna dapat melakukan terapi yang berfokus pada otot lengan dan pergelangan tangan. Dengan menggunakan LMC pengguna dapat mensimulasikan gerakan sesuai petunjuk dalam aplikasi. Leap Motion akan mendeteksi setiap gerakan tangan pengguna, bila pengguna melakukan gerakan memutar pergelangan tangan atau mengangkat tangan maka gambaran virtual dalam aplikasi juga akan melakukan gerakan yang sama. Fokus utama aplikasi ini yaitu menyediakan media baru untuk melakukan terapi bagi pengguna yang di dalamnya telah disusun dan di urutkan secara baik dan benar gerakan-gerakan yang menstimulasi otot tangan untuk dapat saling bekerja sama membentuk suatu gerakan tangan, hal ini dimaksudkan karena ketika otot tidak dapat digerakkan atau mengalami gangguan gerak dari otak dan menyebabkan tidak terjadinya gerakan yang searah maka akan membuat otot bergerak

di luar kendali atau bahkan tidak dapat digerakkan. Oleh sebab itu pada aplikasi ini telah disusun dan disiapkan setiap gerakan dari yang termudah sampai tahap yang membutuhkan keseimbangan otot tangan.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana rancang bangun dan implementasi terapi gerak pada jari tangan pascastroke ?
2. Bagaimana cara merancang virtual terapi pada aplikasi ini sehingga dapat dimainkan sekaligus menerapi penderita ?
3. Bagaimana cara menyimpan data setiap tahap dan menghasilkan file baru berupa laporan hasil terapi dari tahap pertama hingga tahap terakhir sehingga dapat memantau perkembangan pasien ?

1.3. Batasan Permasalahan

Beberapa batasan masalah yang terdapat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Aplikasi yang dibuat merupakan aplikasi berbasis *desktop*.
2. Lingkungan pengembangan yang digunakan menggunakan aplikasi Unity 3D dengan bahasa pemrograman C#.
3. Aplikasi dibuat menggunakan Leap Motion SDK.
4. Tidak ada batasan pada setiap tahap.
5. Menggunakan organ gerak lengan tangan.
6. Digunakan pada pasien pascastroke yang tidak mengalami kelumpuhan tangan total dan dapat menggerakkan jari serta lengan bagian bawah.

1.4. Tujuan

Tujuan dari pembuatan Tugas Akhir ini yaitu membangun suatu perangkat lunak yang dapat menjadi salah satu media terapi gerak tangan dan jari tangan bagi pasien pascastroke secara virtual berbasis *game* simulasi agar terapi dapat berjalan lebih santai, diminati, dan dapat menambah motivasi pasien dalam melakukan terapi Tujuan lain dari pembuatan Tugas Akhir ini yaitu melakukan pengembangan Leap Motion khususnya sebagai alat bantu medis.

1.5. Manfaat

Manfaat yang diharapkan dari pengembangan aplikasi ini diantaranya:

1. Terciptanya sebuah aplikasi yang dapat menjalankan terapi pascastroke secara virtual berbasis *game* dengan memanfaatkan LMC.
2. Dapat menjadi referensi untuk penelitian lebih lanjut mengenai pengembangan Leap Motion dalam bidang medis.

1.6. Metodologi

Ada beberapa tahap dalam proses pengerjaan Tugas Akhir ini. Berikut ini adalah tahap-tahap dalam pembuatan tugas akhir :

1.6.1. Penyusunan proposal Tugas Akhir

Tahapan pertama dalam penyusunan Tugas Akhir ini adalah menyusun proposal Tugas Akhir. Pada proposal Tugas Akhir berisi mengenai rencana pengembangan aplikasi terapi pascastroke yaitu aplikasi untuk menjalankan terapi gerak jari tangan pascastroke dengan memanfaatkan LMC.

1.6.2. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan

pengumpulan informasi yang digunakan untuk mengimplementasikan Tugas Akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi, dan pustaka serta pemahaman dasar teori yang digunakan pada topik Tugas Akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

- a. Pemodelan Tiga Dimensi (3D)
- b. Leap Motion Controller (LMC)
- c. SDK Leap Motion Developer
- d. Unity 3D
- e. Blender

1.6.3. Analisis dan Perancangan Sistem

Pada tahap ini dilakukan analisis dan pendefinisian kebutuhan sistem untuk masalah yang sedang dihadapi. Selanjutnya, dilakukan perancangan sistem dengan beberapa tahap sebagai berikut:

- a. Perancangan diagram kasus penggunaan.
- b. Perancangan model 3D.
- c. Perancangan antarmuka sistem.
- d. Perancangan proses aplikasi.

1.6.4. Implementasi

Pada tahap ini dilakukan implementasi pemodelan objek-objek 3D dan implementasi proses yang telah didefinisikan pada bab Analisis dan Perancangan Sistem. Kemudian dilakukan integrasi aplikasi dengan perangkat Leap Motion. Aplikasi ini dibangun dengan Unity.

1.6.5. Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian aplikasi kepada pengguna secara langsung. Pengujian aplikasi dilakukan untuk mengevaluasi hasil program. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional. Tahapan-tahapan dari pengujian adalah sebagai berikut:

- a. Pengujian kebutuhan fungsional.
- b. Pengujian aplikasi dari pengguna.
- c. Kuesioner terhadap aplikasi dari pengguna.

1.6.6. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan, dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses, dan perancangan antarmuka pada aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi serta mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Bab ini menjelaskan tentang tinjauan pustaka yang menjadi dasar pembuatan Tugas Akhir. Beberapa teori, pustaka, dan teknologi yang mendasari pengerjaan Tugas Akhir ini diantaranya meliputi definisi stroke, terapi fisik anggota gerak tangan pascastroke, fisioterapi, Unity 3D, Leap Motion Controller, SDK Leap Motion, Blender, dan aplikasi terdahulu. Penjelasan secara khusus masing-masing tinjauan pustaka dapat dilihat pada masing-masing subbab berikut ini.

2.1. Rancang Bangun Perangkat Lunak

Rancang bangun perangkat lunak suatu ilmu yang mempelajari proses pembuatan aplikasi yang melingkupi analisis permasalahan dan kebutuhan, perencanaan, analisis sistem, implementasi, serta aktivitas pengujian, dan pemeliharaan perangkat lunak [3]. Rancang bangun perangkat lunak ini sendiri sangat penting dalam proses penentuan konsep-konsep yang akan di gunakan dalam proses pembuatan perangkat lunak, apabila diterapkan diharapkan perangkat lunak yang tercipta akan memiliki kualitas yang tinggi, manajemen waktu dan kerja yang tertata serta perangkat yang mudah dalam pemeliharannya. Model-model yang digunakan diantaranya adalah model *waterfall* dan model iterasi.

2.2. Stroke

Stroke adalah suatu kejadian rusaknya sebagian dari otak. Terjadi jika pembuluh darah arteri yang mengalirkan darah ke otak tersumbat atau jika robek atau bocor, stroke dibagi menjadi dua jenis yaitu stroke iskemik maupun stroke hemorragik. Sebuah prognosis hasil sebuah penelitian di Korea menyatakan bahwa, 75,2% stroke iskemik diderita oleh kaum pria dengan prevalensi berupa hipertensi, kebiasaan merokok dan konsumsi alkohol.

Berdasarkan sistem TOAST, komposisi terbagi menjadi 20,8% LAAS, 17,4% LAC, 18,1% CEI, 16,8% UDE dan 26,8% ODE [2]. Deteksi secepatnya dalam masa “*Golden Period*” beberapa jam setelah serangan stroke sangat berarti bagi kesehatan pasien pascastroke. Stroke iskemik, karena penyumbatan harus diberikan obat pengencer darah untuk melancarkan sumbatan dalam waktu tidak lebih dari 3 jam setelah serangan stroke, sedangkan stroke hemorragik dimana terjadi pendarahan harus segera dilakukan pembedahan untuk membersihkan darah dari otak. Jika terlambat penangannya, maka pasien akan menderita pascastroke yang lebih berat.

2.3. Terapi Fisik Anggota Gerak Tangan Pascastroke

Terapi fisik anggota gerak tangan pascastroke merupakan suatu upaya yang dapat ditempuh melalui berbagai cara seperti pelatihan, penggunaan modalitas, dan alat-alat. Makin dini terapi dimulai, maka dampaknya akan semakin baik. Manfaat yang bisa diperoleh antara lain mengoptimalkan pemulihan, menghindari kontraktur sendi, mencegah pengecilan otot, dan mencegah komplikasi akibat tirah baring terlalu lama [4]. Latihan fisik anggota gerak tangan pascastroke tidak boleh sembarangan dan melebihi kekuatan pasien. Beberapa hal yang perlu diperhatikan dalam melakukan terapi yaitu:

1. Menitikberatkan pada latihan kekuatan, koordinasi, keseimbangan, dan kestabilan otot.
2. Memulai latihan dengan pemanasan terlebih dahulu
3. Tidak memaksakan kemampuan
4. Pada tahap awal dapat menggunakan alat bantu gerak dan secara perlahan berlatih untuk melepas alat bantu tersebut.

Pada latihan terapi anggota gerak tangan terdapat beberapa pengelompokan gerak seperti *upper arm function*, *hand movement*, dan *advance hand activities* [5]. Semua pengelompokan tersebut disusun dan dikelompokkan untuk memperoleh kekuatan,

koordinasi, keseimbangan, dan kestabilan maka terdapat gerakan gerakan yang dapat dijadikan sebagai referensi untuk rancangan tahapan terapi, beberapa gerakan tersebut seperti:

1. Gerakkan kaki dari sisi kanan ke kiri dan sebaliknya [6]
2. Gerakkan kaki untuk mendapatkan objek yang dicari [6]
3. Gerakan mengangkat tangan [7] [5]
4. Gerakan menurunkan tangan [7] [5]
5. Gerakan menekuk dan meluruskan sendi bahu [7]
6. Gerakan menekuk dan meluruskan siku [7]
7. Gerakan menekuk dan meluruskan telapak tangan [7]
8. Gerakan memutar pergelangan tangan [8] [7]
9. Gerakan menggenggam dan melepas [8] [5]
10. Menggambar garis horizontal [5]

Pada aplikasi yang akan dibangun akan memiliki perangkat *eksternal* berupa Leap Motion Controller yang akan dijelaskan pada subbab 2.6. Untuk menunjang setiap pergerakan terapi sesuai dengan beberapa referensi maka penempatan posisi perangkat sangatlah penting. Posisi ideal perangkat pada saat melakukan terapi yaitu berada pada bawah telapak tangan dengan jarak kurang lebih 15 cm dan diletakkan pada sebuah meja, hal ini dilakukan bila pasien dapat menggerakkan lengan bagian atas, apabila pasien tidak dapat menggerakkan lengan bagian atas maka Leap Motion Controller akan diletakkan pada paha pasien dibawah telapak tangan dengan jarak kurang lebih 15 cm sehingga walaupun lengan bagian atas tidak dapat digerakkan akan tetapi lengan bagian bawah masih dapat di tangkap pergerakannya oleh Leap Motion yang terletak pada bagian paha pasien. Sedangkan posisi ideal tubuh pasien yaitu dengan posisi duduk sempurna, apabila pasien tidak dapat duduk dengan sempurna maka posisi Leap Motion Controller akan disesuaikan dengan posisi pasien dengan ketentuan Leap Motion Controller berada di bawah telapak tangan pasien dengan jarak kurang lebih 15 cm. Hal ini dilakukan

agar pasien dapat memiliki posisi yang ideal untuk melakukan setiap gerakan terapi.

2.4. Fisioterapi

Fisioterapi adalah bentuk pelayanan kesehatan professional yang ditujukan kepada individu dan kelompok untuk mengembangkan, memelihara dan memulihkan gerak serta fungsi tubuh sepanjang daur kehidupan dengan menggunakan penanganan secara manual, peralatan (fisik dan mekanik) dan komunikasi [9]. Fisioterapi juga berguna untuk mencegah komplikasi pada fungsi paru akibat tirah baring yang lama, menghambat *spastisitas*, pola sinergis ketika ada peningkatan tonus, mengurangi *oedem* pada anggota gerak atas dan bawah sisi sakit, merangsang timbulnya tonus ke arah normal, dan meningkatkan kemampuan aktifitas fungsional. [8].

Dimensi Pelayanan fisioterapi meliputi upaya peningkatan kesehatan, pencegahan penyakit, penyembuhan, dan pemulihan gangguan sistem gerak dan fungsi dalam rentang kehidupan dari praseminasi sampai ajal, yang terdiri dari upaya-upaya:

- a. Peningkatan dan pencegahan (promotif dan preventif), Pelayanan fisioterapi dapat dilakukan pada pusat kebugaran, pusat kesehatan kerja, sekolah, kantor, pusat panti usia lanjut, pusat olahraga, tempat kerja, industri dan pada pusat-pusat pelayanan umum.
- b. Penyembuhan dan pemulihan (Kuratif dan Rehabilitatif), pelayanan fisioterapi dapat dilakukan pada rumah sakit, rumah perawatan, panti asuhan, pusat rehabilitasi, tempat praktik, klinik privat, klinik rawat jalan, puskesmas, rumah tempat tinggal, pusat pendidikan, dan penelitian.

Pada upaya penyembuhan dan pemulihan dalam fisioterapi terdapat metode pengambilan data berupa rentang maksimal yang dapat dilakukan oleh pasien pada bagian tubuh tertentu, metode ini disebut *Range Of Motion* (ROM). Pengambilan data dilakukan

ketika suatu segmen tubuh bergerak dalam *Range Of Motion*, semua struktur pada bagian tubuh tersebut akan dipengaruhi antara lain otot, permukaan sendi, kapsul, ligamen, pembuluh darah dan saraf yang nantinya akan dicatat. Tabel yang menggambarkan ROM pada anggota gerak tangan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Rentang Gerak

Gerakan	Penjelasan	Rentang
Fleksi	Menggerakkan telapak tangan ke sisi bagian dalam lengan bawah	rentang 80-90°
Ekstensi	Mengerakan jari-jari tangan sehingga jari-jari, tangan, lengan bawah berada dalam arah yang sama	rentang 80-90°
Hiperekstensi	Membawa permukaan tangan dorsal ke belakang sejauh mungkin	rentang 89-90°
Abduksi 1	Menekuk pergelangan tangan miring ke ibu jari	rentang 30°
Abduksi 2	Menekuk pergelangan tangan miring ke arah lima jari	rentang 30-50°

2.5. Unity 3D

Unity3D adalah *software* yang baik yang merupakan *cross-platform* 3D dan *user-friendly*, mudah bagi pemula dan sangat berguna untuk para ahli. Aplikasi ini sangat membantu bagi

siapa saja yang akan membangun sebuah *game* baik itu *desktop*, *mobile*, *website* ataupun *console* [10]. Unity memiliki banyak sekali kegunaan seperti dapat merangkai objek-objek menjadi sebuah lingkungan yang dapat berinteraksi, menambahkan pencahayaan, suara, efek, dan animasi, akan tetapi Unity tidak dapat digunakan untuk mendesain objek atau *modelling*, jika memiliki kebutuhan serupa maka pengguna menggunakan *software* lain untuk melakukan desain seperti Blender dan 3dsmax.



Gambar 2.1 Unity3D

Terlepas dari itu Unity mendukung banyak sekali *extensi* dan format file lain untuk memperkaya objek dari *game* yang akan dibangun seperti file 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks dan Allegorithmic Zat. *Extensi* ini dapat diimport pada proyek Unity yang kemudian akan diproses oleh grafis Unity. Unity juga telah menyediakan aset sendiri secara gratis maupun berbayar pada Unity Store yang telah tersedia. Di dalam *asset store* terdapat banyak sekali objek, skybox, material, dll secara gratis maupun berbayar. *Scripting* pada Unity secara *default* dibangun dalam mesin *scripting* MonoDevelop. *Scripting* pada Unity dapat

dilakukan dengan 3 bahasa yang berbeda seperti C#, JavaScript, dan Boo [11].

2.6. Leap Motion Controller (LMC)

Leap Motion Controller merupakan suatu perangkat yang dikembangkan oleh Leap Motion, Inc pada tahun 2008 oleh David Holz, LMC sendiri digunakan sebagai input dari komputer tanpa sentuh, bisa dikatakan bahwa Leap Motion ini merupakan pengganti *mouse* karena mempunyai tujuan dan fungsi yang sama. Pada tahun 2010, untuk pertama kalinya LMC ini diperkenalkan kepada publik [12]. Perkembangan dari LMC dapat dilihat pada Gambar 2.2.



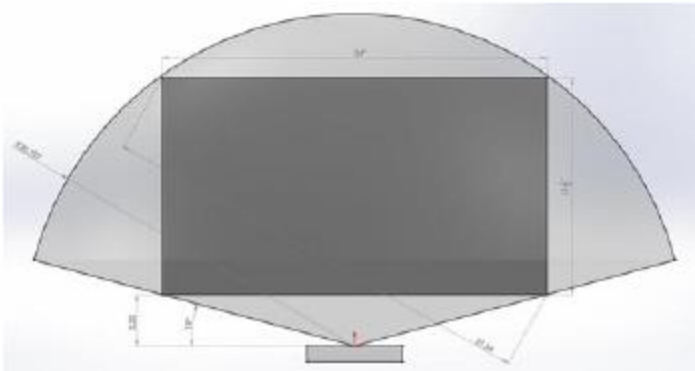
Gambar 2.2 Perkembangan Leap Motion Controller

Leap Motion seperti Gambar 2.3 sendiri merupakan penemuan penting dimana alat ini adalah alat sensor perangkat keras komputer yang mendukung gerakan tangan dan jari sebagai masukan, yang dapat disamakan fungsinya seperti *mouse*, namun tidak membutuhkan kontak langsung dengan tangan atau sentuhan sehingga diharapkan adanya sebuah pengalaman baru dalam dunia virtual

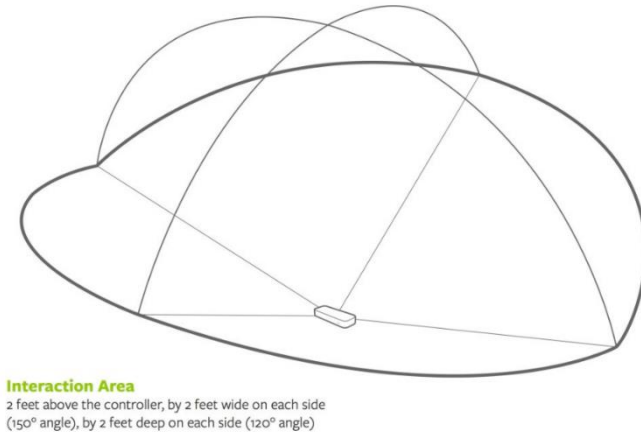


Gambar 2.3 Leap Motion Controller

Leap Motion sendiri memiliki *hardware* yang cukup simpel. Hal utama yang terdapat pada Leap Motion yaitu 2 buah kamera dan 3 buah inframerah LED (*Light-Emitting Diode*) [13]. Inframerah ini memiliki panjang gelombang 850. Jadi pada tahap ini inframerah akan menyebar untuk membentuk sebuah area seperti setengah lingkaran dengan jarak jangkauan maksimal 50 cm. Area jangkauan dari Leap Motion ini dapat dilihat pada Gambar 2.4 dan Gambar 2.5 [14].



Gambar 2.4 Area Jangkauan Inframerah dalam Dua Dimensi



Gambar 2.5 Area Jangkauan Inframerah dalam Tiga Dimensi

LMC dapat meningkatkan interaksi manusia dengan komputer. Dalam Tugas Akhir ini, Leap Motion dimanfaatkan untuk melacak gerakan tangan dan jari. Di saat sistem dijalankan, pengguna dapat melakukan berbagai aktivitas dengan gerakan tangan. Gerakan tangan ini diamati oleh sensor pada LMC sebagai input untuk melakukan berbagai hal yang dapat memicu terjadinya kejadian-kejadian dalam program.

2.7. SDK Leap Motion Developer

Sebagai perangkat keras yang inovatif yang terjun pada dunia teknologi maka dibutuhkan dukungan penuh dari pengembang aplikasi ke pengguna. Selain itu juga sebuah teknologi tidak akan berjalan bila tidak ada pengguna yang dapat menggunakannya walaupun teknologi itu sangat bagus.

Untuk itulah Leap Motion, Inc mengeluarkan Leap SDK yang dapat digunakan dengan mudah dan gratis. Leap SDK memberikan kemudahan-kemudahan dalam mengembangkan aplikasinya, seperti bahasa pemrograman yang disediakan memiliki banyak bahasa diantaranya C, C++, C#, Unity, Objective-

C, Java, Python, JavaScript. Juga mendukung beberapa system operasi seperti Windows, Linux, dan OSX [15] .

2.7.1. Pointables

Merupakan objek yang memiliki ujung yang terdeteksi oleh Leap Motion sehingga dapat digunakan sebagai *pointing*, *screentap*, *scaling*, *rotating*, dan sebagainya. *Pointables* sendiri memiliki dua representasi, yaitu *Finger* dan *Tools*. Adapun objek yang terdeteksi *Finger* adalah objek yang tidak lurus, mempunyai persendian seperti pada Gambar 2.6. Sedangkan *Tools* sendiri akan terdeteksi apabila objek tersebut memiliki panjang, lurus dan berdiameter dan berwarna cerah seperti pada Gambar 2.7 [16].



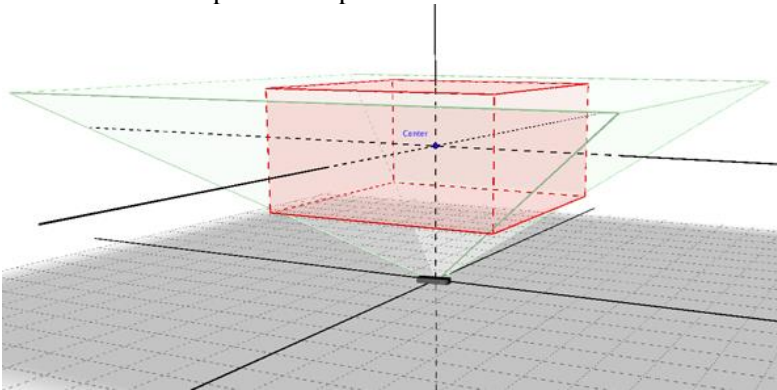
Gambar 2.6 Representasi Finger



Gambar 2.7 Representasi Tool

2.7.2. InteractionBox

InteractionBox merupakan sebuah representasi dari sebuah bidang dalam jarak pandang Leap Motion, dimana direpresentasikan seperti sebuah kubus. Dimana hal ini digunakan untuk mempermudah penentuan koordinat dari objek baik secara dua dimensi maupun tiga dimensi yang tertangkap dalam area Leap Motion. Pada bagian ini Leap Motion merepresentasikan tiap millimeter dengan koordinat tiga dimensi yaitu x, y, dan z sesuai kebutuhan masing-masing dengan normalisasi vektor [16]. *InteractionBox* dapat dilihat pada Gambar 2.8.

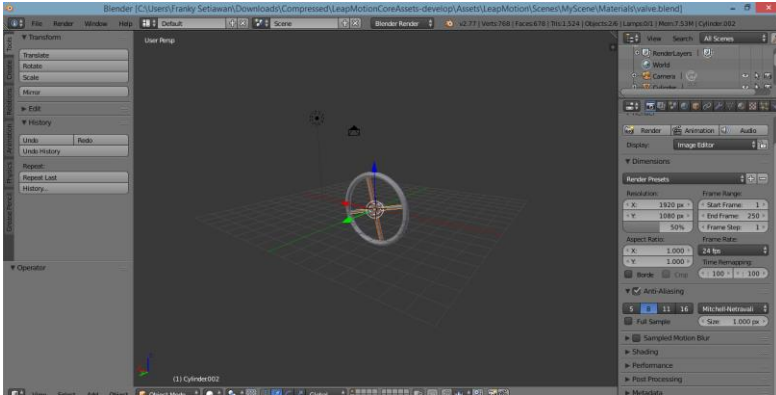


Gambar 2.8 Representasi Wilayah dari *InteractionBox*

2.8. Blender

Blender adalah sebuah perangkat lunak yang berfungsi untuk pemodelan 3 dimensi dan bersifat *open source*. Blender dapat digunakan pada berbagai jenis sistem operasi. Keluaran dari Blender adalah objek-objek 3 dimensi dengan berbagai format, seperti .obj, .3ds, dan lain-lain [17]. Objek keluaran dari Blender dapat digunakan sebagai material dasar pembuatan aplikasi atau permainan dengan menggunakan *game engine* atau perangkat lunak pembuat aplikasi 3 dimensi. Blender memiliki beberapa kegunaan termasuk pemodelan 3D, memberikan tekstur, penyunting gambar bitmap, penulangan, simulasi cairan, simulasi

partikel, animasi, penyunting video, pemahat digital dan *rendering*. Blender sendiri pun memiliki beberapa kelebihan seperti memiliki edit video, *game engine*, dan *Node Compositing* *Sculpting*. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.9.



Gambar 2.9 Tampilan Antarmuka Blender

2.9. Teknologi Informasi untuk Terapi Rehabilitasi Penderita Pascastroke

Teknologi informasi yang berkembang pesat memungkinkan dapat membantu pada bidang kedokteran, salah satu contoh yaitu terapi pascastroke. Kelebihan teknologi informasi dalam terapi yaitu dapat membuat terapi menjadi lebih menarik, menyenangkan, dan menambah motivasi penderita dalam melakukan terapi. Terapi pascastroke terdahulu yang telah menggunakan teknologi informasi sebagai media terapi yang dikemas dalam sebuah *game* yang menarik yaitu memanfaatkan teknologi seperti *kinect*, *augmented reality*, dan konsol *wii*.

2.9.1. *Kinect*

Kinect dapat menciptakan suatu lingkungan virtual yang menarik dan memberikan motivasi kepada pasien pascastroke

untuk melakukan terapi. Salah satu pelatihan dengan menggunakan teknologi *kinect* yang sudah dibuat adalah karya dari Ervina Handayani [18] yang memiliki kelebihan yaitu dapat menangkap keseluruhan tubuh penderita dengan baik dan akurat akan tetapi *kinect* tidak dapat membaca secara detail setiap gerakan jari tangan dari pasien karena ditujukan untuk menangkap pasien dalam gerak seperti duduk, berdiri, menggerakkan tangan serta kaki secara keseluruhan.

2.9.2. Augmented Reality

Augmented Reality (AR) merupakan teknologi yang memanfaatkan objek 3D yang menyatu dengan lingkungan sekitar di dunia nyata dan dapat saling memberikan aksi dan reaksi. Salah satu teknologi terapi pascastroke yang menggunakan teknologi AR adalah karya Nurina Aisyah Fitriani [19] yang memiliki kekurangan yaitu posisi gerak tangan penderita yang tidak stabil serta objek yang terkadang hilang bila marker tidak dalam posisi sempurna, sehingga menyulitkan AR untuk menangkap gerakan tangan pengguna. Hal ini mengakibatkan hasil uji coba yang dilakukan tidak sepenuhnya akurat berdasarkan kemampuan dari pengguna.

2.9.3. Konsol Wii

Konsol *wii* dapat digunakan sebagai salah satu media terapi pascastroke. Aplikasi yang telah menerapkan konsol *wii* sebagai media terapi yaitu *Wii-Based Movement Therapy* yang dikembangkan oleh *NeuRa (Neuroscience Research Australia)* [20]. Aplikasi ini sangat bagus dalam salah satu penyedia media terapi namun memiliki kelemahan mendasar yaitu pasien harus harus menggenggam alat dengan erat yang memiliki dimensi cukup besar yaitu panjang 148 mm, lebar 36.2 mm, dan tebal 30.8 mm.

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis dan perancangan aplikasi terapi gerak lengan tangan dan pergelangan tangan pascastroke dengan memanfaatkan teknologi LMC. Pembahasan yang akan dilakukan meliputi analisis kegunaan yang dibutuhkan dan perancangan perangkat lunak.

3.1. Analisis Perangkat Lunak

Subbab ini menjelaskan tentang hasil analisis kebutuhan perangkat lunak serta arsitektur aplikasi terapi gerak lengan tangan dan pergelangan tangan pascastroke dengan memanfaatkan LMC. Tiap-tiap subbab menjelaskan tentang deskripsi umum perangkat lunak, spesifikasi kebutuhan perangkat lunak, analisis aktor, arsitektur perangkat lunak, dan skenario kasus penggunaan.

3.1.1. Deskripsi Umum Perangkat Lunak

Pada Tugas Akhir ini dilakukan pengembangan aplikasi terapi gerak lengan tangan dan pergelangan tangan pascastroke dengan memanfaatkan teknologi LMC. Aplikasi ini bertujuan bagi pengguna pascastroke yang tidak mengalami kelumpuhan tangan total tetapi pengguna yang masih memiliki terbatasan gerak serta memiliki kekuatan dalam menggerakkan tangan yang masih perlu dilatih secara periodik untuk dapat kembali normal. Dengan memanfaatkan fitur yang dimiliki LMC yaitu *tools and finger detection*, maka pengguna dapat menggerakkan tangan ke atas, bawah, menggeser, mengambil benda serta menggenggam benda yang kemudian di gambarkan secara virtual oleh deteksi tangan dari LMC. Terdapat beberapa hal yang diperlukan agar pengguna dapat menggunakan aplikasi ini. Pertama pengguna harus memastikan bahwa komputer yang dipakai sesuai dengan spesifikasi yang dibutuhkan. Perangkat lain yang harus disediakan yaitu LMC dan seorang admin sebagai pembimbing dengan

keadaan sehat. Dengan mekanisme yang telah diuraikan, aplikasi ini dapat difungsikan sebagai aplikasi untuk terapi gerak tangan pengguna pascastroke dengan memanfaatkan LMC.

3.1.2. Spesifikasi Kebutuhan Fungsional Perangkat Lunak

Kebutuhan fungsional berisi proses-proses yang harus dimiliki sistem. Kebutuhan fungsional mendefinisikan layanan yang harus disediakan dan reaksi sistem terhadap masukan pengguna. Daftar kebutuhan fungsional dapat dilihat pada Tabel 3.1.

Tabel 3.1 Daftar Spesifikasi Kebutuhan Fungsional Perangkat Lunak

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-0001	Melihat penjelasan aplikasi	Pengguna dapat melihat penjelasan aplikasi tahap satu, dua, tiga dan empat
F-0002	Melakukan pelatihan	Pengguna dapat melakukan latihan terapi stroke dimulai dari tahap pertama sampai terakhir sesuai instruksi sistem
F-0003	Melihat hasil latihan	Pengguna dapat melihat hasil latihan yang dapat dilihat pada folder aplikasi dalam bentuk teks yang memungkinkan untuk menjadi bahan acuan perkembangan otot pengguna

3.2. Perancangan Perangkat Lunak

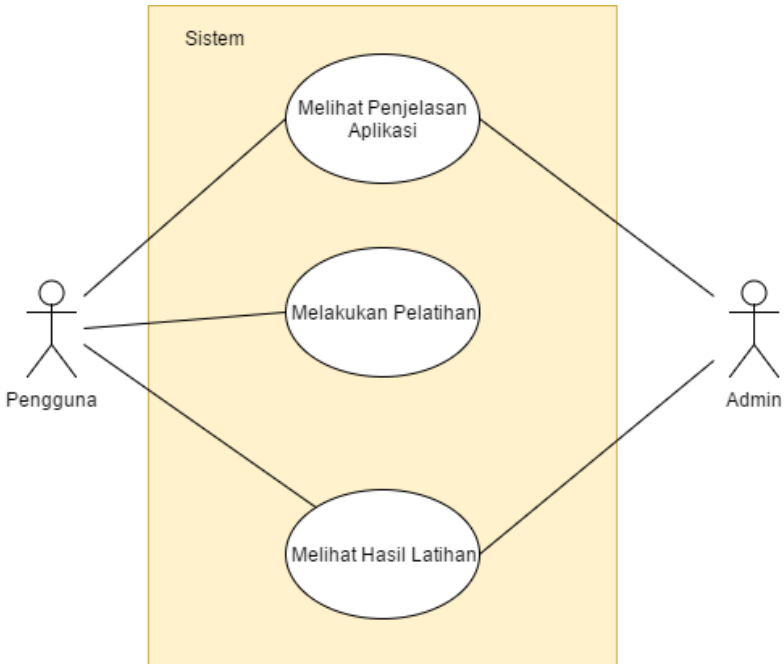
Subbab ini membahas bagaimana rancangan dari aplikasi Tugas Akhir ini. Perancangan perangkat lunak akan membahas model kasus penggunaan, definisi aktor, definisi kasus penggunaan, arsitektur umum sistem, rancangan antarmuka aplikasi, dan rancangan proses aplikasi.

3.2.1. Model Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem dibuat kasus penggunaan sistem. Kasus-kasus penggunaan dalam sistem ini akan dijelaskan secara rinci pada subbab ini. Kasus penggunaan digambarkan dalam sebuah diagram kasus penggunaan. Diagram kasus penggunaan dapat dilihat pada Tabel 3.2 dan Gambar 3.1.

Tabel 3.2 Daftar Kasus Penggunaan

Kode Kasus Penggunaan	Nama	Aktor
UC-0001	Melihat penjelasan aplikasi	Admin dan Pengguna
UC-0002	Melakukan pelatihan	Pengguna
UC-0003	Melihat hasil latihan	Admin dan Pengguna



Gambar 3.1 Diagram Kasus Penggunaan

3.2.2. Definisi Kasus Penggunaan

Definisi Kasus Penggunaan merupakan penjelasan setiap kasus penggunaan yang terdapat pada sistem. Detail mengenai kasus penggunaan tersebut dapat dilihat pada subbab berikut ini.

3.2.2.1. Melihat Penjelasan Aplikasi

Spesifikasi kasus penggunaan melihat penjelasan aplikasi dapat dilihat pada Tabel 3.3.

Tabel 3.3 Spesifikasi Kasus Penggunaan Melihat Penjelasan Aplikasi

Nama	Melihat Penjelasan Aplikasi
Kode	UC-0001
Deskripsi	Pengguna dapat melihat penjelasan aplikasi tahap satu, dua, tiga dan empat
Aktor	Pengguna dan Admin
Kondisi Awal	Sistem berada pada menu awal pada <i>homescreen</i> . Admin sudah mengintegrasikan leap motion dengan aplikasi.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Admin menekan tombol “Pelajari Selengkapnya”. 2. Aplikasi akan menampilkan instruksi dan penjelasan secara singkat pada setiap tahap. 3. Admin dan pengguna melihat dan mempelajari aplikasi. 4. Admin menekan tombol menu untuk kembali ke halaman awal dan menghentikan tampilan instruksi.
- Kejadian Alternatif	Tidak ada
Kondisi Akhir	Sistem akan mengeluarkan instruksi dan penjelasan aplikasi terapi gerak pada setiap tahap.

3.2.2.2. Melakukan Pelatihan

Spesifikasi kasus penggunaan melakukan pelatihan menggunakan LMC dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kasus Penggunaan Melakukan Pelatihan

Nama	Melakukan Pelatihan
Kode	UC-0002
Deskripsi	Pengguna dapat melakukan latihan terapi stroke dimulai dari tahap pertama sampai terakhir sesuai instruksi sistem.
Aktor	Pengguna
Kondisi Awal	Sistem berada pada menu awal pada <i>homescreen</i> .
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Admin menekan tombol “mulai” 2. Pengguna dan admin melihat instruksi untuk tahap 1 pada sisi kiri layar 3. Admin menekan tombol “mulai” 4. Pengguna memulai terapi tahap 1 level 1 hingga selesai 5. Pengguna memulai terapi tahap 1 level 2 hingga selesai 6. Pengguna memulai terapi tahap 1 level 3 hingga selesai 7. Pengguna memulai terapi tahap 1 level 4 hingga selesai 8. Pengguna memulai terapi tahap 1 level 5 hingga selesai 9. Admin menekan tombol “Selanjutnya” 10. Pengguna dan admin melihat instruksi tahap 2 pada sisi kiri layar 11. Admin menekan tombol “mulai” 12. Pengguna memulai terapi tahap 2 hingga selesai 13. Admin menekan tombol “Selanjutnya” 14. Pengguna dan admin melihat instruksi tahap 3 pada sisi kiri layar 15. Admin menekan tombol “mulai” 16. Pengguna memulai terapi tahap 3 hingga selesai

	17. Admin menekan tombol “Selanjutnya” 18. Pengguna melihat instruksi tahap 4 pada sisi kiri layar 19. Admin menekan tombol “mulai” 20. Pengguna memulai terapi tahap 4 level 1 hingga selesai 21. Admin menekan tombol “Selanjutnya” 22. Pengguna memulai terapi tahap 4 level 2 hingga selesai 23. Admin menekan tombol “Selanjutnya” 24. Pengguna memulai terapi tahap 4 level 3
- Kejadian Alternatif	Tidak Ada
Kondisi Akhir	Sistem telah memberikan keseluruhan tahap terapi dan menampilkan nilai pengguna di tiap tahap terapi

3.2.2.3. Melihat Hasil Latihan

Spesifikasi kasus penggunaan melihat hasil latihan dapat dilihat pada Tabel 3.5.

Tabel 3.5 Spesifikasi Kasus Penggunaan Melihat Hasil Latihan

Nama	Melihat Hasil Latihan
Kode	UC-0003
Deskripsi	Pengguna dapat melihat hasil latihan pada folder aplikasi dalam bentuk teks
Aktor	Pengguna dan Admin
Kondisi Awal	Pengguna berada pada tahap 4 level 3
Aliran: - Kejadian Normal	1. Pengguna menyelesaikan tahap 4 level 3 2. Admin menekan tombol “menu”
- Kejadian Alternatif	Tidak Ada

Kondisi Akhir	Sistem akan kembali pada halaman <i>homescreen</i> dan mengeluarkan file berupa teks yang berisi hasil terapi pada folder aplikasi
----------------------	--

3.2.3. Definisi Aktor

Aktor yang terlibat dalam aplikasi terapi gerak dengan memanfaatkan LMC ada dua yaitu pengguna sebagai pasien dan admin sebagai pendamping yang berada dalam kesehatan yang normal. Deskripsi pengguna secara lengkap dapat dilihat pada Tabel 3.6.

Tabel 3.6 Deskripsi Pengguna

No	Nama	Deskripsi
1	Pengguna	- Merupakan aktor sebagai pasien yang bertugas untuk menjalankan aplikasi sebagai cara untuk menjalankan terapi
2	Admin	- Merupakan aktor yang membantu pengguna dalam menjalankan aplikasi, tugas admin adalah memberikan inputan <i>mouse</i> pada aplikasi untuk memulai tahapan terapi dan memberikan saran sesuai nilai yang dikeluarkan aplikasi pada setiap akhir tahapan

3.2.4. Perancangan Model 3D

Terdapat beberapa perancangan model yang diambil dari *standard asset* Unity yang dapat diakses pada <https://www.assetstore.unity3d.com/> dan selanjutnya dimodifikasi menggunakan Blender versi 2.73. Aset-aset yang didapat dari *standard asset* Unity dapat dilihat pada Tabel 3.7 serta terdapat

beberapa objek 3D meliputi tombol on, tuas, tuas geser, tuas putar, bintang, dan kotak penampung asteroid. Detil rancangan model ditampilkan pada subbab berikutnya.

Tabel 3.7 Daftar Objek Dari Unity Asset Store

No	Nama Aset	Deskripsi	Sumber
1	Cockpit_Frame_C	Kokpit pesawat luar angkasa yang digunakan sebagai virtualisasi dan penggambaran ruang kokpit yang sesungguhnya.	Unity Asset Store
2	DemoAsteroids	Demo Asteroid adalah benda-benda luar angkasa berupa batuan yang digunakan sebagai virtualisasi asteroid.	Unity Asset Store
3	Demo_Planet	Demo Planet adalah sebuah penggambaran planet di luar angkasa yang digunakan sebagai penambah suasana menjadi lebih realistis.	Unity Asset Store
4	Demo_SkySphere	Demo Skysphere adalah suatu penggambaran lingkungan luar angkasa yang tak terbatas.	Unity Asset Store

3.2.4.1. Perancangan Model 3D Tombol On

Rancangan model 3D tombol on yang akan ditampilkan merupakan tombol on dengan jenis membulat. Perancangan model tombol on tampak pada Gambar 3.2. Perancangan model 3D tombol on pada aplikasi ini nantinya terdiri dari 2 bagian penting diantaranya:

1. Tombol
2. Lingkaran luar tombol sebagai tempat tombol berpijak



Gambar 3.2 Perancangan Model 3D Tombol On

3.2.4.2. Perancangan Model 3D Tuas

Pada subbab ini digambarkan rancangan model 3D Tuas. Gambar 3.3 merupakan rancangan model 3D Tuas yang akan ditampilkan pada sistem. Tuas ini memiliki bentuk yang umum digunakan pada pesawat terbang atau pesawat luar angkasa. Perancangan model 3D tuas pada aplikasi ini nantinya terdiri dari 2 bagian penting diantaranya:

1. Tuas
2. Tempat tuas berputar pada porosnya



Gambar 3.3 Perancangan Model 3D Tuas 3D

3.2.4.3. Perancangan Model 3D Tuas Geser

Rancangan model 3D tuas geser yang akan ditampilkan merupakan tuas geser / *switch* yang umum terdapat pada serangkaian alat kendali pesawat luar angkasa, peralatan pada kendali mobil, atau alat kendali musik yang berguna untuk memutar lagu atau menggabungkan suatu lagu yang satu dengan lagu yang lainnya. Perancangan model tuas geser tampak pada Gambar 3.4. Perancangan model 3D tuas geser pada aplikasi ini nantinya terdiri dari 2 bagian penting diantaranya:

1. Tuas
2. Tempat tuas bergeser



Gambar 3.4 Perancangan Model 3D Tuas Geser 3D

3.2.4.4. Perancangan Model 3D Tuas Putar

Rancangan model 3D tuas putar yang akan ditampilkan merupakan tuas putar yang umum terdapat pada serangkaian alat kendali pesawat luar angkasa yang pada umumnya terdapat pada pintu kabin, pintu pesawat, *valve*, *trigger* untuk menyalakan mesin atau tuas putar yang seperti pada kendali mobil pada umumnya. Perancangan model 3D tuas putar tampak pada Gambar 3.5 dan memiliki dua bagian penting diantaranya:

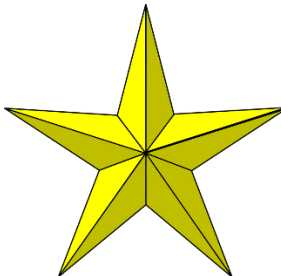
1. Lingkaran putar
2. Silinder yang berputar pada porosnya



Gambar 3.5 Perancangan Model 3D Tuas Putar 3D

3.2.4.5. Perancangan Model 3D Bintang

Rancangan model 3D bintang yang akan ditampilkan yaitu model bintang pada umumnya yang memiliki lima sudut dan berwarna kuning. Perancangan model 3D bintang tampak pada Gambar 3.6.



Gambar 3.6 Perancangan Model 3D Bintang

3.2.4.6. Perancangan Model 3D Kotak Sebagai Penampung Asteroid

Rancangan model kotak 3D penampung asteroid yang akan ditampilkan kotak sederhana yang umum digunakan untuk mengamankan sesuatu yang berbahaya. Perancangan model kotak penampung asteroid tampak pada Gambar 3.7. Perancangan model 3D tuas geser pada aplikasi ini nantinya terdiri dari 2 bagian penting diantaranya:

1. Tutup atas kotak
2. Bidang kotak bagian samping

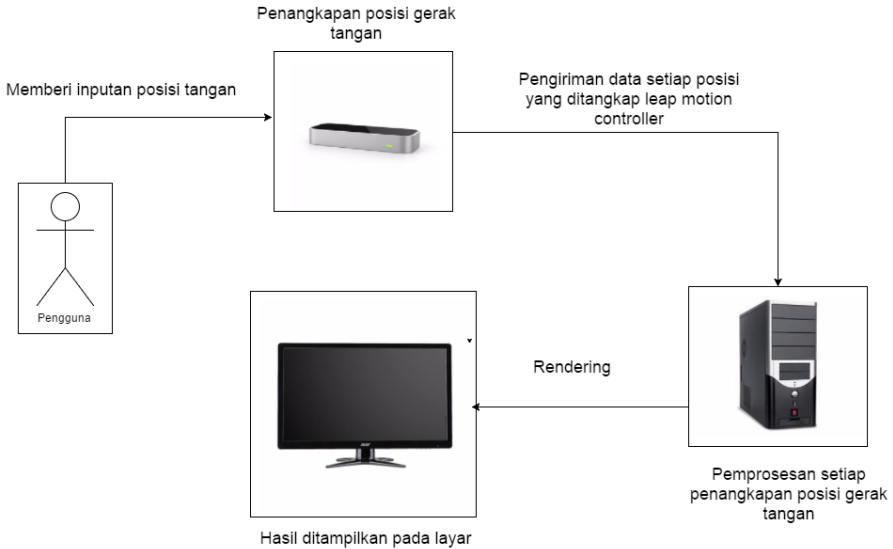


Gambar 3.7 Perancangan Model 3D Box Penampung Asteroid

3.2.5. Arsitektur Umum Sistem

Arsitektur sistem pada aplikasi terapi gerak tangan dengan memanfaatkan LMC ini didukung oleh beberapa perangkat yaitu komputer dan LMC. Implementasi aplikasi juga memanfaatkan salah satu aplikasi *game engine* yang sudah terkenal

keunggulannya yaitu Unity. Untuk pembuatan model 3 dimensi menggunakan aplikasi *3d modeling* Blender. Arsitektur secara umum aplikasi ini terlihat pada Gambar 3.8.



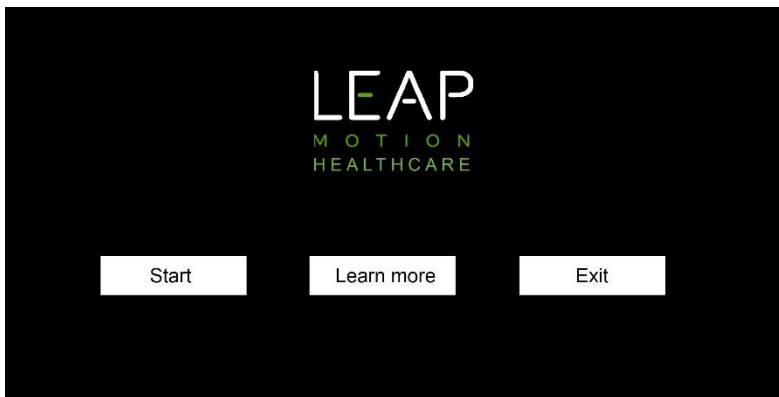
Gambar 3.8 Rancangan Sederhana Arsitektur Aplikasi

3.2.6. Rancangan Antarmuka Aplikasi

Rancangan antarmuka aplikasi diperlukan untuk memberikan gambaran umum kepada pengguna bagaimana sistem yang ada dalam aplikasi ini berinteraksi dengan pengguna. Pada subbab ini akan dijelaskan rancangan aplikasi untuk menggambarkan rancangan antarmuka, rancangan fungsionalitas berupa deteksi *trigger*, dan spesifikasi atribut didalamnya.

3.2.6.1. Rancangan Antarmuka Menu Utama

Halaman ini merupakan tampilan awal saat aplikasi pertama dijalankan. Pada tampilan Gambar 3.9 Rancangan Antarmuka Menu Utama terdapat tiga menu yang dapat dipilih oleh pengguna. Spesifikasi atribut antarmuka untuk halaman menu utama dapat dilihat pada Tabel 3.8.



Gambar 3.9 Rancangan Antarmuka Menu Utama

Tabel 3.8 Spesifikasi Atribut Antarmuka Menu Utama

No	Nama	Jenis	Kegunaan
1	BStart	Button	Memulai menjalankan terapi
2	BLearnMore	Button	Menampilkan antarmuka pengenalan tiap tahap
3	BExit	Button	Keluar dari aplikasi

3.2.6.2. Rancangan Antarmuka Pelajari Selanjutnya

Halaman ini merupakan tampilan “pelajari selanjutnya” saat. Pada tampilan Gambar 3.10 Rancangan Antarmuka Tutorial Terapi terdapat empat panel yang menjelaskan secara garis besar tahapan terapi serta terdapat satu tombol “kembali ” yang berguna untuk kembali ke menu awal. Spesifikasi atribut antarmuka untuk halaman “pelajari selanjutnya” dapat dilihat pada Tabel 3.9.



Gambar 3.10 Rancangan Antarmuka Tutorial Terapi

Tabel 3.9 Spesifikasi Atribut Antarmuka Tutorial Terapi

No	Nama	Jenis	Kegunaan
1	BBack	Button	Navigasi untuk kembali ke menu utama

3.2.6.3. Rancangan Antarmuka Menu Tahap 1

Halaman ini merupakan tampilan menu tahap pertama terapi gerak lengan tangan. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 1 dapat dilihat pada Tabel 3.10 dan Gambar 3.11 merupakan gambar antarmuka aplikasi.



Gambar 3.11 Rancangan Antarmuka Menu Tahap 1

Tabel 3.10 Spesifikasi Atribut Antarmuka Menu Tahap 1

No	Nama	Jenis	Kegunaan
1	BStart	Button	Menjalankan aplikasi tahap 1

3.2.6.4. Rancangan Antarmuka Menu Tahap 2

Halaman ini merupakan tampilan menu tahap kedua terapi gerak lengan tangan. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 2 dapat dilihat pada Tabel 3.11 dan Gambar 3.12 yang merupakan gambar antarmuka aplikasi.



Gambar 3.12 Rancangan Antarmuka Menu Tahap 2

Tabel 3.11 Spesifikasi Atribut Antarmuka Menu Tahap 2

No	Nama	Jenis	Kegunaan
1	BStart	Button	Menjalankan aplikasi tahap 2

3.2.6.5. Rancangan Antarmuka Menu Tahap 3

Halaman ini merupakan tampilan menu ketiga terapi gerak lengan tangan. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 3 dapat dilihat pada Tabel 3.12 dan Gambar 3.13 merupakan gambar antarmuka aplikasi.

**Gambar 3.13 Rancangan Antarmuka Menu Tahap 3****Tabel 3.12 Spesifikasi Atribut Antarmuka Menu Tahap 3**

No	Nama	Jenis	Kegunaan
1	BStart	Button	Menjalankan aplikasi tahap 3

3.2.6.6. Rancangan Antarmuka Menu Tahap 4

Halaman ini merupakan tampilan menu tahap keempat terapi gerak lengan tangan. Spesifikasi atribut antarmuka untuk

halaman antarmuka tahap 4 dapat dilihat pada Tabel 3.13 dan Gambar 3.14 merupakan gambar antarmuka aplikasi.



Gambar 3.14 Rancangan Antarmuka Menu Tahap 4

Tabel 3.13 Spesifikasi Atribut Antarmuka Menu Tahap 4

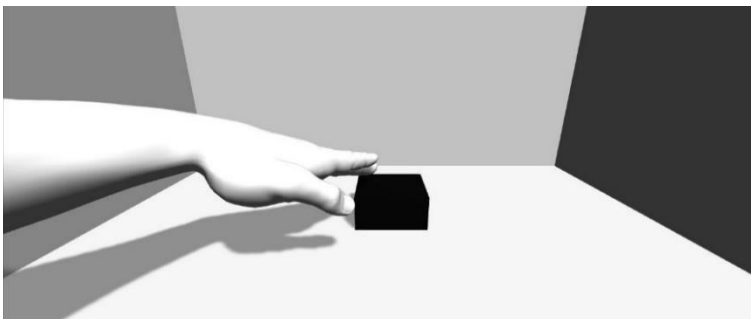
No	Nama	Jenis	Kegunaan
1	BStart	Button	Menjalankan aplikasi tahap 4

3.2.6.7. Rancangan Antarmuka Tahap 1

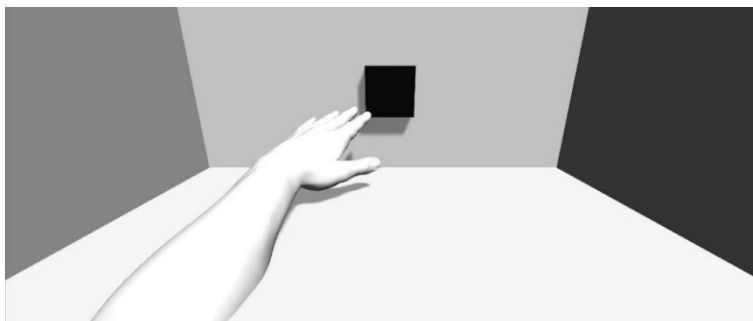
Halaman ini merupakan tampilan tahap pertama terapi gerak lengan tangan. Berisi animasi dan gambar lingkungan yang mendukung terapi, mulai desain hingga suara. Tahap 1 memiliki 5 level yang harus diselesaikan. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 1 dapat dilihat pada Tabel 3.14 serta gambaran aplikasi tahap 1 level 1 sampai 5 dapat dilihat pada Gambar 3.15, Gambar 3.16, Gambar 3.17, Gambar 3.18, dan Gambar 3.19.



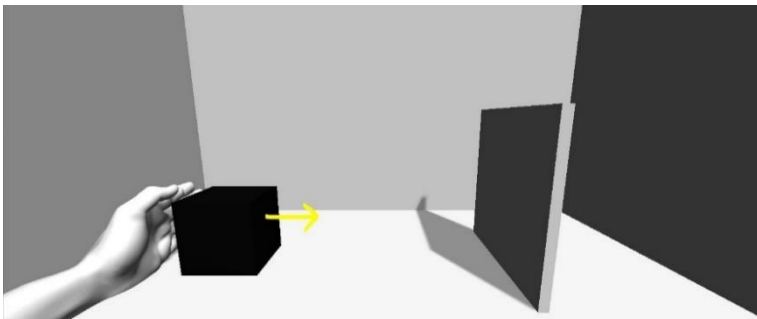
Gambar 3.15 Rancangan Antarmuka Tahap 1 Level 1



Gambar 3.16 Rancangan Antarmuka Tahap 1 Level 2



Gambar 3.17 Rancangan Antarmuka Tahap 1 Level 3



Gambar 3.18 Rancangan Antarmuka Tahap 1 Level 4



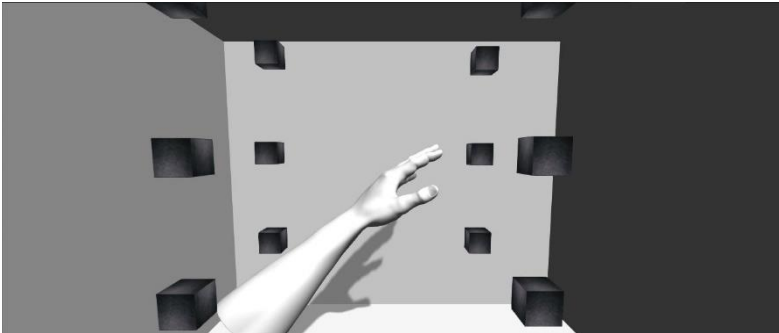
Gambar 3.19 Rancangan Antarmuka Tahap 1 Level 5

Tabel 3.14 Spesifikasi Atribut Antarmuka Tahap 1

No	Nama	Jenis	Kegunaan
1	BNext	Button	Menjalankan fungsi menuju ke menu tahap 2 dan melakukan <i>parsing</i> hasil latihan tahap 1

3.2.6.8. Rancangan Antarmuka Tahap 2

Halaman ini merupakan tampilan tahap kedua terapi gerak lengan tangan. Berisi animasi dan gambar lingkungan yang mendukung terapi, mulai desain hingga suara. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 2 dapat dilihat pada Tabel 3.15 dan Gambar 3.20 merupakan antar muka aplikasi tahap 2



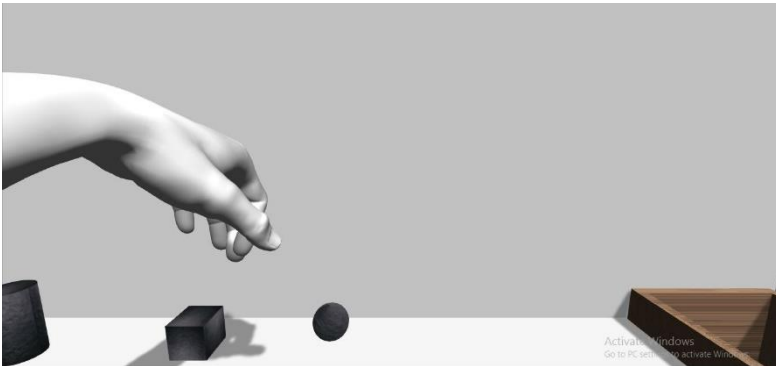
Gambar 3.20 Rancangan Antarmuka Tahap 2

Tabel 3.15 Spesifikasi Atribut Antarmuka Tahap 2

No	Nama	Jenis	Kegunaan
1	BNext	Button	Menjalankan fungsi menuju ke menu tahap 3 dan melakukan <i>parsing</i> hasil latihan tahap 2

3.2.6.9. Rancangan Antarmuka Tahap 3

Halaman ini merupakan tampilan tahap ketiga terapi gerak lengan tangan. Berisi animasi dan gambar lingkungan yang mendukung terapi, mulai desain hingga suara. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 3 dapat dilihat pada Tabel 3.16 dan Gambar 3.21 merupakan antar muka aplikasi tahap 3.



Gambar 3.21 Rancangan Antarmuka Tahap 3

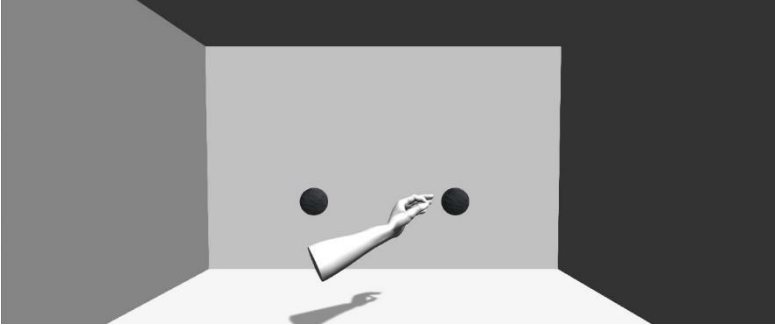
Tabel 3.16 Spesifikasi Atribut Antarmuka Tahap 3

No	Nama	Jenis	Kegunaan
1	BNext	Button	Menjalankan fungsi menuju ke menu tahap 4 dan melakukan <i>parsing</i> hasil latihan tahap 3

3.2.6.10. Rancangan Antarmuka Tahap 4 Level 1

Halaman ini merupakan tampilan tahap keempat pada level pertama terapi gerak lengan tangan. Berisi animasi dan gambar

lingkungan yang mendukung terapi, mulai desain hingga suara. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 4 level 1 dapat dilihat pada Tabel 3.17 dan Gambar 3.22 merupakan antar muka aplikasi tahap 4 level 1.



Gambar 3.22 Rancangan Antarmuka Tahap 4 Level 1

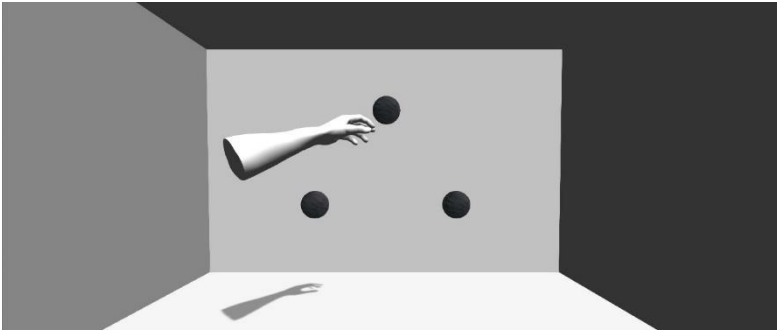
Tabel 3.17 Spesifikasi Atribut Antarmuka Tahap 4 Level 1

No	Nama	Jenis	Kegunaan
1	BNext	Button	Menjalankan fungsi menuju ke menu tahap 4 level 2 dan melakukan <i>parsing</i> hasil latihan tahap 4 level 1

3.2.6.11. Rancangan Antarmuka Tahap 4 Level 2

Halaman ini merupakan tampilan tahap keempat pada level kedua terapi gerak lengan tangan. Berisi animasi dan gambar lingkungan yang mendukung terapi, mulai desain hingga suara. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 4

level 2 dapat dilihat pada Tabel 3.18 dan Gambar 3.23 merupakan antar muka aplikasi tahap 4 level 2.



Gambar 3.23 Rancangan Antarmuka Tahap 4 Level 2

Tabel 3.18 Spesifikasi Atribut Antarmuka Tahap 4 Level 2

No	Nama	Jenis	Kegunaan
1	BNext	Button	Menjalankan fungsi menuju ke menu tahap 4 level 3 dan melakukan <i>parsing</i> hasil latihan tahap 4 level 2

3.2.6.12. Rancangan Antarmuka Tahap 4 Level 3

Halaman ini merupakan tampilan tahap keempat pada level ketiga terapi gerak lengan tangan. Berisi animasi dan gambar lingkungan yang mendukung terapi, mulai desain hingga suara. Spesifikasi atribut antarmuka untuk halaman antarmuka tahap 4 level 3 dapat dilihat pada Tabel 3.19 dan Gambar 3.24 merupakan antar muka aplikasi tahap 4 level 3.



Gambar 3.24 Rancangan Antarmuka Tahap 4 Level 3

Tabel 3.19 Spesifikasi Atribut Antarmuka Tahap 4 Level 3

No	Nama	Jenis	Kegunaan
1	BNext	Button	Menjalankan fungsi menuju ke menu utama atau <i>homescreen</i>

3.2.7. Perancangan Proses Aplikasi

Pada subbab ini akan dijelaskan mengenai rancangan proses yang digunakan untuk pencapaian fungsionalitas yang sudah dirancang pada aplikasi. *Flowchart* rancangan proses aplikasi dapat dilihat pada Gambar A.0.1.

3.2.7.1. Rancangan Proses Penentuan Level Pelatihan

Rancangan proses penentuan level pelatihan bertujuan untuk mendapatkan fokus latihan organ gerak tangan secara tepat, dan berurutan. Bentuk gerakan pada setiap tahap pelatihan terinspirasi dari gerakan terapi stroke di dalam dunia kedokteran berdasarkan pada Subbab 2.3, untuk tingkat kesulitan disusun berdasarkan dari kompleksitas gerakan yang dihasilkan, macam tahapan dalam aplikasi ini yaitu:

Tahap 1:

1. Gerakan mengangkat tangan
2. Gerakan menurunkan tangan
3. Gerakan menggerakkan tangan ke depan
4. Gerakan menggeser tangan
5. Memutar pergelangan tangan

Tahap 2:

1. Kombinasi semua gerakan tahap pertama

Tahap 3:

1. Gerakan mengambil benda

Tahap 4:

1. Gerakan kestabilan dengan menggambar garis lurus
2. Gerakan kestabilan dengan menggambar segitiga
3. Gerakan kestabilan dengan menggambar persegi

Setiap tahapan yang telah dirancang dan disusun dengan tujuan dan fokus tersendiri. Fokus gerakan untuk latihan gerak seluruh tangan atau lengan tangan pada tahap awal yaitu tahap 1 dan 2 memberikan latihan untuk otot dapat terlatih bekerja sama secara sinkron untuk melakukan suatu gerakan tangan. Rancangan penentuan tahap 3 dan 4 bertujuan untuk mendapatkan hasil berupa penguatan dan peningkatan akurasi dari otot-otot pergelangan tangan setelah otot lengan tangan telah dilatih pada tahapan sebelumnya. Rancangan ini diharapkan dapat meningkatkan fungsionalitas otot lengan dan pergelangan tangan pengguna secara bertahap. Untuk proses terapi pasien diharuskan melakukan tahapan terapi dari tahap 1 sampai 4 secara berurutan, hal ini bertujuan agar pasien dapat melakukan terapi secara bertahap dan runtut karena pada setiap tahap memiliki fokus tersendiri dan apabila pasien mengalami kendala pada suatu tahap maka admin sebagai pembimbing dapat membantu pasien dengan gerakan-gerakan pendukung agar pasien dapat tetap menggerakkan tangan walaupun secara pasif.

3.2.7.2. Rancangan Proses Sistem Tiap Tahap

Rancangan proses sistem tiap tahap bertujuan untuk merancang sistem hingga mendapatkan fungsionalitas yang benar dan sesuai kebutuhan pengguna. Rancangan proses sistem didasarkan pada rancangan penentuan level pelatihan pada subbab 3.2.7.1.

3.2.7.2.1. Rancangan Proses Sistem Tahap 1

Rancangan proses sistem tahap 1 bertujuan untuk memberikan fungsionalitas yang sesuai. Tahap 1 merupakan tahapan paling awal dalam proses terapi yang berisi fungsi-fungsi sederhana berupa *trigger* dan masukan sudut rotasi yang diberikan dari gerak tangan oleh pengguna.

3.2.7.2.2. Rancangan Proses Sistem Tahap 2

Rancangan proses sistem tahap 2 bertujuan untuk memberikan fungsionalitas yang sesuai. Rancangan proses tahap 2 menggunakan fungsi *Random.Range()* sebagai cara untuk melakukan *random* lokasi dari objek bintang. Pengacakan lokasi bertujuan agar pengguna dapat seolah menggerakkan tangan ke arah atas, bawah, kanan, dan kiri sehingga pengguna dapat menggabungkan beberapa gerakan tangan dari tahap pertama terapi.

3.2.7.2.3. Rancangan Proses Sistem Tahap 3

Rancangan proses sistem tahap 3 bertujuan untuk memberikan fungsionalitas yang sesuai. Tahap 3 merupakan tahapan awal dalam terapi jari tangan berupa terapi menggenggam atau mencubit dan melepaskannya.

3.2.7.2.4. Rancangan Proses Sistem Tahap 4

Rancangan proses sistem tahap 3 bertujuan untuk memberikan fungsionalitas yang sesuai. Pada rancangan proses sistem tahap 4 terdapat proses rancangan dimana ketika pengguna melakukan *pinching* maka garis akan ter-*instantiate* yang kemudian sistem akan menambahkan *collider* sebagai *child* dari garis tersebut. Pembuatan *collider* ini menggunakan cara *AddComponent* dimana untuk melakukan *update* sudut kemiringan memerlukan rumus (1) yang kemudian nilainya akan digunakan dalam rumus (2) untuk mendapatkan sudut yang telah diubah ke dalam bentuk derajat, nilai yang didapatkan pada rumus ke (2) akan di *parsing* sebagai sudut putar sumbu z dari garis dimana y2 dan x2 adalah titik akhir sumbu y sedangkan y1 dan x1 adalah titik awal dalam sumbu y dan sumbu x. Dan mengubahnya dalam derajat yang kemudian menjadi *value* untuk rotasi sumbu z dari objek.

$$\tan \alpha = |y2 - y1|/|x2 - x1| \quad (1)$$

$$\text{angle} = 57,2958 * \text{arc-tan } \alpha \quad (2)$$

3.2.7.3. Rancangan Proses Pemilihan Objek Latihan

Rancangan proses pemilihan objek latihan bertujuan untuk menyesuaikan objek dengan benda-benda yang sering dijumpai pengguna pada kehidupan sehari-hari. Latar tempat yang diambil juga merupakan latar tempat yang umum dan memiliki gambar yang menarik. Dengan pemilihan objek dan latar tempat ini diharapkan pengguna dapat merasakan rasa nyaman ketika melakukan terapi yang kemudian mendorong motivasi melakukan terapi secara rutin hingga tujuan utama yaitu menyembuhkan otot-otot motorik pengguna dapat terwujud.

3.2.7.4. Rancangan Proses Hasil Pelatihan

Rancangan proses hasil pelatihan ini menjelaskan mengenai proses hasil latihan terapi pengguna. Pada setiap latihan

pengguna harus dapat menjalankan setiap instruksi yang didapat dari aplikasi mulai dari melakukan gerak rotasi pergelangan tangan, menggeser objek buatan, tumbukan ke suatu objek buatan dan menggambar beberapa benda. Hasil latihan akan ditampilkan dalam sebuah file tersendiri berbentuk teks yang berisi waktu pengguna menjalankan setiap tahap dari tahap awal hingga akhir yang dapat digunakan sebagai acuan untuk perkembangan otot pengguna dalam menyelesaikan setiap tahapan.

3.2.7.5. Rancangan Proses Melihat Penjelasan Tiap Tahap

Rancangan proses melihat penjelasan tiap tahap bertujuan untuk menjelaskan aktifitas apa saja yang akan dilakukan pengguna pada setiap tahap terapi. Akan ditampilkan beberapa deskripsi pada layar berupa teks yang menjelaskan tiap tiap tahap serta suatu gambar target otot untuk diterapi. Proses ini bertujuan untuk mengedukasi admin dan pengguna dalam menjalankan dan mengerti tujuan dari aplikasi dengan baik.

BAB IV IMPLEMENTASI

Bab ini akan menjelaskan tentang implementasi Tugas Akhir berdasarkan rancangan perangkat lunak yang diuji oleh penulis. Proses implementasi mengacu pada rancangan perangkat yang telah dilakukan sebelumnya, namun juga dimungkinkan terjadinya perubahan-perubahan jika dirasa perlu. Implementasi dilakukan dalam bahasa C#.

4.1. Lingkungan Implementasi

Subbab ini menjelaskan tentang lingkungan implementasi perangkat lunak yang dibangun. Lingkungan selama proses implementasi aplikasi terapi gerak pascastroke dengan menggunakan LMC sebagai berikut:

Tabel 4.1 Spesifikasi Lingkungan Implementasi

Perangkat Keras	<ul style="list-style-type: none">- Prosesor Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz- Memori 8 GB- Monitor- Leap motion Controller
Perangkat Lunak	<ul style="list-style-type: none">- Sistem Operasi Microsoft Windows 8.1 Enterprise 64-bit- Blender 2.7.7- Unity 5.5.1f1- Leap Motion SDK- Adobe Photoshop CC- Adobe Premiere CC

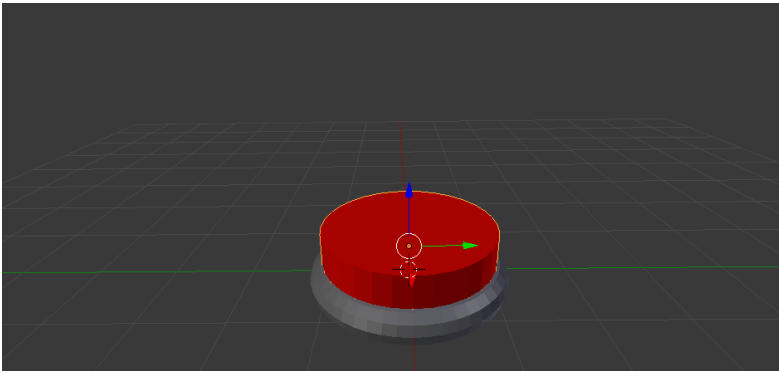
4.2. Implementasi Model 3D

Subbab ini akan menjelaskan tentang implementasi dari model 3D yang digunakan. Implementasi model 3D terdiri dari

enam model 3D sesuai dengan rancangan model 3D pada Subbab 3.2.4.

4.2.1. Implementasi Model 3D Tombol On

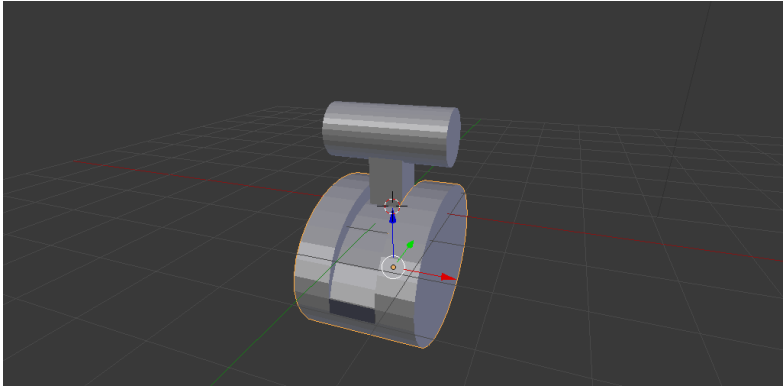
Sesuai Subbab 3.2.4.1 model 3D tombol on yang akan digunakan memiliki dua komponen yaitu tombol dan lingkaran luar tombol. Tombol on yang digunakan memiliki warna merah seperti di kehidupan sehari-hari agar pengguna terbiasa dengan objek tombol on ini. Implementasi tombol on dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi Model 3D Tombol On

4.2.2. Implementasi Model 3D Tuas

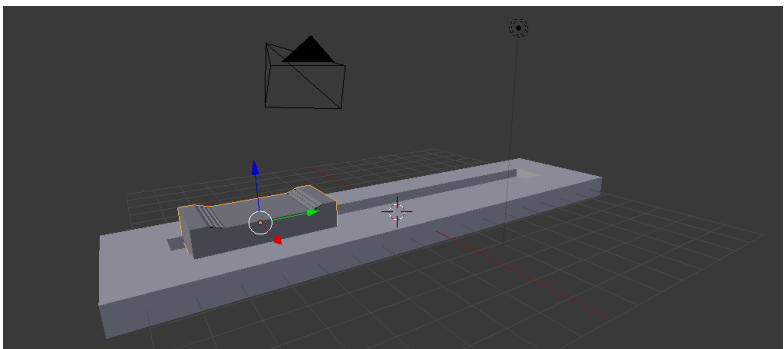
Sesuai Subbab 3.2.4.2 model 3D tuas yang akan digunakan memiliki dua komponen yaitu tuas dan tempat tuas berputar pada porosnya. Tuas yang digunakan memiliki warna abu-abu seperti di dalam rangkaian alat kemudi pesawat atau mobil pada umumnya agar pengguna terbiasa dengan objek tuas ini. Implementasi tuas dapat dilihat pada Gambar 4.2.



Gambar 4.2 Implementasi Model 3D Tuas

4.2.3. Implementasi Model 3D Tuas Geser

Sesuai Subbab 3.2.4.3 model 3D tuas geser yang akan digunakan memiliki dua komponen yaitu tuas dan tempat tuas bergeser. Tuas geser yang digunakan memiliki warna abu-abu seperti di dalam susunan rangkaian alat kendali pesawat luar angkasa atau di dalam mobil pada umumnya agar pengguna terbiasa dengan objek tuas geser ini. Implementasi tuas geser dapat dilihat pada Gambar 4.3.



Gambar 4.3 Implementasi Model 3D Tuas Geser

4.2.4. Implementasi Model 3D Tuas Putar

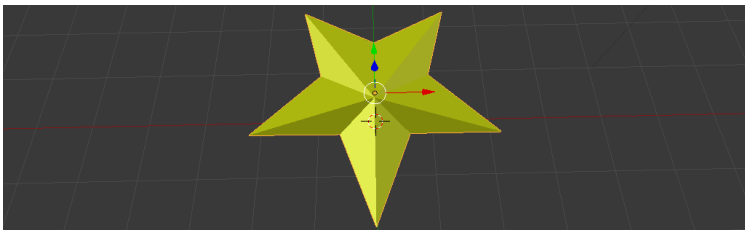
Sesuai Subbab 3.2.4.4 model 3D tuas putar yang akan digunakan memiliki warna abu-abu seperti di pintu kabin dan pintu pesawat pada umumnya agar pengguna terbiasa dengan objek tombol on ini. Implementasi tuas putar dapat dilihat pada Gambar 4.4.



Gambar 4.4 Implementasi Model 3D Tuas Putar

4.2.5. Implementasi Model 3D Bintang

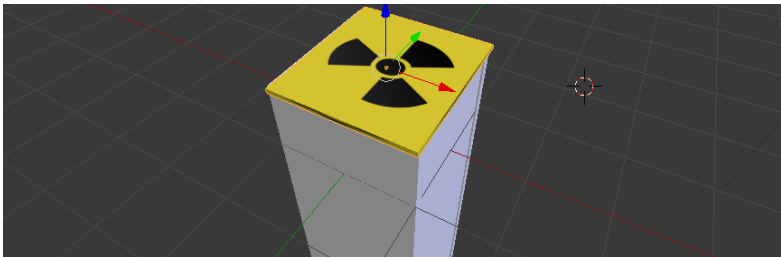
Sesuai Subbab 3.2.4.5 model 3D bintang yang akan digunakan memiliki lima sudut dan berwarna kuning sesuai model bintang pada kehidupan sehari-hari. Implementasi bintang dapat dilihat pada Gambar 4.5.



Gambar 4.5 Implementasi Model 3D Bintang

4.2.6. Implementasi Model 3D Kotak Sebagai Penampung Asteroid

Sesuai Subbab 3.2.4.6 model 3D kotak sebagai penampung asteroid yang akan digunakan memiliki dua komponen yaitu tutup atas kotak dan bidang samping kotak. Kotak yang digunakan memiliki warna kuning agar pengguna melihat dengan jelas posisi kotak dalam *game*. Implementasi kotak sebagai penampung asteroid dapat dilihat pada Gambar 4.6.



Gambar 4.6 Implementasi Model 3D Kotak Penampung Asteroid

4.3. Implementasi Antar muka

Subbab ini akan menjelaskan tentang implementasi dari antar muka yang digunakan. Antarmuka aplikasi ini terdiri dari enam belas tampilan.

4.3.1. Implementasi Antarmuka Menu Utama

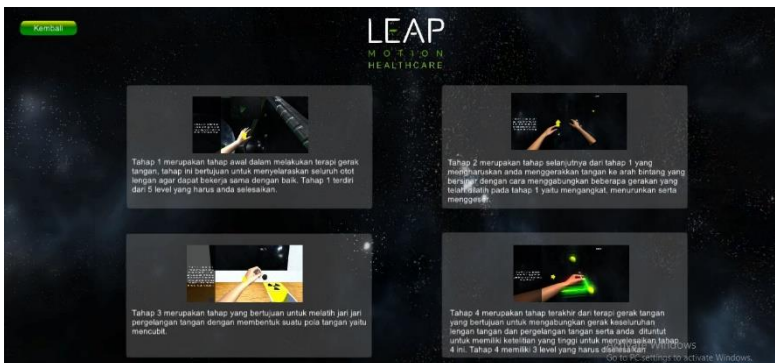
Tampilan pertama merupakan menu utama. Pada tampilan ini terdapat tiga tombol yaitu “mulai” yang berfungsi untuk memulai latihan gerak jari tangan. Tombol kedua merupakan “Pelajari Selengkapnya” yang berfungsi untuk menampilkan secara keseluruhan maksud dan tujuan setiap tahap latihan. Dan tombol yang ketiga yaitu “keluar” untuk keluar dari program. Tampilan antarmuka pada menu utama dapat dilihat pada Gambar 4.7.



Gambar 4.7 Implementasi Antarmuka Menu Utama

4.3.2. Impelementasi Antarmuka Pelajari Selanjutnya

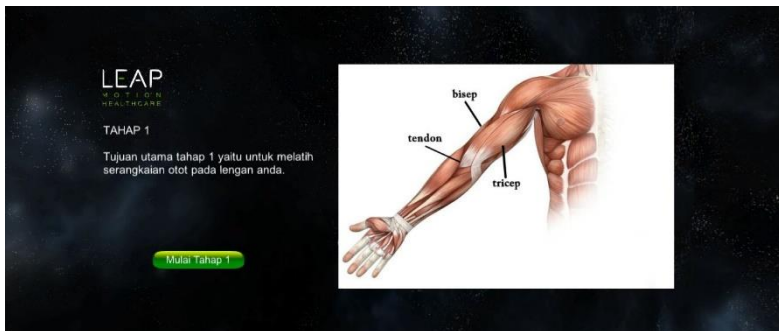
Tampilan kedua merupakan tampilan Pelajari Selanjutnya. Pada tampilan ini terdapat empat buah panel yang berisi teks dan gambar yang menjelaskan fokus tujuan setiap tahapan latihan aplikasi latihan gerak jari tangan dan satu tombol yaitu “kembali” yang berfungsi kembali ke menu utama dan. Tampilan antarmuka Pelajari Selanjutnya dapat dilihat pada Gambar 4.8.



Gambar 4.8 Implementasi Antarmuka Pelajari Selanjutnya

4.3.3. Impelementasi Antarmuka Menu Tahap 1

Tampilan ketiga merupakan tampilan menu tahap 1. Pada tampilan ini terdapat satu tombol yaitu tombol “Mulai” yang berfungsi untuk menuju memasuki latihan tahap 1 terdapat pula gambar bagian tubuh yang di targetkan pada tahapan ini berupa *RawImage* dan teks yang menjelaskan tahapan latihan pertama. Tampilan antarmuka menu tahap 1 dapat dilihat pada Gambar 4.9.



Gambar 4.9 Implementasi Antarmuka Menu Tahap 1

4.3.4. Implementasi Antarmuka Menu Tahap 2

Tampilan keempat merupakan tampilan menu tahap 2. Pada tampilan ini terdapat satu tombol yaitu tombol “Mulai” yang berfungsi untuk menuju memasuki latihan tahap 2 terdapat pula gambar bagian tubuh yang di targetkan pada tahapan ini berupa *RawImage* dan teks yang menjelaskan tahapan latihan kedua. Tampilan antarmuka menu tahap 2 dapat dilihat pada Gambar 4.10.



Gambar 4.10 Implementasi Antarmuka Menu Tahap 2

4.3.5. Impelementasi Antarmuka Menu Tahap 3

Tampilan kelima merupakan tampilan menu tahap 3. Pada tampilan ini terdapat satu tombol yaitu tombol “Mulai” yang berfungsi untuk menuju memasuki latihan tahap 3 terdapat pula gambar bagian tubuh yang di targetkan pada tahapan ini berupa *RawImage* dan teks yang menjelaskan tahapan latihan ketiga. Tampilan antarmuka menu tahap 3 dapat dilihat pada Gambar 4.11.



Gambar 4.11 Implementasi Antarmuka Menu Tahap 3

4.3.6. Impelementasi Antarmuka Menu Tahap 4

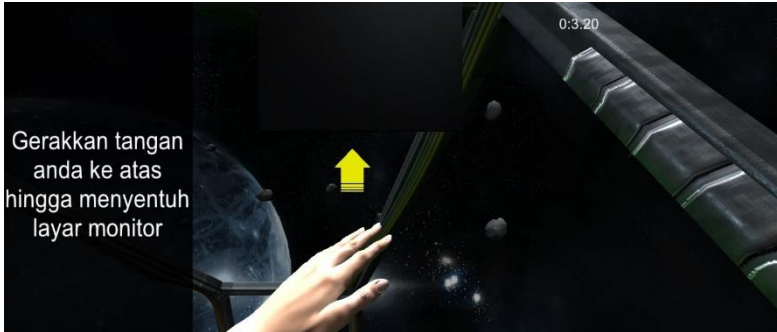
Tampilan keenam merupakan tampilan menu tahap 4. Pada tampilan ini terdapat satu tombol yaitu tombol “Mulai” yang berfungsi untuk menuju memasuki latihan tahap 4 terdapat pula gambar bagian tubuh yang di targetkan pada tahapan ini berupa *RawImage* dan teks yang menjelaskan tahapan latihan keempat. Tampilan antarmuka menu tahap 4 dapat dilihat pada Gambar 4.12.



Gambar 4.12 Implementasi Antarmuka Menu Tahap 4

4.3.7. Impelementasi Antarmuka Tahap 1 Level 1

Tampilan ketujuh merupakan tampilan tahap 1 level 1. Pada tampilan tahap 1 level 1 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 1 level 1 dapat dilihat pada Gambar 4.13.



Gambar 4.13 Implementasi Antarmuka Tahap 1 Level 1

4.3.8. Implementasi Antarmuka Tahap 1 Level 2

Tampilan kedelapan merupakan tampilan tahap 1 level 2. Pada tampilan tahap 1 level 2 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 1 level 2 dapat dilihat pada Gambar 4.14.



Gambar 4.14 Implementasi Antarmuka Tahap 1 Level 2

4.3.9. Implementasi Antarmuka Tahap 1 Level 3

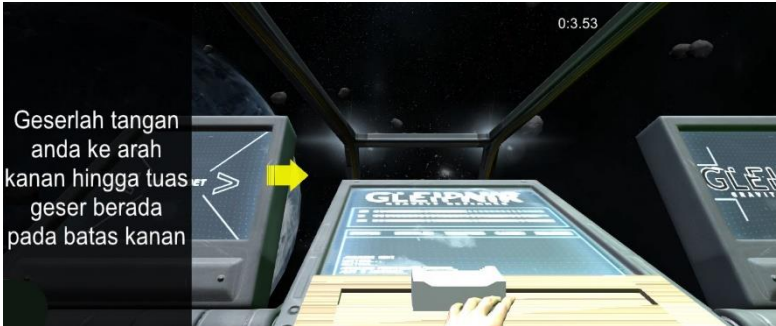
Tampilan kesembilan merupakan tampilan tahap 1 level 3. Pada tampilan tahap 1 level 3 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 1 level 3 dapat dilihat pada Gambar 4.15.



Gambar 4.15 Implementasi Antarmuka Tahap 1 Level 3

4.3.10. Implementasi Antarmuka Tahap 1 Level 4

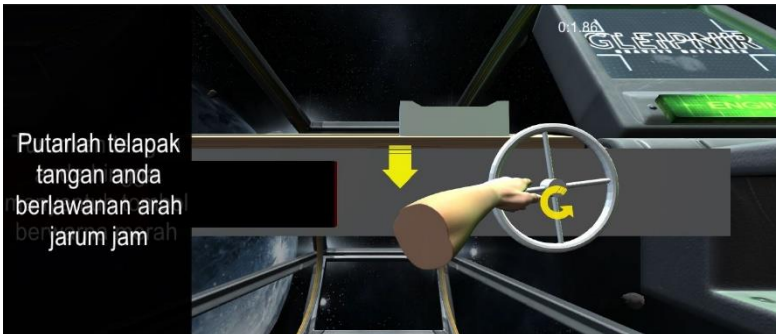
Tampilan kesepuluh merupakan tampilan tahap 1 level 4. Pada tampilan tahap 1 level 4 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 1 level 4 dapat dilihat pada Gambar 4.16.



Gambar 4.16 Implementasi Antarmuka Tahap 1 Level 4

4.3.11. Implementasi Antarmuka Tahap 1 Level 5

Tampilan kesebelas merupakan tampilan tahap 1 level 5. Pada tampilan tahap 1 level 5 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 1 level 5 dapat dilihat pada Gambar 4.17.



Gambar 4.17 Implementasi Antarmuka Tahap 1 Level 5

4.3.12. Implementasi Antarmuka Tahap 2

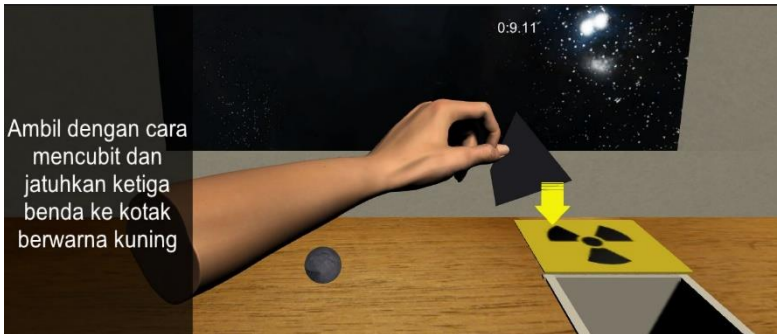
Tampilan kedubelas merupakan tampilan tahap 2. Pada tampilan tahap 2 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 2 dapat dilihat pada Gambar 4.18.



Gambar 4.18 Implementasi Antarmuka Tahap 2

4.3.13. Implementasi Antarmuka Tahap 3

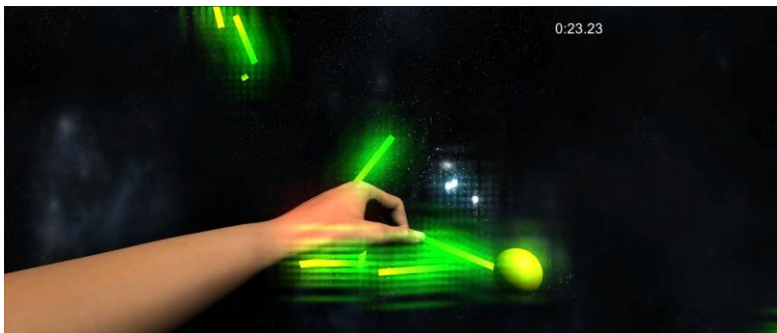
Tampilan ketigabelas merupakan tampilan tahap 3. Pada tampilan tahap 3 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 3 dapat dilihat pada Gambar 4.19.



Gambar 4.19 Implementasi Antarmuka Tahap 3

4.3.14. Implementasi Antarmuka Tahap 4 Level 1

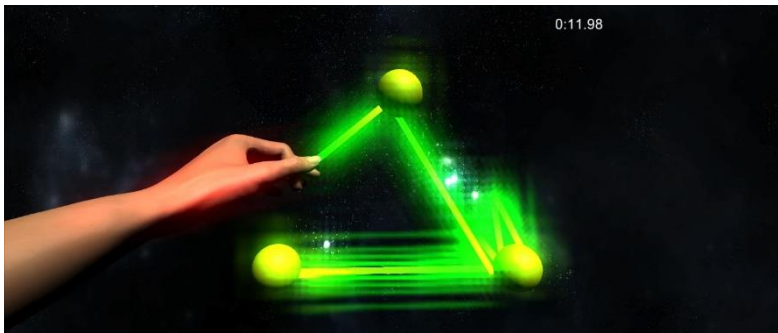
Tampilan keempatbelas merupakan tampilan tahap 4 level 1. Pada tampilan tahap 4 level 1 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 4 level 1 dapat dilihat pada Gambar 4.20.



Gambar 4.20 Implementasi Antarmuka Tahap 4 Level 1

4.3.15. Implementasi Antarmuka Tahap 4 Level 2

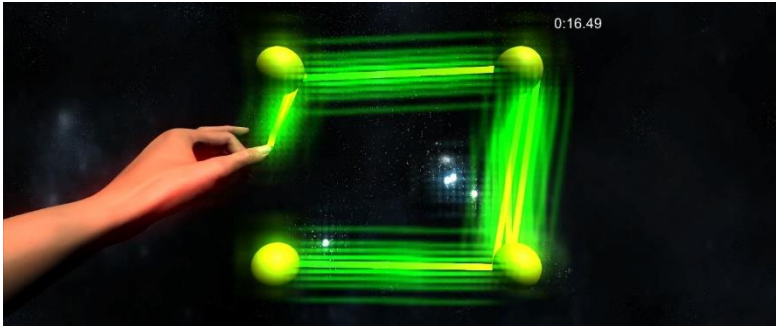
Tampilan kelimabelas merupakan tampilan tahap 4 level 2. Pada tampilan tahap 4 level 2 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 4 level 2 dapat dilihat pada Gambar 4.21.



Gambar 4.21 Implementasi Antarmuka Tahap 4 Level 2

4.3.16. Implementasi Antarmuka Tahap 4 Level 3

Tampilan keenambelas merupakan tampilan tahap 4 level 3. Pada tampilan tahap 4 level 3 terdapat instruksi di sebelah kiri layar mengenai gerak yang harus dilakukan oleh pengguna serta terdapat anak panah yang membantu pengguna untuk mengarahkan tangannya ke suatu posisi. Tampilan antarmuka tahap 4 level 3 dapat dilihat pada Gambar 4.22.



Gambar 4.22 Implementasi Antarmuka Tahap 4 Level 3

4.4. Implementasi Proses

Subbab ini akan menjelaskan tentang implementasi dari rancangan proses yang dijelaskan pada Bab III. Penjelasan mengenai implementasi proses ini dibagi berdasarkan komponen-komponen aplikasi. Pada aplikasi ini setiap proses melibatkan deteksi tangan yang disediakan oleh *Leap Motion SDK* yaitu *class HandController* serta Collider tangan pada *class RigidHand* dan deteksi gerak tangan berupa *pinch* pada *class GrabbingHand*. Berikut ini merupakan penjelasan dari tiap-tiap proses tahapan.

4.4.1. Implementasi Controller Tahap 1

Proses ini merupakan proses yang mengontrol setiap level pada tahap 1 seperti pergantian level, animasi dan penentuan skor. Implementasi controller tahap 1 dapat dilihat pada potongan Kode Sumber 4.1.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class animationscenel : MonoBehaviour {
    Animator animatorscenel;
    public Text timer;
```



```

        float t, starttime, scorebasedtime;
        int
finalscore, count, finalscore1, finalscore2, finalscore3, finalscore4, finalscore5;
        bool done, flag1, flag2, flag3, flag4, flag5;
        public bool enablescript;
        public GameObject CanvasNext;
        public GameObject myReward;
        public GameObject CanvasInst;

        void Start () {
            done = false;
            animatorscenel = GetComponent<Animator> ();
        }

        IEnumerator myRewardCuroutine(){
            if (finalscore >= 85) {
                myReward.SetActive (true);
                yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
                yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
                yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(3).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(3).GetComponent<AudioSource> ().enabled = true;
                yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = false;
                myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = false;
                myReward.transform.GetChild
(3).GetComponent<Image> ().enabled = false;
                myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = false;
                myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = false;

```

```

        myReward.transform.GetChild
(3).GetComponent ().enabled = false;
        myReward.SetActive (false);
    } else if (75 <= finalscore && finalscore <
85) {
        myReward.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = false;
        myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = false;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = false;
        myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = false;
        myReward.SetActive (false);
    } else if (65 <= finalscore && finalscore <
75) {
        myReward.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = false;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = false;
        myReward.SetActive (false);
    } else {
        myReward.SetActive (true);
        yield return new WaitForSeconds
(0.65f);

```

```

        myReward.SetActive (false);
    }

    if (done) {
        StartCoroutine (waits (0));
        CanvasNext.SetActive (true);
    }
}

IEnumerator waits(int i){
    yield return new WaitForSeconds (3f);
    switch (i) {
        case 0:
            break;
        case 1:

            CanvasInst.transform.GetChild(2).gameObject.SetActive
(true);
                break;
            case 2:

            CanvasInst.transform.GetChild(3).gameObject.SetActive
(true);
                break;
            case 3:

            CanvasInst.transform.GetChild(4).gameObject.SetActive
(true);
                break;
            case 4:

            CanvasInst.transform.GetChild(5).gameObject.SetActive
(true);
                break;
    }
}

void Update () {
    finalscore = 0;
    scorebasedtime = 0;
    if
(animatorscene1.GetCurrentAnimatorStateInfo(0).normalizedTime
> 1 && !animatorscene1.IsInTransition(0)){
        t += Time.deltaTime;
        string minutes = ((int)t /
60).ToString ();
        string seconds = (t % 60).ToString
("f2");
        timer.text = minutes + ":" + seconds;
        int temp = (int)t % 60;

```

```

        int fullscore = 100;
        finalscore = fullscore - temp;
    }

    if (GameObject.Find
("screenradar").GetComponent<T1L1radaron> ().finished &&
flag1 == false) {
        t = 0;
        flag1 = true;
        finalscore1 = finalscore;
        string myscore1 = timer.text;

        CanvasInst.transform.GetChild(1).gameObject.SetActive
(false);
        StartCoroutine (myRewardCuroutine());
        PlayerPrefs.SetString ("scenellevel1",
myscore1);
        animatorscenel.SetInteger ("State",
1);
        int i = 1;
        StartCoroutine (waits (i));
    }
    if (GameObject.Find
("button").GetComponent<T1L2pressbutton> ().finished && flag2
== false) {
        t = 0;
        flag2 = true;
        finalscore2 = finalscore;
        string myscore2 = timer.text;

        CanvasInst.transform.GetChild(2).gameObject.SetActive
(false);
        StartCoroutine (myRewardCuroutine());
        PlayerPrefs.SetString ("scenellevel2",
myscore2);
        animatorscenel.SetInteger ("State",
2);
        int i = 2;
        StartCoroutine (waits (i));
    }
    if (GameObject.Find
("togglemain").GetComponent<T1L3toggle> ().finished && flag3
== false) {
        t = 0;
        flag3 = true;
        finalscore3 = finalscore;
        string myscore3 = timer.text;
    }

```

```

        CanvasInst.transform.GetChild(3).gameObject.SetActive
(false);
                                StartCoroutine (myRewardCuroutine());
                                PlayerPrefs.SetString ("scenellevel3",
myscore3);
                                animatorscenel.SetInteger ("State",
3);
                                int i = 3;
                                StartCoroutine (waits (i));
        }
        if (GameObject.Find
("Cube_001").GetComponent<T1L4switch> ().finished && flag4 ==
false) {
                                t = 0;
                                flag4 = true;
                                finalscore4 = finalscore;
                                string myscore4 = timer.text;

        CanvasInst.transform.GetChild(4).gameObject.SetActive
(false);
                                StartCoroutine (myRewardCuroutine());
                                PlayerPrefs.SetString ("scenellevel4",
myscore4);
                                animatorscenel.SetInteger ("State",
4);
                                int i = 4;
                                StartCoroutine (waits (i));
                                enablescript = true;
        }
        if (GameObject.Find
("valve").GetComponent<T1L5valvecheck> ().engineOnrotate &&
flag5 == false) {
                                t = 0;
                                flag5 = true;
                                finalscore5 = finalscore;
                                string myscore5 = timer.text;

        CanvasInst.transform.GetChild(5).gameObject.SetActive
(false);
                                PlayerPrefs.SetString ("scenellevel5",
myscore5);
                                animatorscenel.SetInteger ("State",
5);
                                done = true;
                                StartCoroutine (myRewardCuroutine());
        }
    }
}

```

Kode Sumber 4.1 Controller Tahap 1

4.4.2. Implementasi Tahap 1 Level 1

Tahap ini merupakan tahap pertama yang harus dilakukan pengguna untuk melakukan terapi gerak tangan. Pada tahap ini pengguna harus mengangkat tangan untuk menyentuh monitor hingga menyala. Aplikasi harus dapat mendeteksi tangan dan tumbukan yang terjadi antara monitor dengan tangan pengguna. Untuk mendeteksi tumbukan antara tangan pengguna dengan monitor menggunakan *class TIL1radaron* . Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan monitor dapat dilihat pada Kode Sumber 4.2.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class TIL1radaron : MonoBehaviour {
    public Light movieLight;
    public bool finished;
    MovieTexture movies;

    void Start () {
        movieLight.enabled = false;
        Renderer r = GetComponent<Renderer> ();
        movies =
(MovieTexture)r.material.mainTexture;
    }

    void Update () {
        if (finished)
            return;
    }

    void OnTriggerEnter() {
        finished = true;
        movieLight.enabled = true;
        movies.Play ();
        movies.loop = true;
        this.GetComponent<AudioSource> ().enabled =
true;
    }
}
```

Kode Sumber 4.2 Mendeteksi Collider Tangan Pengguna Dengan Monitor

4.4.3. Implementasi Tahap 1 Level 2

Proses ini merupakan proses kedua yang harus dilakukan oleh pengguna. Pada tahap ini pengguna menurunkan tangan untuk menekan tombol yang berada di bawah tangan pengguna. Terdapat dua proses yang terjadi pada sistem yang pertama yaitu mendeteksi collider tangan pengguna dengan tombol dan mengganti tampilan monitor ketika tombol telah berhasil di sentuh oleh pengguna.

4.4.3.1. Mendeteksi Collider Tangan Pengguna Dengan Tombol

Aplikasi harus dapat mendeteksi tangan dan tumbukan yang terjadi antara tombol dengan tangan pengguna . Untuk mendeteksi tumbukan antara tangan pengguna dengan tombol menggunakan *class T1L2pressbutton*. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.3.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class T1L2pressbutton : MonoBehaviour {
    public Animator anim;
    public bool finished;
    int finalscore;

    void Start () {
        anim = GetComponent<Animator> ();
    }

    void Update() {
        if (finished)
            return;
    }

    void OnTriggerEnter () {
        anim.enabled = true;
        finished = true;
    }
}
```

}

Kode Sumber 4.3 Mendeteksi Collider Tangan Pengguna Dengan Tombol

4.4.3.2. Trigger Pergantian Tampilan Monitor

Aplikasi harus dapat memberikan umpan balik ketika pengguna menekan tombol on, hal ini dapat membuat pengguna merasa terhibur dan menikmati jalannya terapi, untuk melakukan proses ini maka memerlukan *class T1L2monitorchange*. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.4.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class T1L2monitorchangemat : MonoBehaviour {

    void Update () {
        if (GameObject.Find
("button").GetComponent<T1L2pressbutton> ().finished) {
            changeMaterial ();
        }
    }
    void changeMaterial(){
        Material screenchange =
Resources.Load("monitoron", typeof(Material)) as Material;
        gameObject.GetComponent<Renderer>().material
= screenchange;}
}
```

Kode Sumber 4.4 Trigger Pergantian Tampilan Monitor

4.4.4. Implementasi Tahap 1 Level 3

Proses ini merupakan proses ketiga yang harus dilakukan oleh pengguna. Pada tahap ini pengguna dianjurkan untuk mengulurkan tangan ke depan hingga tuas dapat bergerak maju. Proses yang terjadi yaitu dibutuhkannya *class* yang dapat

mengetahui tangan pengguna untuk dapat bersentuhan dengan tuas dan menjalankan animasi pada tuas. Aplikasi harus dapat mendeteksi tangan dan tumbukan yang terjadi antara tuas dengan tangan pengguna. Untuk mendeteksi tumbukan antara tangan pengguna dengan tombol menggunakan *class T1L3toggle*. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.5.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using Leap;

public class T1L3toggle : MonoBehaviour {
    public Animator anim;
    public bool finished;

    void Update () {
        if (finished)
            return;
    }
    void OnTriggerEnter(){
        anim.enabled = true;
        finished = true;
        this.GetComponent<AudioSource> ().enabled =
true;}
}
```

Kode Sumber 4.5 Mendeteksi Collider Tangan Pengguna Dengan Tuas

4.4.5. Implementasi Tahap 1 Level 4

Proses ini merupakan proses keempat yang harus dilakukan oleh pengguna. Pada tahap ini pengguna dianjurkan untuk menggeser tangan ke kanan hingga tuas geser dapat bergerak ke samping kanan sampai penuh. Proses yang terjadi yaitu dibutuhkannya *class* yang dapat mendeteksi tangan pengguna untuk dapat bersentuhan dengan tuas dan deteksi bila tuas geser telah pada posisi penuh. Aplikasi harus dapat mendeteksi tuas geser jika sudah berada pada posisi penuh yang disebabkan oleh aksi dari

tangan pengguna menggeser tuas ke arah samping kanan. Untuk mendeteksi apakah tuas geser berada pada posisi penuh menggunakan *class T1L4switch*. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.6.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class T1L4switch : MonoBehaviour {
    public Rigidbody rb;
    public bool finished;
    Vector3 translateposition;
    Vector3 initialtranslateposition;
    Quaternion lockrotation;
    float
newpositionZ, initialposY, initialposX, initialposZ;

    void Start () {
        rb = GetComponent<Rigidbody> ();
        lockrotation =
Quaternion.Euler (transform.rotation.eulerAngles.x,

                transform.rotation.eulerAngles.y,

                transform.rotation.eulerAngles.z);
        initialposX = transform.localPosition.x;
        initialposY = transform.localPosition.y;
        initialposZ = transform.localPosition.z;
    }

    void FixedUpdate(){
        rb.velocity = new Vector3 (0, 0, -20);
    }

    void Update () {
        transform.rotation =
Quaternion.Euler (270, 90, 0);
        transform.localPosition = new
Vector3(initialposX, initialposY, transform.localPosition.z);
        if (transform.localPosition.z >= 5) {
            transform.localPosition = new
Vector3(initialposX, initialposY, 5);
            finished = true;
        }
    }
}
```

```

        this.GetComponent<AudioSource>
    ).enabled = true;
    }
    else if (transform.localPosition.z <
initialposZ) {
        transform.localPosition = new
Vector3(initialposX, initialposY, initialposZ);
    }

    if (newpositionZ < transform.localPosition.z)
    {
        float newpositionZ =
transform.localPosition.z;
    }
}

```

Kode Sumber 4.6 Mendeteksi Tuas Geser Dalam Posisi Penuh

4.4.6. Implementasi Tahap 1 Level 5

Proses ini merupakan proses kelima yang harus dilakukan oleh pengguna. Pada tahap ini pengguna dianjurkan untuk memutar pergelangan tangan berlawanan arah jarum jam hingga tuas putar dapat mencapai sudut putar tertentu dan dapat menyalakan mesin dari pesawat. Proses yang terjadi yaitu dibutuhkannya *class* yang dapat mendeteksi sudut putar pergelangan tangan pengguna, menyalakan tanda jika pergelangan tangan pengguna mencapai sudut tertentu, dan tanda bila besar sudut sudah terpenuhi.

4.4.6.1. Mendeteksi Besar Sudut Pergelangan Tangan Pengguna

Aplikasi harus dapat mendeteksi besar sudut pergelangan tangan pengguna. Untuk menjalankan proses ini maka perlunya sebuah *class* yang menangkap dan menginterpretasikan sudut putar. Proses ini menggunakan *class TIL5rotate*. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.7.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Leap;

public class T1L5rotate : MonoBehaviour {
    public GameObject myobject;
    public bool engineOnrotate;
    float tempZ = 0;

    void Start () {
        engineOnrotate = false;
    }

    void Update() {
        if (GameObject.Find ("Main
Camera").GetComponent<animationscene1> ().enablescript){

            this.GetComponent<T1L5rotate>().enabled = true;
            HandModel hand_model =
GetComponent<HandModel>();
            Hand leap_hand =
hand_model.GetLeapHand();
            Quaternion myhandrot =
hand_model.GetPalmRotation ();
            myhandrot.x = 0;
            myhandrot.y = 0;
            float handZangular =
myhandrot.eulerAngles.z;
            if(tempZ < handZangular &&
handZangular < 181){
                tempZ =
myobject.transform.rotation.eulerAngles.z;
                myobject.transform.rotation =
myhandrot;
                if (handZangular > 160) {
                    return;
                }
            }
        }
    }
}

```

Kode Sumber 4.7 Mendeteksi Besar Sudut Pergelangan Tangan Pengguna

4.4.6.2. Check Besar Sudut Putar Pergelangan Tangan Pengguna

Aplikasi harus dapat melakukan *checking* besar sudut dari putar pergelangan tangan pengguna untuk menjalankan proses berikutnya yaitu menyalakan lampu mesin. Proses *checking* ini dapat menggunakan *TIL5valvecheck*. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.8.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class TIL5valvecheck : MonoBehaviour {
    public bool engineOnrotate;

    void Start () {
        engineOnrotate = false;
    }

    void Update () {
        if (this.transform.rotation.eulerAngles.z >
160) {
            engineOnrotate = true;
            this.GetComponent<AudioSource>
().enabled = true;
        }
    }
}
```

Kode Sumber 4.8 Check Besar Sudut Putar Pergelangan Tangan Pengguna

4.4.6.3. Trigger Lampu Mesin Berdasarkan Besaran Sudut Putar Pergelangan Tangan Pengguna

Aplikasi harus dapat melakukan *trigger* untuk menyalakan lampu mesin pesawat, hal ini sangat berguna bagi pengguna sebagai timbal balik dari usaha yang telah dilakukan. Proses untuk menyalakan lampu mesin pensawat dapat menggunakan *class*

T1L5engineOnrotate. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.9.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class T1L5engineOnrotate : MonoBehaviour {
    public GameObject myvalve;
    public Light lightoff;
    public Light lighton;
    public float duration = 2F;

    void Start () {
        lightoff.enabled = true;
        lighton.enabled = false;
    }

    void Update () {
        float phi = Time.time / duration * 2 *
Mathf.PI;
        float amplitude = Mathf.Cos(phi) * 1.5F +
1.5F;
        lightoff.intensity = amplitude;

        if (myvalve.GetComponent<T1L5valvecheck>
().engineOnrotate) {
            changeMaterial ();
            lightoff.enabled = false;
            lighton.enabled = true;
            float phi2 = Time.time / duration * 2
* Mathf.PI;
            float amplitude2 = Mathf.Cos(phi) *
1.5F + 1.5F;
            lighton.intensity = amplitude2;
        }

        void changeMaterial(){
            Material screenchange = Resources.Load("on",
typeof(Material)) as Material;
            gameObject.GetComponent<Renderer>().material
= screenchange;
        }
    }
}

```

Kode Sumber 4.9 Trigger Lampu Mesin Berdasarkan Besaran Sudut Putar Pergelangan Tangan Pengguna

4.4.7. Implementasi Tahap 2

Proses ini merupakan proses tahap 2 yang harus dijalankan oleh pengguna untuk bisa menyelesaikan tahapan terapi gerak tangan. Proses ini berfokus pada kombinasi bentuk gerakan yang telah dilakukan pada tahap sebelumnya. Maka proses ini perlu fungsi deteksi *collider* dan fungsi untuk mengacak *spawn* dari benda agar pengguna dapat menggerakkan tangannya secara acak. Proses ini merupakan proses yang dibutuhkan untuk dapat mendeteksi *collider* antara tangan pengguna dengan bintang sebagai objek benda dan mengacak kemunculan bintang. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.10.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class T2 : MonoBehaviour {
    public GameObject randombox;
    public GameObject CanvasNext;
    public GameObject CanvasInst;
    public GameObject CanvasDone;
    public GameObject myReward;
    public GameObject star;
    public Text timer, none;
    public bool finished;
    float scorebasedtime, starttime;
    int finalscore, varrand, value, count = 0;
    List<int> rand = new List<int>();
    List<Vector3> pos = new List<Vector3> ();

    bool[] test = new
bool[] {false, false, false, false, false, false,
        false, false, false, false, false, false,
        false, false, false, false, false, false};
    bool[] check = new
bool[] {false, false, false, false, false, false,
```

```

        false,false,false,false,false,false,
        false,false,false,false,false,false  });

void Start () {
    starttime = Time.time;
    for (int i = 0; i < 18; i++)
        rand.Add (i);
1.58f));
    pos.Add (new Vector3 (-6.74f, 5.96f, -
1.58f));
    pos.Add (new Vector3 (-6.74f, 0.98f, -
1.58f));
    pos.Add (new Vector3 (-6.74f, -3.7f, -
    pos.Add (new Vector3 (8.35f, 5.96f, -1.58f));
    pos.Add (new Vector3 (8.35f, 0.98f, -1.58f));
    pos.Add (new Vector3 (8.35f, -3.7f, -1.58f));
    pos.Add (new Vector3 (-6.74f, 5.96f, 3.66f));
    pos.Add (new Vector3 (-6.74f, 0.98f, 3.66f));
    pos.Add (new Vector3 (-6.74f, -3.7f, 3.66f));
    pos.Add (new Vector3 (8.35f, 5.96f, 3.66f));
    pos.Add (new Vector3 (8.35f, 0.98f, 3.66f));
    pos.Add (new Vector3 (8.35f, -3.7f, 3.66f));
    pos.Add (new Vector3 (-6.74f, 5.96f, 7.93f));
    pos.Add (new Vector3 (-6.74f, 0.98f, 7.93f));
    pos.Add (new Vector3 (-6.74f, -3.7f, 7.93f));
    pos.Add (new Vector3 (8.35f, 5.96f, 7.93f));
    pos.Add (new Vector3 (8.35f, 0.98f, 7.93f));
    pos.Add (new Vector3 (8.35f, -3.7f, 7.93f));
    InvokeRepeating ("SpawnBox", 0, 5);
    StartCoroutine (waits ());
}

IEnumerator waits(){
    yield return new WaitForSeconds (10);

    CanvasInst.transform.GetChild(1).gameObject.SetActive
(false);
}

IEnumerator waits2(int count){
    CanvasDone.SetActive (true);
    none.text = count+"/5";
    yield return new WaitForSeconds (3);
    CanvasDone.SetActive (false);
    // CanvasDone.GetComponent<AudioSource>
().enabled = false;
}

IEnumerator canvasnext(){

```



```

        if (finalscore >= 85) {
            CanvasNext.SetActive (true);
            yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(3).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(3).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);
        } else if (75 <= finalscore && finalscore <
85) {
            CanvasNext.SetActive (true);
            yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);
        } else if (65 <= finalscore && finalscore <
75) {
            CanvasNext.SetActive (true);
            yield return new WaitForSeconds
(0.65f);
                myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
                myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;

```

```

        yield return new WaitForSeconds
(0.65f);
    } else {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
    }
}

void Update () {
    if (count == 5) {
        StartCoroutine (canvasnext ());
        finished = true;
        string myscore = timer.text;
        PlayerPrefs.SetString ("scene2",
myscore);
    }
    if (finished)
        return;
    for (int i = 0; i < test.Length; i++) {
        if (test [i] == true && check [i] ==
false) {
            count++;
            check [i] = true;
            rand.RemoveAt (varrand);
            Vector3 post = pos[value];
            Instantiate (star, new Vector3
(post.x,post.y,post.z),transform.rotation);
            StartCoroutine
(waits2(count));
        }
    };
    float t = Time.time - starttime;
    string minutes = ((int)t / 60).ToString ();
    int intseconds = (int)t % 60;
    string seconds = (t % 60).ToString ("f2");
    timer.text = minutes + ":" + seconds;
    scorebasedtime = Time.timeSinceLevelLoad;
    int temp = (int) scorebasedtime;
    int fullscore = 100;
    finalscore = fullscore - temp + 15;
}

void OnTriggerEnter(Collider other)
{
    if (IsHand(other))
    {
        test[varrand] = true;
    }
}

```

```

    }

    void SpawnBox() {
        varrand = Random.Range (0, rand.Count);
        value = rand [varrand];
        changeposition (pos [value]);
    }

    void changeposition(Vector3 newposition){
        randombox.transform.position = newposition;
    }

    private bool IsHand(Collider other)
    {
        if (other.transform.parent &&
other.transform.parent.parent &&
other.transform.parent.parent.GetComponent<HandModel>())
            return true;
        else
            return false;
    }
}

```

Kode Sumber 4.10 Collider Tangan Pengguna Dengan Bintang Sebagai Objek Benda

4.4.8. Implementasi Tahap 3

Proses ini merupakan proses tahap 3 yang harus dijalankan oleh pengguna untuk bisa menyelesaikan tahapan terapi gerak tangan. Proses ini berfokus pada pergerakan pergelangan jari-jari tangan pengguna. Proses ini perlu fungsi deteksi gerak jari ketika melakukan *pinching*, deteksi keberhasilan memasukkan objek ke tempat yang dituju, dan melakukan *grabbing* objek. Proses ini merupakan proses yang dibutuhkan agar aplikasi dapat menghitung objek yang telah memasuki kotak dari gerak tangan pengguna serta melakukan perhitungan skor pengguna dalam memainkan tahap ini. Proses ini memerlukan dua class yaitu *T3trigger* dan *T3randomopenbasket*. Potongan kode untuk mendeteksi tumbukan antar tangan pengguna dengan tombol dapat dilihat pada Kode Sumber 4.11 dan Kode Sumber 4.12.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class T3trigger : MonoBehaviour {
    public bool flag1,flag2,flag3;
    public static int count;
    public T3randopenclosebasket linktomyscript;
    public GameObject CanvasNext;
    public GameObject myReward;
    public Text timer;
    public AudioSource audio;
    float starttime,scorebasedtime;
    int endsession,finalscore;
    bool finished;

    void Start () {
        starttime = Time.time;
    }

    IEnumerator canvasnext() {
        if (finalscore >= 85) {
            CanvasNext.SetActive (true);
            yield return new WaitForSeconds
(0.65f);
            myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
            myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);
            myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
            myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);
            myReward.transform.GetChild
(3).GetComponent<Image> ().enabled = true;
            myReward.transform.GetChild
(3).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);
        } else if (75 <= finalscore && finalscore <
85) {
            CanvasNext.SetActive (true);
            yield return new WaitForSeconds
(0.65f);

```

```

        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else if (65 <= finalscore && finalscore <
75) {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
    }
}

void Update () {
    if (finished)
        return;
    if (endsession == 3) {
        finished = true;
        string myscore = timer.text;
        PlayerPrefs.SetString ("scene3",
myscore);
        StartCoroutine (canvasnext());
    }
    float t = Time.time - starttime;
    string minutes = ((int)t / 60).ToString ();
    int intseconds = (int)t % 60;
    string seconds = (t % 60).ToString ("f2");
    timer.text = minutes + ":" + seconds;
    scorebasedtime = Time.timeSinceLevelLoad;
    int temp = (int) scorebasedtime;
    int fullscore = 100;
    finalscore = fullscore - temp;

```

```

    }

    void OnTriggerEnter(Collider other){
        if (other.tag == "asteroid") {
            endsession++;
            audio.GetComponent<AudioSource>
().Play ();
        }
    }
}

```

Kode Sumber 4.11 Deteksi Objek Memasuki Kotak Dan Animasi

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class T3randopenclosebasket : MonoBehaviour {

    public GameObject openclose;
    public GameObject CanvasInst;
    bool done;
    int randomopenbox;

    void Start () {
        StartCoroutine (waits ());
    }

    IEnumerator waits(){
        yield return new WaitForSeconds (10);

        CanvasInst.transform.GetChild(1).gameObject.SetActive
(false);
    }

    void Update () {
        if (done) {
            openclose.GetComponent<Animator>
().SetInteger ("State", 1);
        }
    }
}

```

Kode Sumber 4.12 Animasi Kotak

4.4.9. Implementasi Instantiate Line Tahap 4

Proses ini merupakan proses yang berfungsi untuk menggambarkan posisi - posisi tangan pengguna dalam membentuk suatu garis. Pada proses ini terdapat beberapa fungsi yang digunakan yaitu fungsi mendeteksi ketika tangan pengguna dalam kondisi *pinch*, menangkap posisi awal dan posisi akhir *pinch* pengguna dalam bentuk *Vector3*, fungsi untuk membuat garis, dan fungsi untuk memberikan *collider* pada setiap garis yang telah dibuat. Potongan kode untuk menjalankan *Intantiate Line* dapat dilihat pada Kode Sumber 4.13.

```

using UnityEngine;
using System.Collections;
using System.Collections.Generic;
using Leap;

public class T4line : MonoBehaviour {

    public const float TRIGGER_DISTANCE_RATIO = 0.7f;
    bool trigger_pinch, pinching_, makeLines;
    private LineRenderer line;
    private LineRenderer renderer;
    public List<Vector3> pointList;
    Vector3 Start;
    Vector3 EndLine;
    public GameObject newLine;
    private GameObject mycollidernewLine;

    void Awake() {
        line = gameObject.AddComponent<LineRenderer>
    );
        line.material = new Material (Shader.Find
("Particles/Additive"));
        line.startWidth = 0.1f;
        line.endWidth = 0.1f;
        line.startColor = Color.yellow;
        line.endColor = Color.green;
        line.useWorldSpace = true;
        pointList = new List<Vector3> ();
    }

    void Update() {
        trigger_pinch = false;
        HandModel hand_model =
GetComponent<HandModel>();
        Hand leap_hand = hand_model.GetLeapHand();
    }
}

```

```

        if (leap_hand == null)
            return;

        Vector leap_thumb_tip =
leap_hand.Fingers[0].TipPosition;
        float proximal_length =
leap_hand.Fingers[0].Bone(Bone.BoneType.TYPE_PROXIMAL).Length;
        float trigger_distance = proximal_length *
TRIGGER_DISTANCE_RATIO;

        for (int i = 1; i < HandModel.NUM_FINGERS &&
!trigger_pinch; ++i) {
            Finger finger = leap_hand.Fingers[i];

            for (int j = 0; j <
FingerModel.NUM_BONES && !trigger_pinch; ++j) {
                Vector leap_joint_position =
finger.Bone((Bone.BoneType)j).NextJoint;
                if
(leap_joint_position.DistanceTo(leap_thumb_tip) <
trigger_distance)
                    trigger_pinch = true;
            }
        }

        Vector3 pinch_position =
hand_model.fingers[0].GetTipPosition();
        if (trigger_pinch && !pinching_) {
            pinching_ = true;
            Start.x = pinch_position.x;
            Start.y = pinch_position.y;
            Start.z = 0;
        }
        else if (!trigger_pinch && pinching_) {
            EndLine.x = pinch_position.x;
            EndLine.y = pinch_position.y;
            EndLine.z = 0;
            pinching_ = false;
        }
        if (trigger_pinch) {
            makeLines = true;
            if (!pointList.Contains
(pinch_position)) {
                pointList.Add
(pinch_position);

                line.SetVertexCount (2);
                line.SetPosition (0,Start);
            }
        }
    }
}

```



```

                                line.SetPosition
(1,pinch_position);
                                }
                                }

                                if (!trigger_pinch && makeLines) {
                                    renderer =
newLine.GetComponent<LineRenderer> ();
                                    renderer.material = new Material
(Shader.Find ("Particles/Additive"));
                                    renderer.startWidth = 0.1f;
                                    renderer.endWidth = 0.1f;
                                    renderer.startColor = Color.yellow;
                                    renderer.endColor = Color.green;
                                    renderer.useWorldSpace = true;
                                    renderer.SetPosition (0,Start);
                                    renderer.SetPosition (1,new
Vector3(pinch_position.x,pinch_position.y,0));

                                    mycollidernewLine = Instantiate
(newLine, Start, transform.rotation) as GameObject;
                                    addColliderToLine ();
                                    makeLines = false;
                                }
                                }
                                private bool checkPoints (Vector3 pointA, Vector3
pointB){
                                    return (pointA.x == pointB.x && pointA.y ==
pointB.y);
                                }
                                private void addColliderToLine()
                                {
                                    BoxCollider col = new
GameObject("Collider").AddComponent<BoxCollider> ();
                                    string myparenttag = mycollidernewLine.tag;
                                    col.tag = myparenttag;
                                    col.transform.parent =
mycollidernewLine.transform;
                                    float lineLength = Vector3.Distance (Start,
EndLine);
                                    col.size = new Vector3 (lineLength, 0.1f,
1f);

                                    Vector3 midPoint = (Start + EndLine)/2;
                                    col.transform.position = midPoint;
                                    float angle = (Mathf.Abs (Start.y -
EndLine.y) / Mathf.Abs (Start.x - EndLine.x));
                                    if((Start.y<EndLine.y && Start.x>EndLine.x)
|| (EndLine.y<Start.y && Start.x>Start.x))
                                    {

```

```

        angle*=-1;}
    angle = Mathf.Rad2Deg * Mathf.Atan (angle);
    col.transform.Rotate (0, 0, angle);}
}

```

Kode Sumber 4.13 Intantiate Line Tahap 4

4.4.10. Implementasi Tahap 4 Level 1

Proses ini merupakan proses tahap 4 level 1 yang harus dijalankan oleh pengguna untuk bisa menyelesaikan tahapan terapi gerak tangan. Proses ini berfokus pada pergerakan pergelangan jari-jari tangan pengguna dan keakuratan dari pergelangan jari pengguna. Implementasi proses ini akan memberikan titik acuan yang harus dilewati oleh pergelangan tangan pengguna hingga membentuk bentuk yang telah ditetapkan, pada tahap 4 level 1 ini aplikasi akan memberi petunjuk membuat suatu garis lurus yang menghubungkan 2 bintang berbentuk bola dan bila garis lurus tersebut telah menyentuh kedua bola tersebut maka pengguna telah menyelesaikan tahap 4 level 1 ini. Proses ini menggunakan beberapa *class* yaitu *T4Line* dapat dilihat pada Kode Sumber 4.13, *T4Lltriggercolor*, *T4sphere1* dan *T4sphere2*. *Class T4Lltriggercolor* berfungsi sebagai pengecekan bila dua bola sudah terhubung dengan garis dengan menggunakan *collider* pada garis tersebut. *Class T4sphere1* dan *T4sphere2* merupakan *Class* yang berfungsi untuk mendeteksi *collider* dengan *line* yang dibuat oleh pengguna. Potongan kode untuk tahap 4 level 1 dapat dilihat pada Kode Sumber 4.14, Kode Sumber 4.15, dan Kode Sumber 4.16.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;

public class T4Lltriggercolor : MonoBehaviour {
    public float changecolordelay = 0.01f;
}

```

```

    public GameObject capsule1, capsule2, CanvasNext,
myReward, CanvasInst;
    public Text timer;
    GameObject clone;
    int finalscore;
    float starttime, scorebasedtime;
    bool finished;

    void Start () {
        starttime = Time.time;
        StartCoroutine (waits ());
    }

    IEnumerator waits(){
        yield return new WaitForSeconds (10);

        CanvasInst.transform.GetChild(1).gameObject.SetActive
(false);
    }

    IEnumerator canvasnext(){
        if (finalscore >= 85) {
            CanvasNext.SetActive (true);
            yield return new WaitForSeconds
(0.65f);

            myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
            myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);

            myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
            myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);

            myReward.transform.GetChild
(3).GetComponent<Image> ().enabled = true;
            myReward.transform.GetChild
(3).GetComponent<AudioSource> ().enabled = true;
            yield return new WaitForSeconds
(0.65f);

        } else if (75 <= finalscore && finalscore <
85) {
            CanvasNext.SetActive (true);
            yield return new WaitForSeconds
(0.65f);

```

```

        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else if (65 <= finalscore && finalscore <
75) {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
    }
}

void Update(){
    if (GameObject.Find
("Sphere").GetComponent<T4sphere1> ().isflagOn &&
        GameObject.Find
("Sphere2").GetComponent<T4sphere2> ().isflagOn) {
        string myscore = timer.text;
        PlayerPrefs.SetString
("scene4level1", myscore);
        StartCoroutine (canvasnext());
        finished = true;
    }
    if (finished)
        return;
    float t = Time.time- starttime;
    string minutes = ((int)t / 60).ToString ();
    int intseconds = (int)t % 60;
    string seconds = (t % 60).ToString ("f2");
    timer.text = minutes + ":" + seconds;
    scorebasedtime = Time.timeSinceLevelLoad;
}

```

```

        int temp = (int) scorebasedtime;
        int fullscore = 100;
        finalscore = fullscore - temp;
    }
}

```

Kode Sumber 4.14 Pengecekan Collider Dua Benda Terhadap Instantiate Line Pengguna

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class T4sphere1 : MonoBehaviour {
    public float changecolordelay = 0.01f;
    public bool isflagOn = false;

    void OnTriggerEnter (Collider other){
        if (other.tag == "line") {
            isflagOn = true;
        }
        if (other.tag == "mythumb") {
            Material screenchange =
Resources.Load("myyellowstar", typeof(Material)) as
Material;

            gameObject.GetComponent<Renderer>().material =
screenchange;
        }
    }
    void OnTriggerExit(Collider other){
        if (other.tag == "mythumb") {
            Material screenchange =
Resources.Load("naturalyellow", typeof(Material)) as
Material;

            gameObject.GetComponent<Renderer>().material =
screenchange;}
        }
    }
}

```

Kode Sumber 4.15 Deteksi Collider Pada Sphere Pertama

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class T4sphere2 : MonoBehaviour {
    public float changecolordelay = 0.01f;
    public bool isflagOn = false;

    void OnTriggerEnter (Collider other){
        if (other.tag == "line") {
            isflagOn = true;
        }
        if (other.tag == "mythumb") {
            Material screenchange =
Resources.Load("myyellowstar", typeof(Material)) as
Material;

            gameObject.GetComponent<Renderer>().material =
screenchange;
        }
    }
    void OnTriggerExit(Collider other){
        if (other.tag == "mythumb") {
            Material screenchange =
Resources.Load("naturalyellow", typeof(Material)) as
Material;

            gameObject.GetComponent<Renderer>().material =
screenchange;
        }
    }
}

```

Kode Sumber 4.16 Deteksi Collider Pada Sphere Kedua

4.4.11. Implementasi Tahap 4 Level 2

Proses ini merupakan proses tahap 4 level 2 yang harus dijalankan oleh pengguna untuk bisa menyelesaikan tahapan terapi gerak tangan. Proses ini berfokus pada pergerakan pergelangan jari-jari tangan pengguna dan keakuratan dari pergelangan jari pengguna. Implementasi proses ini akan memberikan titik acuan yang harus dilewati oleh pergelangan tangan pengguna hingga membentuk bentuk yang telah ditetapkan, pada tahap 4 level 2 ini

aplikasi akan memberi petunjuk membuat suatu garis lurus yang menghubungkan 3 bola hingga membentuk segitiga dan bila pengguna telah membuat segitiga yang saling dihubungkan dengan garis maka pengguna telah menyelesaikan tahap 4 level 2 ini. Proses ini menggunakan beberapa *class* yaitu *T4Line* dapat dilihat pada Kode Sumber 4.13, *T4L2triggercolor*, *T4sphere1* dapat dilihat pada Kode Sumber 4.15, *T4sphere2* dapat dilihat pada Kode Sumber 4.16 dan *T4sphere3*. *Class T4L2triggercolor* berfungsi sebagai pengecekan bila ketiga bola sudah terhubung dengan garis dengan menggunakan *collider* pada garis tersebut. *Class T4sphere1*, *T4sphere2* dan *T4sphere3* merupakan *Class* yang berfungsi untuk mendeteksi *collider* dengan *line* yang dibuat oleh pengguna. Potongan kode untuk tahap 4 level 2 dapat dilihat pada Kode Sumber 4.17 dan Kode Sumber 4.18.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;

public class T4L2triggercolor : MonoBehaviour {
    public float changecolordelay = 0.01f;
    public GameObject capsule1, capsule2,
capsule3,CanvasNext, myReward, CanvasInst;
    public Text timer;
    GameObject clone;
    int finalscore,t = 0;
    float starttime,scorebasedtime;
    bool finished;

    void Start(){
        starttime = Time.time;
        StartCoroutine (waits ());
    }

    IEnumerator waits(){
        yield return new WaitForSeconds (10);

        CanvasInst.transform.GetChild(1).gameObject.SetActive
(false);
    }
}
```

```

IEnumerator canvasnext() {
    if (finalscore >= 85) {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(3).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(3).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else if (75 <= finalscore && finalscore <
85) {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else if (65 <= finalscore && finalscore <
75) {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;

```



```

        yield return new WaitForSeconds
(0.65f);
    } else {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
    }
}

void Update(){
    if (!GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().enabledstar) {
        if (GameObject.Find
("Sphere").GetComponent<T4sphere1> ().isflagOn &&
            GameObject.Find
("Sphere2").GetComponent<T4sphere2> ().isflagOn) {
            GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().enabledstar = true;
            GameObject.Find
("Sphere").GetComponent<T4sphere1> ().isflagOn = false;
        }
    }

    else if (GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().enabledstar) {
        if (GameObject.Find
("Sphere").GetComponent<T4sphere1> ().isflagOn &&
            GameObject.Find
("Sphere2").GetComponent<T4sphere2> ().isflagOn &&
            GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().isflagOn) {
            string myscore = timer.text;
            PlayerPrefs.SetString
("scene4level2", myscore);

            StartCoroutine (canvasnext());
            finished = true;
        }
    }

    if (finished)
        return;
    float t = Time.time-starttime;
    string minutes = ((int)t / 60).ToString ();
    int intseconds = (int)t % 60;
    string seconds = (t % 60).ToString ("f2");
    timer.text = minutes + ":" + seconds;
    scorebasedtime = Time.timeSinceLevelLoad;
    int temp = (int) scorebasedtime;

```

```

        int fullscore = 100;
        finalscore = fullscore - temp;
    }
}

```

Kode Sumber 4.17 Pengecekan Collider Tiga Benda Terhadap Instantiate Line Pengguna

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class T4sphere3 : MonoBehaviour {
    public float changecolordelay = 0.01f;
    public bool isflagOn = false;
    public bool enabledstar = false;

    void OnTriggerEnter (Collider other){
        if (other.tag == "line") {
            isflagOn = true;
        }
        if (other.tag == "mythumb") {
            Material screenchange =
Resources.Load("myyellowstar", typeof(Material)) as
Material;

            gameObject.GetComponent<Renderer>().material =
screenchange;
        }
    }
    void OnTriggerExit(Collider other){
        if (other.tag == "mythumb") {
            Material screenchange =
Resources.Load("naturalyellow", typeof(Material)) as
Material;

            gameObject.GetComponent<Renderer>().material =
screenchange;
        }
    }
}

```

Kode Sumber 4.18 Deteksi Collider Pada Sphere Ketiga

4.4.12. Implementasi Tahap 4 Level 3

Proses ini merupakan proses tahap 4 level 3 yang harus dijalankan oleh pengguna untuk bisa menyelesaikan tahapan terapi gerak tangan. Proses ini berfokus pada pergerakan pergelangan jari-jari tangan pengguna dan keakuratan dari pergelangan jari pengguna. Implementasi proses ini akan memberikan titik acuan yang harus dilewati oleh pergelangan tangan pengguna hingga membentuk bentuk yang telah ditetapkan, pada tahap 4 level 3 ini aplikasi akan memberi petunjuk membuat suatu garis lurus yang menghubungkan 4 bola hingga membentuk persegi dan bila pengguna telah membuat persegi yang saling dihubungkan dengan garis maka pengguna telah menyelesaikan tahap 4 level 3 ini. Proses ini menggunakan beberapa *class* yaitu *T4Line* dapat dilihat pada Kode Sumber 4.13, *T4L3triggercolor*, *T4sphere1* dapat dilihat pada Kode Sumber 4.15, *T4sphere2* dapat dilihat pada Kode Sumber 4.16 dan *T4sphere3* dapat dilihat pada Kode Sumber 4.18 dan *T4sphere4*. *Class T4L3triggercolor* berfungsi sebagai pengecekan bila ketiga bola sudah terhubung dengan garis dengan menggunakan *collider* pada garis tersebut. *Class T4sphere1*, *T4sphere2*, *T4sphere3*, dan *T4sphere4* merupakan *Class* yang berfungsi untuk mendeteksi *collider* dengan *line* yang dibuat oleh pengguna. Potongan kode untuk tahap 4 level 3 dapat dilihat pada Kode Sumber 4.19, dan Kode Sumber 4.20.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using System.IO;

public class T4L3triggercolor : MonoBehaviour {
    public float changecolordelay = 0.01f;
    public GameObject capsule1, capsule2, capsule3,
capsule4, CanvasNext, myReward, CanvasInst;
    public Text timer;
    GameObject clone;
    int finalscore,t = 0;
    float scorebasedtime,starttime;
```

```

bool finished;

void Start(){
    starttime = Time.time;
    StartCoroutine (waits ());
}

IEnumerator waits(){
    yield return new WaitForSeconds (10);

    CanvasInst.transform.GetChild(1).gameObject.SetActive
(false);
}

IEnumerator canvasnext(){
    if (finalscore >= 85) {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(3).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(3).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else if (75 <= finalscore && finalscore <
85) {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);

```

```

        myReward.transform.GetChild
(2).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(2).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else if (65 <= finalscore && finalscore <
75) {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
        myReward.transform.GetChild
(1).GetComponent<Image> ().enabled = true;
        myReward.transform.GetChild
(1).GetComponent<AudioSource> ().enabled = true;
        yield return new WaitForSeconds
(0.65f);
    } else {
        CanvasNext.SetActive (true);
        yield return new WaitForSeconds
(0.65f);
    }
}

void Update(){
    if (!GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().enabledstar) {
        if (GameObject.Find
("Sphere").GetComponent<T4sphere1> ().isflagOn &&
        GameObject.Find
("Sphere2").GetComponent<T4sphere2> ().isflagOn) {
            GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().enabledstar = true;
            GameObject.Find
("Sphere4").GetComponent<T4sphere4> ().enabledstar = true;
        }
    }
    else if (GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().enabledstar) {
        if (GameObject.Find
("Sphere").GetComponent<T4sphere1> ().isflagOn &&
        GameObject.Find
("Sphere2").GetComponent<T4sphere2> ().isflagOn &&
        GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().isflagOn &&
        !GameObject.Find
("Sphere4").GetComponent<T4sphere4> ().isflagOn ) {
            GameObject.Find
("Sphere").GetComponent<T4sphere1> ().isflagOn = false;

```

```

    }
    else if(GameObject.Find
("Sphere").GetComponent<T4sphere1> ().isflagOn &&
        GameObject.Find
("Sphere2").GetComponent<T4sphere2> ().isflagOn &&
        GameObject.Find
("Sphere3").GetComponent<T4sphere3> ().isflagOn &&
        GameObject.Find
("Sphere4").GetComponent<T4sphere4> ().isflagOn){
        string myscore = timer.text;
        System.IO.File.WriteAllText
(DateTime.UtcNow.ToString("dd MMMM yyyy Pukul HH mm")
+".txt", "Tahap 1 Level 1 : " + PlayerPrefs.GetString
("scenellevel1") + "\r\n" + "Tahap 1 Level 2 : " +
PlayerPrefs.GetString ("scenellevel2") + "\r\n" +
        "Tahap 1 Level 3 : "
+ PlayerPrefs.GetString ("scenellevel3") + "\r\n" +
        "Tahap 1 Level 4 : "
+ PlayerPrefs.GetString ("scenellevel4") + "\r\n" +
        "Tahap 1 Level 5 : "
+ PlayerPrefs.GetString ("scenellevel5") + "\r\n" +
        "Tahap 2 : "
+ PlayerPrefs.GetString ("scene2") + "\r\n" +
        "Tahap 3 : "
+ PlayerPrefs.GetString ("scene3") + "\r\n" +
        "Tahap 4 Level 1 : "
+ PlayerPrefs.GetString ("scene4level1") + "\r\n" +
        "Tahap 4 Level 2 : "
+ PlayerPrefs.GetString ("scene4level2") + "\r\n" +
        "Tahap 4 Level 3 : "
+ myscore);

        StartCoroutine (canvasnext());
        finished = true;}
    }
    if (finished)
        return;
    float t = Time.time-starttime;
    string minutes = ((int)t / 60).ToString ();
    int intseconds = (int)t % 60;
    string seconds = (t % 60).ToString ("f2");
    timer.text = minutes + ":" + seconds;
    scorebasedtime = Time.timeSinceLevelLoad;
    int temp = (int) scorebasedtime;
    int fullscore = 100;
    finalscore = fullscore - temp;}
}

```

**Kode Sumber 4.19 Pengecekan Collider Empat Benda Terhadap
Instantiate Line Pengguna**

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

public class T4sphere4 : MonoBehaviour {
    public float changecolordelay = 0.01f;
    public bool isflagOn = false;
    public bool enabledstar = false;

    void OnTriggerEnter (Collider other){
        if (other.tag == "line") {
            isflagOn = true;
        }
        if (other.tag == "mythumb") {
            Material screenchange =
Resources.Load("myyellowstar", typeof(Material)) as
Material;

            gameObject.GetComponent<Renderer>().material =
screenchange;
        }
    }
    void OnTriggerExit(Collider other){
        if (other.tag == "mythumb") {
            Material screenchange =
Resources.Load("naturalyellow", typeof(Material)) as
Material;

            gameObject.GetComponent<Renderer>().material =
screenchange;
        }
    }
}

```

Kode Sumber 4.20 Deteksi Collider Pada Sphere Keempat

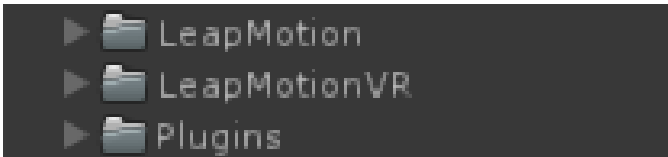
4.5. Implementasi Integrasi Leap Motion dengan Unity

Implementasi integrasi Leap Motion dengan Unity memerlukan suatu penghubung, dalam hal ini adalah SDK Leap Motion. SDK Leap Motion dapat diunduh melalui banyak situs. Salah satu situs yang menyediakan SDK Leap Motion adalah situs

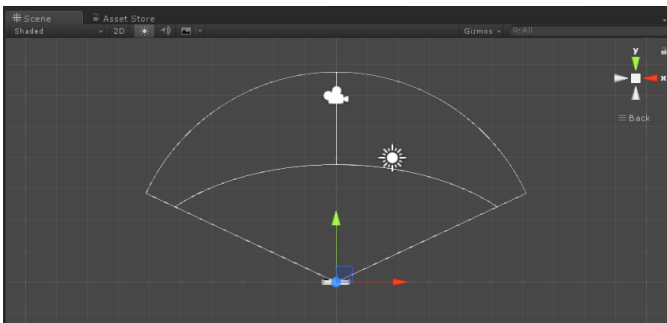
resmi Leap Motion yaitu <https://developer.leapmotion.com/>. Hasil unduh dari SDK ini berupa sebuah *package* untuk mengintegrasikan Leap Motion dengan Unity. *Package* tersebut kemudian diimpor kedalam Unity. Setelah proses impor berhasil, maka akan muncul empat buah folder baru pada bagian *Assets* di Unity. Hasil proses tersebut dapat dilihat pada Gambar 4.23.

Tiga folder baru tersebut adalah folder Leap Motion, Leap Motion VR, dan *Plugins*. Folder Leap Motion berisikan asset-asset yang digunakan untuk membuat aplikasi atau game dengan memanfaatkan Leap Motion. Folder leap Motion VR berisikan asset-asset untuk membangun aplikasi *virtual reality* sedangkan folder *Plugins* berisi pengaturan yang ada pada Leap Motion.

Untuk implementasi sensor area Leap Motion, maka digunakan *Hand Detection Scene* yang ada di dalam folder Leap Motion. *Scene* tersebut dimasukkan ke dalam *scene* Unity. Hasil proses tersebut dapat dilihat pada Gambar 4.24.



Gambar 4.23 Hasil Proses Impor *Package* Leap Motion



Gambar 4.24 Implementasi Input Area Leap Motion

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem dan kegunaan sistem. Pengujian fungsionalitas mengacu pada kasus pada Bab III. Pengujian kegunaan program dilakukan dengan mengetahui tanggapan dari pengguna terhadap sistem. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kaku seperti yang tertera pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Sistem

Perangkat Keras	<ul style="list-style-type: none">- Prosesor Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz- Memori 8 GB- Monitor- Leap motion Controller
Perangkat Lunak	<ul style="list-style-type: none">- Sistem Operasi Microsoft Windows 8.1 Enterprise 64-bit- Blender 2.7.7- Unity 5.5.1f1- Leap Motion SDK- Adobe Photoshop CC- Adobe Premiere CC

5.2. Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk mengetahui kesesuaian keluaran dari tiap tahap atau langkah penggunaan fitur terhadap skenario yang dipersiapkan. Pengujian fungsionalitas aplikasi akan

dilakukan oleh penulis. Selain itu, pada Subbab 5.3 pengujian telah dilakukan pada penderita pasca stroke yang bertujuan untuk menguji bahwa aplikasi ini dapat digunakan sebagai aplikasi terapi anggota gerak tangan. Pengujian dilakukan oleh penderita pascastroke berjenis kelamin laki-laki berumur 60 tahun. Pengujian juga dilakukan oleh seorang ahli fisioterapi, kritik serta pendapat beliau dapat dilihat dalam subbab 6.2 yang berisi kritik dan saran pengembangan aplikasi kedepan.

5.2.1. Skenario Pengujian Fungsionalitas

Pada subbab ini dijelaskan beberapa skenario uji coba perangkat lunak secara mandiri berdasarkan metode kotak hitam sebagai dasar tolok ukur keberhasilan. Pengujian fungsionalitas yang terdapat pada aplikasi dijabarkan sebagai berikut :

- a. Uji coba melihat penjelasan aplikasi
- b. Uji coba latihan terapi
- c. Uji coba melihat hasil terapi

Daftar uji coba tersebut merupakan pengujian yang dilakukan untuk menguji setiap kasus penggunaan pada perangkat lunak yang dibangun. Dibuat beberapa skenario yang dilakukan pada setiap daftar pengujian tersebut. Penjelasan mengenai cara dan hasil pengujian fungsionalitas perangkat lunak dibahas pada subbab hasil uji coba.

5.2.2. Hasil Uji Coba Fungsionalitas

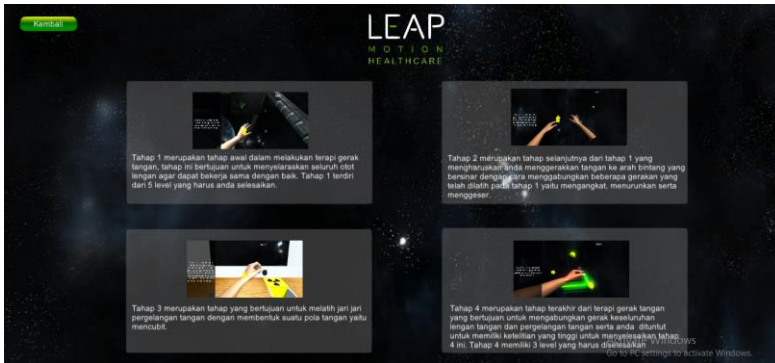
Pada subbab ini dijelaskan secara detail mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas perangkat lunak yang dibangun. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran, hasil yang dicapai, dan kondisi akhir. Berikut ini merupakan penjabaran skenario dan hasil pengujian yang dicapai pada tiap-tiap fungsionalitas perangkat lunak.

5.2.2.1. Uji Coba Melihat Penjelasan Aplikasi

Uji coba melihat penjelasan aplikasi ini berfungsi untuk mengetahui keberhasilan aplikasi dalam menjalankan skenario menampilkan penjelasan tiap-tiap tahap. Hasil dari pengujian tersebut secara lengkap dapat dilihat pada Tabel 5.2. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.1

Tabel 5.2. Hasil Uji Coba Melihat Penjelasan Aplikasi

ID	UJ-P-01
Nama	Uji Coba Melihat Penjelasan Aplikasi
Tujuan Uji Coba	Pengguna dan Admin dapat melihat penjelasan dan kegunaan setiap tahap pada aplikasi sebelum melakukan terapi pada pengguna
Kondisi awal	Pengguna sudah masuk dalam menu utama
Skenario 1	Admin selaku pendamping pengguna menekan tombol “Pelajari Selengkapnya”
Keluaran yang diharapkan	Sistem akan mengeluarkan halaman yang berisi penjelasan dan tujuan setiap tahapan dalam terapi.
Hasil uji coba	Berhasil
Kekurangan	Menurut Tabel 5.9 Kritik dan Saran Penggunamasih perlu penjelasan lebih detail dari setiap level yang ada
Kondisi akhir	Sistem berhasil mengeluarkan halaman yang berisi penjelasan dan tujuan setiap tahapan dalam terapi.



Gambar 5.1 Uji Coba Saat Melihat Penjelasan Aplikasi

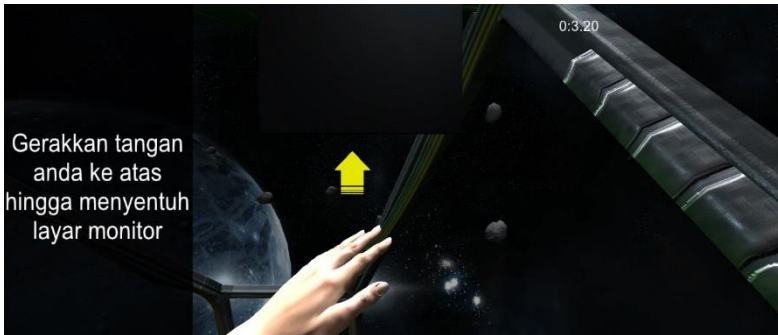
5.2.2.2. Uji Coba Latihan Terapi

Uji coba latihan terapi ini berfungsi untuk mengetahui keberhasilan aplikasi dalam menjalankan terapi anggota gerak tangan untuk pengguna sebagai pasien pascastroke. Hasil dari pengujian tersebut secara lengkap dapat dilihat pada Tabel 5.3. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.2 dan Gambar 5.3.

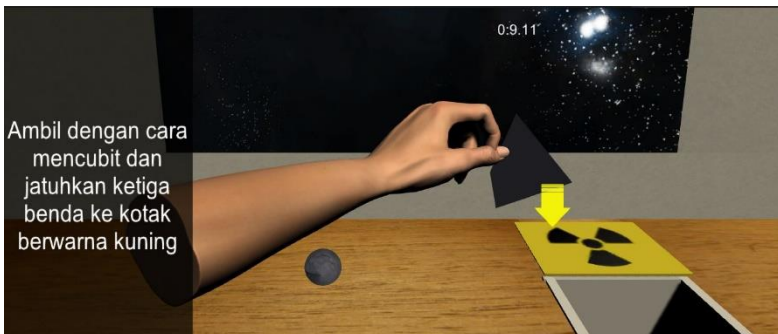
Tabel 5.3 Hasil Uji Coba Latihan Terapi

ID	UJ-P-02
Nama	Uji Coba Latihan Terapi
Tujuan Uji Coba	Mencoba keberhasilan aplikasi dalam memberikan pelatihan pada pengguna sebagai pasien pasca stroke
Kondisi awal	Admin telah memilih tombol “mulai” untuk memulai dan admin harus memastikan alat tambahan sudah terpasang yaitu Leap Motion Controller
Skenario 1	Pengguna sebagai pasien melakukan gerak pada tangan sesuai instruksi pada aplikasi dan melewati tahap 1 sampai 4 pada aplikasi

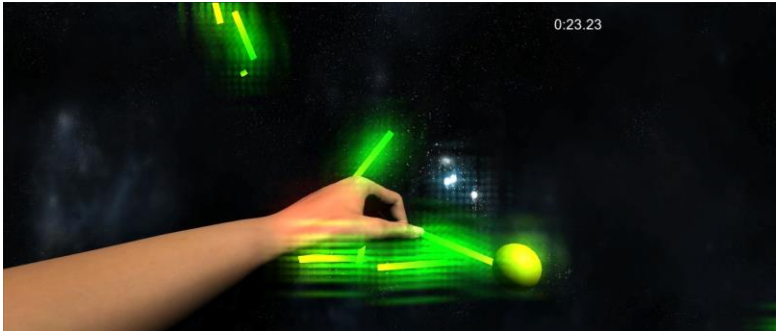
Keluaran yang diharapkan	Sistem akan memberikan instuksi kepada pengguna dan memberikan <i>feedback</i> dari setiap gerakan tangan pengguna dan dapat menjalankan sistem sampai tahap 4 dengan baik
Hasil uji coba	Berhasil
Kekurangan	Tangkapan gerak yang terkadang tidak sesuai dengan gerak tangan yang asli
Kondisi akhir	Sistem berhasil memberikan instuksi kepada pengguna dan memberikan <i>feedback</i> dari setiap gerakan tangan pengguna dan dapat menjalankan sistem sampai tahap 4 dengan baik



Gambar 5.2 Uji Coba Aplikasi Saat Tahap 1 Level 1



Gambar 5.3 Uji Coba Aplikasi Saat Tahap 3



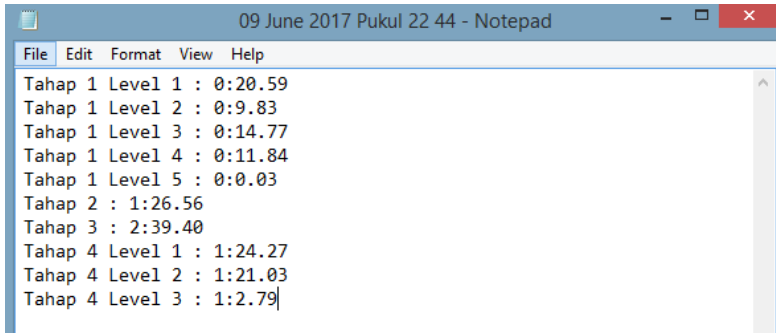
Gambar 5.4 Uji Coba Aplikasi Saat Tahap 4 Level 1

5.2.2.3. Uji Coba Melihat Hasil Terapi

Uji coba melihat terapi berfungsi untuk menguji keberhasilan aplikasi dalam menunjukkan hasil terapi. Uji coba ini dapat dilakukan pada akhir tahap 4. Hasil dari pengujian tersebut secara lengkap dapat dilihat pada Tabel 5.4. Hasil dari skenario uji coba tersebut dapat dilihat pada Gambar 5.5.

Tabel 5.4 Uji Coba Melihat Hasil Terapi

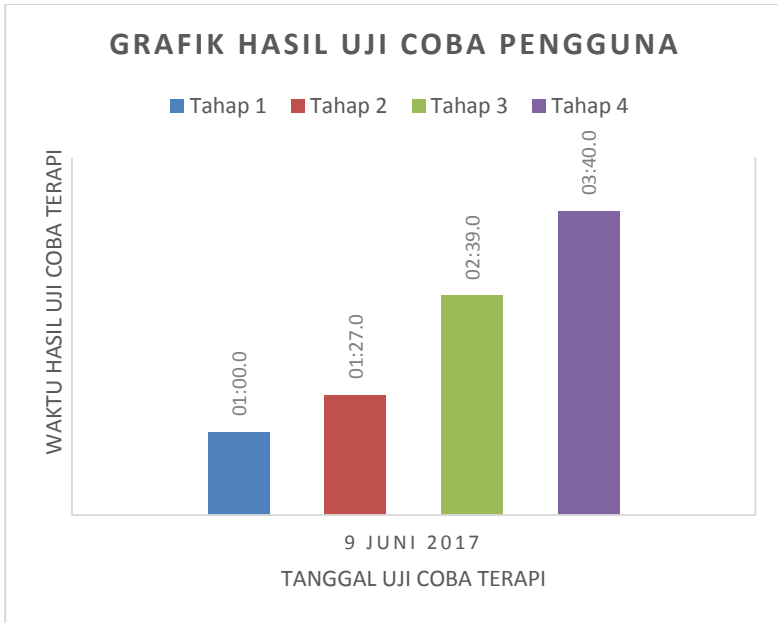
ID	UJ-P-03
Nama	Uji Coba Melihat Hasil Terapi.
Tujuan Uji Coba	Pengguna dan admin dapat melihat hasil dari terapi tiap tahap yang telah dilalui
Kondisi awal	Pengguna berada pada tahap 4 terapi
Skenario 1	Pengguna menyelesaikan tahap 4 terapi
Keluaran yang diharapkan	Sistem akan mengeluarkan waktu yang dicatat dalam bentuk teks yang terdapat pada folder aplikasi
Hasil uji coba	Berhasil.
Kekurangan	-
Kondisi akhir	Sistem berhasil mengeluarkan waktu yang dicatat dalam bentuk teks yang terdapat pada folder aplikasi



Gambar 5.5 Uji Coba Saat Pengguna Dalam Tahap 1

5.3. Pengujian Pengguna

Pengujian aplikasi pada penderita pascastroke bertujuan untuk mengukur apakah aplikasi ini memang dapat digunakan untuk pelatihan gerak penderita pasca stroke. Dalam pengujian ini pasien berjenis kelamin laki-laki dengan umur 60 tahun. Memiliki kekuatan otot yang tergolong normal akan tetapi memiliki keterbatasan gerak. Pengguna melakukan terapi dimulai dari tahap 1 sampai 4 dengan didampingi oleh admin. Hasil pengujian terapi dapat dilihat pada Gambar 5.6 serta proses pengujian setiap tahap dapat dilihat pada Gambar 5.7, Gambar 5.8, Gambar 5.9, dan Gambar 5.10.



Gambar 5.6 Grafik Hasil Uji Coba Pengguna



Gambar 5.7 Pengujian Pengguna Tahap 2



Gambar 5.8 Pengujian Pengguna Tahap 3



Gambar 5.9 Pengujian Pengguna Tahap 4 Level 2



Gambar 5.10 Pengujian Pengguna Tahap 4 Level 3

5.3.1. Kuisiner Pengguna Dalam Pengujian Aplikasi

Dalam melakukan pengujian perangkat lunak, pengguna dan admin diminta mencoba menggunakan perangkat lunak untuk mencoba semua fungsionalitas dan fitur yang ada. Pengujian aplikasi oleh pengguna dilakukan dengan sebelumnya memberikan informasi seputar aplikasi, kegunaan, dan fitur-fitur yang dimiliki. Setelah informasi tersampaikan, pengguna dan admin kemudian diarahkan untuk langsung mencoba aplikasi dengan spesifikasi lingkungan yang sama dengan yang telah diuraikan pada Lingkungan Pengujian.

Jumlah pengguna dan admin yang terlibat dalam pengujian perangkat lunak sebanyak dua orang yang terdiri dari satu admin dan satu pengguna. Dalam memberikan penilaian dan tanggapan, pengguna dan admin diberikan kuisiner pengujian perangkat lunak. Kuisiner pengujian perangkat lunak ini memiliki beberapa aspek penilaian seputar desain antarmuka, kenyamanan, dan tingkat kegunaan aplikasi. Nilai yang diberikan memiliki rentang nilai 1 hingga 6 dengan rincian pada Tabel 5.5. Pada bagian akhir terdapat saran untuk perbaikan fitur. Detail kuisiner pengguna

dapat dilihat pada Tabel 5.6 dan hasil kuisisioner secara keseluruhan dalam persentase dapat dilihat pada Tabel 5.8.

Tabel 5.5 Rentang Nilai

Nilai	Keterangan
1	Sangat Tidak Setuju (STS)
2	Tidak Setuju (TS)
3	Kurang Setuju (KS)
4	Cukup Setuju (CS)
5	Setuju (S)
6	Sangat Setuju (SS)

Tabel 5.6 Kuesioner Pengguna

No	Parameter Antarmuka	Target	Nilai (1-6)
1	Aplikasi memiliki tampilan, warna dan desain yang menarik	Pengguna	
2	Aplikasi mudah dipahami dan dioperasikan	Admin	
3	Aplikasi memiliki objek permainan yang menarik	Pengguna	
No	Parameter Kenyamanan	Target	Nilai (1-6)
4	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	Pengguna	
5	Saya merasa terbantu dengan adanya tutorial yang disediakan	Admin	
6	Saya merasa tertarik untuk menggunakan aplikasi ini	Pengguna	
No	Parameter Kegunaan	Target	Nilai (1-6)
7	Saya merasa terbantu dengan adanya aplikasi ini sebagai salah satu media terapi	Pengguna	
8	Aplikasi dapat memberikan macam gerak terapi yang tepat dan benar	Pengguna	

5.3.2. Hasil Pengujian Pengguna

Kuisisioner uji coba yang dilakukan terhadap pengguna dan admin memiliki beberapa aspek yang dipisahkan berdasarkan antarmuka, kenyamanan dan tingkat kegunaan. Sistem penilaian didasarkan pada skala penghitungan satu sampai enam dimana skala satu menunjukkan nilai terendah dan skala enam menunjukkan skala tertinggi. Hasil uji coba dipaparkan secara lengkap dengan disertai tabel yang dapat dilihat pada Tabel 5.7 serta hasil uji coba dalam bentuk persentase dapat dilihat pada Tabel 5.8.

Tabel 5.7 Hasil Pengujian Pengguna Berumur 60

No	Parameter Antarmuka	Target	Nilai (1-6)
1	Aplikasi memiliki tampilan, warna dan desain yang menarik	Pengguna	4
2	Aplikasi mudah dipahami dan dioperasikan	Admin	5
3	Aplikasi memiliki objek permainan yang menarik	Pengguna	5
No	Parameter Kenyamanan	Target	Nilai (1-6)
4	Aplikasi dapat berjalan lancar tanpa adanya <i>lag</i> dan <i>crash</i>	Pengguna	4
5	Saya merasa terbantu dengan adanya tutorial yang disediakan	Admin	5
6	Saya merasa tertarik untuk menggunakan aplikasi ini	Pengguna	5
No	Parameter Kegunaan	Target	Nilai (1-6)
7	Saya merasa terbantu dengan adanya aplikasi ini sebagai salah satu media terapi	Pengguna	5
8	Aplikasi dapat memberikan macam gerak terapi yang tepat dan benar	Pengguna	5

Tabel 5.8 Hasil Keseluruhan Pengujian Pengguna Dalam Persenentase

No	Parameter	Nilai Total	Persentase
1	Antarmuka	14	77,77%
2	Kenyamanan	14	77,77%
3	Kegunaan	10	83,33%

5.3.3. Kritik dan Saran Pengguna

Dalam memberikan penilaian dan tanggapan, pengguna sekaligus admin diberikan kuesioner pengujian perangkat lunak. Kuesioner pengujian perangkat lunak ini terdapat bagian kritik dan saran untuk perbaikan fitur kedepannya. Kritik dan saran pengguna dapat dilihat pada Tabel 5.9.

Tabel 5.9 Kritik dan Saran Pengguna

No	User	Kritik dan Saran
1	Penderita pasca stroke berusia 60 tahun	Memperbaiki tampilan agar lebih menarik
2	Admin	Memperbaiki tampilan agar lebih menarik

5.4. Konsultasi Dengan Ahli Fisioterapi

Konsultasi dengan ahli bersama Niniek Soetini, M.Fis di klinik fisioterapi pada alamat Jl. Mulyosari Timur No.69, Kalisari, Mulyorejo, Kota SBY, Jawa Timur 60112. Konsultasi dilakukan dengan melakukan wawancara mengenai dasar gerakan yang diterapkan pada aplikasi. Kesimpulan yang diambil dari wawancara dengan ahli yaitu aplikasi telah memiliki gerakan serta tahapan yang baik dimulai dari gerakan dasar hingga gerakan yang fungsional dalam kehidupan sehari-hari. Akan tetapi terdapat kekurangan pada aplikasi yaitu sulitnya lokasi bintang pada tahap 2, perlakukan khusus tiap benda seperti memegang bola dan kotak,

dan terapis tersebut memberikan saran untuk pengembangan selanjutnya memerlukan tambahan gerakan yang melibatkan aktivitas sehari-hari seperti makan, mandi, merapikan rambut, dan menggosok gigi. Gerakan fungsional juga sebaiknya didasari dengan prinsip PNF (*Proprioceptif Neuromuscular Facilitation*).

5.5. Evaluasi Pengujian

Berdasarkan hasil pengujian aplikasi, dan pengujian pengguna yang telah dijelaskan pada subbab sebelumnya, maka didapatkan evaluasi sebagai berikut:

1. Aplikasi berhasil dalam memberikan penjelasan tiap tiap tahap pada bagian “Pelajari Selengkapnya”. Hal ini dibuktikan pada pengujian UJ-P-01.
2. Aplikasi berhasil melakukan serangkaian tahapan yang dapat menjadi salah satu media bagi pengguna untuk melakukan terapi anggota gerak tangan dalam masa pascastroke untuk memulihkan kembali fungsi-fungsi otot. Hal ini dibuktikan pada pengujian UJ-P-02.
3. Aplikasi berhasil memberikan hasil di tiap tahap yang telah dilalui oleh pengguna sebagai nilai acuan perkembangan otot pengguna. Hal ini dibuktikan pada pengujian UJ-P-03.
4. Desain antarmuka aplikasi sudah sangat baik, hal ini terbukti dari presentase hasil penilaian antarmuka yang dapat mencapai 77,77%
5. Tingkat kenyamanan aplikasi ini masih kurang yang disebabkan oleh terdapat *frame drop* pada selang waktu tertentu ketika terapi sedang berjalan dan tangan virtual yang terkadang tidak tampil yang mengharuskan pengguna menarik tangannya kembali keluar dari area tangkapan LMC dan memasukinya lagi. Akan tetapi hal tersebut jarang terjadi, hal tersebut terjadi apabila komputer yang digunakan menjalankan beberapa aplikasi lain yang membebankan proses kerja komputer. Tetapi dari segi tutorial yang mudah serta jaranganya *lag* maka tingkat kenyamanan dapat mencapai 77,77%.

6. Tingkat kegunaan aplikasi ini cukup berguna bagi pasien penderita pasca stroke sebagai salah satu media terapi, akan tetapi masih perlu penambahan gerak dalam terapi. Tingkat kenyamanan dapat mencapai 83,33% dan hal ini dapat dibuktikan dalam Tabel 5.9 Kritik dan Saran Pengguna.

[Halaman ini sengaja dikosongkan

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil selama proses perancangan, implementasi, serta pengujian dapat diambil kesimpulan sebagai berikut :

1. Aplikasi telah berhasil menciptakan sebuah media baru untuk melakukan latihan terapi pasca stroke khusus pada organ gerak tangan.
2. Simulasi menggambar bentuk benda pada terapi tahap 4 masih kurang sempurna karena LMC tidak bisa menangkap setiap gerakan secara lebih detail, masih terdapat gerakan gerakan yang tidak sesuai dengan tangan pengguna pada dunia nyata seperti munculnya *tremor* kecil pada tangan virtual dan beberapa jari yang tidak dapat mendeteksi bentuk jari pengguna secara sempurna.

6.2. Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Diantaranya adalah sebagai berikut:

1. Menambah macam gerak yang memiliki fungsionalitas agar pengguna dapat melakukan terapi tangan lebih efektif.
2. Memperbaiki tampilan agar lebih menarik bagi pengguna serta lebih mudah digunakan.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

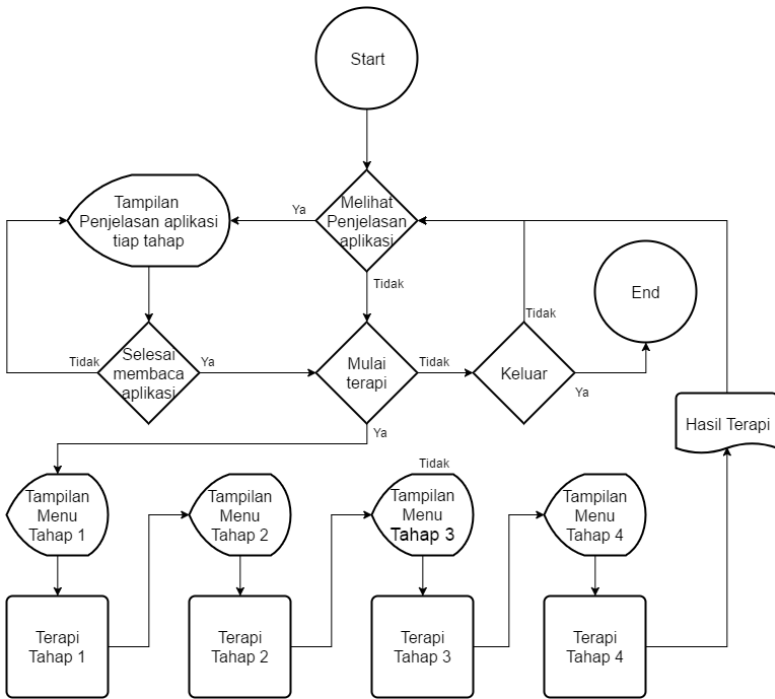
- [1] E. S. Silva, J. A. O. d. Abreu, J. H. P. d. Almeida, V. Teichrieb and G. L. Ramalho, "A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments," in *14^o Simpósio Brasileiro de Computação*, Brasília, Sociedade Brasileira de Computação – SBC, 2013, pp. 59-70.
- [2] Wikipedia, "Stroke," Wikipedia Bahasa Indonesia, [Online]. Available:
<https://id.wikipedia.org/w/index.php?title=Stroke&oldid=12066095> . [Accessed 26 November 2016].
- [3] I. Sommerville, *Software Engineering* (9th ed.), Harlow, England: Pearson Education, 2010.
- [4] Sahlasaida, "Enam Terapi Dasar Pemulihan Pasca Stroke," [Online]. Available: <http://tipkesehatan.com/2015/05/enam-terapi-dasar-pemulihan-pasca-stroke/>. [Accessed 08 06 2017].
- [5] R.S.J.Carr, *Motor Assessment Scale for Stroke*, 2nd, pp.175-180 ed., Physical Therapy, 1994, p. 65.
- [6] W. Khotimah and R. W.Sholikah, "Sitting To Standing and Walking Therapy for Post-Stroke Patients using Virtual Reality System," Surabaya, 2015.
- [7] MahendraIkkhuGincung, "bahan_SAP_senaam_stroke," [Online]. Available:
<https://www.scribd.com/document/349210268/bahan-SAP-senaam-stroke-1-docx>. [Accessed 3 July 2017].
- [8] I. Farida and N. Amalia, *Mengantisipasi stroke petunjuk mudah lengkap dan praktis sehari hari*, Jogjakarta: Buku Biru, 2009.

- [9] R. S. JIH, "Fisioteraphy | Rumah Sakit JIH," [Online]. Available: <https://www.rs-jih.co.id/layanan/fisioteraphy>. [Accessed 21 Desember 2016].
- [10] "Introduction to Unity 3D," Code Envato Tuts+, [Online]. Available: <https://code.tutsplus.com/tutorials/introduction-to-unity3d--mobile-10752>. [Accessed 21 Desember 2016].
- [11] G. Pierce, "Unity IOS Game Development Beginners Guide," Packt Publishing Ltd, 2012. [Online]. [Accessed 21 Desember 2016].
- [12] V. Llopis, "Introducing the Skeletal Tracking Model," The Motion Report, 6 May 2013. [Online]. Available: <http://themotionreport.com/evolution-of-leap-motion-controller/>. [Accessed 25 May 2016].
- [13] L. Motion, "Leap Motion," Leap Motion, [Online]. Available: <https://www.leapmotion.com/product/desktop>. [Accessed 21 Desember 2016].
- [14] A. Colgan, "Introducing the Skeletal Tracking Model," Leap Motion, Inc, 9 August 2014. [Online]. Available: <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>. [Accessed 25 May 2016].
- [15] L. Motion, "Leap Motion Developers," Leap Motion, [Online]. Available: <https://developer.leapmotion.com/documentation/index.html?proglang=current>. [Accessed 21 Desember 2016].
- [16] L. Motion, "Leap Motion API Classes," Leap Motion, Inc, 21 January 2012. [Online]. Available: https://developer.leapmotion.com/documentation/v2/unity/api/Leap_Classes.html#leap-motion-api-classes. [Accessed 25 May 2016].
- [17] Blender, "about-blender," Blender, [Online]. Available: <http://www.blender.org/about/>. [Accessed 10 april 2014].



- [18] E. Handayani, U. L. Yuhana and R. R. Haridi, RANCANG BANGUN APLIKASI TERAPI REHABILITASI GERAK PENDERITA PASCASTROKE DENGAN MENGGUNAKAN TEKNOLOGI KINECT, Surabaya: Jurusan Teknik Informatika-ITS, 2014.
- [19] N. A. Fitriani, U. L. Yuhana and R. R. Haridi, Perancangan dan implementasi augmented reality sebagai alternatif terapi tangan untuk penderita pasca stroke, Surabaya: Jurusan Teknik Informatika-ITS, 2013.
- [20] D. P. McNulty, "Improving Rehabilitation After Stroke," NeuRA, [Online]. Available: <https://www.neura.edu.au/project/improving-rehabilitation-stroke/>. [Accessed 14 juni 2017].

[Halaman ini sengaja dikosongkan]

LAMPIRAN



Gambar A.0.1 Flowchart Rancangan Proses Aplikasi

Rancang Bangun Aplikasi Terapi Pasca Stroke Untuk Latihan Pergerakan Jari Tangan Dengan Menggunakan Leap Motion Controller

Nama : Kuswita
 Umur : 60
 Pekerjaan : -

Silahkan lingkari (O) pada pilihan yang sesuai

1	Apakah anda pernah menggunakan Leap Motion Controller sebelumnya ?	a. Ya	<input type="radio"/> b.) Tidak
2	Apakah anda pernah melakukan terapi stroke secara virtual seperti pada aplikasi ini sebelumnya ?	a. Ya	<input type="radio"/> b.) Tidak
3	Apakah anda pernah memainkan game simulasi sebelumnya ?	a. Ya	<input type="radio"/> b.) Tidak

Kuesioner Tugas Akhir:

Berikut ini silahkan centang (v) di kolom yang sesuai
 1: Sangat Tidak Setuju, 2: Tidak Setuju, 3: Kurang Setuju, 4 : Cukup Setuju, 5: Setuju, 6: Sangat Setuju

No	Parameter Antarmuka	Target	Nilai					
			1	2	3	4	5	6
1	Aplikasi memiliki tampilan, warna dan desain yang menarik	Pengguna				✓		
2	Aplikasi mudah dipahami dan diooperasikan	Admin					✓	
3	Aplikasi memiliki objek permainan yang menarik	Pengguna						✓
Parameter Kenyamanan		Target	Nilai					
No			1	2	3	4	5	6
4	Aplikasi dapat berjalan lancar tanpa adanya lag dan crash	Pengguna				✓		
5	Saya merasa terbantu dengan adanya tutorial yang disediakan	Admin					✓	
6	Saya merasa tertarik untuk menggunakan aplikasi ini	Pengguna						✓
Parameter Kegunaan		Target	Nilai					
No			1	2	3	4	5	6
7	Saya merasa terbantu dengan adanya aplikasi ini sebagai salah satu media terapi	Pengguna					✓	
8	Aplikasi dapat memberikan macam gerak terapi yang tepat dan benar	Pengguna						✓

Gambar A.0.2 Kuesioner Responden Lembar 1



Usability Testing:

Berikut ini isi nilai pencapaian pada aktifitas yang dijalankan

No	Tahap 1 Terapi	Waktu Pencapaian
1	Level 1	0:20.59
2	Level 2	0:9.83
3	Level 3	0:14.77
4	Level 4	0:11.39
5	Level 5	0:0.03
No	Tahap 2 Terapi	Waktu Pencapaian
6	Level 1	1:26.56
No	Tahap 1 Terapi	Waktu Pencapaian
7	Level 1	2:39.40
No	Tahap 1 Terapi	Waktu Pencapaian
8	Level 1	1:24.27
9	Level 2	1:21.05
10	Level 3	1:2.79

Kritik dan Saran untuk pengembangan selanjutnya

Tambah grafik

Surabaya, 9 Juni 2017

Gambar A.0.3 Kuesioner Responden Lembar 2

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, Franky Setiawan Daldiri, lahir di kota Mojokerto pada tanggal 28 November 1994. Penulis dibesarkan di kota Mojokerto, Jawa Timur.

Penulis menempuh pendidikan formal di TK Wijaya Sejati Mojokerto (1999-2001), SD Taruna Nusa Harapan Mojokerto (2001-2007), SMP Taruna Nusa Harapan Mojokerto (2007-2010), SMA Taruna Nusa Harapan Mojokerto (2010-2013). Pada tahun 2013, penulis melanjutkan pendidikan S1 jurusan Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur.

Di jurusan Teknik Informatika, penulis mengambil bidang minat Interaksi Grafika dan Seni atau biasa disingkat menjadi IGS dan memiliki ketertarikan di bidang Website, Manajemen Basis Data, Augmented Reality, Realitas Virtual, dan pembuatan Game. Penulis aktif sebagai staf Departemen Kesejahteraan Mahasiswa Himpunan Mahasiswa Teknik Komputer periode kepengurusan 2014 serta kegiatan organisasi dan event mahasiswa lainnya. Penulis dapat dihubungi melalui alamat email setiawan.franky94@gmail.com