

**PERANCANGAN SIMULASI *VIRTUAL TOUR* PERPUSTAKAAN UIN
SUNAN KALIJAGA MENGGUNAKAN *UNITY ENGINE***

Skripsi

Untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1
Program Studi Teknik Informatika



disusun oleh:

Ahmad Nur Kholiq

10650004

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2016



PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/2812/2016

Skripsi/Tugas Akhir dengan judul : Perancangan Simulasi *Virtual Tour* Perpustakaan UIN Sunan Kalijaga Menggunakan *Unity Engine*

HALAMAN PENGESAHAN

Yang dipersiapkan dan disusun oleh :
Nama : Ahmad Nur Kholiq
NIM : 10650004
Telah dimunaqasyahkan pada : Kamis, 11 Agustus 2016
Nilai Munaqasyah : A -
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Sumarsono, M.Kom
NIP. 19710209 200501 1 003

Penguji I

M. Didik R. Wahyudi, M.T
NIP.19760812 200901 1 015

Penguji II

Nurochman, M.Kom
NIP. 19801223 200901 1 007

Yogyakarta, 16 Agustus 2016
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan



Dr. Murtono, M.Si.
NIP.19641212 200003 1 001



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal :

Lamp :

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Ahrnad Nur Kholiq

NIM : 10650004

Judul Skripsi : Perancangan Simulasi Virtual Tour Perpustakaan Uin Sunan Kalijaga
Menggunakan Unity Engine.

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Teknik Informatika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 5 Agustus 2016

Pembimbing

Sumarsono, S.T., M.Kom

NIP. 19710209 200501 1 003

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Ahmad Nur Kholiq
Nim : 10650004
Program Studi : Teknik Informatika
Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi dengan judul **Perancangan Simulasi *Virtual Tour* Perpustakaan UIN Sunan Kalijaga Menggunakan *Unity Engine*** tidak terdapat pada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi, dan sepengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

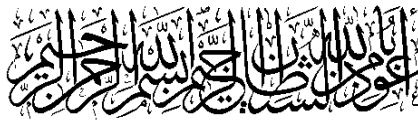
Yogyakarta, 5 Agustus 2016

Yang Menyatakan,



Ahmad Nur Kholiq
NIM : 10650004

KATA PENGANTAR



Alhamdulillah, segala puji bagi Allah, Tuhan semesta alam, tiada daya dan upaya kecuali Allah yang Maha Esa. Puji syukur kehadiran Allah yang Maha Pengasih lagi Maha Penyayang, sholawat serta salam semoga tercurah pada junjungan Nabi Muhammad SAW. Alhamdulillah dengan kasih sayang dan petunjuk-Nya, penulis dapat menyelesaikan penelitian yang berjudul *“Perancangan Simulasi Virtual Tour Perpustakaan Uin Sunan Kalijaga Menggunakan Unity Engine”* sebagai salah satu syarat untuk mencapai gelar sarjana pada program studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta

Tidak lupa penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu proses pengerjaan penelitian ini sehingga laporan dari penelitian ini dapat terselesaikan, diantaranya:

1. Orang tua tercinta, atas segala daya pun upaya yang telah tercurah tanpa mengenal kata lelah.
2. Prof. Drs. KH Yudian Wahyudi, Ph.D selaku Rektor UIN Sunan Kalijaga Yogyakarta.
3. Dr. Murtono, M.Si selaku Dekan Fakultas Sains & Teknologi UIN Sunan Kalijaga
4. Bapak Sumarsono, S.T., M.Kom, selaku Ketua Program Studi Teknik Informatika UIN Sunan Kalijaga serta Dosen Pembimbing yang telah

senantiasa meluangkan waktu untuk memberikan arahan dan bimbingan selama penelitian.

5. Bapak Nurrochman, M.Kom selaku Dosen pembimbing akademik yang senantiasa ada, mengawal serta mengingatkan selama jenjang perkuliahan.
6. Segenap dosen Teknik Informatika UIN Sunan Kalijaga Yogyakarta yang senantiasa mendukung serta memfasilitasi penyelesaiannya skripsi.
7. Mami Lestari & Papi Andi, Serta keluarga kecil Basecamp: Alex, Fahmi, Bobo, April, Rasyid, Dipo, Mujib, Samson, Fajar, Hana, Idus, Agus, Ndut, Wawa dkk.
8. Keluarga Monster2010, tiada yang lebih mencerahkan dari senyuman kalian, dan maaf ya saya lulus belakangan.
9. Keluarga #80KP45 Semaken: Bapak Sukari & Ibuk, Kikik, Opal, Fati, Kholis, Mima, Uus, Nissa, Yanu, Mami Siti, Dewi, Ai. Jangan lupa sebuah sapa pada pertemuan-pertemuan selanjutnya.
10. Untukmu, terima kasih sudah bersedia menjadi rumah, ketika hati pun raga mulai lelah.

Masih begitu banyak kekurangan dalam penulisan laporan ini, Oleh karena itu, kritik dan saran senantiasa penulis harapkan. Dan akhirnya, semoga laporan tugas akhir ini dapat bermanfaat bagi kita.

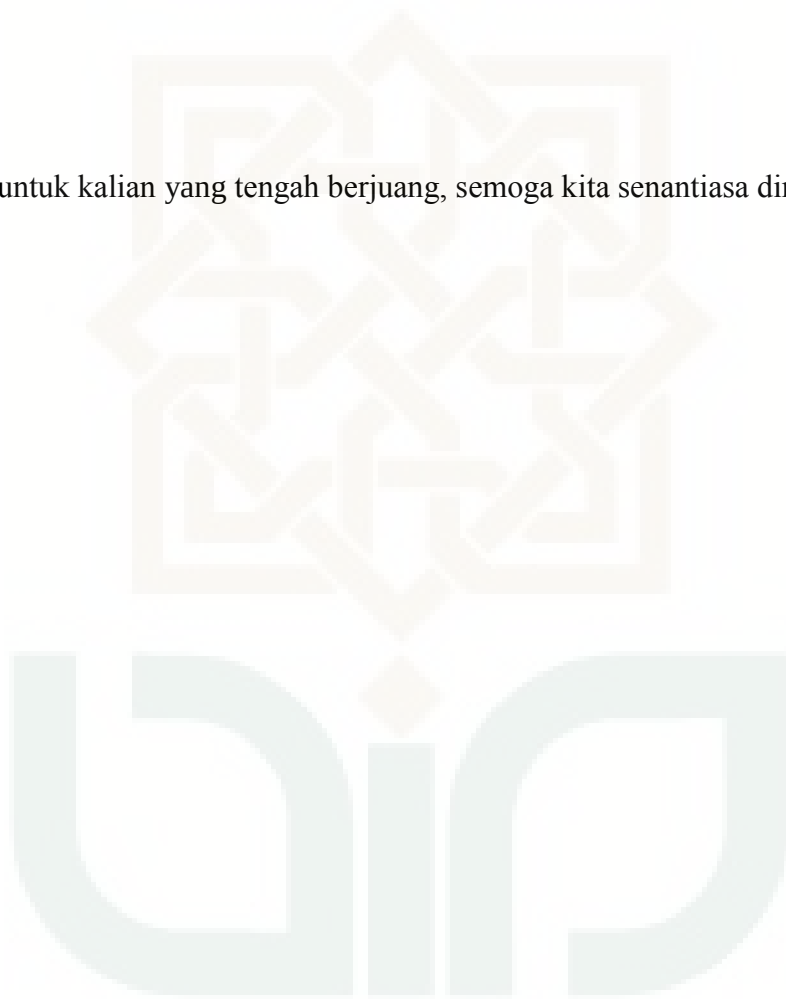
Yogyakarta, 5 Agustus 2016

Penyusun

Ahmad Nur Kholiq
NIM. 10650004

HALAMAN PERSEMBAHAN

“Teruntuk kalian yang tengah berjuang, semoga kita senantiasa dimampukan”



HALAMAN MOTTO

“If you’re going to be weird, be confident about it” –Unknown

“Kehilangan? Apakah kita pernah benar-benar memiliki?”

“Bahwa kebaikan, sesederhana apapun itu, akan menuntun kita pada
kebaikan-kebaikan yang lain”

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
HALAMAN PERSETUJUAN SKRIPSI.....	iii
HALAMAN KEASLIAN SKRIPSI	iv
KATA PENGANTAR	v
HALAMAN PERSEMBAHAN	vii
HALAMAN MOTTO	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Landasan Teori	8
2.2.1 Game	8
2.2.2 Game Design dan Development.....	12
2.2.3 Model Pengembangan Perangkat Lunak <i>Prototyping</i>	13
2.2.4 Engine Game	15
2.2.5 UNITY	17
2.2.6 Blender	18
BAB III METODE PENGEMBANGAN SISTEM.....	21
3.1 Metode Pengembangan Sistem	21
3.2 Kebutuhan Pengembangan Sistem	24

BAB IV ANALISIS DAN PERANCANGAN SISTEM.....	25
4.1 Analisis Kebutuhan	25
4.1.1 Kebutuhan Non Fungsional.....	25
4.1.2 Kebutuhan Fungsional	26
4.2 Perancangan Sistem.....	26
4.2.1 Gambaran Umum Sistem	26
4.2.2 Usecase Diagram.....	27
4.2.3 Activity Diagram.....	29
4.3 Perancangan Antarmuka.....	31
4.3.1 Splash Screen	31
4.3.2 Main Menu Screen	32
4.3.3 Credits Screen	32
4.3.4 Stage Screen.....	33
BAB V IMPLEMENTASI DAN PENGUJIAN	35
5.1 Implementasi	35
5.1.1 3D Modelling	36
5.1.2 Texturing.....	40
5.1.3 Splash Screen	43
5.1.4 Main Menu Screen	43
5.1.5 Credits Screen	45
5.1.6 Stage Screen.....	46
5.1.7 Pause Menu Screen	47
5.2 Pengujian	48
5.2.1 Pengujian Alpha	48
5.2.2 Pengujian Beta	50
BAB VI HASIL DAN PEMBAHASAN	57
6.1 Hasil Pengujian Sistem.....	57
6.2 Hasil dan Pembahasan Pengujian Alpha	57
6.3 Hasil dan Pembahasan Pengujian Beta	59
BAB VII PENUTUP	63
7.1 Kesimpulan.....	63

7.2	Saran	63
DAFTAR PUSTAKA		65
LAMPIRAN		67



DAFTAR GAMBAR

Gambar 4.1 Digram alur perancangan secara umum	27
Gambar 4.2 <i>Usecase Diagram</i>	28
Gambar 4.3 <i>Activity Diagram Play/Pause</i>	29
Gambar 4.4 <i>Activity Diagram Credit (loop)</i>	30
Gambar 4.5 <i>Activity Diagram QUIT</i>	31
Gambar 4.6 Rancangan <i>Splash Screen</i>	31
Gambar 4.7 Rancangan <i>Main Menu Screen</i>	32
Gambar 4.8 Rancangan <i>Credits Screen</i>	33
Gambar 4.9 Rancangan <i>Stage Screen</i>	34
Gambar 4.10 Rancangan <i>Pause Menu Screen</i>	34
Gambar 5.1 <i>Design 3D model</i> bangunan	36
Gambar 5.2 <i>Design 3D Model</i> bangunan dengan <i>texture</i>	37
Gambar 5.3 <i>Design 3D model</i> rak buku	37
Gambar 5.4 <i>Design 3D model</i> kursi dan meja	38
Gambar 5.5 Implementasi <i>Splash Screen</i>	43
Gambar 5.6 <i>Main Menu Screen</i>	44
Gambar 5.7 <i>Credits Screen</i>	46
Gambar 5.8 <i>Stage Screen</i>	47
Gambar 5.9 <i>Pause Menu Screen</i>	47

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian.....	7
Tabel 5.1 Rencana Pengujian Alpha	49
Tabel 5.2 Pertanyaan Usabilitas Aplikasi	52
Tabel 5.3 Pertanyaan Fungsionalitas Aplikasi	53
Tabel 5.4 Contoh Skor Skala Likert atau Tingkatan Pengukuran Ordinal.....	54
Tabel 5.5 Contoh Persentase Interval Skala Likert dengan lima skala	55
Tabel 6.1 Daftar Responden	57
Tabel 6.2 Hasil pengujian Alpha.....	58
Tabel 6.3 Hasil Pengujian Usabilitas Aplikasi.....	59
Tabel 6.4 Nilai Likert, Tingkat Kepuasan pada setiap index dan Tingkat Kepuasan Total	60
Tabel 6.5 Hasil Pengujian Fungsionalitas Aplikasi	60

Perancangan Simulasi *Virtual Tour* Perpustakaan Uin Sunan Kalijaga

Menggunakan *Unity Engine*

Ahmad Nur Kholiq

NIM. 10650004

INTISARI

Perpustakaan merupakan unsur penunjang yang penting dalam dunia pendidikan terutama pada jenjang perguruan tinggi. Untuk memaksimalkan fungsi tersebut, perpustakaan UIN Sunan Kalijaga tiap tahunnya mengadakan *User education* yang merupakan upaya pengenalan perpustakaan kepada mahasiswa baru. Namun dalam pelaksanaannya kerap timbul masalah yaitu kenyamanan pengunjung perpustakaan lain yang terganggu, salah satu cara untuk menanggulangi hal tersebut adalah dengan melakukan simulasi *tour* secara digital atau *virtual tour*. Simulasi *virtual tour* bertujuan sebagai program penunjang *user education*. *Virtual tour* dipilih karena *user* dapat berinteraksi dengan lingkungan nyata perpustakaan yang disimulasikan oleh komputer sehingga *user* seolah-olah terlibat secara fisik.

Menggunakan *blender* dalam perancangan *3D modelling*nya serta *unity engine* yang merupakan *multi-platform engine game* untuk menciptakan konten 3D yang menarik dalam diharapkan dapat menyuguhkan informasi dengan interaktif dan edukatif.

Berdasarkan hasil pengujian usabilitas sebesar 89,2% dan fungsionalitas sebesar 100%, *virtual tour* ini merupakan aplikasi yang interaktif, mudah digunakan dan dipahami serta dapat mensimulasikan pengenalan ruangan perpustakaan maupun pengenalan lokasi koleksi perpustakaan dengan baik.

Kata Kunci: *Blender*, Perpustakaan, Simulasi, *User Education*, *Unity Engine*,

Virtual Tour

Virtual Tour Simulation Designing of Sunan Kalijaga State Islamic University's Library Using Unity Engine

Ahmad Nur Kholiq

NIM. 10650004

ABSTRACT

The library is an important supporting element in education especially at the college level. To maximize these functions, The library of Sunan Kalijaga University annually held User education which is the introduction of the library to their new students. However, in the implementation of it often arise a problem that bother another visitor's comfort, one way to solve it is by doing digitally simulated tour or virtual tour. virtual tour simulation is intended as a supporting user education programs. Virtual tour selected because user can interact with the real environment of the library that simulated by computer so as if the user physically involved.

Using a blender in 3D modelling design and unity engine which is a multi-platform game engine for creating a creative 3D content is expected to be interesting in presenting information with interactive and educative way.

Based on the results of usability testing amounted to 89.2% and 100% of functionality testing, the virtual tour is an interactive application, easy to use and understand and capable to simulate the introduction of a library's room as well as the introduction of the location of collections.

Keywords: Blender, Library, Simulation, User Education, Unity Engine, Virtual Tour

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perpustakaan sebagaimana yang ada dan berkembang sekarang telah dipergunakan sebagai salah satu pusat informasi, sumber ilmu pengetahuan, penelitian, rekreasi, pelestarian khasanah budaya bangsa, serta memberikan berbagai layanan jasa lainnya. Perpustakaan pada prinsipnya mempunyai tiga kegiatan pokok, yaitu: mengumpulkan (*to collect*) semua informasi yang sesuai dengan bidang kegiatan dan misi organisasi dan masyarakat yang dilayaninya. Kedua, melestarikan, memelihara, dan merawat seluruh koleksi perpustakaan, agar tetap dalam keadaan baik, utuh, dan layak pakai dan tidak lekas rusak, baik karena pemakaian maupun karena usianya (*to preserve*). Ketiga, menyediakan dan menyajikan informasi untuk siap dipergunakan dan diberdayakan (*to make available*) seluruh koleksi yang dihimpun di perpustakaan untuk dipergunakan pemakainya (Sutarno, 2003).

Perpustakaan merupakan salah satu unsur penunjang yang penting dalam proses belajar mengajar dalam dunia pendidikan terutama pada perguruan tinggi, pemanfaatan perpustakaan secara optimal berperan penting dalam mengantarkan keberhasilan studi mahasiswanya. Oleh karena itu, perpustakaan Universitas Islam Negeri Sunan Kalijaga tiap tahunnya menyelenggarakan kegiatan *User education* kepada mahasiswa barunya.

User education merupakan bentuk pengenalan kepada mahasiswa baru terhadap lingkungan perpustakaan yang bertujuan untuk mengedukasi dan memotivasi mahasiswa baru dalam mengoptimalkan pemanfaatan layanan dan fasilitas perpustakaan seperti pengenalan ruang, pengenalan koleksi, peminjaman & pengembalian koleksi, peminjaman & pengembalian loker, *corner spot* dan fasilitas lainnya. *User education* dilaksanakan dalam beberapa sesi dan kelompok berdasarkan kelas serta prodi mahasiswa yang bersangkutan dengan didampingi oleh pustakawan yang kompeten dan berpengalaman. Namun dengan banyaknya jumlah kelompok tiap sesinya, tentu saja akan berdampak pada efektifitas penerimaan informasi oleh tiap peserta serta berpengaruh juga terhadap kenyamanan pengunjung perpustakaan yang lain (lingkungan perpustakaan menjadi kurang kondusif).

Kenyamanan pengunjung perpustakaan merupakan hal yang wajib dipenuhi, salah satu solusi untuk mewujudkan suasana kondusif dalam lingkungan perpustakaan tanpa mengesampingkan efektifitas penerimaan informasi adalah melakukan kegiatan *user education* melalui sebuah simulasi *virtual tour*. *Virtual tour* dipilih karena pengguna dapat berinteraksi dengan lingkungan nyata perpustakaan yang disimulasikan oleh komputer sehingga pengguna seolah-olah terlibat langsung secara fisik. Ini tentu saja akan lebih praktis karena peserta tidak perlu berkeliling secara langsung layaknya *tour konvensional* serta menjaga lingkungan perpustakaan agar tetap kondusif.

Banyak hal yang perlu dipertimbangkan dalam merancang sebuah simulasi *virtual tour*, salah satunya adalah pada presentasi grafisnya (2

Dimensi atau 3 Dimensi), ini berpengaruh pada menarik tidaknya penyampaian sebuah virtual tour nantinya, ada yang menggunakan media foto, video, teks, hingga dalam bentuk sebuah *game*. Perkembangan grafis 3D sekarang begitu luar biasa, bahkan beberapa *game* yang beredar dipasaran mampu menyuguhkan grafis yang mendekati *real* atau nyata, perkembangan ini tentu saja merupakan dampak dari semakin gencarnya *developer* dalam mengembangkan sebuah *game engine*.

Game Engine merupakan sebuah framework yang digunakan oleh *developer* dalam merancang serta mengembangkan sebuah video *game*. Beberapa contoh engine *game* yang terus berkembang saat ini adalah Unreal Engine, Unity, CryEngine, Blender serta lainnya dengan kelebihan dan kekurangan masing-masing. *Unity* yang merupakan *multi-platform engine game* untuk menciptakan konten 3D interaktif, dipilih karena selain kemudahan penggunaannya, dukungan forum yang solid, kemudahan dalam ketersediaan *assets*, juga diharapkan dapat menyuguhkan informasi dengan lebih interaktif namun tetap edukatif.

1.2 Rumusan Masalah

Berdasarkan uraian diatas, maka rumusan masalah adalah bagaimana cara merancang suatu simulasi *virtual tour* perpustakaan menggunakan *unity engine*.

1.3 Batasan Masalah

Karena terbatasnya waktu penelitian dan berbagai faktor lain yang mempengaruhi pengerjaan, maka penulis menetapkan beberapa batasan masalah sebagai berikut:

1. *Virtual tour* hanya untuk pengunjung perpustakaan dan tidak mempertimbangkan gender, grafis serta detail bangunan.
2. *Virtual tour* dibangun untuk *single user*.
3. *Virtual Tour* bersifat *Open Wolrd* dan tidak menggunakan NPC/AI.
4. *Virtual Tour* hanya mencakup pengenalan lokasi koleksi dan pengenalan ruangan perpustakaan.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah diuraikan sebelumnya, penelitian ini bertujuan untuk merancang simulasi *virtual tour* perpustakaan UIN Sunan Kalijaga dengan menggunakan *Unity Engine* yang interaktif dan edukatif.

1.5 Manfaat Penelitian

Manfaat dari aplikasi ini adalah mempermudah (sebagai pengganti *tour* manual) pustakawan dalam memberikan informasi serta gambaran nyata terhadap user mengenai ruangan dan lokasi koleksi perpustakaan UIN Sunan Kalijaga.

BAB VII

PENUTUP

7.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan pada sepuluh responden, maka didapat kesimpulan sebagai berikut: Simulasi *virtual tour* perpustakaan UIN Sunan Kalijaga menggunakan *Unity Engine* berhasil dibuat dan berdasarkan pengujian usabilitas dengan hasil uji 89.2%, *virtual tour* dinyatakan *userfriendly* serta edukatif sehingga dapat disimpulkan bahwa aplikasi mudah digunakan dan dipahami serta dapat mensimulasikan pengenalan ruangan perpustakaan maupun pengenalan lokasi koleksi perpustakaan dengan baik.

Sedangkan hasil uji fungsionalitas diperoleh hasil 100%, yang berarti aplikasi interaktif, dapat berjalan dengan optimal serta cukup menarik untuk diimplementasikan sebagai alat bantu pemustaka dalam *user education* perpustakaan.

7.2 Saran

Penelitian ini tentu saja tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu untuk pengembangan penelitian selanjutnya, berikut adalah beberapa saran yang mungkin dapat dipertimbangkan:

1. Penggunaan Scene yang berbeda pada tiap lantai agar beban aplikasi berkurang.

2. Peningkatan texturing dengan menggunakan material yang sudah di-bake, supaya game engine tidak render secara real time terus menerus.
3. Adanya human model yang juga ditambahkan NPC atau AI serta animasi agar aplikasi menjadi lebih menarik dan tidak terkesan sepi.
4. Tambahan fitur layanan-layanan perpustakaan yang lain seperti peminjaman dan pengembalian koleksi juga loker, drop box, gate dan lain-lain.
5. Menggunakan versi Unity yang stable untuk meminimalisir bug atau error dari engine game.
6. Virtual Tour dengan skala yang lebih besar, mungkin bisa fakultas atau bahkan seluruh universitas.
7. Tambahan coding pop up untuk menampilkan informasi atau deskripsi objek melalui trigger zone, baik itu objek ruangan, koleksi maupun properti.

DAFTAR PUSTAKA

- Abhique.blogspot.com*. (2012). Dipetik 12 08, 2013, dari
<http://abhique.blogspot.com/2012/11/metode-prototyping-dalam-pengembangan.html>
- Al-Fatta, H. (2007). *Analisis dan perancangan system informasi untuk keunggulan perusahaan dan organisasi kelas dunia*. Yogyakarta: Andi Offset.
- Alfiansyah, Muhammad. (2014). *Simulasi Mobile Gerak Lurus Berubah Beraturan Dengan Metode Finite State Machine Untuk Pembelajaran*. Yogyakarta: UIN Sunan Kalijaga.
- Asfari, U., Setiawan, B., & Sani, N. A. (2012). *Pembuatan Aplikasi Tata Ruang Tiga Dimensi Gedung Serba Guna Menggunakan Teknologi Virtual Reality*. Surabaya: ITS.
- Basori, A. H. (2007). *Simulasi Virtual Reality Pada Rumah Sakit Graha Amerta Surabaya*. Surabaya: ITS.
- Belson, H., & Ho, J. (2012). A Fresh Graduate's Guide to Software Development Tools and Technologies. Dalam D. C. Rajapakse, *Usability* (2 ed., Vol. 2). Singapore: National University of Singapore.
- Bermudez, Julio & Kevin King. (2000). *Media Interaction and Design Process: Establishing A Knowledge Base*, Journal of Automation in Construction (9, p.37-56).
- ilmukomputer.com. (2011). Dipetik November 5, 2014, dari
<http://ilmukomputer.org/2011/11/29/kelebihan-blender/>
- Jebba, A. M., Ocfera, A., & Yoannita. (2014). *Rancang Bangun Aplikasi Edugame Museum Sultan Mahmud Badarrudin II Palembang Berbasis Unity 3D*. Palembang: STMIK GI MDP.
- Kim, H.-J. J., Graesser, A., Jackson, T., Olney, A., & Chipman, P. (t.thn.). The Effectiveness of Computer Simulations in a Computer -based Learning Environment.
- Ladjamudin, A.-b. (2006). *Rekayasa Perangkat Lunak*. Yogyakarta: Graha Ilmu.
- Mulyanto, Agus. (2009). *Sistem Informasi Konsep dan Aplikasi*. Yogyakarta: Pustaka Pelajar
- Nalwan, Agustinus. (1996). *Seri Aplikasi Pemrograman: Pemrograman Animasi dan Game Profesional*. Yogyakarta: Elex Media Komputindo.

- Nielsen, J., & Budiu, R. (2013). *Mobile Usability*. Berkeley: New Riders.
- N S, Sutarno., (2003). *Perpustakaan dan Masyarakat*. Jakarta: Yayasan Obor Indonesia
- O'Brien, James. (2005). *Introduction to Information System Twelfth Edition*. New York: McGraw-Hill
- Risnita. (2012). Pengembangan Skla Model Likert. *Edu-Bio Vol.3* .
- Rickykurn.wordpress.com. (2012). Dipetik Oktober, 4, 2015 dari <https://rickykurn.wordpress.com/2012/03/08/apa-itu-game-engine/>
- Schell, J. (2008). The Art of Game Design. Dalam *What is Game Design* (hal. xxiv). Burlington: Morgan Kaufmann Publishers.
- Sommerville, I. (2003). Software Engineering. Dalam *Rekayasa Perangkat Lunak* (hal. 86-105). Jakarta: Erlangga.
- Tavinor, G. (2009). Dipetik November 5, 2014, dari http://aesthetics-online.org/articles/index.php?articles_id=44&print=1
- Techterms.com. (2014). Dipetik November 5, 2014, dari <http://www.techterms.com/definition/user-friendly>
- Textures.com. (2016). Dipetik Agustus 2016, dari <http://www.textures.com/>

LAMPIRAN



```
//Source Code First Person Controller
```

```
using System;
using UnityEngine;
using UnityEngine.StandardAssets.CrossPlatformInput;
using UnityEngine.StandardAssets.Utility;
using Random = UnityEngine.Random;

namespace UnityEngine.StandardAssets.Characters.FirstPerson
{
    [RequireComponent(typeof(CharacterController))]
    [RequireComponent(typeof(AudioSource))]
    public class FirstPersonController : MonoBehaviour
    {
        [SerializeField] private bool m_IsWalking;
        [SerializeField] private float m_WalkSpeed;
        [SerializeField] private float m_RunSpeed;
        [SerializeField] [Range(0f, 1f)] private float
m_RunstepLengthen;
        [SerializeField] private float m_JumpSpeed;
        [SerializeField] private float m_StickToGroundForce;
        [SerializeField] private float m_GravityMultiplier;
        [SerializeField] private MouseLook m_MouseLook;
        [SerializeField] private bool m_UseFovKick;
        [SerializeField] private FOVKick m_FovKick = new
FOVKick();
        [SerializeField] private bool m_UseHeadBob;
        [SerializeField] private CurveControlledBob
m_HeadBob = new CurveControlledBob();
        [SerializeField] private LerpControlledBob
m_JumpBob = new LerpControlledBob();
        [SerializeField] private float m_StepInterval;
        [SerializeField] private AudioClip[]
m_FootstepSounds; // an array of footstep sounds that will
be randomly selected from.
        [SerializeField] private AudioClip m_JumpSound;
// the sound played when character leaves the ground.
        [SerializeField] private AudioClip m_LandSound;
// the sound played when character touches back on ground.

        private Camera m_Camera;
        private bool m_Jump;
        private float m_YRotation;
        private Vector2 m_Input;
        private Vector3 m_MoveDir = Vector3.zero;
        private CharacterController m_CharacterController;
        private CollisionFlags m_CollisionFlags;
        private bool m_PreviouslyGrounded;
        private Vector3 m_OriginalCameraPosition;
        private float m_StepCycle;
        private float m_NextStep;
        private bool m_Jumping;
        private AudioSource m_AudioSource;

        // Use this for initialization
        private void Start()
        {
            m_CharacterController =
GetComponent<CharacterController>();
            m_Camera = Camera.main;
            m_OriginalCameraPosition =
m_Camera.transform.localPosition;
            m_FovKick.Setup(m_Camera);
            m_HeadBob.Setup(m_Camera, m_StepInterval);
            m_StepCycle = 0f;
            m_NextStep = m_StepCycle/2f;
            m_Jumping = false;
            m_AudioSource = GetComponent<AudioSource>();
            m_MouseLook.Init(transform ,
m_Camera.transform);

```

```

        }

        private void Update()
        {
            RotateView();
            // the jump state needs to read here to make sure it is
not missed
            if (!m_Jump)
            {
                m_Jump =
CrossPlatformInputManager.GetButtonDown("Jump");
            }

            if (!m_PreviouslyGrounded &&
m_CharacterController.isGrounded)
            {
                StartCoroutine(m_JumpBob.DoBobCycle());
                PlayLandingSound();
                m_MoveDir.y = 0f;
                m_Jumping = false;
            }
            if (!m_CharacterController.isGrounded &&
!m_Jumping && m_PreviouslyGrounded)
            {
                m_MoveDir.y = 0f;
            }
            m_PreviouslyGrounded =
m_CharacterController.isGrounded;
        }

        private void PlayLandingSound()
        {
            m_AudioSource.clip = m_LandSound;
            m_AudioSource.Play();
            m_NextStep = m_StepCycle + .5f;
        }

        private void FixedUpdate()
        {
            float speed;
            GetInput(out speed);
            // always move along the camera forward as it is the
direction that it being aimed at
            Vector3 desiredMove =
transform.forward*m_Input.y +
transform.right*m_Input.x;

            // get a normal for the surface that is being touched
to move along it
            RaycastHit hitInfo;
            Physics.SphereCast(transform.position,
m_CharacterController.radius, Vector3.down, out hitInfo,
m_CharacterController.height/2f);
            desiredMove =
Vector3.ProjectOnPlane(desiredMove,
hitInfo.normal).normalized;

            m_MoveDir.x = desiredMove.x*speed;
            m_MoveDir.z = desiredMove.z*speed;

            if (m_CharacterController.isGrounded)
            {
                m_MoveDir.y = -m_StickToGroundForce;

                if (m_Jump)
                {
                    m_MoveDir.y = m_JumpSpeed;
                    PlayJumpSound();
                    m_Jump = false;
                    m_Jumping = true;

```

```

    }
    }
    else
    {
        m_MoveDir +=
Physics.gravity*m_GravityMultiplier*Time.fixedDeltaTime
;
    }
    m_CollisionFlags =
m_CharacterController.Move(m_MoveDir*Time.fixedDelta
aTime);

    ProgressStepCycle(speed);
    UpdateCameraPosition(speed);
}

private void PlayJumpSound()
{
    m_AudioSource.clip = m_JumpSound;
    m_AudioSource.Play();
}

private void ProgressStepCycle(float speed)
{
    if (m_CharacterController.velocity.sqrMagnitude >
0 && (m_Input.x != 0 || m_Input.y != 0))
    {
        m_StepCycle +=
(m_CharacterController.velocity.magnitude +
(speed*(m_IsWalking ? 1f : m_RunstepLenghten)))*
Time.fixedDeltaTime;
    }

    if (!(m_StepCycle > m_NextStep))
    {
        return;
    }

    m_NextStep = m_StepCycle + m_StepInterval;

    PlayFootStepAudio();
}

private void PlayFootStepAudio()
{
    if (!m_CharacterController.isGrounded)
    {
        return;
    }
    // pick & play a random footstep sound from the
array,
    // excluding sound at index 0
    int n = Random.Range(1,
m_FootstepSounds.Length);
    m_AudioSource.clip = m_FootstepSounds[n];
    m_AudioSource.PlayOneShot(m_AudioSource.clip);
    // move picked sound to index 0 so it's not picked
next time
    m_FootstepSounds[n] = m_FootstepSounds[0];
    m_FootstepSounds[0] = m_AudioSource.clip;
}

private void UpdateCameraPosition(float speed)
{
    Vector3 newCameraPosition;
    if (!m_UseHeadBob)
    {
        return;
    }

```

```

        if (m_CharacterController.velocity.magnitude > 0
&& m_CharacterController.isGrounded)
        {
            m_Camera.transform.localPosition =
m_HeadBob.DoHeadBob(m_CharacterController.velocity.
magnitude +
                (speed*(m_IsWalking ? 1f :
m_RunstepLenghten)));
            newCameraPosition =
m_Camera.transform.localPosition;
            newCameraPosition.y =
m_Camera.transform.localPosition.y -
m_JumpBob.Offset();
        }
        else
        {
            newCameraPosition =
m_Camera.transform.localPosition;
            newCameraPosition.y =
m_OriginalCameraPosition.y - m_JumpBob.Offset();
        }
        m_Camera.transform.localPosition =
newCameraPosition;
    }

    private void GetInput(out float speed)
    {
        // Read input
        float horizontal =
CrossPlatformInputManager.GetAxis("Horizontal");
        float vertical =
CrossPlatformInputManager.GetAxis("Vertical");

        bool waswalking = m_IsWalking;

#if !MOBILE_INPUT
        // On standalone builds, walk/run speed is modified
by a key press.
        // keep track of whether or not the character is
walking or running
        m_IsWalking = !Input.GetKey(KeyCode.LeftShift);
#endif
        // set the desired speed to be walking or running
        speed = m_IsWalking ? m_WalkSpeed :
m_RunSpeed;
        m_Input = new Vector2(horizontal, vertical);

        // normalize input if it exceeds 1 in combined length:
        if (m_Input.sqrMagnitude > 1)
        {
            m_Input.Normalize();
        }

        // handle speed change to give an fov kick
        // only if the player is going to a run, is running and
the fovkick is to be used
        if (m_IsWalking != waswalking && m_UseFovKick
&& m_CharacterController.velocity.sqrMagnitude > 0)
        {
            StopAllCoroutines();
            StartCoroutine(!m_IsWalking ?
m_FovKick.FOVKickUp() : m_FovKick.FOVKickDown());
        }

        private void RotateView()
        {
            m_MouseLook.LookRotation (transform,
m_Camera.transform);

```

```

    }

    private void
    OnControllerColliderHit(ControllerColliderHit hit)
    {
        Rigidbody body = hit.collider.attachedRigidbody;
        //dont move the rigidbody if the character is on top
        if (m_CollisionFlags == CollisionFlags.Below)
        {
            return;
        }

        if (body == null || body.isKinematic)
        {
            return;
        }

        body.AddForceAtPosition(m_CharacterController.velocity
        *0.1f, hit.point, ForceMode.Impulse);
    }
}

```

//Source Code Head Bob

```

using System;
using UnityEngine;
using UnityStandardAssets.Utility;

namespace UnityStandardAssets.Characters.FirstPerson
{
    public class HeadBob : MonoBehaviour
    {
        public Camera Camera;
        public CurveControlledBob motionBob = new
        CurveControlledBob();
        public LerpControlledBob jumpAndLandingBob = new
        LerpControlledBob();
        public RigidbodyFirstPersonController
        rigidbodyFirstPersonController;
        public float StrideInterval;
        [Range(0f, 1f)] public float RunningStrideLengthen;

        // private CameraRefocus m_CameraRefocus;
        private bool m_PreviouslyGrounded;
        private Vector3 m_OriginalCameraPosition;

        private void Start()
        {
            motionBob.Setup(Camera, StrideInterval);
            m_OriginalCameraPosition =
            Camera.transform.localPosition;
            // m_CameraRefocus = new CameraRefocus(Camera,
            transform.root.transform, Camera.transform.localPosition);
        }

        private void Update()
        {
            // m_CameraRefocus.GetFocusPoint();
            Vector3 newCameraPosition;
            if (rigidbodyFirstPersonController.Velocity.magnitude
            > 0 && rigidbodyFirstPersonController.Grounded)
            {
                Camera.transform.localPosition =
                motionBob.DoHeadBob(rigidbodyFirstPersonController.Veloci
                ty.magnitude*(rigidbodyFirstPersonController.Running ?
                RunningStrideLengthen : 1f));
            }
        }
    }
}

```

```

        newCameraPosition =
        Camera.transform.localPosition;
        newCameraPosition.y =
        Camera.transform.localPosition.y -
        jumpAndLandingBob.Offset();
    }
    else
    {
        newCameraPosition =
        Camera.transform.localPosition;
        newCameraPosition.y = m_OriginalCameraPosition.y
        - jumpAndLandingBob.Offset();
    }
    Camera.transform.localPosition = newCameraPosition;

    if (!m_PreviouslyGrounded &&
    rigidbodyFirstPersonController.Grounded)
    {
        StartCoroutine(jumpAndLandingBob.DoBobCycle());
    }

    m_PreviouslyGrounded =
    rigidbodyFirstPersonController.Grounded;
    // m_CameraRefocus.SetFocusPoint(); }}}

```

//Source Code Mouse Look

```
using System;
using UnityEngine;
using UnityEngine.CrossPlatformInput;

namespace UnityStandardAssets.Characters.FirstPerson
{
    [Serializable]
    public class MouseLook
    {
        public float XSensitivity = 2f;
        public floatYSensitivity = 2f;
        public bool clampVerticalRotation = true;
        public float MinimumX = -90F;
        public float MaximumX = 90F;
        public bool smooth;
        public float smoothTime = 5f;

        private Quaternion m_CharacterTargetRot;
        private Quaternion m_CameraTargetRot;

        public void Init(Transform character, Transform camera)
        {
            m_CharacterTargetRot = character.localRotation;
            m_CameraTargetRot = camera.localRotation;
        }

        public void LookRotation(Transform character, Transform camera)
        {
            float yRot = CrossPlatformInputManager.GetAxis("Mouse X") * XSensitivity;
            float xRot = CrossPlatformInputManager.GetAxis("Mouse Y") * YSensitivity;

            m_CharacterTargetRot *= Quaternion.Euler (0f, yRot, 0f);
            m_CameraTargetRot *= Quaternion.Euler (-xRot, 0f, 0f);

            if(clampVerticalRotation)
                m_CameraTargetRot = ClampRotationAroundXAxis (m_CameraTargetRot);

            if(smooth)
            {
                character.localRotation = Quaternion.Slerp (character.localRotation, m_CharacterTargetRot, smoothTime * Time.deltaTime);
                camera.localRotation = Quaternion.Slerp (camera.localRotation, m_CameraTargetRot, smoothTime * Time.deltaTime);
            }
            else
            {
                character.localRotation = m_CharacterTargetRot;
                camera.localRotation = m_CameraTargetRot;
            }
        }

        Quaternion ClampRotationAroundXAxis(Quaternion q)
        {
            q.z /= q.w;
            q.w = 1.0f;

            float angleX = 2.0f * Mathf.Rad2Deg * Mathf.Atan (q.x);

            angleX = Mathf.Clamp (angleX, MinimumX, MaximumX);

            q.x = Mathf.Tan (0.5f * Mathf.Deg2Rad * angleX);

            return q;
        }
    }
}
```

//Source Code Rigidbody FirstPersonController

```
using System;
using UnityEngine;
using UnityEngine.InputSystem;

namespace UnityStandardAssets.Characters.FirstPerson
{
    [RequireComponent(typeof(Rigidbody))]
    [RequireComponent(typeof(CapsuleCollider))]
    public class RigidbodyFirstPersonController :
        MonoBehaviour
    {
        [Serializable]
        public class MovementSettings
        {
            public float ForwardSpeed = 8.0f; // Speed when
            walking forward
            public float BackwardSpeed = 4.0f; // Speed when
            walking backwards
            public float StrafeSpeed = 4.0f; // Speed when
            walking sideways
            public float RunMultiplier = 2.0f; // Speed when
            sprinting
            public KeyCode RunKey = KeyCode.LeftShift;
            public float JumpForce = 30f;
            public AnimationCurve SlopeCurveModifier = new
            AnimationCurve(new Keyframe(-90.0f, 1.0f), new
            Keyframe(0.0f, 1.0f), new Keyframe(90.0f, 0.0f));
            [HideInInspector] public float CurrentTargetSpeed = 8f;
        }

        #if !MOBILE_INPUT
            private bool m_Running;
        #endif

        public void UpdateDesiredTargetSpeed(Vector2 input)
        {
            if (input == Vector2.zero) return;
            if (input.x > 0 || input.x
            < 0)
            {
                //strafe
                CurrentTargetSpeed = StrafeSpeed;
            }
            if (input.y < 0)
            {
                //backwards
                CurrentTargetSpeed = BackwardSpeed;
            }
            if (input.y > 0)
            {
                //forwards
                //handled last
                as if strafing and moving forward at the same time forwards
                speed should take precedence
                CurrentTargetSpeed = ForwardSpeed;
            }
        }

        #if !MOBILE_INPUT
            if (Input.GetKey(RunKey))
            {
                CurrentTargetSpeed *=
                RunMultiplier;
                m_Running = true;
            }
            else
            {
                m_Running = false;
            }
        #endif
    }
}
```

```
    }
}

#endif

#if !MOBILE_INPUT
    public bool Running
    {
        get { return m_Running; }
    }
#endif

[Serializable]
public class AdvancedSettings
{
    public float groundCheckDistance = 0.01f; // distance
    for checking if the controller is grounded ( 0.01f seems to work
    best for this )
    public float stickToGroundHelperDistance = 0.5f; //
    stops the character
    public float slowDownRate = 20f; // rate at which the
    controller comes to a stop when there is no input
    public bool airControl; // can the user control the
    direction that is being moved in the air
}

public Camera cam;
public MovementSettings movementSettings = new
MovementSettings();
public MouseLook mouseLook = new MouseLook();
public AdvancedSettings advancedSettings = new
AdvancedSettings();

private Rigidbody m_RigidBody;
private CapsuleCollider m_Capsule;
private float m_YRotation;
private Vector3 m_GroundContactNormal;
private bool m_Jump, m_PreviouslyGrounded,
m_Jumping, m_IsGrounded;

public Vector3 Velocity
{
    get { return m_RigidBody.velocity; }
}

public bool Grounded
{
    get { return m_IsGrounded; }
}

public bool Jumping
{
    get { return m_Jumping; }
}

public bool Running
{
    get
    {
        #if !MOBILE_INPUT
            return
            movementSettings.Running;
        #else
            return false;
        #endif
    }
}
```

```

    }

    private void Start()
    {
        m_RigidBody = GetComponent<Rigidbody>();
        m_Capsule = GetComponent<CapsuleCollider>();
        mouseLook.Init (transform, cam.transform);
    }

    private void Update()
    {
        RotateView();

        if
        (CrossPlatformInputManager.GetButtonDown("Jump") &&
        !m_Jump)
        {
            m_Jump = true;
        }
    }

    private void FixedUpdate()
    {
        GroundCheck();
        Vector2 input = GetInput();

        if ((Mathf.Abs(input.x) > float.Epsilon ||
        Mathf.Abs(input.y) > float.Epsilon) &&
        (advancedSettings.airControl || m_IsGrounded))
        {
            // always move along the camera forward as it is the
            direction that it being aimed at
            Vector3 desiredMove =
            cam.transform.forward*input.y + cam.transform.right*input.x;
            desiredMove =
            Vector3.ProjectOnPlane(desiredMove,
            m_GroundContactNormal).normalized;

            desiredMove.x =
            desiredMove.x*movementSettings.CurrentTargetSpeed;
            desiredMove.z =
            desiredMove.z*movementSettings.CurrentTargetSpeed;
            desiredMove.y =
            desiredMove.y*movementSettings.CurrentTargetSpeed;
            if (m_RigidBody.velocity.sqrMagnitude <
            (movementSettings.CurrentTargetSpeed*movementSettings.Cu
            rrentTargetSpeed))
            {
                m_RigidBody.AddForce(desiredMove*SlopeMultiplier(),
                ForceMode.Impulse);
            }
        }

        if (m_IsGrounded)
        {
            m_RigidBody.drag = 5f;

            if (m_Jump)
            {
                m_RigidBody.drag = 0f;
                m_RigidBody.velocity = new
                Vector3(m_RigidBody.velocity.x, 0f,
                m_RigidBody.velocity.z);
                m_RigidBody.AddForce(new Vector3(0f,
                movementSettings.JumpForce, 0f), ForceMode.Impulse);
            }
        }
    }

    private float SlopeMultiplier()
    {
        float angle = Vector3.Angle(m_GroundContactNormal,
        Vector3.up);
        return
        movementSettings.SlopeCurveModifier.Evaluate(angle);
    }

    private void StickToGroundHelper()
    {
        RaycastHit hitInfo;
        if (Physics.SphereCast(transform.position,
        m_Capsule.radius, Vector3.down, out hitInfo,
        ((m_Capsule.height/2f) -
        m_Capsule.radius) +
        advancedSettings.stickToGroundHelperDistance))
        {
            if (Mathf.Abs(Vector3.Angle(hitInfo.normal,
            Vector3.up)) < 85f)
            {
                m_RigidBody.velocity =
                Vector3.ProjectOnPlane(m_RigidBody.velocity,
                hitInfo.normal);
            }
        }
    }

    private Vector2 GetInput()
    {
        Vector2 input = new Vector2
        {
            x =
            CrossPlatformInputManager.GetAxis("Horizontal"),
            y =
            CrossPlatformInputManager.GetAxis("Vertical")
        };

        movementSettings.UpdateDesiredTargetSpeed(input)
        ;

        return input;
    }

    private void RotateView()
    {
        m_Jumping = true;
    }

    if (!m_Jumping && Mathf.Abs(input.x) <
    float.Epsilon && Mathf.Abs(input.y) < float.Epsilon &&
    m_RigidBody.velocity.magnitude < 1f)
    {
        m_RigidBody.Sleep();
    }
    else
    {
        m_RigidBody.drag = 0f;
        if (m_PreviouslyGrounded && !m_Jumping)
        {
            StickToGroundHelper();
        }
        m_Jump = false;
    }

```



```

    {
        //avoids the mouse looking if the game is effectively
        paused
        if (Mathf.Abs(Time.timeScale) < float.Epsilon) return;

        // get the rotation before it's changed
        float oldYRotation = transform.eulerAngles.y;

        mouseLook.LookRotation (transform, cam.transform);

        if (m_IsGrounded || advancedSettings.airControl)
        {
            // Rotate the rigidbody velocity to match the new
            direction that the character is looking
            Quaternion velRotation =
            Quaternion.AngleAxis(transform.eulerAngles.y - oldYRotation,
            Vector3.up);
            m_RigidBody.velocity =
            velRotation*m_RigidBody.velocity;
        }
    }

    /// sphere cast down just beyond the bottom of the capsule
    to see if the capsule is colliding round the bottom
    private void GroundCheck()
    {
        m_PreviouslyGrounded = m_IsGrounded;
        RaycastHit hitInfo;
        if (Physics.SphereCast(transform.position,
        m_Capsule.radius, Vector3.down, out hitInfo,
        ((m_Capsule.height/2f) -
        m_Capsule.radius) + advancedSettings.groundCheckDistance))
        {
            m_IsGrounded = true;
            m_GroundContactNormal = hitInfo.normal;
        }
        else
        {
            m_IsGrounded = false;
            m_GroundContactNormal = Vector3.up;
        }
        if (!m_PreviouslyGrounded && m_IsGrounded &&
        m_Jumping)
        {
            m_Jumping = false;
        }
    }
}

```

LEMBAR ANGKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tuliskan Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : Agung
Pekerjaan : PMS
Instansi : UIN Sunan Kalijaga

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?	✓				
2.	Apakah aplikasi menarik untuk dimainkan ?	✓				
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?	✓				
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?		✓			
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti ?		✓			

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik?	✓	
3.	Apakah sistem dapat melakukan pause?	✓	
4.	Apakah sistem dapat melakukan resume?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas?	✓	

LEMBAR ANGKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tulislah Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : Minto warono
Pekerjaan : Staff TU
Instansi : OPT Perpustakaan UIN Sukra YK

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?	✓				
2.	Apakah aplikasi menarik untuk dimainkan ?	✓				
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?		✓			
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?	✓				
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti ?	✓				

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik ?	✓	
3.	Apakah sistem dapat melakukan pause ?	✓	
4.	Apakah sistem dapat melakukan resume ?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas ?	✓	

LEMBAR ANGKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tulislah Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : ARINA DAMAYANTI

Pekerjaan : RESEARCH ADMIN

Instansi : FKT UGM

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?	✓				
2.	Apakah aplikasi menarik untuk dimainkan ?	✓				
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?	✓				
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?		✓			
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti ?		✓			

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat bekerja dengan baik?	✓	
3.	Apakah sistem dapat melakukan pause?	✓	
4.	Apakah sistem dapat melakukan resume?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas?	✓	

LEMBAR ANGKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tulislah Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : ALEX
Pekerjaan : Game Tester
Instansi : Gameloft

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?	✓				
2.	Apakah aplikasi menarik untuk dimainkan ?	✓				
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?	✓				
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?		✓			
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti ?		✓			

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik?	✓	
3.	Apakah sistem dapat melakukan pause ?	✓	
4.	Apakah sistem dapat melakukan resume ?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas ?	✓	

LEMBAR ANKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tulislah Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : Fadholi
Pekerjaan : Swasta
Instansi : Citranet

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?	✓				
2.	Apakah aplikasi menarik untuk dimainkan ?		✓			
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?		✓			
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?			✓		
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti ?		✓			

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik ?	✓	
3.	Apakah sistem dapat melakukan pause ?	✓	
4.	Apakah sistem dapat melakukan resume ?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas ?	✓	

LEMBAR ANGKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tulislah Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : Ahmad Hanif Arsyad Fata
Pekerjaan : Mahasiswa
Instansi : UIN Sunan Kalijaga (Manajemen Pend. Islam)

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?		✓			
2.	Apakah aplikasi menarik untuk dimainkan ?		✓			
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?			✓		
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?		✓			
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti ?		✓			

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik ?	✓	
3.	Apakah sistem dapat melakukan pause ?	✓	
4.	Apakah sistem dapat melakukan resume ?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas ?	✓	

LEMBAR ANKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tuliskan Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : AMI NAHDIA P
Pekerjaan : MAHASISWI
Instansi : UIN SUNAN KALUAGA YK (KPI)

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?		✓			
2.	Apakah aplikasi menarik untuk dimainkan ?	✓				
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?	✓				
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?	✓				
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti?		✓			

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik?	✓	
3.	Apakah sistem dapat melakukan pause ?	✓	
4.	Apakah sistem dapat melakukan resume ?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	✗
11.	Karakter tidak dapat menembus objek Meja?	✓	✗
12.	Karakter tidak dapat menembus objek Rak buku?	✓	✗
13.	Karakter tidak dapat menembus objek Pintu?	✓	✗
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas ?	✓	

LEMBAR ANGKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tulislah Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : Rahmadhani M

Pekerjaan : Mahasiswa

Instansi : UIN

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?	✓				
2.	Apakah aplikasi menarik untuk dimainkan ?	✓				
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?	✓				
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?	✓				
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti?	✓				

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik?	✓	
3.	Apakah sistem dapat melakukan pause ?	✓	
4.	Apakah sistem dapat melakukan resume ?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas ?	✓	

LEMBAR ANKET PENGUJIAN SISTEM

Contoh pengisian :

1. Tulislah Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : Rasyra YS

Pekerjaan : Freelance

Instansi : -

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?		✓			
2.	Apakah aplikasi menarik untuk dimainkan ?	✓				
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?		✓			
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?			✓		
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti?		✓			

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik?	✓	
3.	Apakah sistem dapat melakukan pause?	✓	
4.	Apakah sistem dapat melakukan resume?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas?	✓	

LEMBAR ANKET PENGUJIAN SISTEM

Petunjuk pengisian :

1. Tulislah Nama, Pekerjaan dan Instansi di tempat yang telah disediakan
2. Berilah tanda centang (✓) pada kolom jawaban yang anda pilih, yaitu Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS) dan Sangat Tidak Setuju (STS)

Nama : Ifvan Nur Cahyanfo
Pekerjaan : Mahasiswa
Instansi : UIN Sunan Kalijaga (KPI)

5.1 Tabel Pertanyaan Usabilitas Aplikasi

No.	Pertanyaan	SS	S	N	TS	STS
1.	Apakah aplikasi mudah digunakan ?	✓				
2.	Apakah aplikasi menarik untuk dimainkan ?		✓			
3.	Apakah aplikasi dapat mensimulasikan pengenalan ruangan perpustakaan dengan baik ?	✓				
4.	Apakah aplikasi dapat mensimulasikan pengenalan lokasi koleksi perpustakaan dengan baik?	✓				
5.	Apakah informasi yang terdapat dalam sub-menu credit mudah dimengerti?		✓			

5.2 Tabel Pertanyaan Fungsionalitas Aplikasi

No.	Pertanyaan	Ya	Tidak
1.	Apakah aplikasi dapat dimainkan dengan baik?	✓	
2.	Apakah semua tombol dapat berjalan dengan baik?	✓	
3.	Apakah sistem dapat melakukan pause ?	✓	
4.	Apakah sistem dapat melakukan resume ?	✓	
5.	Apakah sistem dapat menampilkan Pause Menu Screen?	✓	
6.	Apakah sistem dapat menampilkan Main Menu Screen?	✓	
7.	Apakah sistem dapat menampilkan informasi dalam credit menu?	✓	
8.	Apakah karakter dapat melakukan JUMP/Loncat?	✓	
9.	Apakah karakter dapat melakukan RUN/Lari?	✓	
10.	Karakter tidak dapat menembus objek Dinding?	✓	✗
11.	Karakter tidak dapat menembus objek Meja?	✓	
12.	Karakter tidak dapat menembus objek Rak buku?	✓	
13.	Karakter tidak dapat menembus objek Pintu?	✓	
14.	Apakah Keterangan penulisan Label pada tiap Rak Koleksi jelas ?	✓	

CURRICULUM VITAE



Nama : Ahmad Nur Kholiq

Tempat, Tanggal Lahir : Jepara, 27 November 1992

Jenis Kelamin : Laki – laki

Agama : Islam

Alamat Asal : Dk Kembul Sari, Suwawal Timur RT 05 RW 02,
Pakis Aji, Jepara, 59452

No Handphone : +6282189271572

Email : kuhoriku@gmail.com

Riwayat Pendidikan :

1. MI Miftahul Huda Suwawal Timur (1998 - 2004)
2. MTs. Mamba’ul Ulum Mambak (2004 - 2007)
3. SMK N 1 Jepara (2007 - 2010)
4. Teknik Informatika UIN Sunan Kalijaga (2010 - 2016)