

LAPORAN KERJA PRAKTEK
Pembangunan Game Mixed Reality



Dipersiapkan oleh :
Eric Simahan / 130707313

Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta
2018

HALAMAN PENGESAHAN

Laporan Kerja Praktek



Dosen Pembimbing,



Y. Sigit Purnomo, S.T., M.Kom.

Pembimbing Lapangan,



Davin Pratama

KATA PENGANTAR

Puji syukur kepada Tuhan yang maha esa, karena atas kasih karunianya saya dapat menyelesaikan laporan pelaksanaan magang yang berjudul “Mixed Reality Game Development”. Laporan pelaksanaan magang ini disusun sebagai syarat untuk menyelesaikan magang yang merupakan salah satu mata kuliah wajib pada program studi Teknik Informatika Universitas Atma Jaya Yogyakarta.

Terima kasih saya ucapkan kepada pihak dari Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta, pihak dari Catlil Studio Indonesia, dan pihak – pihak lain yang terlibat secara langsung maupun tidak langsung dalam pembuatan laporan ini. Saya berharap laporan ini dapat bermanfaat bagi anda.

Saya menyadari keterbatasan pengetahuan yang saya miliki ketika menulis laporan ini, kritik dan saran akan sangat membantu saya untuk menjadi penulis yang lebih baik dari sekarang.

Jakarta, 1 Desember 2017

Penulis,

Eric Simahan

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	v
1. BAB I PENDAHULUAN.....	1
1.1. Sekilas Perusahaan	1
1.2. Sejarah Perusahaan	1
1.3. Visi, Misi, Dan Tujuan Perusahaan	2
1.4. Struktur Organisasi	2
1.5. Deskripsi Tugas Struktur Organisasi	3
1.6. Departemen IT Dalam Perusahaan.....	4
2. BAB II PELAKSANAAN MAGANG	5
2.1. Penjelasan Logbook	5
2.2. Hasil Pekerjaan Secara Umum.....	15
2.3. Bukti Hasil Pekerjaan.....	16
3. BAB III HASIL PEMBELAJARAN	30
3.1. Manfaat Magang	30
3.2. Penerapan Ilmu Dalam Magang.....	30
4. BAB IV KESIMPULAN	32

DAFTAR GAMBAR

Gambar 1.4.1. Bagan Struktur Organisasi	2
Gambar 2.3.1. Proses Pengerjaan.....	25
Gambar 2.3.2. Prontera	25
Gambar 2.3.3. Sograt Desert.....	26
Gambar 2.3.4. Senjata di Ragnarok VR.....	27
Gambar 2.3.5. Musuh di Ragnarok VR.....	27
Gambar 2.3.6. Cross Fire VR.....	28
Gambar 2.3.7. Senjata di Cross Fire VR.....	29

BAB I

PENDAHULUAN

1.1. Sekilas Perusahaan

Catlil Studio Indonesia adalah perusahaan yang bergerak dibidang video game. Kegiatan sehari – hari dalam perusahaan ini meliputi pembuatan konsep dan cerita dalam game, desain karakter dan environment dalam game, pembuatan animasi, dan programming. Beberapa game terkenal yang dirilis oleh perusahaan ini adalah Robowar dan Boboiboy.

1.2. Sejarah Perusahaan

Catlil Studio Indonesia didirikan pada tahun 2014 dan beroperasi di daerah ibu kota Indonesia yaitu Jakarta. Catlil dibangun oleh pengusaha lokal Alex Gunawan dan didukung oleh publisher game terkenal Lytogame. Pada tahun 2016, Catlil bekerja sama dengan Dongseo University dan Mintsphere berhasil merilis game berjudul RoboWar – Robot VS Alien. Sesuai dengan namanya Robowar merupakan sebuah game action yang menampilkan pertarungan antar robot. Pada awal tahun 2017, Catlil bekerja sama dengan 8elements berhasil merilis game BoBoiBoy: Galactic Heroes RPG. Game tersebut diadaptasi dari serial televisi berjudul BoBoiBoy Galaxy. Kedua game tersebut dirilis pada platform mobile Android dan Apple. Selain game mobile, Catlil juga berhasil membangun game pada platform lainnya, seperti web dan virtual reality. Game Virtual Reality yang dibangun oleh Catlil dapat dimainkan pada event – event besar yang diadakan oleh pihak Lytogame.

1.3. Visi, Misi, Dan Tujuan Perusahaan

1. Visi

Untuk menjadi perusahaan game terbaik dan terbesar di dunia.

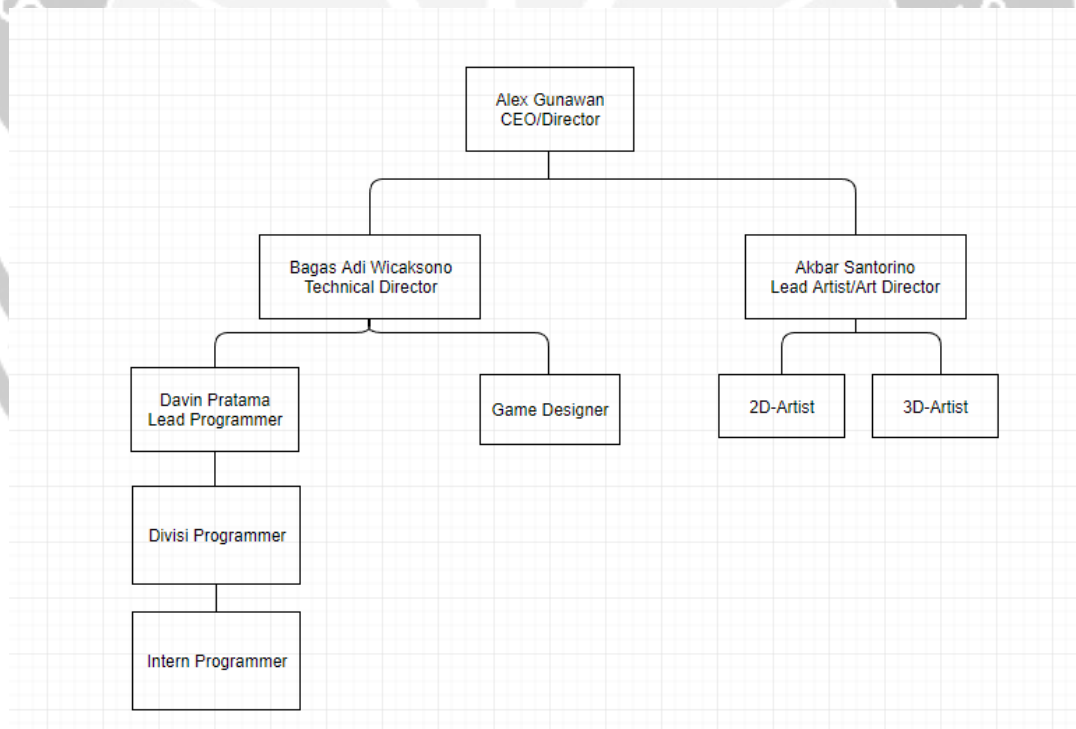
2. Misi

Untuk menciptakan pengalaman bermain yang terbaik bagi para pemain.

3. Tujuan

Untuk dapat bersaing dalam bidang industri game di pasar internasional.

1.4. Struktur Organisasi



Gambar 1.4.1. Bagan Struktur Organisasi

1.5. Deskripsi Tugas Struktur Organisasi

1. CEO / Director

Bertugas untuk menentukan strategi perusahaan agar perusahaan tersebut dapat bergerak sesuai dengan tujuan.

2. Technical Director

Bertugas untuk memonitor perkembangan proyek untuk memastikan bahwa proyek yang dikerjakan oleh semua staff sesuai dengan rencana.

3. Art Director

Bertugas untuk memonitor perkembangan artist untuk memastikan bahwa aset yang dikerjakan sesuai dengan rencana.

4. Lead Programmer

Bertugas sebagai programmer utama dan memberikan arahan bagi programmer lain mengenai bagian proyek yang dikerjakan.

5. Lead Artist

Bertugas sebagai artist utama dan memberikan arahan bagi artist lain mengenai bagian aset yang dikerjakan.

6. Game Designer

Bertugas untuk mendesain gameplay sesuai dengan gambaran hasil akhir.

7. Artist

Bertugas untuk membuat desain aset yang dibutuhkan.

8. Programmer

Bertugas untuk merealisasikan seluruh bagian dalam game dan memastikan tidak adanya error dalam gameplay.

1.6. Departemen IT Dalam Perusahaan

Catlil Studio Indonesia memiliki 2 divisi yaitu Artist dan Programmer. Artist berperan sebagai penyedia aset gambar dan animasi, sedangkan programmer berperan sebagai pengembang game berdasarkan aset yang disediakan. Adapun bagian divisi lain dalam perusahaan Lytogame yang turut berperan dalam perkembangan Catlil yaitu divisi Human Resource, Server, Website, Marketing Communication, Game Master, dan Community Manager. Divisi Human Resource berperan sebagai pengelola sumber daya manusia. Divisi Server berperan sebagai teknisi hardware & penyedia layanan internet, server, dan basis data. Divisi Website berperan sebagai pengelola situs Catlil. Divisi Marketing Communication berperan sebagai pengorganisir, penyedia kebutuhan, dan publikasi event. Divisi Game Master berperan sebagai game tester. Divisi Community Manager berperan sebagai pengelola hubungan dengan komunitas pemain.

BAB II

PELAKSANAAN KERJA PRAKTEK

2.1. Penjelasan Logbook

Logbook merupakan buku yang diberikan kepada setiap mahasiswa yang akan menjalankan magang untuk mencatat agenda kegiatan yang dilakukan selama proses magang berlangsung. Adapun penjelasan isi logbook kegiatan magang yang saya miliki sebagai berikut.

NO	TANGGAL	HAL YANG DIKERJAKAN
1	18 Agustus 2017	Kegiatan hari ini adalah mengerjakan tutorial mengenai virtual reality menggunakan software Unity3D. Hasilnya berupa sebuah game tutorial simple berisikan seorang pemain dan area permainan.
2	21 Agustus 2017	Kegiatan hari ini adalah membuat karakter dalam game tutorial berjalan. Hasilnya adalah karakter dalam game tutorial tersebut dapat berjalan ketika pemain berjalan ditempat.
3	22 Agustus 2017	Kegiatan hari ini adalah membuat tampilan split screen pada game tutorial. Hasilnya adalah game tutorial tersebut dapat menampilkan 2 tampilan permainan, 1 tampilan untuk pemain dan 1 tampilan lainnya untuk penonton.
4	23 Agustus 2017	Kegiatan hari ini adalah mengubah game Ragnarok Virtual Reality menjadi mixed reality. Hasilnya adalah Ragnarok Mixed Reality dimana penonton dapat melihat game beserta pemainnya di dunia game tersebut.

5	24 Agustus 2017	Kegiatan hari ini adalah menyusun letak kamera pada Ragnarok VR untuk dapat menampilkan game secara mixed reality. Hasilnya adalah posisi kamera berhasil diatur sedemikian rupa untuk menampilkan pemain dengan benar.
6	25 Agustus 2017	Kegiatan hari ini adalah mencoba teknik Chroma Key pada latar abu – abu. Hasilnya adalah kamera berhasil merekam objek nyata dengan latar yang dikosongkan.
7	28 Agustus 2017	Kegiatan hari ini adalah menyediakan aset panah untuk game Ragnarok VR. Hasilnya adalah aset dalam game Ragnarok VR berupa panah dan anak panah.
-	29 Agustus 2017	Izin tidak masuk kerja.
8	30 Agustus 2017	Kegiatan hari ini adalah membuat aset panah dapat digunakan oleh karakter dalam game Ragnarok VR. Hasilnya berupa karakter dalam game Ragnarok VR yang dapat menembakkan panah.
9	31 Agustus 2017	Kegiatan hari ini adalah memasukkan aset tongkat ke dalam game Ragnarok VR. Hasilnya adalah karakter dalam game Ragnarok VR dapat memegang tongkat.
10	4 September 2017	Kegiatan hari ini adalah membuat tongkat di dalam game Ragnarok VR dapat mengeluarkan api. Hasilnya adalah tongkat dalam game Ragnarok VR dapat mengeluarkan efek berupa api untuk menyerang.
11	5 September 2017	Kegiatan hari ini adalah pengenalan proyek game lain yang bertema tower defense bernama

		Plek Plek. Saya mempelajari aset dan metode aset dari proyek game Plek Plek tersebut.
12	6 September 2017	Kegiatan hari ini adalah pengenalan script dari game Plek Plek. Saya mempelajari script dan metode pemrograman yang digunakan dalam game Plek Plek tersebut.
13	7 September 2017	Kegiatan hari ini adalah memperbaiki error pada game Ragnarok VR. Hasilnya adalah game Ragnarok VR dapat berjalan dengan lancar tanpa adanya error.
14	8 September 2017	Kegiatan hari ini adalah mengubah tongkat dalam game Ragnarok VR untuk menyesuaikan dengan desain yang telah dibuat. Hasilnya berupa tongkat emas yang memiliki motif naga.
15	11 September 2017	Kegiatan hari ini adalah mengelola Cardbase pada game Plek Plek. Hasilnya adalah script mengenai cardbase yang telah disederhanakan.
16	12 September 2017	Kegiatan hari ini adalah membuat karakter dalam game Plek Plek dapat dipanggil menggunakan tap dari jari.
-	13 September 2017	Izin tidak masuk kerja.
17	14 September 2017	Kegiatan hari ini adalah membuat karakter pada game Plek Plek dapat dipilih menggunakan tap. Hasilnya adalah kartu karakter dalam game Plek Plek menjadi transparan ketika di tap oleh pemain.
18	15 September 2017	Kegiatan hari ini adalah membuat Object Pool pada game Plek Plek agar aset yang digunakan dapat di recycle dan tidak menghabiskan resource ketika bermain. Hasilnya adalah game

		Plek Plek memiliki object pool untuk karakter, ledakan, dan panah.
19	18 September 2017	Kegiatan hari ini adalah menyederhanakan dependensi script pada game Plek Plek. Hasilnya adalah penyederhanaan dependensi dari script cardbase, card, hero, dan game controller.
20	19 September 2017	Kegiatan hari ini adalah menambahkan 2 buah tower pada game Plek Plek. Hasilnya adalah masing – masing tim memiliki 3 buah tower. 2 tower kecil harus dihancurkan terlebih dahulu sebelum menuju tower terakhir.
21	20 September 2017	Kegiatan hari ini adalah menambahkan area tempat munculnya karakter pada game Plek Plek. Hasilnya adalah penambahan zona spawn karakter ketika tower kecil berhasil dihancurkan.
22	22 September 2017	Kegiatan hari ini adalah memisahkan script spawner, spawn controller, dan cardbase. Hasilnya adalah masing – masing script dapat berjalan dengan sendirinya tanpa dependensi yang tinggi antar script.
23	25 September 2017	Kegiatan hari ini adalah memperbaiki error pada script spawner. Hasilnya adalah script spawner yang dapat digunakan untuk memunculkan karakter pada game Plek Plek tanpa adanya error.
24	26 September 2017	Kegiatan hari ini adalah menghilangkan glitch berupa karakter yang tidak bergerak, pemilihan karakter yang tersangkut, dan ledakan yang

		terjadi di tempat yang lain. Hasilnya adalah game Plek Plek dapat berjalan tanpa adanya glitch tersebut.
-	27 September 2017	Izin tidak masuk kerja.
25	28 September 2017	Kegiatan hari ini adalah pengenalan konsep game virtual reality yang baru. Game ini berupa game first person shooting dan diadaptasi dari game Cross Fire Online.
26	29 September 2017	Kegiatan hari ini adalah menyediakan aset yang akan digunakan pada Cross Fire VR, aset yang dibutuhkan berupa environment, target, dan gun. Hasilnya adalah scene lengkap berisi sebuah gudang berisi target – target tembak dan senjata jenis glock.
27	2 Oktober 2017	Kegiatan hari ini adalah membuat script mengenai gameplay Cross Fire VR. Hasilnya berupa script game controller yang mengatur mengenai scoring, munculnya target, dan lama permainan.
28	3 Oktober 2017	Kegiatan hari ini adalah membuat script mengenai shooting. Hasilnya berupa script gun controller yang mengatur fungsi menembak pada senjata.
29	4 Oktober 2017	Kegiatan hari ini adalah membuat script untuk menggerakkan target. Hasilnya berupa script target controller yang mengatur pergerakan target.
30	5 Oktober 2017	Kegiatan hari ini adalah review kembali game Ragnarok VR. Hasilnya adalah beberapa perubahan pada aset dan perbaikan minor bug.

31	6 Oktober 2017	Kegiatan hari ini adalah mengubah panah dalam Ragnarok VR. Hasilnya adalah aset panah dalam game Ragnarok VR diubah menjadi aset panah yang telah disediakan oleh tim Catlil.
32	9 Oktober 2017	Kegiatan hari ini adalah mengubah game Ragnarok VR menjadi mixed reality sepenuhnya. Hasilnya adalah game Ragnarok VR dapat mendeteksi kontroler ke tiga dan mengubah tampilan menjadi mixed reality secara otomatis.
33	10 Oktober 2017	Kegiatan hari ini adalah mengubah map yang terdapat di game Ragnarok VR menjadi map yang telah disediakan oleh tim Catlil. Hasilnya adalah game yang sebelumnya dimainkan di Prontera diubah menjadi Sograt Desert.
34	11 Oktober 2017	Kegiatan hari ini adalah memperbaiki map yang diubah sebelumnya. Hasilnya adalah map Sograt Desert saat ini dilengkapi dengan sungai, pepohonan, dan objek – objek lainnya sesuai dengan map yang terdapat di game Ragnarok.
35	12 Oktober 2017	Kegiatan hari ini adalah menambahkan jenis monster lain yang akan menjadi mini boss dalam game Ragnarok VR. Hasilnya adalah penambahan Angeling, Archangeling, Deviling, dan Ghostring ke dalam gameplay.
36	13 Oktober 2017	Kegiatan hari ini adalah menambahkan monster baru kedalam gameplay dan merapikan peletakkan monster. Hasilnya adalah penambahan Marin sebagai common monster

		dan penempatan monster yang merata di seluruh map.
37	16 Oktober 2017	Kegiatan hari ini adalah menyesuaikan kembali posisi kamera agar tepat dengan lokasi kontroler. Hasilnya adalah kamera posisi kontroler ketiga menjadi sesuai dengan posisi kamera dalam Virtual Reality.
38	17 Oktober 2017	Kegiatan hari ini adalah testing untuk game Ragnarok VR dan Cross Fire VR, beberapa game master dipanggil untuk mencoba game tersebut. Hasilnya adalah bug list yang terjadi ketika game dimainkan dan berhasil diperbaiki hari ini.
39	18 Oktober 2017	Kegiatan hari ini adalah menjadi operator VR saat ada kunjungan dari sekolah dasar, anak – anak dan wali mengantri untuk mencoba Cross Fire VR. Hasilnya adalah beberapa masukan dari para pemain mengenai gameplay.
40	19 Oktober 2017	Kegiatan hari ini adalah mengimplementasi masukan dari siswa siswi sekolah dasar kemarin dan merapikan lokasi VR karena adanya kunjungan dari universitas. Hasilnya adalah penyesuaian gameplay untuk game Cross Fire VR.
41	20 Oktober 2017	Kegiatan hari ini adalah testing untuk game Cross Fire VR, beberapa game master dipanggil kembali untuk mencoba beberapa penyesuaian gameplay. Hasilnya, gameplay Cross Fire VR telah siap untuk dimainkan publik.
-	23 Oktober 2017	Izin tidak masuk kerja.

42	24 Oktober 2017	Kegiatan hari ini adalah menambahkan papan score dan waktu untuk mempermudah pemain dan publik melihat score dan waktu saat ini. Hasilnya, penambahan papan score dan waktu yang diletakkan disetiap sisi pemain.
43	25 Oktober 2017	Kegiatan hari ini adalah menyesuaikan rotasi lokasi game Cross Fire VR untuk dapat direkam menggunakan software OBS. Hasilnya, lokasi gameplay dirotasi agar publik dapat melihat score pemain.
44	26 Oktober 2017	Kegiatan hari ini adalah mencoba software OBS untuk merekam dan menampilkan permainan kepada publik. Hasilnya, software OBS dapat menampilkan gameplay dengan sedikit delay ketika kamera digunakan terlalu lama.
45	27 Oktober 2017	Kegiatan hari ini adalah mempersiapkan seluruh device yang digunakan untuk event besok. Hasilnya, seluruh device VR termasuk PC berhasil dipacking dan siap untuk dikirimkan ke lokasi event.
46	29 Oktober 2017	Kegiatan hari ini adalah menjadi operator stand VR di acara Cross Fire Star Final 2017. Hasilnya, banyak pemain yang mengantri untuk bermain termasuk para finalis Cross Fire.
-	30 Oktober 2017	Izin tidak masuk kerja.
47	31 Oktober 2017	Kegiatan hari ini adalah memasang kembali perangkat VR di kantor dan mengatur lokasi musuh pada game Ragnarok VR. Hasilnya, lokasi musuh dan mini boss diletakkan

		diberbagai sisi map agar pemain dapat menemukan musuh disetiap lokasi yang ada.
48	1 November 2017	Kegiatan hari ini adalah menghadiri technical meeting untuk event Ragnarok Festival 2017. Hasilnya, adalah rundown acara dan pembagian person in charge untuk masing – masing booth, saya sebagai PIC booth VR.
49	2 November 2017	Kegiatan hari ini adalah menambahkan display publik pada game Ragnarok VR. Hasilnya, penambahan display seperti pada game Cross Fire VR.
50	3 November 2017	Kegiatan hari ini adalah memperbaiki ukuran display pada game Ragnarok VR. Hasilnya, ukuran display diperbesar dan disesuaikan dengan resolusi layar.
51	6 November 2017	Kegiatan hari ini adalah mengubah metode spawn anak panah pada game Ragnarok VR. Hasilnya, perubahan metode spawn anak panah yang sebelumnya spawn destroy menjadi object pool spawner.
52	7 November 2017	Kegiatan hari ini adalah memperbaiki bug yang muncul akibat perubahan metode. Hasilnya, bug mengenai panah yang tidak dapat spawn berhasil diperbaiki dan panah dapat ditembak secara normal lagi.
53	8 November 2017	Kegiatan hari ini adalah packing dan pemasangan alat – alat VR di Mall of Indonesia. Hasilnya, booth VR berhasil didirikan namun komputer belum dapat dipasang karena kendala listrik.

54	9 November 2017	Kegiatan hari ini adalah menjaga booth VR pada event Ragnarok Festival 2017 di Mall of Indonesia. Hasilnya, terjadi kesulitan pemasangan dua kontroler karena banyaknya pemain yang bermain, bug akan diperbaiki esok hari sebelum event dimulai.
55	10 November 2017	Kegiatan hari ini adalah menjaga booth VR dan mempermudah pemasangan kontroler. Hasilnya, penambahan warna yang berbeda untuk kontroler kiri dan kanan agar pemasangan kontroler menjadi mudah dan cepat.
-	11 November 2017	Izin tidak masuk kerja.
-	12 November 2017	Izin tidak masuk kerja.
-	13 November 2017	Izin tidak masuk kerja.
56	14 November 2017	Kegiatan hari ini adalah installasi perangkat VR di kantor. Hasilnya, perangkat VR dapat digunakan dengan lancar di kantor.
57	15 November 2017	Kegiatan hari ini adalah testing untuk game Robo IO. Hasil kegiatan hari ini adalah penyesuaian gameplay pada Robo IO.
58	16 November 2017	Kegiatan hari ini adalah testing untuk game Robo IO. Hasil kegiatan hari ini adalah list bug yang terdapat dalam game.
59	17 November 2017	Kegiatan hari ini adalah testing untuk game Robo IO. Hasil kegiatan hari ini adalah memberitahukan hasil test dan bug didalam game.
60	20 November 2017	Kegiatan hari ini adalah testing Robo IO. Hasilnya, terjadi bug mengenai karakter yang tidak hilang ketika mati.

61	21 November 2017	Kegiatan hari ini adalah testing Robo IO. Hasilnya, terjadi bug mengenai karakter yang bisa menembus dinding pembatas.
62	22 November 2017	Kegiatan hari ini adalah testing Robo IO. Hasilnya, terjadi bug mengenai senjata yang tidak mengenai target.
63	23 November 2017	Kegiatan hari ini adalah testing Robo IO. Hasilnya, terjadi bug mengenai nama pemain yang menggunakan karakter simbol.
64	24 November 2017	Kegiatan hari ini adalah final testing Robo IO. Hasilnya, game Robo IO siap dipublish tanpa adanya bug.

2.2. Hasil Pekerjaan Secara Umum

Proses magang dimulai dari tanggal 18 Agustus 2017 sampai dengan 18 November 2017, namun karena kebutuhan perusahaan akan tester, saya melanjutkan magang hingga tanggal 24 November 2017. Selama menjalani proses tersebut, saya berhasil menyelesaikan tugas untuk 5 proyek game, adapun proyek tersebut adalah : Ragnarok VR, Cross Fire VR, Cross Fire VR untuk anak – anak, Plek – plek, dan Robo IO. 3 dari 5 proyek tersebut berupa VR, Plek – plek merupakan game 2D tower defense, dan Robo IO merupakan game top down shooter.

Beberapa tugas yang dikerjakan adalah redesign, develop, dan testing. Untuk game Ragnarok VR, saya ditugaskan untuk redesign game yang dibuat sebelumnya untuk mengubah seluruh aset, memperbaiki bug, dan menambah fitur. Untuk game Cross Fire VR, saya bertugas menjadi developer utama game tersebut, saya dibantu oleh tim designer untuk melengkapi game first person shooter tersebut. Cross Fire VR kemudian di redesign untuk dimainkan oleh anak – anak sekolah dasar dengan mengubah beberapa aset.

Pada game Plek – plek, saya bertugas sebagai programmer pembantu, tugas saya adalah testing gameplay dan menambahkan beberapa fitur yang diberikan programmer utama. Pada game Robo IO, saya bertugas sebagai tester, tugas saya adalah mencari bug sebanyak – banyaknya dan memberikan list bug – bug tersebut kepada programmer utama untuk diperbaiki.

2.3. Bukti Hasil Pekerjaan

Script atau code yang digunakan dalam pembuatan game ini cukup banyak dan saling bergantung satu sama lain, salah satu script penting yang mengubah game biasa menjadi virtual reality adalah script mengenai SteamVR.

```
//===== Copyright (c) Valve Corporation, All rights reserved.
//=====
//
// Purpose: Access to SteamVR system (hmd) and compositor
// (distort) interfaces.
//
//=====
//=====

using UnityEngine;
using Valve.VR;

public class SteamVR : System.IDisposable
{
    // Use this to check if SteamVR is currently active
    // without attempting
    // to activate it in the process.
    public static bool active { get { return _instance !=
    null; } }

    // Set this to false to keep from auto-initializing when
    // calling SteamVR.instance.
    private static bool _enabled = true;
    public static bool enabled
    {
        get
        {
            if (!UnityEngine.XR.XRSettings.enabled)
                enabled = false;
            return _enabled;
        }
        set
        {
            _enabled = value;
        }
    }
}
```

```

        if (!_enabled)
            SafeDispose();
    }
}

private static SteamVR _instance;
public static SteamVR instance
{
    get
    {
#if UNITY_EDITOR
        if (!Application.isPlaying)
            return null;
#endif
        if (!_enabled)
            return null;

        if (_instance == null)
        {
            _instance = CreateInstance();

            // If init failed, then auto-disable so
            // scripts don't continue trying to re-initialize things.
            if (_instance == null)
                _enabled = false;
        }

        return _instance;
    }
}

public static bool usingNativeSupport
{
    get { return
        UnityEngine.XR.XRDevice.GetNativePtr() != System.IntPtr.Zero; }
}

static SteamVR CreateInstance()
{
    try
    {
        var error = EVRInitError.None;
        if (!SteamVR.usingNativeSupport)
        {
            Debug.Log("OpenVR initialization failed.
            Ensure 'Virtual Reality Supported' is checked in Player
            Settings, and OpenVR is added to the list of Virtual Reality
            SDKs.");

            return null;
        }

        // Verify common interfaces are valid.

        OpenVR.GetGenericInterface(OpenVR.IVRCompositor_Version,
        ref error);
    }
}

```

```

        if (error != EVRInitError.None)
        {
            ReportError(error);
            return null;
        }

        OpenVR.GetGenericInterface(OpenVR.IVROverlay_Version, ref
error);

        if (error != EVRInitError.None)
        {
            ReportError(error);
            return null;
        }
    }
    catch (System.Exception e)
    {
        Debug.LogError(e);
        return null;
    }

    return new SteamVR();
}

static void ReportError(EVRInitError error)
{
    switch (error)
    {
        case EVRInitError.None:
            break;
        case
EVRInitError.VendorSpecific_UnableToConnectToOculusRuntime:
            Debug.Log("SteamVR Initialization
Failed! Make sure device is on, Oculus runtime is installed,
and OVRService_*.exe is running.");
            break;
        case EVRInitError.Init_VRClientDLLNotFound:
            Debug.Log("SteamVR drivers not found!
They can be installed via Steam under Library > Tools. Visit
http://steampowered.com to install Steam.");
            break;
        case EVRInitError.Driver_RuntimeOutOfDate:
            Debug.Log("SteamVR Initialization
Failed! Make sure device's runtime is up to date.");
            break;
        default:
            Debug.Log(OpenVR.GetStringForHmdError(error));
            break;
    }
}

// native interfaces
public CVRSystem hmd { get; private set; }
public CVRCompositor compositor { get; private set; }
public CVROverlay overlay { get; private set; }

```

```

// tracking status
static public bool initializing { get; private set; }
static public bool calibrating { get; private set; }
static public bool outOfRange { get; private set; }

static public bool[] connected = new
bool[OpenVR.k_unMaxTrackedDeviceCount];

// render values
public float sceneWidth { get; private set; }
public float sceneHeight { get; private set; }
public float aspect { get; private set; }
public float fieldOfView { get; private set; }
public Vector2 tanHalfFov { get; private set; }
public VRTextureBounds_t[] textureBounds { get; private
set; }
public SteamVR_Utils.RigidTransform[] eyes { get; private
set; }
public ETextureType textureType;

// hmd properties
public string hmd_TrackingSystemName { get { return
GetStringProperty(ETrackedDeviceProperty.Prop_TrackingSystemName
_String); } }
public string hmd_ModelNumber { get { return
GetStringProperty(ETrackedDeviceProperty.Prop_ModelNumber_String
); } }
public string hmd_SerialNumber { get { return
GetStringProperty(ETrackedDeviceProperty.Prop_SerialNumber_Strin
g); } }

public float hmd_SecondsFromVsyncToPhotons { get { return
GetFloatProperty(ETrackedDeviceProperty.Prop_SecondsFromVsyncToP
hotons_Float); } }
public float hmd_DisplayFrequency { get { return
GetFloatProperty(ETrackedDeviceProperty.Prop_DisplayFrequency_Fl
oat); } }

public string GetTrackedDeviceString(uint deviceId)
{
    var error =
ETrackedPropertyError.TrackedProp_Success;
    var capacity =
hmd.GetStringTrackedDeviceProperty(deviceId,
ETrackedDeviceProperty.Prop_AttachedDeviceId_String, null, 0,
ref error);
    if (capacity > 1)
    {
        var result = new
System.Text.StringBuilder((int)capacity);
        hmd.GetStringTrackedDeviceProperty(deviceId,
ETrackedDeviceProperty.Prop_AttachedDeviceId_String, result,
capacity, ref error);
        return result.ToString();
    }
}

```



```

        return null;
    }

    public string GetStringProperty(ETrackedDeviceProperty
prop, uint deviceId = OpenVR.k_unTrackedDeviceIndex_Hmd)
    {
        var error =
ETrackedPropertyError.TrackedProp_Success;
        var capacity =
hmd.GetStringTrackedDeviceProperty(deviceId, prop, null, 0, ref
error);
        if (capacity > 1)
        {
            var result = new
System.Text.StringBuilder((int)capacity);
            hmd.GetStringTrackedDeviceProperty(deviceId,
prop, result, capacity, ref error);
            return result.ToString();
        }
        return (error !=
ETrackedPropertyError.TrackedProp_Success) ? error.ToString() :
"<unknown>";
    }

    public float GetFloatProperty(ETrackedDeviceProperty prop,
uint deviceId = OpenVR.k_unTrackedDeviceIndex_Hmd)
    {
        var error =
ETrackedPropertyError.TrackedProp_Success;
        return hmd.GetFloatTrackedDeviceProperty(deviceId,
prop, ref error);
    }

    #region Event callbacks

    private void OnInitializing(bool initializing)
    {
        SteamVR.initializing = initializing;
    }

    private void OnCalibrating(bool calibrating)
    {
        SteamVR.calibrating = calibrating;
    }

    private void OnOutOfRange(bool outOfRange)
    {
        SteamVR.outOfRange = outOfRange;
    }

    private void OnDeviceConnected(int i, bool connected)
    {
        SteamVR.connected[i] = connected;
    }

    private void OnNewPoses(TrackedDevicePose_t[] poses)

```

```

    {
        // Update eye offsets to account for IPD changes.
        eyes[0] = new
SteamVR_Utils.RigidTransform(hmd.GetEyeToHeadTransform(EVREye.Ey
e_Left));
        eyes[1] = new
SteamVR_Utils.RigidTransform(hmd.GetEyeToHeadTransform(EVREye.Ey
e_Right));

        for (int i = 0; i < poses.Length; i++)
        {
            var connected = poses[i].bDeviceIsConnected;
            if (connected != SteamVR.connected[i])
            {
                SteamVR_Events.DeviceConnected.Send(i,
connected);
            }
        }

        if (poses.Length >
OpenVR.k_unTrackedDeviceIndex_Hmd)
        {
            var result =
poses[OpenVR.k_unTrackedDeviceIndex_Hmd].eTrackingResult;

            var initializing = result ==
ETrackingResult.Uninitialized;
            if (initializing != SteamVR.initializing)
            {
                SteamVR_Events.Initializing.Send(initializing);
            }

            var calibrating =
                result ==
ETrackingResult.Calibrating_InProgress ||
                result ==
ETrackingResult.Calibrating_OutOfRange;
            if (calibrating != SteamVR.calibrating)
            {
                SteamVR_Events.Calibrating.Send(calibrating);
            }

            var outOfRange =
                result ==
ETrackingResult.Running_OutOfRange ||
                result ==
ETrackingResult.Calibrating_OutOfRange;
            if (outOfRange != SteamVR.outOfRange)
            {
                SteamVR_Events.OutOfRange.Send(outOfRange);
            }
        }
    }
}

```

```

#endregion

private SteamVR()
{
    hmd = OpenVR.System;
    Debug.Log("Connected to " + hmd_TrackingSystemName +
":" + hmd_SerialNumber);

    compositor = OpenVR.Compositor;
    overlay = OpenVR.Overlay;

    // Setup render values
    uint w = 0, h = 0;
    hmd.GetRecommendedRenderTargetSize(ref w, ref h);
    sceneWidth = (float)w;
    sceneHeight = (float)h;

    float l_left = 0.0f, l_right = 0.0f, l_top = 0.0f,
l_bottom = 0.0f;
    hmd.GetProjectionRaw(EVREye.Eye_Left, ref l_left,
ref l_right, ref l_top, ref l_bottom);

    float r_left = 0.0f, r_right = 0.0f, r_top = 0.0f,
r_bottom = 0.0f;
    hmd.GetProjectionRaw(EVREye.Eye_Right, ref r_left,
ref r_right, ref r_top, ref r_bottom);

    tanHalfFov = new Vector2(
        Mathf.Max(-l_left, l_right, -r_left, r_right),
        Mathf.Max(-l_top, l_bottom, -r_top,
r_bottom));

    textureBounds = new VRTextureBounds_t[2];

    textureBounds[0].uMin = 0.5f + 0.5f * l_left /
tanHalfFov.x;
    textureBounds[0].uMax = 0.5f + 0.5f * l_right /
tanHalfFov.x;
    textureBounds[0].vMin = 0.5f - 0.5f * l_bottom /
tanHalfFov.y;
    textureBounds[0].vMax = 0.5f - 0.5f * l_top /
tanHalfFov.y;

    textureBounds[1].uMin = 0.5f + 0.5f * r_left /
tanHalfFov.x;
    textureBounds[1].uMax = 0.5f + 0.5f * r_right /
tanHalfFov.x;
    textureBounds[1].vMin = 0.5f - 0.5f * r_bottom /
tanHalfFov.y;
    textureBounds[1].vMax = 0.5f - 0.5f * r_top /
tanHalfFov.y;

    // Grow the recommended size to account for the
overlapping fov

```

```

        sceneWidth = sceneWidth /
Mathf.Max(textureBounds[0].uMax - textureBounds[0].uMin,
textureBounds[1].uMax - textureBounds[1].uMin);
        sceneHeight = sceneHeight /
Mathf.Max(textureBounds[0].vMax - textureBounds[0].vMin,
textureBounds[1].vMax - textureBounds[1].vMin);

        aspect = tanHalfFov.x / tanHalfFov.y;
        fieldOfView = 2.0f * Mathf.Atan(tanHalfFov.y) *
Mathf.Rad2Deg;

        eyes = new SteamVR_Utils.RigidTransform[] {
            new
SteamVR_Utils.RigidTransform(hmd.GetEyeToHeadTransform(EVREye.Ey
e_Left)),
            new
SteamVR_Utils.RigidTransform(hmd.GetEyeToHeadTransform(EVREye.Ey
e_Right)) };

        switch (SystemInfo.graphicsDeviceType)
        {
#if (UNITY_5_4)
            case
UnityEngine.Rendering.GraphicsDeviceType.OpenGL2:
#endif
            case
UnityEngine.Rendering.GraphicsDeviceType.OpenGLCore:
            case
UnityEngine.Rendering.GraphicsDeviceType.OpenGLES2:
            case
UnityEngine.Rendering.GraphicsDeviceType.OpenGLES3:
                textureType = ETextureType.OpenGL;
                break;
#if !(UNITY_5_4)
            case
UnityEngine.Rendering.GraphicsDeviceType.Vulkan:
                textureType = ETextureType.Vulkan;
                break;
#endif
            default:
                textureType = ETextureType.DirectX;
                break;
        }

        SteamVR_Events.Initializing.Listen(OnInitializing);
        SteamVR_Events.Calibrating.Listen(OnCalibrating);
        SteamVR_Events.OutOfRange.Listen(OnOutOfRange);

        SteamVR_Events.DeviceConnected.Listen(OnDeviceConnected);
        SteamVR_Events.NewPoses.Listen(OnNewPoses);
    }

    ~SteamVR()
    {
        Dispose(false);
    }
}

```

```

public void Dispose()
{
    Dispose(true);
    System.GC.SuppressFinalize(this);
}

private void Dispose(bool disposing)
{
    SteamVR_Events.Initializing.Remove(OnInitializing);
    SteamVR_Events.Calibrating.Remove(OnCalibrating);
    SteamVR_Events.OutOfRange.Remove(OnOutOfRange);

    SteamVR_Events.DeviceConnected.Remove(OnDeviceConnected);
    SteamVR_Events.NewPoses.Remove(OnNewPoses);

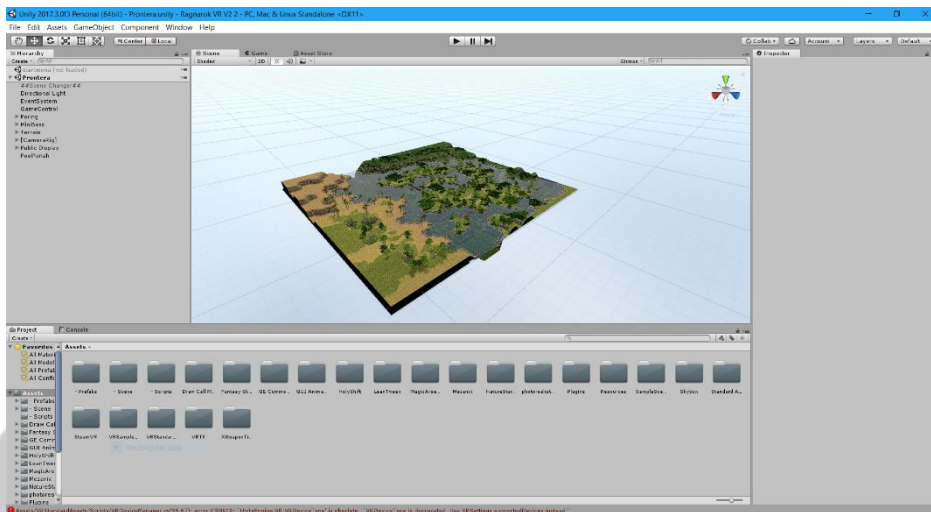
    _instance = null;
}

// Use this interface to avoid accidentally creating the
// instance in the process of attempting to dispose of it.
public static void SafeDispose()
{
    if (_instance != null)
        _instance.Dispose();
}
}

```

Script tersebut merupakan script SteamVR, script ini berfungsi sebagai pengatur jalannya virtual reality dalam game. Dalam script ini terdapat fungsi yang mengatur mengenai posisi headset dan controller VR, fungsi mengenai proyeksi tampilan game virtual reality pada headset vr dan monitor, fungsi yang mengatur mengenai tombol pada controller, dan error handling untuk perangkat virtual reality. Script ini memiliki dependensi yang tinggi dengan script individual mengenai kamera, controller, sound, dan script lain mengenai rendering.

Gambar – gambar berikut merupakan screenshot sebagai bukti hasil pekerjaan yang saya lakukan selama proses magang berlangsung.



Gambar 2.3.1. Proses pengerjaan

Untuk semua game yang saya kerjakan, langkah pertama adalah membuat scene baru melalui hierarchy, lalu memasukkan objek – objek yang digunakan didalam game. Objek wajib yang harus ada pada sebuah scene aktif adalah camera dan directional light, objek lainnya berupa karakter, musuh, dan environment. Sebuah objek dapat memiliki beberapa komponen, komponen dalam sebuah objek bisa berupa fisik, warna, audio, video, atau script yang merupakan tempat code disimpan.



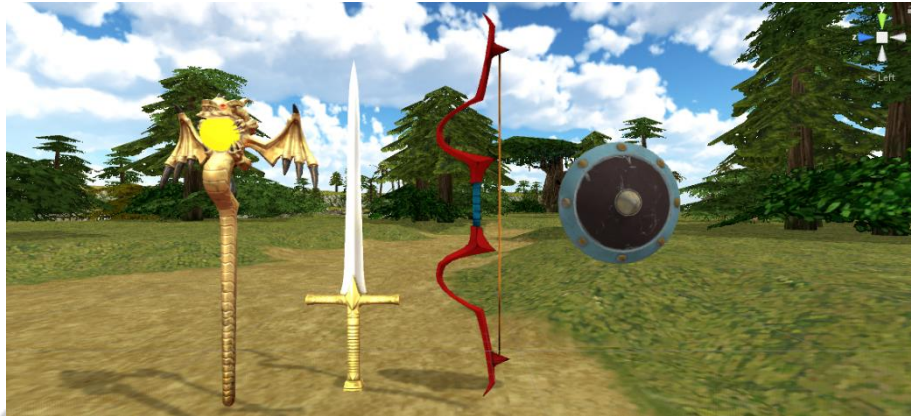
Gambar 2.3.2. Prontera

Prontera merupakan salah satu map dalam permainan Ragnarok, map ini merupakan proyek pertama yang saya kerjakan. Ketika saya diberikan proyek ini saya ditugaskan untuk mencoba mengerti metode pengerjaan game ini, saya juga menggunakan map ini sebagai proyek tutorial. Setelah mengerti metode pembuatan game ini, saya ditugaskan untuk memperbaiki beberapa bug yang masih ada didalam game ini dan menambahkan fitur – fitur baru.



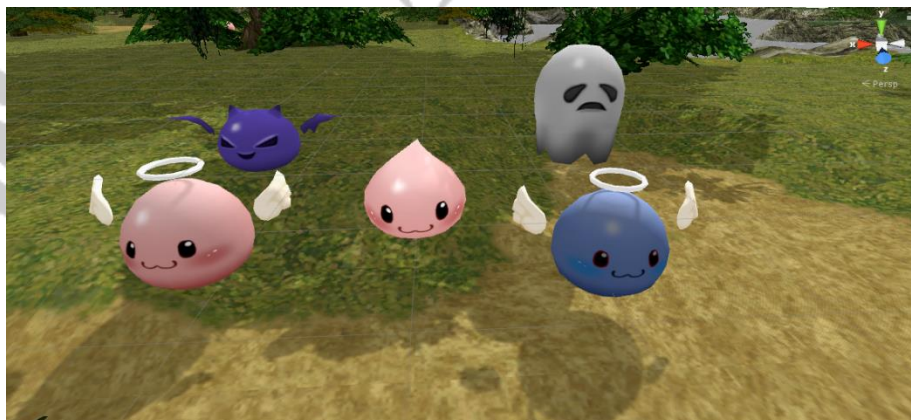
Gambar 2.3.3. Sograt Desert

Sograt Desert merupakan salah satu map dalam game Ragnarok, map ini merupakan hasil akhir dari proyek Ragnarok VR yang saya kerjakan. Hampir seluruh aset diganti dengan aset yang baru, beberapa script disederhanakan untuk menyesuaikan dengan perubahan gameplay. Lama permainan adalah 180 detik (3 menit) dan tujuan game ini adalah mengalahkan musuh sebanyak mungkin. Game ini dipersiapkan untuk event tahunan Ragnarok Festival.



Gambar 2.3.4. Senjata di Ragnarok VR

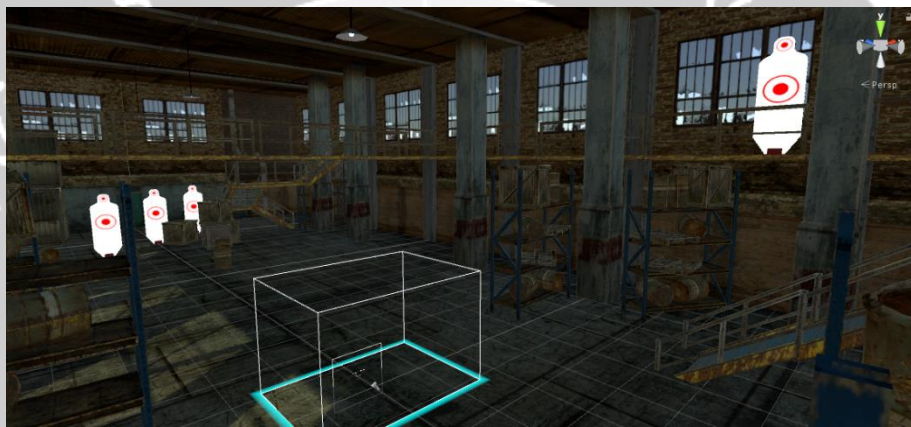
Awalnya senjata yang ada hanya pedang ganda (dual sword), saya menambahkan tongkat sihir yang dapat menembakkan bola api, panah, dan perisai. Script untuk tongkat tidak begitu sulit karena metode penembakkan sama seperti senjata api, namun yang paling sulit adalah panah, menembak anak panah membutuhkan beberapa script mengenai physics, dan untuk mencegah anak panah berlebihan yang membuat game menjadi lag, saya harus mendaur ulang anak panah dengan object pool.



Gambar 2.3.5. Musuh di Ragnarok VR

Awalnya musuh yang terdapat didalam game ini hanyalah poring dan poporing, poring dapat dilihat pada gambar diatas (tengah), poporing merupakan poring yang berwarna hijau. Saya ditugaskan untuk menambahkan marin yaitu

poring yang berwarna biru, dan membuat mereka dapat dikalahkan dengan satu kali pukulan dari senjata, sebelumnya poporing harus dipukul beberapa kali untuk dikalahkan. Kemudian saya ditugaskan untuk menambah jenis musuh baru yang harus dipukul beberapa kali untuk mengalahkannya, musuh ini berfungsi sebagai mini boss dan bertujuan untuk menarik perhatian para pemain, adapun nama musuh – musuh tersebut adalah Angeling (kiri bawah), Archangeling (kanan bawah), Deviling (kiri atas), dan Ghostring (kanan atas).



Gambar 2.3.6. Cross Fire VR

Cross Fire adalah salah satu game online bertema First Person Shooter, saya ditugaskan untuk membuat game VR adaptasi dari game tersebut untuk acara event turnamen nasional Cross Fire Star. Game yang saya buat bertemakan target practice didalam gudang, sebagian aset disediakan oleh tim design dan sebagian lagi dari unity asset store. Lama permainan adalah 90 detik dan tujuan permainan adalah menjatuhkan target sebanyak mungkin.



Gambar 2.3.7. Senjata di Cross Fire VR

Aset senjata disediakan oleh tim design, senjata sebelah kiri digunakan untuk event Cross Fire Festival dan senjata sebelah kanan digunakan ketika adanya kunjungan dari siswa siswi sekolah dasar.

BAB III

HASIL PEMBELAJARAN

3.1. Manfaat Kerja Praktek

Selama menjalani proses magang di Catlil Studio Indonesia, saya mempelajari banyak hal mengenai gaming industry. Proses magang telah memberikan pelajaran dan manfaat yang bernilai bagi saya untuk melanjutkan kehidupan menuju dunia kerja. Beberapa manfaat dan pelajaran yang saya dapat selama proses magang tersebut adalah :

1. Seluruh mata kuliah yang saya pelajari selama proses perkuliahan berlangsung hanyalah dasar dan tidak cukup untuk digunakan dalam dunia kerja. Diluar kuliah, kita harus tetap belajar terus menerus agar tidak tertinggal dalam dunia teknologi.
2. Manajemen waktu dan keuangan, lokasi kerja yang sering terjadi macet membuat saya harus bangun lebih pagi dari biasanya dan kondisi keuangan di ibukota memberikan saya pelajaran mengenai penghematan.
3. Ketika kita bekerja sesuai dengan hobi kita, tidak ada seharipun kita merasa pekerjaan kita terbebani.

3.2. Penerapan Ilmu Dalam Kerja Praktek

Ilmu yang saya terapkan selama berlangsungnya kerja praktek sebagian besarnya adalah Object Oriented Programming. Dasar dari sebuah game adalah karakter yang dimainkan, lokasi permainan, dan cara bermain, konsep OOP diberikan kepada masing – masing elemen tersebut. Script yang dibuat untuk masing – masing elemen tersebut adalah mengenai bagaimana karakter tersebut dimainkan, bagaimana lokasi berinteraksi dengan permainan, dan bagaimana permainan tersebut berlangsung. Sebuah objek bisa memiliki lebih dari satu script, dan sebuah script bisa memiliki banyak kelas.

Menurut saya hal yang penting dalam game programming adalah bagaimana mengecilkan resource yang digunakan oleh game tersebut seperti penggunaan CPU dan RAM tanpa mengorbankan gameplay. Hal penting lainnya yang saya pelajari adalah bagaimana mempersingkat code dan bagaimana menghubungkan satu script dengan yang lainnya.



BAB IV

KESIMPULAN

Setelah menyelesaikan proses magang di Catlil Studio Indonesia, maka kesimpulan yang dapat saya ambil adalah :

1. Magang dapat menjadi sebuah media pembelajaran dalam mengembangkan kemandirian mental mahasiswa di dunia kerja.
2. Magang dapat melatih kepekaan mahasiswa untuk mencari solusi masalah yang dihadapi dalam dunia kerja.
3. Magang mampu mengenalkan mahasiswa terhadap suasana kerja yang sebenarnya, sehingga dapat membangun etos kerja yang baik, serta sebagai upaya dalam memperluas pengalaman bekerja.
4. Magang dapat memberikan manfaat bagi mahasiswa, perguruan tinggi, dan bagi perusahaan tempat dilaksanakannya proses kerja praktek atau magang.
5. Magang dapat menjadi tolak ukur sejauh mana mahasiswa mampu menerapkan ilmu yang sudah dipelajari selama berada dibangku perkuliahan.