

Rancang Bangun Game Edukasi Go Green “Save Our Earth” Dengan Menggunakan Metode Finite State Machine (FSM)

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

Oleh:

AZIS FAUZAN MUHADZDZIB NIM. 1341180081



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2017

Rancang Bangun Game Edukasi Go Green “Save Our Earth” Dengan Menggunakan Metode Finite State Machine (FSM)

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

Oleh:

AZIS FAUZAN MUHADZDZIB NIM. 1341180081



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2017

HALAMAN PENGESAHAN

Rancang Bangun Game Edukasi Go Green “Save Our Earth” Dengan Menggunakan Metode Finite State Machine (FSM)

Disusun oleh:

AZIS FAUZAN MUHADZDZIB NIM. 1341180081

Skripsi ini telah diuji pada tanggal Juli 2017

Disetujui oleh:

1.	Penguji I	:	
2.	Penguji II	:	
3.	Pembimbing I	:	<u>Dhebys Suryani Hormansyah,S.KOM.,MT</u> NIP. 198311092014042001
4.	Pembimbing II	:	<u>Dyah Ayu Irawati, S.T., M.Cs.</u> NIP. 19840708 200812 2 001

Mengetahui,

Ketua Jurusan
Teknologi Informasi

Ketua Program Studi
Teknik Informatika

Rudy Ariyanto, ST., MCs.
NIP. 19711110 199903 1 002

Ir. Deddy Kusbianto P., M.MKom.
NIP. 19621128 198811 1 001

PERNYATAAN

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya/kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, Agustus 2017

Azis Fauzan Muhadzdzib

ABSTRAK

Azis Fauzan Muhadzdzib. “Implementasi FSM (Finite State Machine) pada Game “Go Green Save Our Earth”. **Pembimbing: (1) Dhebys Suryani Hormansyah, S.Kom, MT, (2) Dyah Ayu Irawati ST.,M.Cs**

Skripsi, Program Studi Teknologi Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.

Beraneka macam genre *game* dapat dimainkan pada platform yang tersedia. Tetapi sayangnya banyak *game* yang beredar kurang menonjolkan manfaat dan tujuan, bahkan terdapat beberapa *game* yang mengandung unsur pornografi dan kekerasan yang tidak baik untuk anak dibawah umur. Kebanyakan anak – anak, remaja bahkan dewasa menjadikan *game* sebagai pelepas penat, hiburan ditengah kesibukan, bahkan hobi.

Dari permasalahan tersebut, penulis akan membangun sebuah aplikasi *game* edukasi dengan mengangkat tema *go green save our earth*. Pada aplikasi ini akan diimplementasikan kecerdasan buatan dengan metode *Finite State Machine* (FSM) dimana akan mempermudah mengatur perilaku dari objek saat terjadi perubahan keadaan atau kondisi dalam permainan. Seiring dengan berkembangnya *game*, maka seseorang lebih memilih untuk memanfaatkan *dekstop* sebagai salah satu sarana bermain *game*, dimana aplikasi ini akan menggunakan sistem operasi *Dekstop*. Aplikasi ini dibangun sebagai sarana hiburan dan edukasi mengenal lebih dalam tentang *go green* melindungi bumi.

Hasil akhir dari *game* ini adlah sebuah alur cerita permainan perlindungan bumi yang dilengkapi dengan fitur penambahan materi *global warming* dan efek *rumahkaca*, cetak hasil dan juga karakter sampah yang dikendalikan oleh kecerdasan buatan.

Keyword : kecerdasan buatan, *finite state machine*, edukasi

ABSTRACT

Azis Fauzan Muhadzdzib. “Implementasi FSM (Finite State Machine) pada Game “Go Green Save Our Earth”. **Pembimbing: (1) Dhebys Suryani Hormansyah, S.Kom, MT, (2) Dyah Ayu Irawati ST.,M.Cs**

Final Report, Informatics Management Study Programme, Department of Information Technology, State Polytechnic of Malang, 2017.

A wide variety of game genres can be played on available platforms. Unfortunately many games in circulation less emphasize the benefits and goals, there are even some games that contain elements of pornography and violence that is not good for minors. Most children, teenagers and even adults make the game as a tired release, entertainment amid busyness, even a hobby.

From these problems, the author will build an educational game app by raising the go green save our earth. In this application will be implemented artificial intelligence with Finite State Machine method (FSM) which will make it easier to set the behavior of the object when there is a change of state or condition in the game. Along with the development of the game, then someone would prefer to use the mobile device as one means of playing games, where this application will use the Desktop operating system. This application was built as a means of entertainment and education to know more about the go green save our earth

The end result of this game was a game story line of the Earth's shield cover fitted with the addition of the material and the effects of global warming than the print results, and also the garbage characters controlled by artificial intelligence.

Keyword : *artificial intelligence, finite state machine, education*

KATA PENGANTAR

Puji Syukur kami panjatkan kehadiran Allah AWT atas segala rahmat dan hidayah-Nya, shalawat serta salam tetap tercurahkan kepada Nabi Muhammad SAW, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun *game* Monopoli Edukasi dengan Latar Belakang Pengetahuan Adat Istiadat di Indonesia”.

Penyusunan skripsi ini dimaksudkan untuk memenuhi salah satu syarat dalam menyelesaikan program D4 Teknik Informatika. Keberhasilan penyusunan skripsi ini juga tidak lepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada berbagai pihak yang telah membantu dalam penyusunan skripsi ini, diantaranya:

1. Bapak Ir. Tundung Subali Patma, M.T., selaku Direktur Politeknik Negeri Malang.
2. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi
3. Bapak Ir. Deddy Kusbianto P., M.MKom., selaku ketua program studi Teknik Informatika
4. Bapak Ridwan Rismanto, SST., M.Kom selaku Dosen Pembina Akademik Teknik Informatika Politeknik Negeri Malang.
5. Ibu Dhebys Suryani Hormansyah, S.Kom., MT selaku pembimbing pertama.
6. Ibu Dyah Ayu Irawati ST., M.Cs selaku pembimbing kedua.
7. Kedua orang tua yang selalu mendoakan dan memberikan dukungan baik moril maupun material sehingga diberikan kelancaran dalam menyelesaikan skripsi dengan baik.
8. Seluruh teman-teman seperjuangan D4 Teknik Informatika 2013 yang telah memberikan support serta masukan sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
9. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Skripsi dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Dengan menyadari keterbatasan pada skripsi ini, tentu laporan ini masih jauh dari sempurna. Untuk itu penulis dengan senang hati menerima kritik dan saran yang bersifat membangun dari para pembaca laporan ini. Semoga laporan skripsi ini dapat memberi manfaat bagi kita semua.

Malang, Agustus 2017

Azis Fauzan Muhadzdzib

DAFTAR ISI

HALAMAN PENGESAHAN.....	xiv
PERNYATAAN.....	xv
DAFTAR ISI.....	xx
DAFTAR TABEL.....	xxiii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
BAB I. PENDAHULUAN	1
1.1 Rumusan Masalah	2
1.2 Tujuan Penelitian.....	2
1.3 Batasan Masalah.....	2
BAB II. LANDASAN TEORI.....	5
2.1 Definisi Implementasi	5
2.2 FSM (<i>Finite State Machine</i>).....	5
2.3 Definisi Game.....	6
2.4 Kecerdasan Buatan	7
2.5 Go Green	7
2.6 Work Breakdown Structure.....	7
2.7 <i>Use Case</i> Diagram.....	8
2.8 <i>Flowchart</i>	8
2.9 <i>Storyboard</i>	9
2.10 <i>GDD (Game Design Document)</i>	10
BAB III. METODOLOGI PENELITIAN	13
3.1 Studi Pustaka	13
3.2 Metodologi Pengambilan Data	13
3.3 Perancangan Sistem.....	13
3.3.1 Konsep (<i>Concept</i>).....	13
3.3.2 Desain/Perancangan (<i>Design</i>)	14
3.3.3 Pengumpulan Bahan (<i>Material Collecting</i>)	14
3.3.4 Pembuatan (<i>Assembly</i>)	15
3.3.5 Pengujian (<i>Testing</i>)	15
3.3.6 Distribusi (<i>Distribution</i>).....	15
BAB IV. ANALISIS DAN PERANCANGAN	16

4.1.	Analisis	16
4.1.1	Gambaran Umum Sitem.....	16
4.2	GDD (Game Design Document)	16
4.2.1	Gambaran Umum Game	16
4.2.2	Alur Permainan	17
4.2.3	Penampilan Umum Game	17
4.3	Perancangan.....	18
4.3.1	Perancangan Edukasi	18
4.3.2	Perancangan Karakter	18
4.3.3	Perancangan <i>Gameplay</i>	20
4.3.4	Perancangan kondisi musuh.....	20
4.3.6	<i>Work Breakdown Structure</i>	22
4.3.7	<i>Use case Diagram</i>	23
4.5.8	Perancangan <i>Storyboard</i>	26
BAB V.	IMPLEMENTASI.....	31
5.1	Implementasi Pembuatan <i>Asset</i> Gambar	31
5.2.1	Pembuatan Karakter pada Corel Draw X7.....	31
5.3	Implemetasi Pada Unity	34
5.3.1	Menu Utama.....	34
5.3.2	Edukasi.....	36
5.3.3	Pengaturan Kamera	37
5.3.4	Pengaturan <i>Player</i>	38
5.3.5	Pengaturan <i>Enemy Sampah Anorganik</i>	40
5.3.6	Pengaturan <i>Waktu Enemy</i>	41
5.3.7	Pengaturan Score, <i>suhu</i> dan Healthbar	42
5.3.8	Pengaturan <i>Pause dan Game Over</i>	42
5.3.9	Implementasi Metode Finite State Machine (FSM) pada <i>Game</i>	44
BAB VI.	PENGUJIAN DAN PEMBAHASAN	48
6.1	Pengujian Sistem	48
6.1.1	Rencana Pengujian	48
6.1.2	Pengujian <i>Alpha</i>	49
6.1.3	Pengujian <i>Betha</i>	51
6.1.4	Hasil Pengujian Metode <i>Finite State Machine (FSM)</i>	52
6.1.5	Hasil Pengujian Perangkat	53
6.2	Pembahasan	55

6.2.1 Hasil Pengujian Alpha.....	55
6.2.2 Hasil Pengujian <i>Betha</i>	55
BAB VII. PENUTUP.....	59
7.1 Kesimpulan.....	59
7.2 Saran	59
DAFTAR PUSTAKA	xiii

DAFTAR TABEL

Tabel 4. 1 Tabel Karakter Player	18
Tabel 4. 2 Tabel Karakter Sampah.....	19
Tabel 4. 3 Tabel Karakter Enemy	19
Tabel 4. 4 menjelaskan rancangan alur state Finite State Machine permainan go green save our earth pada sampah anorganik.....	21
Tabel 4. 5 Use case play.....	24
Tabel 4. 6 Use case Tutorial.....	25
Tabel 4. 7 Use case Setting	25
Tabel 4. 8 Use case Exit.....	26
Tabel 4. 9 Storyboard Game Go Green “Save Our Earth”	26
Tabel 5. 1 Penjelasan Keterangan Diagram State pada sampah	44
Tabel 5. 2 Tabel Transisi Diagram State Sampah.....	46
Tabel 6 1 Daftar Rencana Pengujian.....	48
Tabel 6 2 Daftar Pengujian Alpha Game “Go Green “Save Our Earth”	49
Tabel 6 3 Daftar Pertanyaan Kuesioner	52
Tabel 6 4 Daftar perangkat uji coba.....	53
Tabel 6 5 Pengujian User	55
Tabel 6 6 Hasil Kuesioner Penilaian Game	56
Tabel 6 7 Hasil Pengujian Pertanyaan Pertama	56
Tabel 6 8 Hasil Pengujian Pertanyaan Kedua.....	57
Tabel 6 9 Hasil Pengujian Pertanyaan Ketiga.....	57
Tabel 6 10 Hasil Pengujian Pertanyaan Keempat.....	58

DAFTAR GAMBAR

Gambar 2 1 Diagram state Finite State Machine	6
Gambar 2 2 Contoh Work Breakdown Structure	8
Gambar 2 3 Contoh use case	8
Gambar 2 4 Contoh notasi Flowchart	9
Gambar 2 5 Contoh Storyboard	10
Gambar 2 6 Diagram Multimedia Development Life Cycle.....	13
Gambar 4 2 Alur Metode Pada Game	21
Gambar 4 3 Flowchart Game	22
Gambar 4 4 Work Breakdown Structure.....	23
Gambar 4 5 Use case Diagram Game Go Green “Save Our Earth”	24
Gambar 5 1 Sketsa Player	31
Gambar 5 2 Hasil Gambar Player	32
Gambar 5 3 Gambar Sprites Player.....	32
Gambar 5 4 Hasil Gambar Sampah Organik.....	33
Gambar 5 5 Hasil Gambar Sampah Anorganik.....	33
Gambar 5 6 Gambar Sprites Ledakan Api	34
Gambar 5 7 Tampilan Awal Unity 5.5	34
Gambar 5 8 Tampilan Inspector Menu Utama.....	35
Gambar 5 9 Inspector Animasi Setting	35
Gambar 5 10 Animator untuk Animasi Setting.....	36
Gambar 5 11 Tampilan Hasil Menu Utama	36
Gambar 5 12 Inspector Edukasi	36
Gambar 5 13 Tampilan Edukasi.....	37
Gambar 5 14 Inspector Kamera	37
Gambar 5 15 Tampilan Hasil Kamera	38
Gambar 5 16 Inspector Player.....	38
Gambar 5 17 RigidBody 2D Player	39
Gambar 5 18 Animasi player idle dan run	39
Gambar 5 19 Animator Player	40
Gambar 5 20 Inspector Enemy Sampah Anorganik.....	40
Gambar 5 21 Tampilan Circle Collider 2D Enemy	41
Gambar 5 22 Animasi Enemy Sampah Anorganik	41
Gambar 5 23 Animator Enemy Sampah Anorganik	41
Gambar 5 24 Inspector Waktu Musuh	42
Gambar 5 25 Tampilan jumlah pohon, Jumlah Hutan, dan HealthBar, suhu.	42
Gambar 5 26 Inspector Pause Menu	43
Gambar 5 27 Tampilan Panel Pause	43
Gambar 5 28 Tampilan Panel Game Over	43
Gambar 5 29 Tampilan Diagram FSM Pada sampah.....	44

DAFTAR LAMPIRAN

- Lampiran 1 Coding
- Lampiran 2 Contoh Kuisisioner
- Lampiran 3 Profil Penulis

BAB I. PENDAHULUAN

Salah satu fenomena alam yang sering terjadi dan banyak dibicarakan oleh masyarakat adalah terjadinya perubahan cuaca yang tidak mengikuti pola pergantian musim. Di Indonesia sendiri terdapat dua musim, yaitu musim penghujan dan musim kemarau. Kedua musim tersebut dijadikan sebagai acuan untuk menentukan cuaca. Umumnya pada bulan April sampai September di Indonesia akan terjadi musim kemarau dan pada bulan Oktober sampai Maret akan mengalami musim penghujan. Pemanasan global (global warming) merupakan suatu peristiwa meningkatnya temperatur di bumi yang dapat berakibat pada kelangsungan hidup makhluk hidup di bumi. Terjadinya peningkatan temperatur di bumi disebabkan oleh adanya efek rumah kaca karena adanya peningkatan gas-gas seperti karbon dioksida (CO_2), metana (CH_4), nitro oksida (NO_2), dan CFC yang menyebabkan panas dari matahari yang masuk ke bumi terperangkap di dalam atmosfer bumi sehingga temperatur di bumi meningkat. Peningkatan temperatur bumi yang disebabkan oleh efek rumah kaca merupakan masalah serius yang dihadapi oleh manusia. Apabila masalah ini tidak segera diselesaikan dengan solusi yang tepat maka dapat menyebabkan masalah-masalah yang lebih serius seperti mencairnya es di kutub yang dapat berakibat naiknya volume air laut sehingga pulau-pulau kecil akan menghilang karena tertutup oleh air laut dan tergenangnya kawasan pesisir dan pantai.

Untuk mengatasi permasalahan mengenai pemanasan global semua kembali ke tangan manusia karena manusialah sumber utama terjadinya pemanasan global. Sebagai manusia yang berakal, kita harus segera membatasi perbuatan kita karena fenomena pemanasan global semakin nyata dampaknya. Disini saya akan mengambil langkah kecil untuk mengatasi masalah pemanasan global, langkah yang saya akan ambil yaitu dengan cara membuat game edukasi tentang go green yang bertemakan save our earth yang mempunyai arti melindungi bumi, game yang akan saya buat ditujukan untuk anak-anak kecil. Karena pembelajaran tentang edukasi ini harus diterapkan sejak dini supaya terbiasa akan tatacara melindungi bumi dan menyayangi bumi supaya terbiasa hidup sehat dan tenang.

Dari permasalahan tersebut, penulis akan membangun sebuah aplikasi *game* edukasi dengan mengangkat *go green* tentang pemanasan global. Pada aplikasi ini akan diimplementasikan kecerdasan buatan dengan metode *Finite State Machine* (FSM) dimana akan mempermudah mengatur perilaku dari objek saat terjadi perubahan keadaan atau kondisi dalam permainan. Dimana aplikasi ini akan menggunakan sistem operasi *Dekstop*. Aplikasi ini dibangun sebagai sarana hiburan dan edukasi mengenal bagaimana cara melindungi bumi dengan mudah .

1.1 Rumusan Masalah

Berdasarkan latar belakang masalah yang diuraikan, maka dapat dirumuskan masalah sebagai berikut :

1. Bagaimana merancang dan membangun *game* edukasi *go green* berlatar pemanasan global “Save Our Earth” berbasis *Dekstop* sehingga dapat menyampaikan nilai edukasi penyelamatan bumi dari pemanasan global?
2. Bagaimana menerapkan metode *Finite State Machine* pada *Game* Edukasi Go Green “Save Our Earth” Berbasis *Dekstop*?

1.2 Tujuan Penelitian

Tujuan dari penelitian ini:

1. Merancang dan membangun sebuah *game* sebagai media pembelajaran untuk mencegah dari pemanasan global di mana *game* ini berjenis *platformer* dengan misi menyelamatkan bumi dari pemanasan global.
2. Menerapkan Metode *Finite State Machine* pada *Game* Edukasi Go Green “Save Our Earth” Berbasis *Dekstop*.

1.3 Batasan Masalah

Agar pembahasan lebih terarah, maka penulis memberikan batasan-batasan pembahasan masalah, yaitu:

1. *Game* ini Membahas tentang edukasi Go Green “Save Our Earth”.
2. *Game* berformat 2 Dimensi (2D)
3. *Game* bergenre *platformer*.
4. Metode yang digunakan yaitu *Finite State Machine*.

5. *Game* ini dimainkan satu pemain (*Single Player*)
6. *Game* ini diperuntukan untuk anak di atas usia 9 – 12 Tahun.

1.4 Sistematika Penulisan

1. BAB I PENDAHULUAN

Pada bab ini dijelaskan mengenai hal-hal yang bersifat umum seperti latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

2. BAB II LANDASAN TEORI

Pada bab ini berisi teori-teori yang mendasari dan berkaitan dengan masalah perencanaan dan pembuatan aplikasi yang digunakan untuk memudahkan pemahaman dan pemecahan terhadap masalah yang ada.

3. BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan langkah-langkah yang digunakan penulis untuk memilih metode, teknik, prosedur apa yang tepat, dan *tools* apa yang akan digunakan sehingga setiap tahap penelitian dapat dilakukan dengan tepat.

4. BAB IV ANALISIS DAN PERANCANGAN

Berisi pembahasan mengenai analisis perencanaan dan pembuatan terhadap sistem. Termasuk di dalamnya adalah *Flowchart*, *Work Breakdown Structure*, *Use Case*, *GDD (Game Desain Document)*, Perancangan *asset* gambar.

5. BAB V IMPLEMENTASI

Membahas tentang uraian mengenai proses implementasi dari sistem yang telah dirancang sebelumnya ke dalam bahasa pemrograman. Disertai dengan gambaran dengan *interface* aplikasi beserta penjelasan.

6. BAB VI PENGUJIAN DAN PEMBAHASAN

Membahas tentang hasil uji coba aplikasi, jalannya uji coba bertahap sistem yang dikembangkan. Dapat terdiri dari metode uji coba, tujuan uji coba, proses uji coba serta analisa hasil uji coba.

7. BAB VII KESIMPULAN

Bab ini berisi kesimpulan yang diperoleh dari hasil pengujian dan saran-saran yang dibutuhkan untuk kesempurnaan sistem sehingga sistem tersebut dapat disempurnakan dan dikembangkan kemudian hari.

BAB II. LANDASAN TEORI

Pada bab ini akan membahas mengenai teori – teori yang digunakan dalam pembuatan aplikasi Implementasi Metode FSM (*Finite State Machine*) pada Game Go Green Save Our Earth.

2.1 Definisi Implementasi

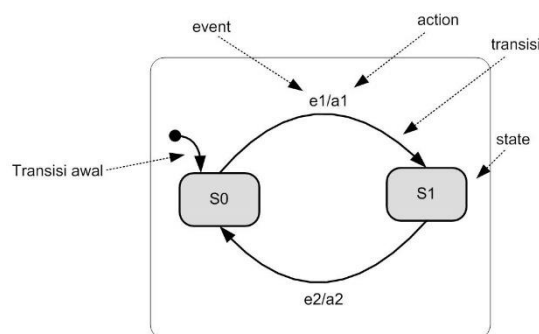
Implementasi berasal dari bahasa Inggris yaitu *to implement* yang berarti mengimplementasikan. Implementasi adalah suatu tindakan atau pelaksanaan rencana yang telah disusun dengan cermat dan rinci. Implementasi ini biasanya selesai setelah dianggap permanen.

Implementasi adalah bermuara pada aktivitas, aksi, tindakan, atau adanya mekanisme suatu sistem. Implementasi bukan sekedar aktivitas, tetapi suatu kegiatan yang terencana dan untuk mencapai tujuan kegiatan (Usman, Nurdin, 2012).

2.2 FSM (*Finite State Machine*)

Finite State Machine (FSM) terdiri dari serangkaian *state* yang menentukan pengambilan keputusan. Setiap *state* dapat berpindah ke *state* lainnya jika memenuhi kondisi yang telah ditentukan sebelumnya (Rostianingsih Silvia , 2011)

Finite State Machine (FSM) merupakan sebuah mesin abstrak yang berfungsi untuk mendefinisikan sekumpulan kondisi yang menentukan kapan suatu *state* harus berubah. Setiap *state* yang sedang dijalankan tersebut menentukan perilaku yang terjadi pada objek yang bersangkutan. Diagram state FSM digambarkan pada gambar 2.1 sebagai berikut (Feisal Ferdian, 2015)



Gambar 2 1 Diagram state Finite State Machine

Finite State Automata (FSA) bukanlah mesin fisik tetapi suatu model matematika dari suatu sistem yang menerima input dan output. FSA merupakan mesin automata dari bahasa regular (tipe 3). Suatu FSA memiliki state yang banyaknya berhingga, dan dapat berpindah dari suatu state ke state lain. Perubahan state dinyatakan oleh fungsi transisi. Suatu FSA secara formal dinyatakan oleh 5 (lima) tuple $M = (Q, \Sigma, \delta, S, F)$ dimana:

- Q = Himpunan State/ kedudukan
- Σ = himpunan symbol input/ masukan
- Δ = Fungsi transisi
- S = State awal/ kedudukan awal
- F = Himpunan state akhir

FSA berdasarkan pada pendefinisian kemampuan perubahan statenya bisa dikelompokkan kedalam *Deterministic Finite Automata* (DFA) dan *Non Deterministic Finite Automata* (NFA). Pada DFA dari suatu state ada tepat satu state berikutnya untuk setiap simbol input (masukan) yang diterima. Pada NFA dari suatu state bisa terdapat nol (0) atau satu (1), atau lebih busur keluar (transisi) berlabel symbol yang sama. Jadi setiap pasangan state input, bisa memiliki 0 atau lebih pilihan untuk state berikutnya.

2.3 Definisi Game

Permainan (games) adalah koneksi antara pemain yang berinteraksi satu sama lain dengan mengikuti aturan-aturan tertentu untuk mencapai tujuan tertentu pula. Jadi permainan adalah cara bermain dengan mengikuti aturan-aturan tertentu yang dapat dilakukan secara individu maupun berkelompok guna mencapai tujuan tertentu. Alat permainan adalah semua alat bermain yang dapat digunakan oleh peserta didik untuk memenuhi naluri bermainnya dan memiliki berbagai macam sifat, seperti bongkar pasang, mengelompokkan, memadukan, mencari padanannya, merangkai, membentuk, atau menyusun sesuai dengan bentuk aslinya (Marimin, 2004).

2.4 Kecerdasan Buatan

Definisi Kecerdasan Buatan atau sering disebut Artificial Intelligence (AI) merupakan cabang dari ilmu komputer yang dalam merepresentasi pengetahuan lebih banyak menggunakan bentuk simbol-simbol daripada bilangan, dan memproses informasi berdasarkan metode heuristic atau dengan berdasarkan sejumlah aturan. Ada tiga tujuan kecerdasan buatan, yaitu, membuat komputer lebih cerdas, mengerti tentang kecerdasan, dan membuat mesin yang lebih. Kecerdasan adalah kemampuan untuk belajar dan mengerti dari pengalaman, memahami pesan yang kontradiktif dan ambigu, menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah serta menyelesaikannya dengan efektif (Suparman, 2008).

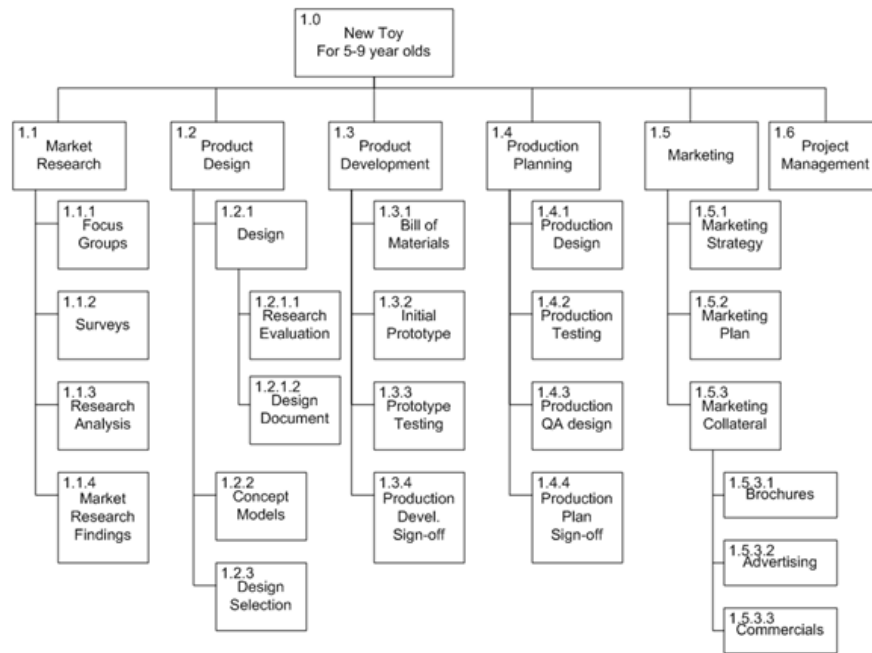
2.5 Go Green

Go green dalam artian luas lingkungan go green berarti ajakan “mari hidup hijau”, dengan arti mari kita hidup lebih ramah lingkungan & berbuat sesuatu yang baik baik untuk bumi

2.6 Work Breakdown Structure

Suatu metode perorganisasian proyek menjadi struktur pelaporan hierarkis. WBS digunakan untuk melakukan *breakdown* atau memecahkan tiap proses pekerjaan menjadi lebih detail. Hal ini dimaksudkan agar proses perencanaan proyek memiliki tingkat yang lebih baik.[6]

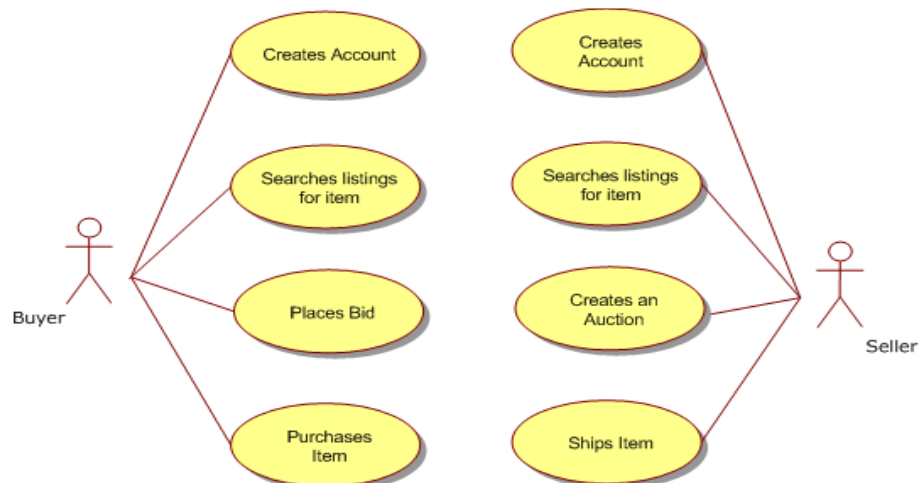
Contoh *Work Breakdown Structure* dijelaskan pada Gambar 2.2.



Gambar 2.2 Contoh Work Breakdown Structure

2.7 Use Case Diagram

Untuk mengetahui apa saja yang dapat dilakukan oleh pengguna, diperlukan *use case*. *Use case* adalah suatu pola atau gambaran yang menunjukkan perilaku atau kebiasaan sistem. *Use case* Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.[7] Contoh *use case* dijelaskan pada Gambar 2.3.



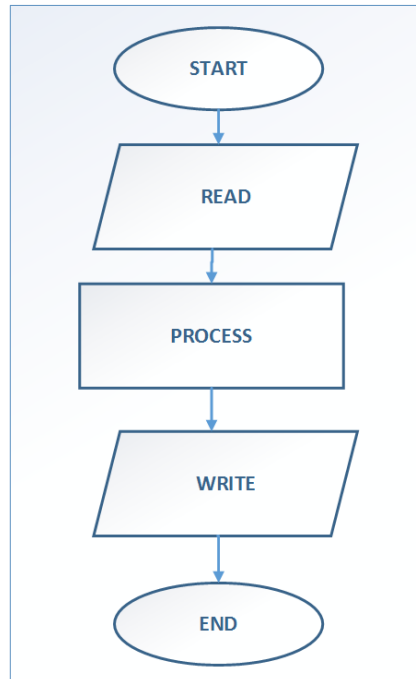
Gambar 2.3 Contoh use case

2.8 Flowchart

Flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Tujuan dari

pembuatan *flowchart* ini untuk memudahkan *user* dalam menggunakan aplikasi. Dimulai dengan *opening game* dan proses jalannya aplikasi.[8]

Contoh notasi *flowchart* dijelaskan pada Gambar 2.4.

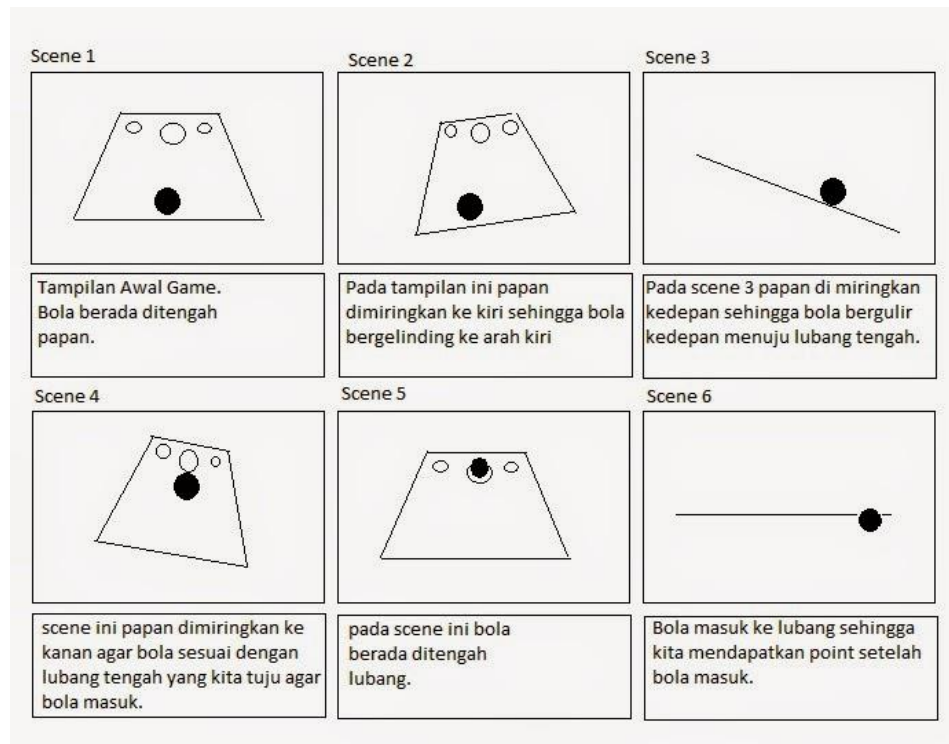


Gambar 2 4 Contoh notasi Flowchart

2.9 Storyboard

Sebelum pembuatan *game* , maka perlu dibuat sebuah *storyboard* agar pembuatan *game* lebih mudah. *Storyboard* adalah visualisasi ide dari aplikasi yang akan dibangun, sehingga dapat memberikan gambaran dari aplikasi yang akan dihasilkan. *Storyboard* dapat dikatakan juga *visual script* yang akan dijadikan *outline* dari sebuah proyek, ditampilkan *shot by shot* yang biasa disebut dengan istilah *scene*. *Storyboard* sekarang lebih banyak digunakan untuk membuat kerangka pembuatan *website* dan proyek media interaktif lainnya seperti iklan, film pendek, *game* s, media pembelajaran interaktif ketika dalam tahap perancangan atau desain.[9]

Contoh *Storyboard* digambarkan pada Gambar 2.5.



Gambar 2 5 Contoh Storyboard

2.10 GDD (Game Design Document)

Desain permainan adalah proses mendesain atau merancang konten dan peraturan permainan dalam tahap pra-pembuatan dan perancangan tata permainan, lingkungan, alur cerita, dan karakter selama tahap pembuatan. Pada tahap pembuatan game terdapat sebuah dokumen yang menampung seluruh gambaran dari game yang akan didesain. Dokumen desain game atau Game Design Document (GDD) adalah gambaran bagaimana sebuah game komputer akan dibuat. Setiap hal-hal yang detail di dalam game harus disebutkan di dalamnya. Jika tidak terdapat di dalam dokumen ini, berarti seharusnya tidak ada dalam game. Permasalahannya, tidak ada dokumen desain game (GDD) umum yang cocok untuk semua genre game. Contohnya game Pacman, SimCity atau Doom.[10]

2.11 Unity Game Engine

Pada penelitian ini penulis memilih *Unity* yang merupakan salah satu *game engine*. *Unity game engine* adalah sebuah *game engine multiplatform*. *Unity* digunakan untuk membuat *game* untuk *web plug in*, *dekstop platform*,

console, dan perangkat *mobile*. Unity dirilis pada tahun 2005, oleh David Helgason, Nicholas Francis, dan Joachim Ante.

Pada pembuatan *game*, terdapat beberapa istilah umum yang ada pada *Unity*, diantaranya:

- *Scenes*, merupakan istilah pada *Unity* yang digunakan untuk menyebutkan sebuah *level* atau *menu*.
- *Game Object*, merupakan sebuah object yang ada pada *Unity*. *Game object* memiliki beberapa komponen yang dapat ditambahkan atau dikurangi dengan cara *drag and drop*.
- *Components*, merupakan reaksi baru, bagi objek seperti *collison* memunculkan partikel, dan sebagainya.
- *Asset Store*, merupakan sebuah *website* yang disediakan oleh *Unity*, yang menyediakan *desain*, *plug in*, hingga *sounds* baik yang *open source* maupun berbayar.
- *Prefabs*, merupakan tempat untuk menyimpan satu jenis *game objects*, sehingga mudah untuk diperbanyak.

Pada setiap obyek di *Unity*, terdapat beberapa *component* yang bisa dilihat pada *inspector*. Pada kolom *hierarchy* menampilkan daftar obyek yang ada pada suatu *scene* berserta dengan *hierarchy* dari *parent-child* pada setiap obyek. Pada kolom *project*, menampilkan *asset* dari *project* sesuai dengan penampilan *game* yang dibuat secara keseluruhan.

Setiap *script* diturunkan dari *parent class* utama yang ada pada *library Unity* yang bernama *MonoBehaviour*. Pada *MonoBehaviour* terdapat method dasar yang biasa digunakan pada pemrograman *game*, seperti *method Start* yang dieksekusi pada awal *game* dijalankan, dan *method Update* yang dieksekusi pada setiap frame pada *game*. Script pada *Unity* dikategorikan sebagai *component* sehingga setiap *script* dapat di masukan ke dalam obyek seperti *component* lain.

Dalam *game engine Unity* terdapat berbagai macam bahasa pemrograman seperti *javascript*, *C#*, dan *boo*. Di *Unity* tidak dapat melakukan desain atau modeling, dikarenakan *game engine* ini dikembangkan bukan untuk mendesain *asset-asset* untuk *game*. Ada beberapa fitur yang terdapat pada *Unity* seperti *audio reverb*

zone, sky box, particle effect dan masih banyak yang lain. Pada *Unity* juga dapat edit texture dari editor seperti *photoshop, illustrator, dll.*[11]

Features (*Scripting*) di dalam *Unity* adalah sebagai berikut:

- Mendukung 3 bahasa pemrograman, JavaScript, C#, dan Boo.
- *Flexible and EasyMoving, rotating*, dan *objects* hanya perlu sebaris kode. Begitu juga dengan duplicating, removing, dan changing properties.
- *Multiplatform game* bisa di *deploy* di *PC, Mac, Wii, iPhone, iPad*, dan *Browser, Android*.
- *Visual Properties Variables* yang di definisikan dengan scripts ditampilkan pada editor. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker*.
- Berbasis .NET, perjalanan program dilakukan dengan *Open Source .NET platform, Mono*.

BAB III. METODOLOGI PENELITIAN

Dalam metode penelitian ini akan dijelaskan langkah-langkah yang dilakukan untuk merancang game Go Green Save Our Earth sebagai berikut:

3.1 Studi Pustaka

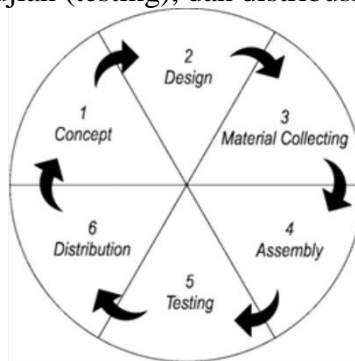
Studi pustaka ini sebagai dasar teori yang akan melandasi pengerjaan aplikasi ini dengan mencari studi pustaka yang berkaitan pada buku - buku. Sebagai bahan pertimbangan untuk pembuatan proposal aplikasi ini.

3.2 Metodologi Pengambilan Data

Metode pengambilan data yang digunakan dalam penelitian ini adalah pengumpulan data penunjang yang dilakukan dengan pengambilan data – data dari buku atau jurnal, buku mengenai go green dan dari internet yang digunakan sebagai landasan teori dalam penulisan laporan ini.

3.3 Perancangan Sistem

Pada tahap ini metode yang digunakan adalah metode Multimedia Development Life Cycle (MDLC). Dimana metode ini terdiri dari beberapa fase yaitu konsep (concept), design (design), pengumpulan bahan (material collecting), pembuatan (assembly), pengujian (testing), dan distribusi (distribution).



Gambar 2 6 Diagram Multimedia Development Life Cycle

3.3.1 Konsep (Concept)

Game 2D ini bergenre *platformer* dan merupakan *game* edukatif yang bertujuan untuk menyampaikan salah satu Penyelamatan bumi dari Pemanasan

Global.Game ini memiliki misi untuk menyelamatkan Bumi dari Pemanasan global dengan cara mengambil sampah organik dan anorganik[8].

3.3.2 Desain/Perancangan (*Design*)

Setelah konsep terbentuk langkah selanjutnya adalah perancangan. Berdasarkan tema yang diambil yaitu Pemanasan Global dan cara melindungi Bumi, maka dibuatlah judul *Game* Edukasi Go Green “Save Our Earth” Berbasis *Dekstop*. Proses pada tahap ini juga meliputi perancangan karakter dan latar belakang yang akan digunakan. Adapun karakter yang terdapat dalam *game* meliputi: 1 *player* dan 2 jenis sampah dan beberapa rintangan yang ditempatkan berbeda di setiap *platforme*. Latar belakang yang akan digunakan meliputi: Hutan dan kota dengan suasana siang hari yang digambarkan pada *gameplay*.

Setelah proses-proses diatas selesai, maka proses yang akan dilakukan selanjutnya adalah pemodelan karakter 2D, pemodelan latar belakang 2D, pembuatan animasi karakter, penulisan *source code game*, serta implementasi metode FSM (*Finite State Machine*). Salah satu unsur yang berperan penting dalam sebuah *game* adalah memiliki unsur tantangan bagi user untuk memainkan *game* tersebut. Dalam *game* Save Our Earth waktu musuh, suhu temperatur di jalan menggunakan metode FSM (*Finite State Machine*). Hal ini bertujuan agar, apabila pemain melakukan permainan, maka posisi sampah tidak akan sama sehingga pemain tidak mudah menebak letak sampah yang harus didapatkan dan dilewati dalam permainan tersebut.

3.3.3 Pengumpulan Bahan (*Material Collecting*)

Adapun bahan-bahan yang dibutuhkan adalah sebagai berikut:

- a. Teks : Tipe *font* yang digunakan pada *game* Go Green “Save Our Earth”
- b. Gambar : Semua gambar yang digunakan bertipe .jpg, .jpeg, .png
- c. Audio : File audio yang digunakan dalam *game* ini bertipe .mp3. Meliputi *music* dan *sound*.
- d. Animasi : Obyek animasi yang digunakan adalah animasi pada karakter yang ada pada *game*.

3.3.4 Pembuatan (*Assembly*)

Tahap *assembly* adalah tahap pembuatan semua objek atau bahan multimedia. Pembuatan aplikasi berdasarkan *storyboard*, bagan alur (*flowchart*), dan struktur navigasi yang berasal pada tahap *design*. Proses ini dimulai dengan pemodelan karakter dan lingkungan *game*, pembuatan animasi pada karakter, dan pembuatan *game* serta penulisan *source code*.

3.3.5 Pengujian (*Testing*)

Pengujian dilakukan dengan menggunakan metode *White Box* dan metode *Black Box*:

- Metode pengujian *White Box* menggunakan struktur kontrol desain prosedural (structural testing) untuk memperoleh *test case*. Pengujian ini mengasumsikan bahwa logik spesifik adalah penting dan harus diuji untuk menjamin sistem melakukan fungsi dengan benar.
- Metode pengujian *Black Box* merupakan pengujian *user interface* kepada pengguna apakah sistem dapat dioperasikan atau tidak. Pengujian ini dilakukan dengan memberikan *data sample* sebagai nilai masukan dan dibandingkan dengan informasi yang dihasilkan, disebut juga pengujian berbasis skenario.

3.3.6 Distribusi (*Distribution*)

Distribusi adalah tahapan di mana aplikasi disimpan dalam suatu media penyimpanan. Hasil evaluasi ini dapat digunakan sebagai masukan untuk tahap concept pada produk selanjutnya.

BAB IV. ANALISIS DAN PERANCANGAN

Pada bab ini, akan dibahas secara detail dan terperinci mengenai analisis dan perancangan *game* edukasi Go Green “Save Our Earth” berbasis *Dekstop* yang akan diimplementasikan dengan menerapkan kerangka konsep dan metode penelitian yang telah dijelaskan pada Bab III.

4.1. Analisis

4.1.1 Gambaran Umum Sitem

Game edukasi Go Green “Save Our Earth” menggunakan metode *FSM* (*Finite State Machine*). Metode ini digunakan untuk state pada sampah anorganik.

4.1.2 Analisis Kebutuhan

Kebutuhan atau spesifikasi software untuk membangun *game* edukasi Go Green “Save Our Earth” berbasis *Android* adalah:

- Sistem Operasi Windows 7 32-bit
- Corel Draw versi X7
- Adobe Photoshop Cs5
- UNITY *Game Engine* versi 5.5.2

Kebutuhan *hardware* yang digunakan dalam membangun *game* edukasi Go Green “Save Our Earth” berbasis *Dekstop* adalah:

- Processor Intel Core i3, 2.50 GHz
- RAM 4 GB
- *Mouse*
- *Keyboard*

4.2 GDD (Game Design Document)

4.2.1 Gambaran Umum Game

Permainan ini termasuk dalam katagori Platfrom. Dalam permainan ini terdapat satu karakter (single player). Permainan ini memiliki tujuan untuk mencari sampah organik dan anorganik, dan menyelamatkan bumi dari pemanasan global. Disamping player harus mencari sampah player harus

mempertahankan suhu temperatur. Permainan Go Green “Save Our Earth” ini rancang 2D (Dua Dimensi).

4.2.2 Alur Permainan

Pada permainan single player ini, dan game ini hanya 1 level ,pemain akan memasuki sebuah halaman dimana permainan ini memiliki tujuan mencari sampah organik dan anorganik. jika player mengambil sampah organik maka player akan mendapatkan 1 pohon dan apabila player sudah mengambil 5 organik maka player akan mendapatkan 1 hutan dan player harus mengambil sampah anorganik dan membuang ke tong sampah untuk bisa menurunkan suhu temperatur,player harus mengambil sampah secara berurutan, sampah anorganik ini mempunyai waktu 5 detik apabila player tidak bisa mengambil sampah anorganik maka sampah anorganik akan mengeluarkan aksi terbakar maka suhu akan naik, jika player bisa mengambil sampah anorganik maka suhu akan turun 5 derajat ,apabila sampah anorganik terbakar 5 maka player akan game over.Player mempunyai 3 nyawa.

4.2.3 Penampilan Umum Game

Gambaran umum dari aplikasi Implementasi FSM (*Finite State Machine*) pada Game Go Green Save Our Earth antara lain,

- a. Aplikasi ini dimainkan oleh satu orang saja atau *Single Player*
- b. Aplikasi ini berbasis Dekstop atau dijalankan pada perangkat *PC*.
- c. Aplikasi ini menerapkan metode FSM (*Finite State Machine*) untuk mengatur perilaku objek pada kondisi tertentu. Metode ini diterapkan pada sampah anorganik, dimana terdapat NPC (*Non Player Character*) yang diimplementasikan kecerdasan buatan dengan metode FSM, seperti pada perilaku sampah anorganik dan kondisi untuk berpindah ke state berikutnya. Pengimplementasian metode ini berupa rangkaian tahap atau state dengan masukan *event* dan *action* sehingga menghasilkan *output* yang diharapkan sesuai kondisi yang ada. Dari kondisi – kondisi tersebut maka membentuk suatu pola *state* yang saling berhubungan. Perpindahan *state* dapat terjadi apabila mendapat masukan berupa *event* tertentu yang disertai aksi yang dilakukan oleh sistem.

4.3 Perancangan

4.3.1 Perancangan Edukasi


Berikut ini merupakan perancangan edukasi ,ada 2 pilihan pembelajaran belajar materi dan menonton video, pada tampilan edukasi ini player bisa belajar tentang global warming dan efek rumah kaca,yang berisi tentang arti,dampak,dan solusi mencegah global warming dan efek rumah kaca.

4.3.2 Perancangan Karakter

Karakter dalam *game* dibedakan menjadi 3, yaitu *Player*,*Enemy* dan *Sampah* Karakter *player* dimainkan oleh pemain dengan menyentuh menekan arah atas,bawah,kanan dan kiri, untuk *enemy* adalah karakter yang harus dihindari oleh *player*.

Karakter *player* yang akan ditampilkan dalam *game* dapat dilihat pada tabel 4.1.

Tabel 4. 1 Tabel Karakter Player


No.	Gambar	Keterangan
1.		<i>Player</i> Save Our Earth

Terdapat 2 karakter *sampah* yang akan digunakan dalam *game* dapat dilihat pada tabel 4.2.

Tabel 4. 2 Tabel Karakter Sampah

No.	Gambar	Keterangan
1.		Contoh Karakter Sampah Organik, Sampah organik adalah sampah / limbah yang berasal dari sisa-sisa makhluk hidup yang ada di alam ini. Semua sampah yang berasal dari tumbuhan (flora) dan hewan (fauna) serta segala macam produk olahannya merupakan jenis sampah organik.
2.		Contoh Karakter Sampah Anorganik Adalah sampah / limbah yang dihasilkan dari berbagai macam proses, di mana jenis sampah ini tidak dapat terurai oleh bakteri secara alami dan pada umumnya memerlukan waktu yang sangat lama di dalam penguraiannya. Dan bisa didaur ulang.

Tabel 4. 3 Tabel Karakter Enemy

No.	Gambar	Keterangan
1.		<i>Karakter</i> Enemy,Enemy ini mempunyai serangan yaitu menaikkan suhu 2 derajat jika player menabrak enemy ini.

4.3.3 Perancangan *Gameplay*

Pada *gameplay* *player* memasuki *ground* berlatar belakang hutan dengan suasana siang hari, *player* harus mempertahankan suhu temperature tetap dingin, dan *player* pada *gameplay* ini harus mengambil sampah organik dan anorganik. *Player* harus mengambil sampah anorganik dengan cara menabrak sampah anorganik, jika *player* mendapatkan satu sampah organik *player* akan mendapatkan 1 pohon, jika *player* mendapatkan 5 pohon maka *player* akan dapat *bonus 1 hutan*. *Player* juga harus mengambil sampah anorganik dengan waktu yang sudah ditentukan dan mengambil sampah anorganik ini dengan berurutan, apabila *player* tidak mengambil sampah anorganik suhu *player* akan bertambah. Apabila sampah anorganik sudah terbakar 5 maka *player* akan *game over*. Pada *gameplay* *player* dapat mengambil *sampah organik* untuk mengumpulkan *pohon dan hutan*. Dan *player* dapat mengambil sampah anorganik untuk di masukan ke tong sampah supaya suhu bisa turun 4 derajat. Kondisi menang pada game ini *player* harus bisa mengumpulkan 25 pohon dan 5 hutan, dengan kondisi nyawa masih tersisa. *Player* disini mempunyai 3 nyawa, nyawa ini bisa berkurang jika suhu temperature mencapai 100 derajat, jika nyawa *player* habis maka *player* akan *game over*.

4.3.4 Perancangan kondisi musuh

Pada aplikasi ini metode FSM diimplementasikan pada rangkaian tahap atau state dengan masukan event dan action sehingga menghasilkan output yang diharapkan sesuai kondisi yang ada. Dari kondisi – kondisi tersebut maka membentuk suatu pola state yang saling berhubungan. Perpindahan state dapat terjadi apabila mendapat masukan berupa event tertentu yang disertai aksi yang dilakukan oleh sistem. Output state sebelumnya menjadi inputan untuk state berikutnya.

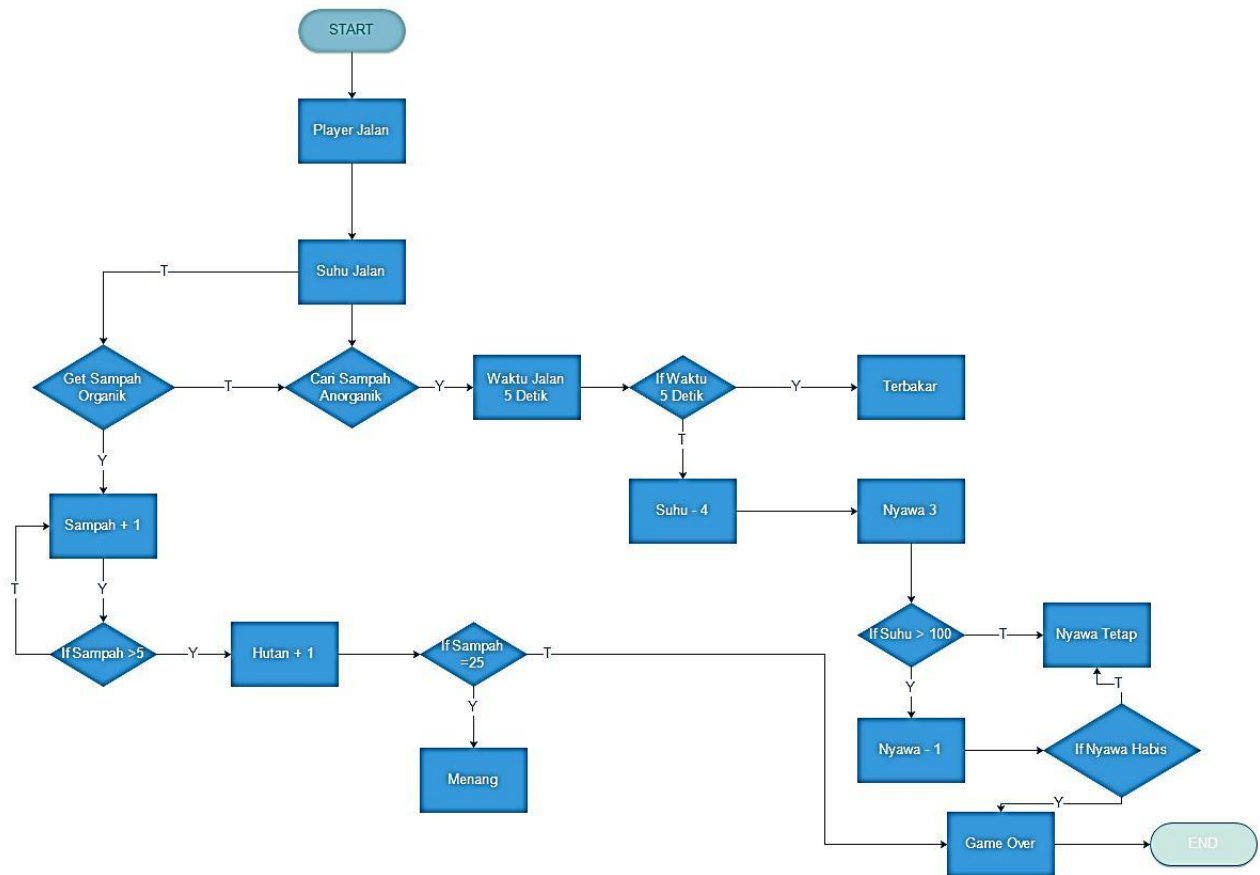
Dalam aplikasi ini metode FSM diterapkan pada sampah anorganik dalam permainan. Penerapan FSM pada aplikasi ini dilakukan agar user dapat mengenali perilaku musuh dengan sistem. Rancangan alur state FSM *go green save our earth* dibagi menjadi beberapa bagian dengan mencantumkan kebutuhan permainan dan aksi yang perlu dilakukan oleh pengguna untuk dapat mengenali perilaku

sampah anorganik dengan baik. Diantaranya perilaku state, action, event. Penentuan perilaku sampah tersebut diterapkan dengan menggunakan kecerdasan buatan FSM. Berikut merupakan penerapan FSM untuk kondisi sampah anorganik pada state game:

Tabel 4. 4 menjelaskan rancangan alur state Finite State Machine permainan go green save our earth pada sampah anorganik

<i>State (Q)</i>	<i>Event (e)</i>	<i>Action (a)</i>	<i>Next State</i>
<i>Start</i>	Jika tidak ada kondisi yang memenuhi	<i>lopping</i>	idle
<i>Idle</i>	Jika jarak $x > 9$ dan Jarak $y < 16$	Perilaku objek musuh <i>Idle</i>	<i>Time Sampah</i>
<i>Time Sampah</i>	Jika jarak $x < 16$ dan Jarak $y < 8$	Time Musuh 5 detik Berkurang	<i>Terbakar</i>
<i>Terbakar</i>	Jika time = 0 maka terbakar dan jika Sampah di ambil maka suhu turun	Perilaku objek musuh <i>terbakar</i>	<i>suhu</i>
<i>suhu</i>	Jika waktu ≤ 5 di ambil = true dan jika tidak maka menjalankan animasi terbakar	Objek suhu turun 5 derajat	<i>Hasil</i>
Hasil	Eksekusi dampak objek mati destroy gameobject	Objek musuh hancur	<i>Stop</i>

4.3.5 Flowchart *game Go Gren Save Our Earth* dalam *game* ini ditunjukkan pada gambar 4.3.

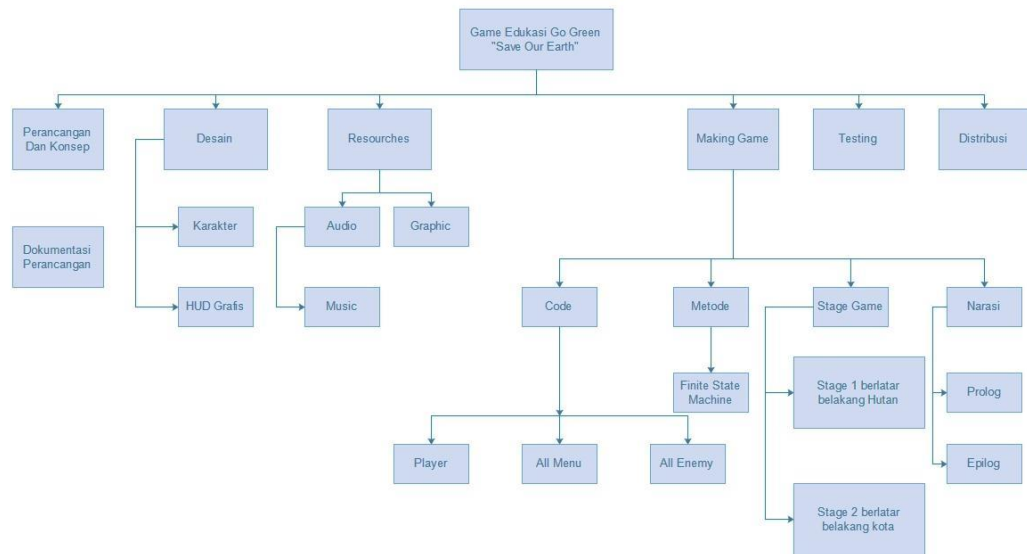


Gambar 4 2Flowchart Game

4.3.6 Work Breakdown Structure

Perancangan *Game Go Green “Save Our Earth”* meliputi tahap membuat alur *game*, perancangan *game*, pembuatan karakter maupun object yang di tampilkan, pembuatan *game*, serta implementasi metode Finite State Machine (FSM) untuk sampah anorganik.

Work Breakdown Structure game Go Green “Save Our Earth” dalam *game* ini ditunjukkan pada gambar 4.4.



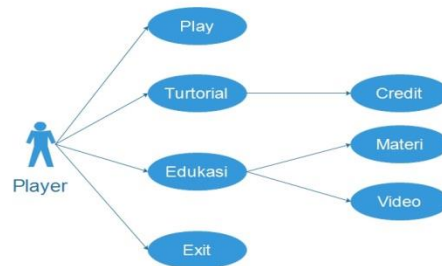
Gambar 4 3Work Breakdown Structure

4.3.7 Use case Diagram

Kegiatan yang dilakukan player digambarkan menggunakan *Use case diagram*. Hak akses yang dapat dilakukan oleh player:

1. Menu *Play*, pemain dapat langsung bermain *game* Go Green “Save Our Earth”.
2. Menu *Tutorial*, pemain dapat melihat cara bermain *game* Go Green “Save Our Earth” dan Kredit, pemain dapat melihat panel yang berisikan data pembuat *game*.
3. Menu *Setting*, pemain dapat *on/off Back sound* musik
4. Menu *Exit*, pemain dapat keluar dari *game*.

Berikut merupakan *use case* diagram dari game Go Green “Save Our Earth”:



Gambar 4 4Use case Diagram Game Go Green “Save Our Earth”

4.3.7.1 Deskripsi *Use Case*

Tabel 4. 5 Use case play

<i>Use case id</i>	1
<i>Use case name</i>	<i>Play</i>
<i>Created by</i>	
<i>Actor</i>	Pemain
<i>Description</i>	<i>Use case ini berfungsi untuk memulai permainan</i>
<i>Trigger</i>	Pemain klik tombol <i>play</i>
<i>Pre-Condition</i>	Pemain memulai permainan
<i>Exception</i>	
<i>Includes</i>	
<i>Priority</i>	<i>High</i>
<i>Frequency of use</i>	<i>High</i>
<i>Bussiness rule</i>	
<i>Special requirement</i>	
<i>Assumption</i>	
<i>Note issues</i>	

Tabel 4. 6 Use case Tutorial

<i>Use case id</i>	2
<i>Use case name</i>	<i>Tutorial</i>
<i>Created by</i>	
<i>Actor</i>	Pemain
<i>Description</i>	<i>Use case ini berfungsi untuk melihat cara bermain game</i>
<i>Trigger</i>	Pemain klik tombol tutorial
<i>Pre-Condition</i>	Pemain memilih klik tombol <i>Back</i>
<i>Post-Condition</i>	Masuk tampilan tutorial
<i>Normal Flow</i>	1. Pemain klik tombol tutorial 2. Pemain memilih tombol <i>Back</i> 3. Pemain memilih tombol <i>close</i>
<i>Exception</i>	
<i>Includes</i>	
<i>Priority</i>	<i>Low</i>
<i>Frequency of use</i>	<i>Low</i>
<i>Bussiness rule</i>	
<i>Special requirement</i>	
<i>Assumption</i>	
<i>Note issues</i>	

Tabel 4. 7 Use case Setting

<i>Use case id</i>	3
<i>Use case name</i>	<i>Setting</i>
<i>Created by</i>	
<i>Actor</i>	Pemain
<i>Description</i>	<i>Use case ini berfungsi untuk on/off Back sound musik game</i>
<i>Trigger</i>	Pemain klik tombol pengaturan
<i>Pre-Condition</i>	Pemain memilih klik musik
<i>Post-Condition</i>	<i>On/off musik</i>
<i>Normal Flow</i>	4. Pemain klik tombol <i>setting</i> 5. Pemain memilih <i>on/off</i> musik
<i>Exception</i>	
<i>Includes</i>	
<i>Priority</i>	<i>Low</i>
<i>Frequency of use</i>	<i>Low</i>
<i>Bussiness rule</i>	
<i>Special requirement</i>	
<i>Assumption</i>	

<i>Note issues</i>	
--------------------	--


Tabel 4. 8 Use case Exit

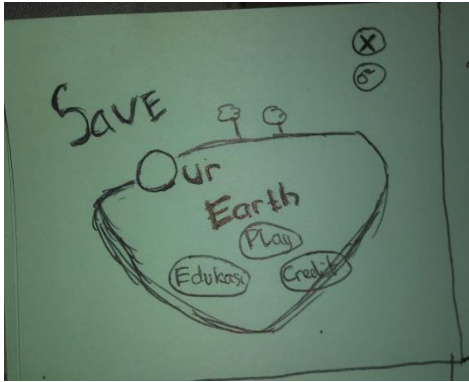
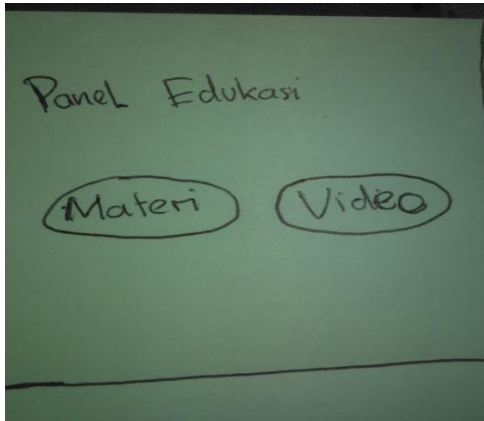
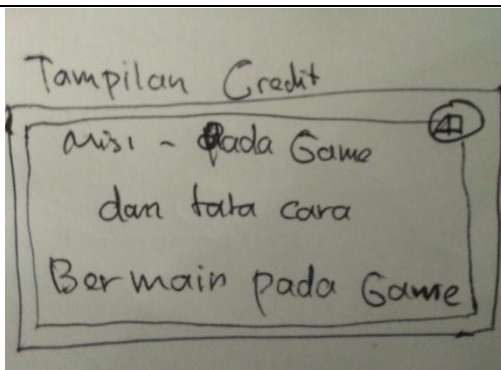

<i>Use case id</i>	5
<i>Use case name</i>	Exit
<i>Created by</i>	
<i>Actor</i>	Pemain
<i>Description</i>	<i>Use case</i> ini berfungsi untuk keluar dari aplikasi
<i>Trigger</i>	Pemain klik tombol <i>exit</i>
<i>Pre-Condition</i>	Pemain keluar dari aplikasi
<i>Post-Condition</i>	keluar dari aplikasi
<i>Normal Flow</i>	1. Pemain klik tombol keluar 2. Keluar dari aplikasi
<i>Exception</i>	
<i>Includes</i>	
<i>Priority</i>	<i>Low</i>
<i>Frequency of use</i>	<i>Low</i>
<i>Bussiness rule</i>	
<i>Special requirement</i>	
<i>Assumption</i>	
<i>Note issues</i>	

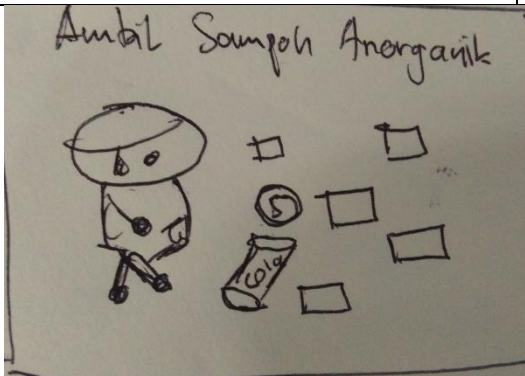
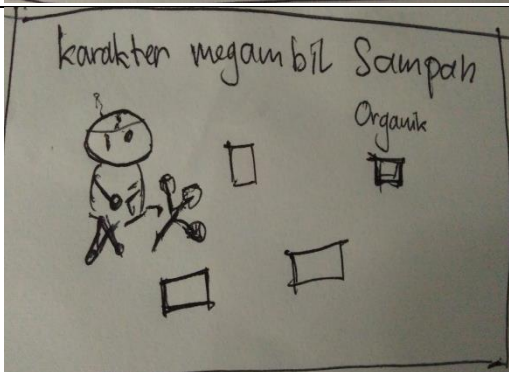
4.5.8 Perancangan *Storyboard*

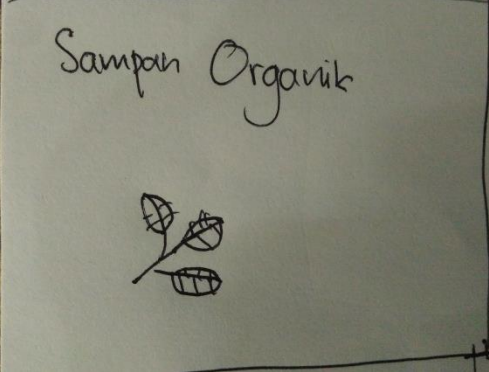
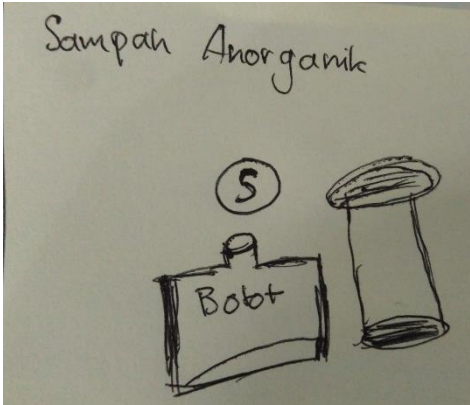
Storyboard merupakan naskah alur narasi tiap scene sebuah *game* yang dibuat dalam bentuk serangkaian gambar dan dijadikan sebagai panduan dalam mengerjakan, sehingga akan memperkecil tingkat kesalahan. Untuk menggambarkan tampilan tiap *scene* pada *game* edukasi Go Green “Save Our Earth” maka akan dijelaskan menggunakan *storyboard* pada Tabel 4.11.

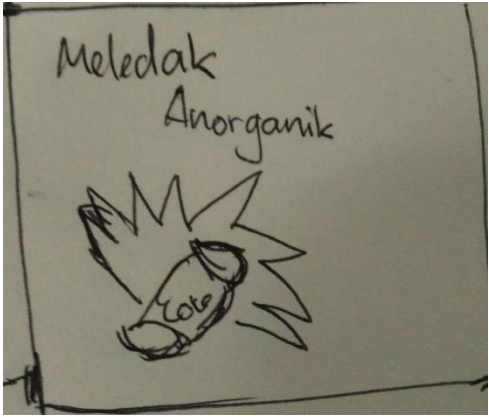
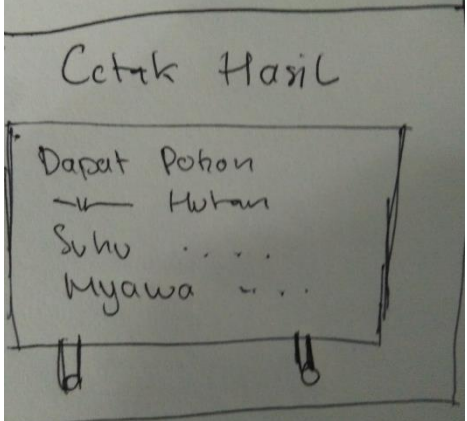
Tabel 4. 9 *Storyboard* Game Go Green “Save Our Earth”

No.	Gambar	Keterangan
1.		Tampilan Loading Tampilan <i>Loading</i> adalah tampilan awal <i>game</i> saat pertama kali dijalankan. Selama <i>game</i> di-load, pemain akan menunggu beberapa saat. <i>Loading</i> <i>Back ground</i> dibuat semenarik mungkin agar pemain

		tidak bosan menunggu.
2.		<p>Tampilan Main Menu</p> <ul style="list-style-type: none"> • Setelah menampilkan tampilan loading, akan muncul tampilan menu yang teredia pada <i>game</i> ini, seperti gambar diatas. • Pada menu utama terdapat tombol <i>Play</i>, <i>Edukasi</i>, <i>credit</i>, dan <i>Exit</i>.
3.		<p>Tampilan Edukasi</p> <p>Tampilan Edukasi adalah tampilan yang menampilkan edukasi materi dan video, pembelajaran bisa pilih materi dengan tema pemanasan global dan efek rumah kaca dan panel video bisa belajar dengan nonton video.</p>
4.		<p>Tampilan Credit</p> <p>Tampilan Credit disini menjelaskan tentang misi pada game go green save pur earth caa cara memainkan game.</p>
5.		<p>Tampilan Game Play</p> <ul style="list-style-type: none"> • Tampilan scene dimana terdapat backround cerah di pagi hari ,berlatar belakang pegunungan. • Panel Tong sampah adalah wadah player jika mendapatkan

		<p>sampah anorganik</p> <ul style="list-style-type: none"> • Panel gambar pohon wadah player jika mendapatkan sampah organik • Panel gambar Hutan wadah player jika mendapatkan sampah organic 5 dan mendapatkan 1 hutan • Panel gambar Love menunjukkan nyawa player • Panel Kebakaran menunjukkan berapa banyak sampah anorganik yang terbakar. Jika 5 sampah terbakar maka player akan game over. • Panel Suhu menunjukkan suhu pada temperature pada game.
6.		<p>Ambil Sampah Anorganik</p> <p>Cara mengambil nya adalah dengan menabrak sampah anorganik dan tong sampah akan bertambah 1.</p>
7.		<p>Ambil Sampah Organik dengan cara menabrak sampah, dan panel pohon akan bertambah 1.</p>

8.		<p>Sampah Organik</p> <p>Sampah Organik Adalah sampah buangan dari hasil manusia.</p> <p>Ini salah satu contoh sampah organik seperti daun,kulit pisang,apel ,kue.</p>
9.		<p>Sampah Anorganik</p> <ul style="list-style-type: none"> • Sampah Anorganik adalah sampah yang bukan dari perbuatan manusia,sampah yang bisa di daur ulang. • Salah satu contoh sampah anorganik yaitu botol,kresek,baterai. • Metode FSM di terapkan pada sampah anorganik ini. • Dimana sampah mempunyai event dan action terbakar. • Player harus mengambil sampah anorganik secara berurutan dari sampah yang nomer 1 dahulu. • Jika player mendapatkan sampah maka suhu bumi akan turun 5 derajat. • Sampah ini juga mempunyai time 5 detik,jika player dengan jarak > 1 cm maka time automatic akan jalan.

10.		<p>Ledakan Terbakar Sampah Anorganik.</p> <ul style="list-style-type: none"> • Animasi Ledekan Sampah Anorganik .jika player tidak bisa mengambil sampah anorganik dengan waktu <5 detik maka sampah akan terbakar. • Efek dari sampah terbakar ini adalah suhu semakin naik.
11.		<p>Tampilan Cetak Hasil</p> <p>Tampilan cetak hasil disini menampilkan atau mencetak hasil player mendapatkan pohon,hutan,tong,dan sisa nyawa,dan suhu berapa derajat.</p>

BAB V. IMPLEMENTASI

Setelah melakukan tahapan perancangan aplikasi, maka pada bab ini akan dibahas tentang proses implementasi, yaitu realisasi perancangan menjadi nyata. Bagian pada bab ini meliputi implementasi pembuatan asset gambar, implementasi sistem, dan implementasi metode.

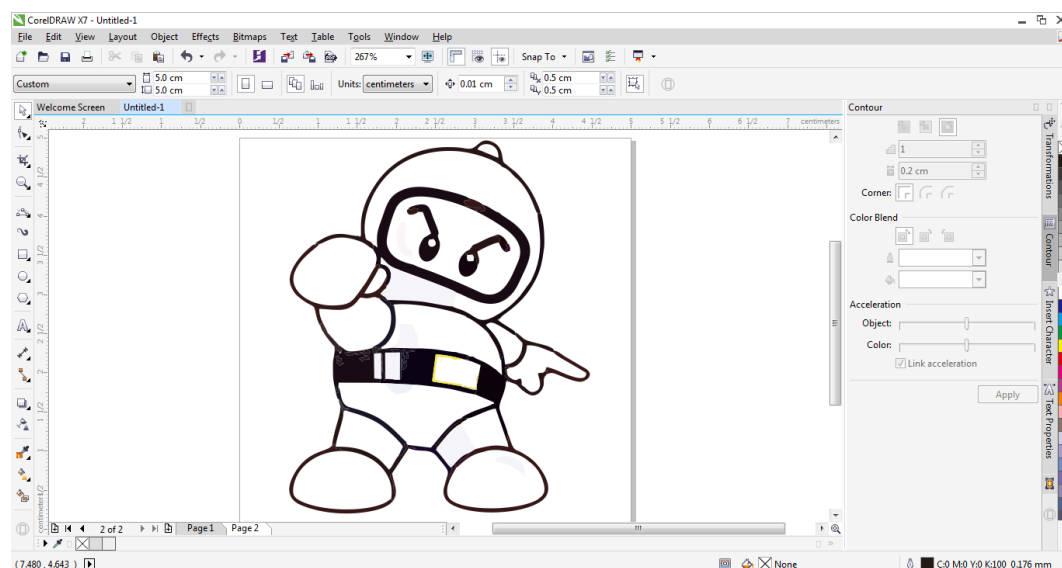
5.1 Implementasi Pembuatan *Asset* Gambar

Berikut ini merupakan langkah-langkah pembuatan *asset* gambar 2 dimensi yang terdapat pada *game*:

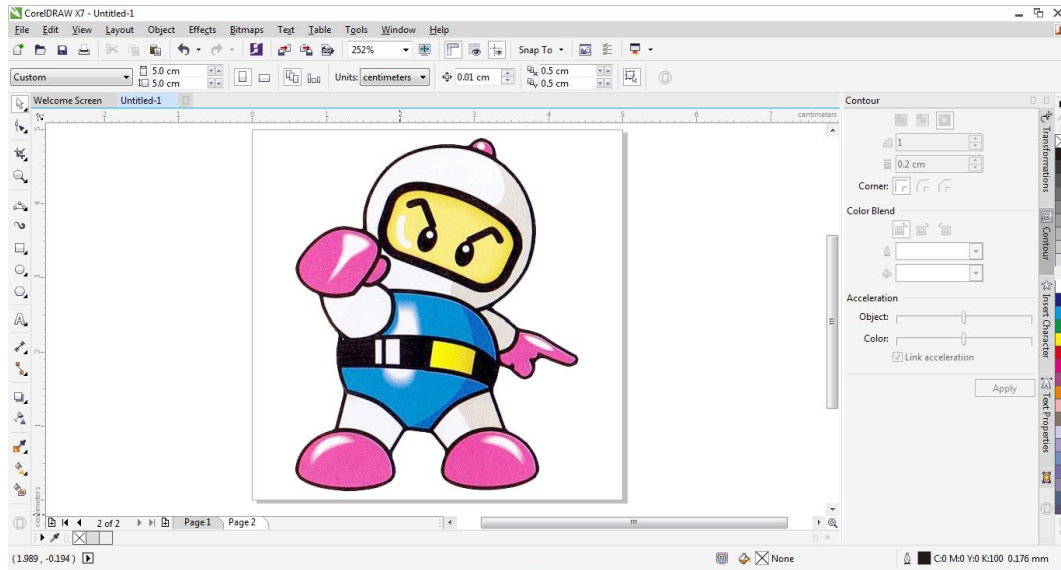
5.2.1 Pembuatan Karakter pada Corel Draw X7

5.2.1.1 Pembuatan untuk karakter *player*.

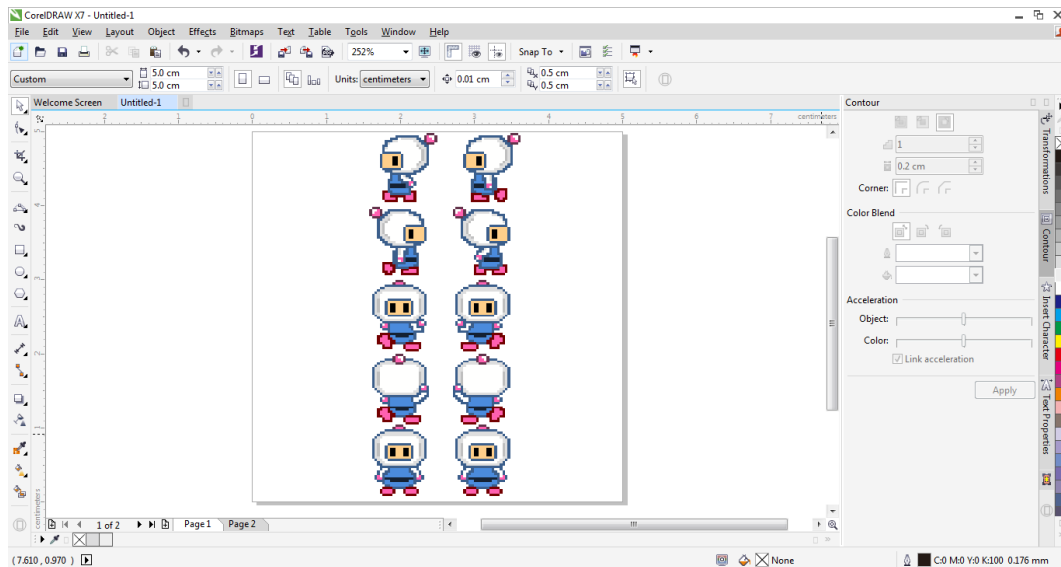
Pembuatan karakter *player* dilakukan dengan menggunakan *software* Corel Draw X7. *Tool* yang digunakan adalah *Pen Tool*, *Shape Tool*, *Eclipse Tool*, *Basic Shape Tool*. Pembuatan efek bayangan maupun *teksture* dilakukan dengan memasukkan memasukkan gambar efek pada gambar yang ingin diberikan efek, dengan cara Menu *Effects-> Powerclip->Place Inside Frame*. Pemberian warna dilakukan dengan menggunakan *Fill Tool*, khususnya *Uniform Fill* untuk memberi warna *solid*.



Gambar 5 1 Sketsa Player



Gambar 5 2Hasil Gambar Player

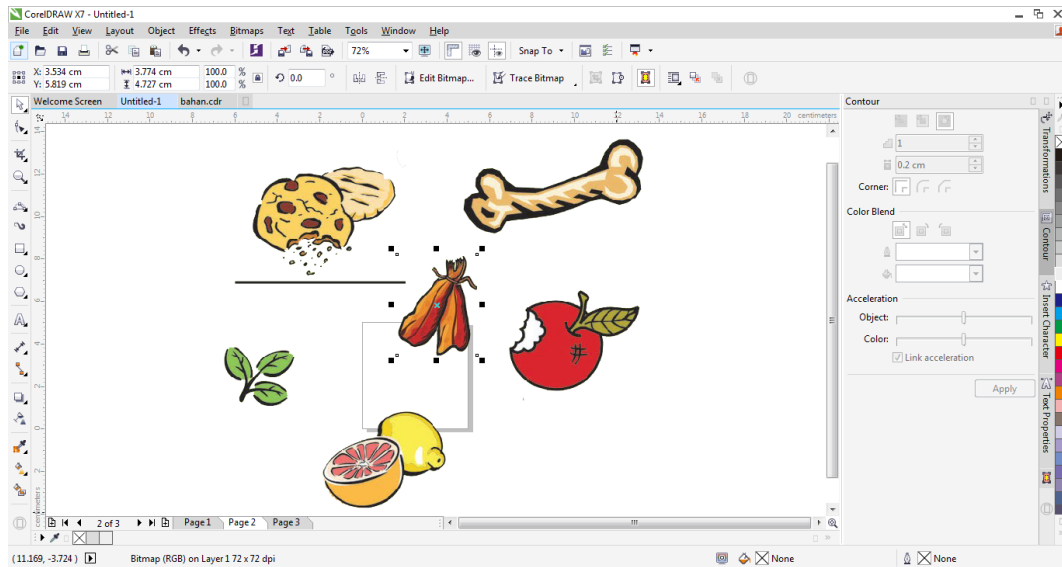


Gambar 5 3Gambar Sprites Player

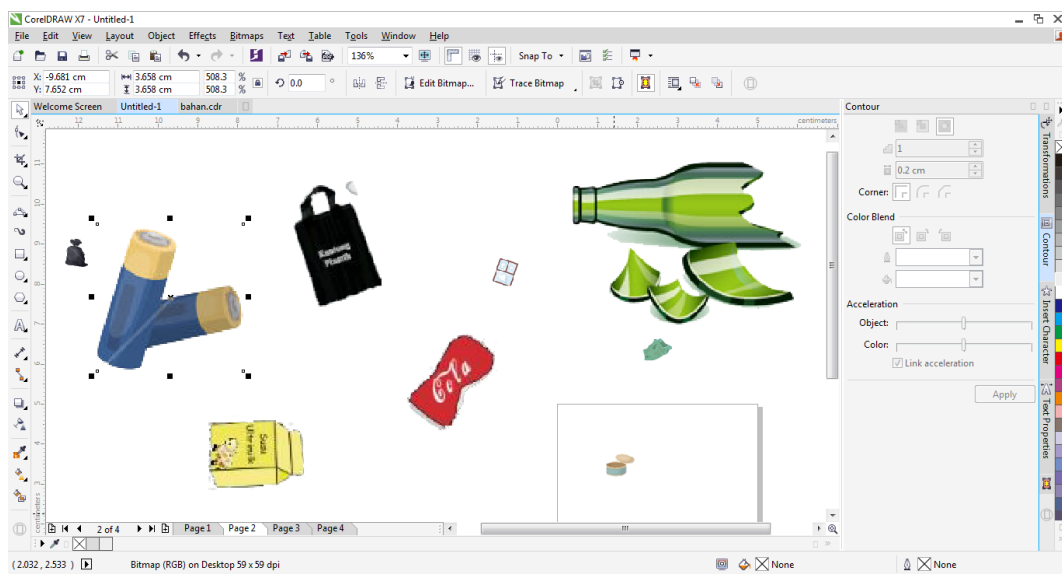
5.2.2.2 Pembuatan untuk karakter *enemy Raksasa*.

Pembuatan karakter *enemy* dilakukan dengan menggunakan *software* Corel Draw X7. *Tool* yang digunakan adalah *Pen Tool*, *Shape Tool*, *Eclipse Tool*, *Basic Shape Tool*. Pembuatan efek bayangan maupun tekstur dilakukan dengan memasukkan memasukkan gambar efek pada gambar yang ingin diberikan efek, dengan cara Menu *Effects-> Powerclip->Place Inside Frame*. Pemberian warna dilakukan dengan menggunakan *Fill Tool*, khususnya *Uniform Fill* untuk memberi warna *solid*. Pinggiran dari setiap gambar diberi garis tebal dengan

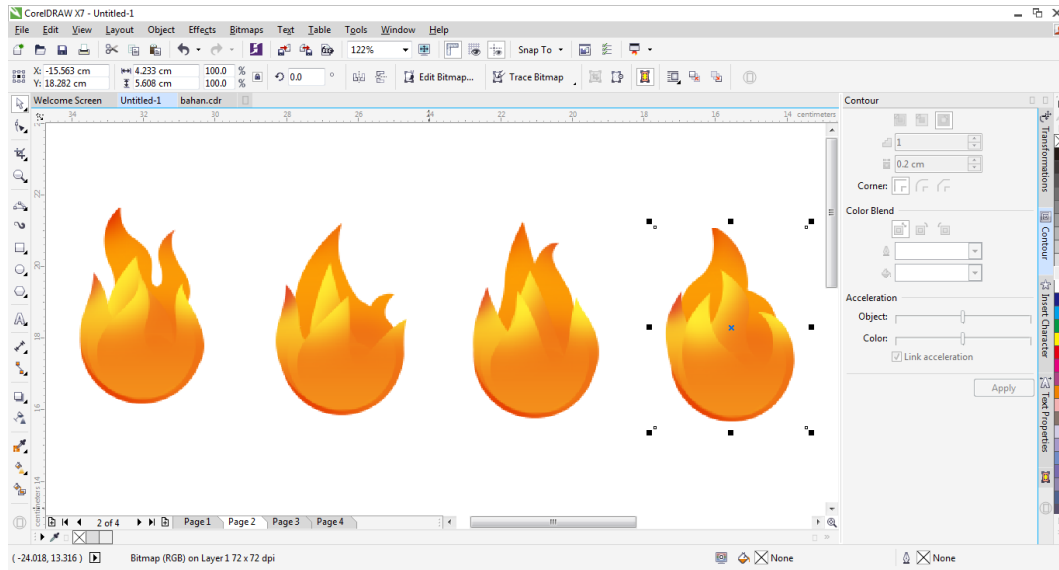
menggunakan *Contour* yang dapat dibuka di *Window Menu -> Dockers -> Contour*.



Gambar 5 4 Hasil Gambar Sampah Organik



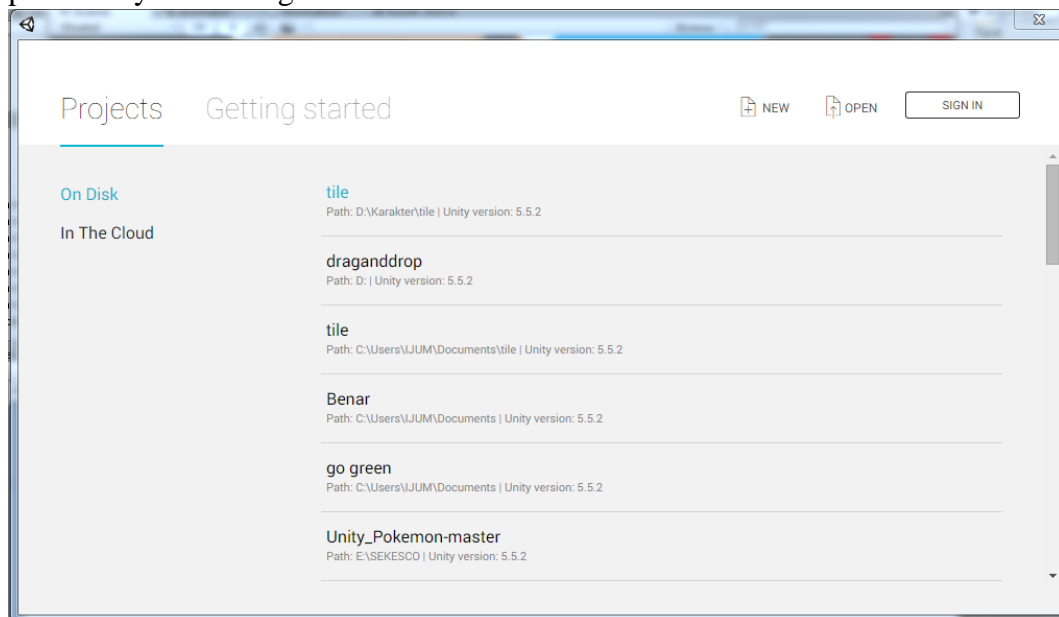
Gambar 5 5 Hasil Gambar Sampah Anorganik



Gambar 5 6 Gambar Sprites Ledekan Api

5.3 Implemetasi Pada Unity

Berikut ini merupakan langkah-langkah implementasi pembuatan *game* pada Unity *Game Engine*:



Gambar 5 7 Tampilan Awal Unity 5.5

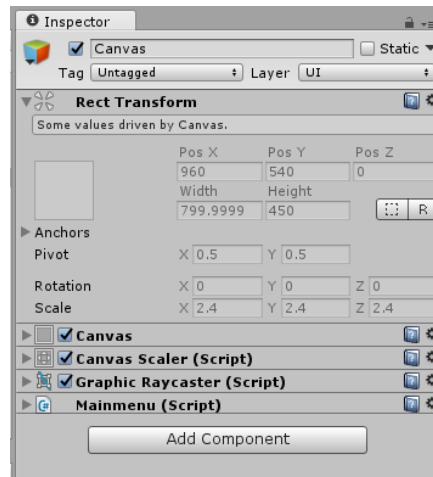
5.3.1 Menu Utama

Menu utama memiliki 5 tombol yang masing-masing dari tombol berfungsi sebagai berikut:

- Tombol Play : Untuk memulai permainan.

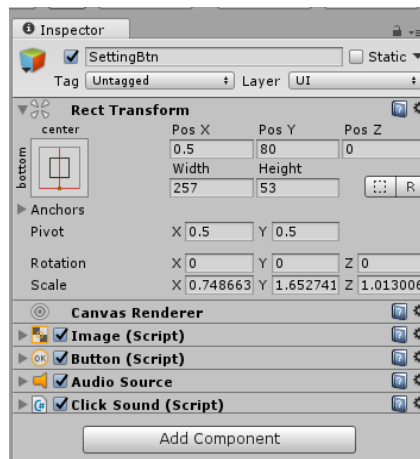
- Tombol Edukasi : Untuk menampilkan cara bermain
- Tombol Setting : Berisi pengaturan *background* musik
- Tombol Credit
- Tombol Exit : Untuk keluar dari aplikasi *game*.

Untuk dapat membuat menu utama berjalan dengan yang diinginkan, maka diperlukan pengaturan pada unity sebagai berikut:

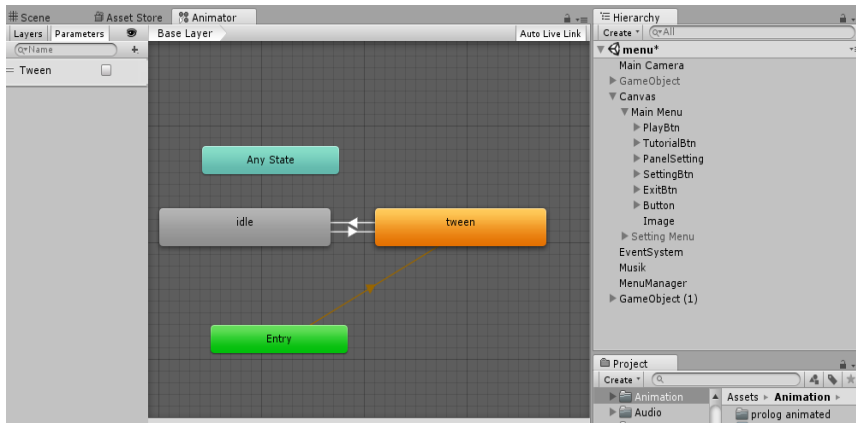


Gambar 5 8 Tampilan Inspector Menu Utama

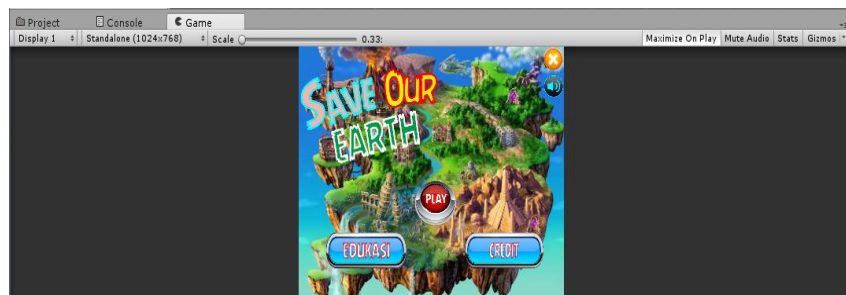
Untuk membuat animasi pada setting, agar hasil sesuai dengan apa yang diinginkan, maka diperlukan pengaturan sebagai berikut:



Gambar 5 9 Inspector Animasi Setting



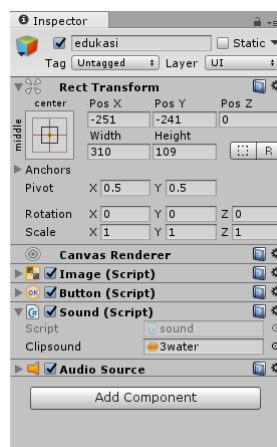
Gambar 5 10 Animator untuk Animasi Setting



Gambar 5 11 Tampilan Hasil Menu Utama

5.3.2 Edukasi

Edukasi dalam *game* menyajikan pembelajaran tentang global warming dan efek rumah kaca yang di kemas dengan bentuk majalah digital dan video . Untuk mendapatkan hasil tampilan edukasi yang diinginkan, maka yang diperlukan adalah mengatur pada unity sebagai berikut:



Gambar 5 12 Inspector Edukasi

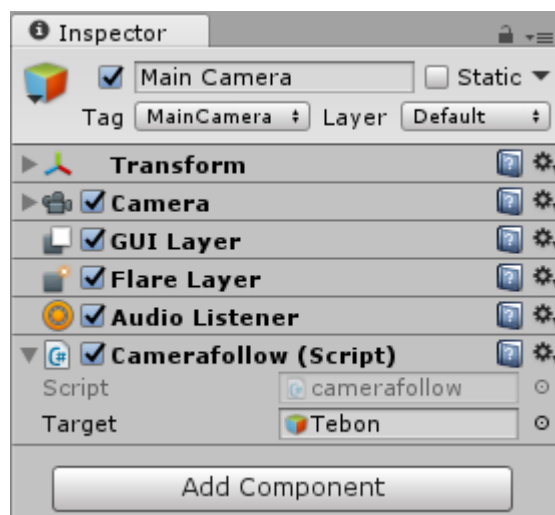


Gambar 5 13 Tampilan Edukasi

Pada prolog tersedia kotak yang berisi pembelajaran. Dan juga tersedia tombol untuk menuju video selanjutnya ataupun ingin kembali ke halaman utama.

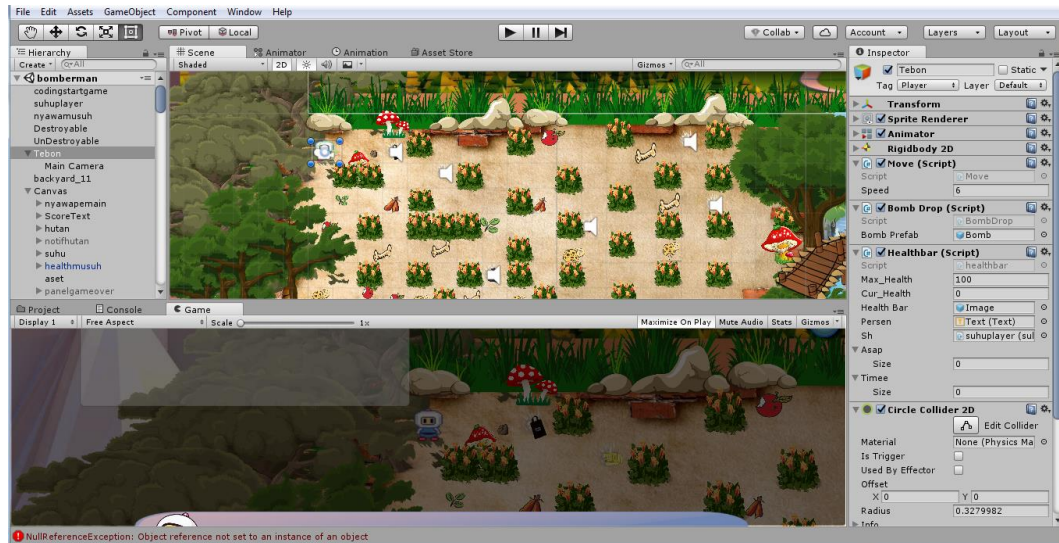
5.3.3 Pengaturan Kamera

Kamera yang akan di gunakan adalah *Orthographic*. Kamera akan mengikuti kemanapun *player* pergi. Agar dapat berjalan sesuai dengan yang diinginkan, maka diperlukan perngaturan sebagai berikut:



Gambar 5 14 Inspector Kamera

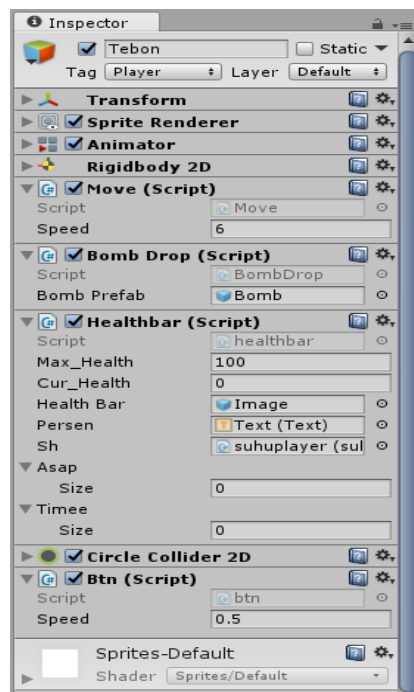
The Player pada camera Script digunakan agar agar kamera mengikuti letak *player*, seperti yang di tunjukan pada gambar 5.15.



Gambar 5 15 Tampilan Hasil Kamera

5.3.4 Pengaturan *Player*

Player memiliki *playerHealth* sebanyak 100f. Animasi yang digunakan pada *player* meliputi *idle* dan *run* Agar jalannya *player* sesuai dengan yang diinginkan, maka diperlukan pengaturan pada unity sebagai berikut:



Gambar 5 16 Inspector Player

Player memiliki *playerHealth* sebesar 100f, kecepatan/speed awal *player* untuk berlari sebesar 6f.

Agar *player* tidak bertabrakan dengan *asset* lainnya dan dapat mendeteksi *asset* lainnya, maka perlu diberikan *RigidBody2D* dan *Circle Collider 2D* seperti berikut ini:

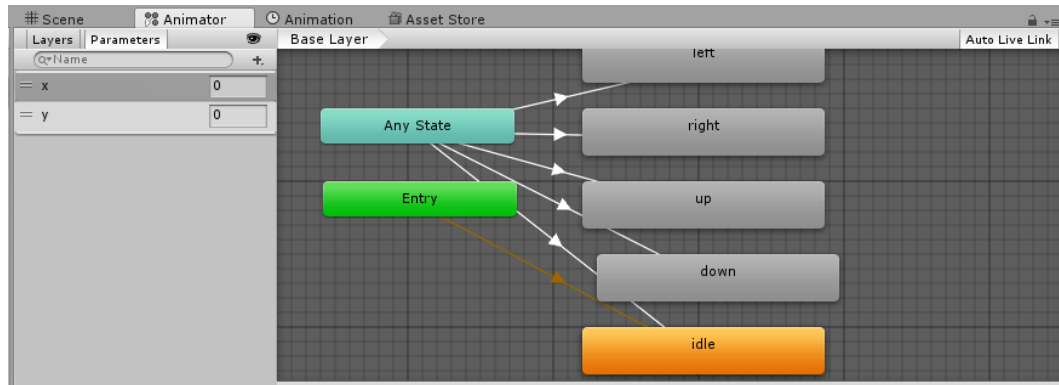


Gambar 5 17 RigidBody 2D Player

Animasi yang dibutuhkan oleh *player* yaitu *idle*, berlari. Berikut *Animation* dan *animator* yang ada pada *player*:



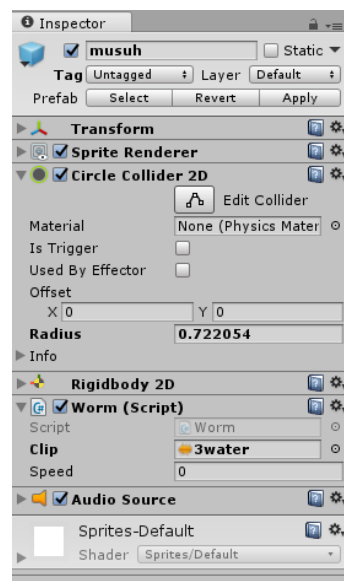
Gambar 5 18 Animasi player idle dan run



Gambar 5 19 Animator Player

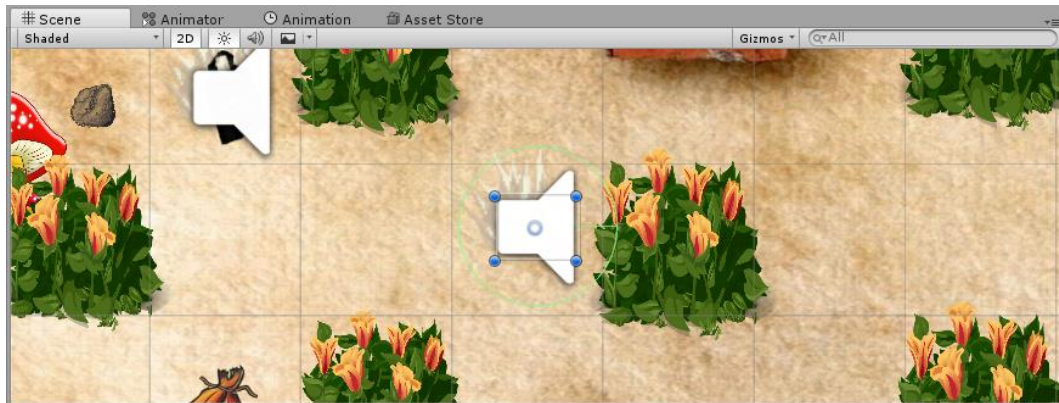
5.3.5 Pengaturan *Enemy Sampah Anorganik*

Tingkah laku yang dilakukan *enemy Sampah Anorganik* yaitu idle. Untuk membuat *enemy* sesuai dengan yang diinginkan maka diperlukan pengaturan pada unity sebagai berikut:



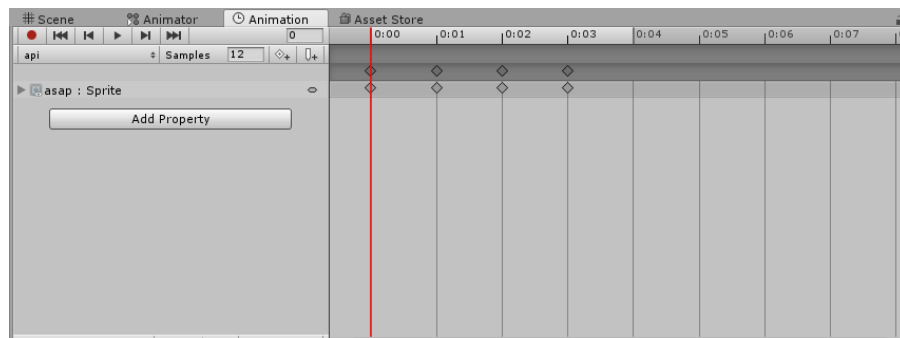
Gambar 5 20 Inspector Enemy Sampah Anorganik

Agar *enemy* tidak bertabrakan /collision dengan *asset* lainnya dan dapat mendeteksi *asset* lainnya, maka perlu diberikan *circell collider 2D* dan *RigidBody 2D*.

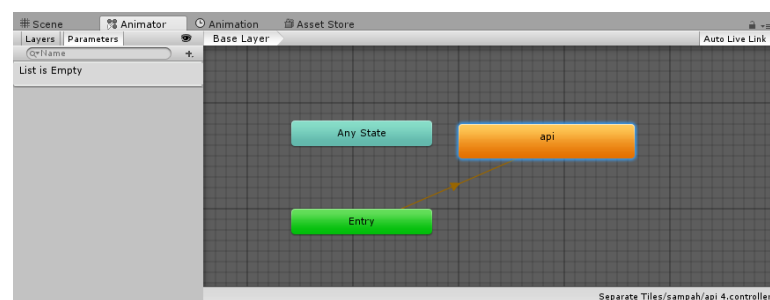


Gambar 5 21 Tampilan Circell Collider 2D Enemy

Animasi yang dibutuhkan oleh *enemy* yaitu ledakan. Berikut *Animation* dan *animator* yang ada pada *enemy*:



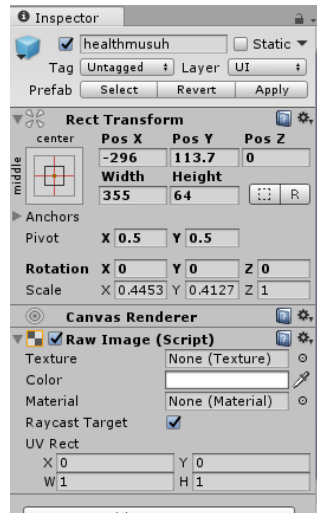
Gambar 5 22 Animasi Enemy Sampah Anorganik



Gambar 5 23 Animator Enemy Sampah Anorganik

5.3.6 Pengaturan Waktu *Enemy*

Waktu pada setiap musuh berbeda-beda, setiap musuh diberi waktu untuk aksi meledak sebanyak 5 detik. Sehingga apabila *player* tidak menabrak musuh maka suhu *player* akan semakin naik, oleh karena itu *player* harus menabrak musuh. Untuk membuat rintangan sesuai dengan yang diinginkan maka diperlukan pengaturan pada unity sebagai berikut:

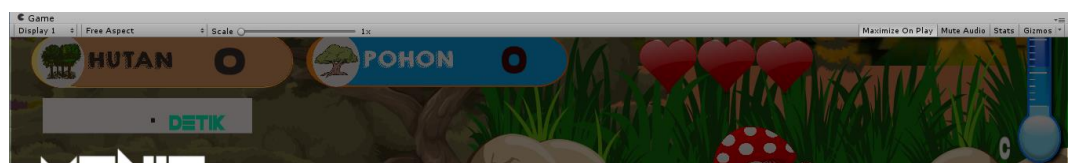


Gambar 5 24 Inspector Waktu Musuh

5.3.7 Pengaturan Score, suhu dan Healthbar

Player memiliki *healthPlayer* 3 love. *HealthPlayer* akan berkurang sebanyak 1 love setiap *player* mencapai suhu 100 derajat. Ketika *healthPlayer* telah habis maka *player* akan mati dan tidak dapat melanjutkan permainan lagi apabila nyawa yang di miliki adalah 0.

Dalam *game* ini *player* akan mendapatkan pohon dengan cari mengambil sampah organik yang ada pada stage, *player* akan mendapatkan 1 pohon setiap mengambil 1 sampah organik. Poin yang di dapat tidak hanya berasal dari pohon, *player* akan mendapatkan 1 hutan apabila *player* berhasil mendapatkan 5 *pohon* dengan mengumpulkan sampah organik.



Gambar 5 25 Tampilan jumlah pohon, Jumlah Hutan, dan HealthBar,suhu.

5.3.8 Pengaturan Pause dan Game Over

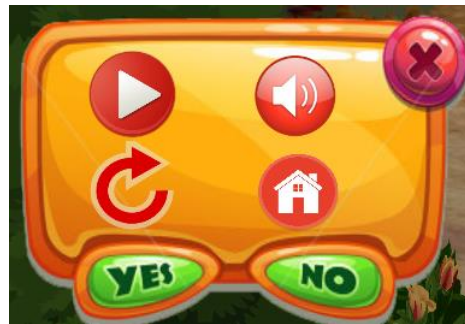
Apabila user menekan tombol pause yang terletak di kanan atas, maka game akan diberhentikan sementara, kemudian panel pause akan keluar. Ada beberapa menu pada panel pause yaitu:

1. Tombol *Resume* : Untuk *player* melanjutkan permainan
2. Tombol *Retry* : Untuk *player* mengulang permainan dari awal
3. Tombol *Home* : Untuk *player* kembali ke menu awal

Berikut adalah pengaturan inspector *pause menu* pada unity :



Gambar 5 26 Inspector Pause Menu



Gambar 5 27 Tampilan Panel Pause

Apabila *player* gagal menyelesaikan misi hingga akhir maka akan keluar panel *game over*. Pada panel *game over* akan ditampilkan *high score*, *score*, tombol untuk bermain lagi, dan tombol home untuk kembali ke menu awal.

Berikut gambar 5.28 Tampilan Panel *Game Over*:

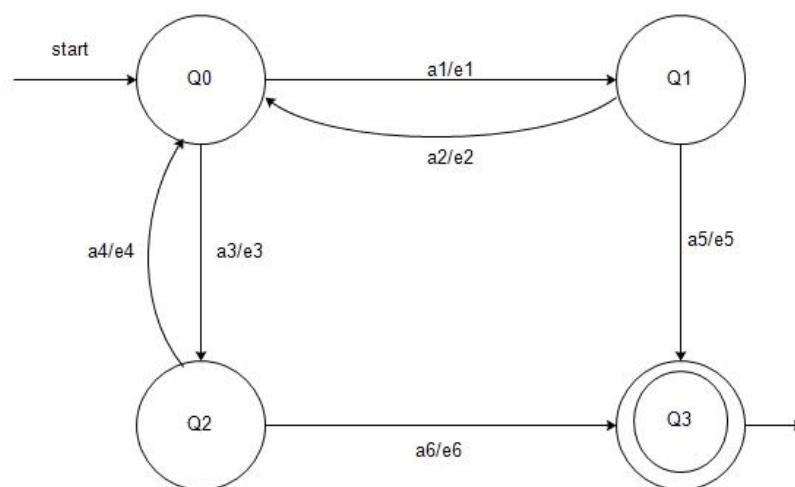


Gambar 5 28 Tampilan Panel Game Over

5.3.9 Implementasi Metode Finite State Machine (FSM) pada *Game*

a. Diagram State Sampah Anorganik

Gambar 5.31 berikut merupakan diagram *state* permainan edukasi go green “Save Our Earth” dengan penerapan kecerdasan buatan *Finite State Machine* (FSM). Pada diagram *state* ini menjelaskan alur yang terjadi saat musuh berada dalam permainan. *State* dalam diagram ini meliputi ketentuan yang berlaku dan kegiatan yang dapat terjadi yang dipengaruhi oleh *event* (e) dan *action* (a) yang telah dijabarkan pada tabel perancangan alur untuk mengatur jalannya FSM pada musuh. Adapun *state* tersebut meliputi idle (Q0), sampah organik (Q1), sampah anorganik (Q2), winning (Q3),



Gambar 5 29 Tampilan Diagram FSM Pada sampah

Keterangan :

- *State* (Q) adalah keadaan
- *Event* (e) adalah kondisi
- *Action* (a) adalah aksi

Berikut keterangan setiap lambang yang ada dalam bentuk tabel,

Tabel 5. 1 Penjelasan Keterangan Diagram State pada sampah

Lambang	Penjelasan
Q0	<i>State Idle</i> dilambangkan (Q0)
Q1	<i>State Sampah organik</i> dilambangkan (Q1)
Q2	<i>State Sampah Anorganik</i> dilambangkan (Q2)
Q3	<i>State Winning</i> dilambangkan (Q3)

e0/a0	Event 0 (e0) dan action 0 (a0)
e1/a1	Event 1 (e1) dan action 1 (a1)
e2/a2	Event 2 (e2) dan action 2 (a2)
e3/a3	Event 3 (e3) dan action 3 (a3)
e4/a4	Event 4 (e4) dan action 4 (a4)
e5/a5	Event 5 (e5) dan action 5 (a5)
e6/a6	Event 6 (e6) dan action 6 (a6)

b. Fungsi Transisi Diagram State Sampah

Tabel transisi dapat dijabarkan kedalam bentuk fungsi dengan. Lambang δ merupakan lambang dari fungsi transisi. *State* merupakan *state* awal dengan input parameter *event* dan *action* akan menghasilkan transisi menuju ke *state* yang dituju atau *state* hasil transisi. Berikut penjabaran diagram transisi,

a. Transisi State Q0

Transisi pada *state* Q0 dituliskan dalam bentuk fungsi sebagai berikut :

- $\delta(Q0, a0) = Q0$
- $\delta(Q0, a1) = Q1$
- $\delta(Q0, a3) = Q2$

b. Transisi State Q1

Transisi pada *state* Q1 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q1, a2) = Q0$
- $\delta(Q1, a5) = Q2$

c. Transisi State Q2

Transisi pada *state* Q2 dituliskan dalam bentuk fungsi sebagai berikut,

- $\delta(Q2, a4) = Q0$
- $\delta(Q2, a6) = Q3$

Keterangan :

- *State* (Q) adalah keadaan
- *Event* (e) adalah kondisi
- *Action* (a) adalah aksi
- $\delta (State, event/action) = State$ transisi

Berikut keterangan setiap lambang yang ada pada penjabaran diagram transisi,

c. Tabel Transisi Diagram State Musuh *Shooter*

Diagram *state* sampah pada gambar menjelaskan alur permainan Go Green Save Our Earth menggunakan metode FSM. Berikut transisi *state* yang dirangkum dalam bentuk tabel dari diagram *state* sampah ditunjukkan dalam tabel berikut,

Tabel 5. 2 Tabel Transisi Diagram State Sampah

Event (e) / Action	Tabel Transisi			
	State	Q0	Q1	Q2
	0	Q0	Q1	Q2
	1	Q1	Q1	Q2
	2	Q0	Q0	Q2
	3	Q3	Q1	Q2
	4	Q0	Q2	Q0
	5	Q0	Q3	Q2
	6	Q0	Q1	Q5

Keterangan :

- *State* (Q) adalah keadaan
- *Event* (e) adalah kondisi
- *Action* (a) adalah aksi

Berikut ini merupakan potongan script dari Metode FSM:

```
public void nyawamusuh1(){
    seconds += Time.deltaTime;

    if (seconds > 60) {
        seconds = 0;
    }
    if (musuh[1].activeSelf == false) {
        asap [1].SetActive (false);
    }
    if (seconds >= 5) {
        seconds = 5;
        asap [1].SetActive (true);
        musuh [1].SetActive (false);
        timemusuh [1].SetActive (false);
        jumlahkebakaran += 1;
        txtjumlahkebakaran.text = "" + jumlahkebakaran;
        Debug.Log ("kebakaran :" + jumlahkebakaran);
    }
}
```

```

    }
    Debug.Log("Detik : " + System.Math.Round(seconds,0));
}
void Update () {

    if (jumlahkebakaran >= 5) {
        panelgameover.SetActive (true);
        Time.timeScale = 0;
    }

    if(keluarwaktu[1].activeSelf == true){
        nyawamusuh1 ();
    }
    if (coll.gameObject.name == "block1" ) {
        timemusuh[1].SetActive (true);
        Destroy (coll.gameObject);
    }
    //ambilmusuh1
    if (coll.gameObject.name == "musuh1") {
        cur_Health -= 10f;
        sh.okee ();
        source.PlayOneShot (clip);
        timemusuh[1].SetActive (false);

        Destroy (coll.gameObject);
        countTong +=1;
        txttong.text = ""+ countTong;
    }
}

```

BAB VI. PENGUJIAN DAN PEMBAHASAN

Pengujian sistem merupakan tahap selanjutnya setelah aplikasi selesai dalam pembuatannya. Pengujian sistem yang dilakukan pada *game* ini meliputi dua tahapan, yaitu pengujian *Alpha* dan *Betha*. Pengujian tersebut dilakukan untuk mengevaluasi hasil sistem yang dibuat.

6.1 Pengujian Sistem

Pengujian sistem merupakan tahap selanjutnya setelah aplikasi selesai dalam pembuatannya. Pengujian sistem yang dilakukan pada *game* “Go Green “Save Our Earth”” meliputi dua tahapan, yaitu pengujian *Alpha* dan *Betha*. Pengujian tersebut dilakukan untuk mengevaluasi hasil sistem yang dibuat.

6.1.1 Rencana Pengujian

Adapun rancangan pengujian menu *game* yang akan diuji dengan teknik pengujian *Black Box*. Pengujian *Black Box* terfokus pada persyaratan fungsional perangkat lunak. Daftar rencana pengujian dapat dilihat pada tabel 6.1.

Tabel 6 1 Daftar Rencana Pengujian

No.	Komponen yang di Uji	Skenario Butir Uji	Jenis Uji
1.	Tampilan awal	Terdapat tampilan menu utama	<i>Black Box</i>
2.	Menu <i>Play</i>	Memilih tombol <i>play</i> untuk memulai permainan	<i>Black Box</i>
3.	Menu <i>Setting</i>	Memilih tombol pengaturan untuk mengatur musik	<i>Black Box</i>
4.	Menu <i>Credit</i>	Memilih tombol <i>credit</i> untuk melihat petunjuk bagaimana cara bermain <i>game</i> .	<i>Black Box</i>
5.	Menu <i>Exit</i>	Memilih tombol <i>exit</i> untuk keluar dari aplikasi	<i>Black Box</i>
6.	Tombol Musik	Memilih tombol musik untuk mematikan/menghidupkan musik	<i>Black Box</i>
7.	Tampilan Materi	Terdapat tampilan majalah digital	<i>Black Box</i>
8.	Tombol Selanjutnya Video	Memilih tombol selanjutnya untuk melanjutkan pembelajaran melalui video	<i>Black Box</i>

9.	Tampilan <i>Stage</i>	Tampilan <i>stage</i> permainan dari awal hingga akhir	<i>Black Box</i>
11.	Perolehan <i>High Score</i>	Memperoleh <i>High Score</i>	<i>Black Box</i>
12.	Tombol <i>Pause</i>	Memilih tombol <i>pause</i> untuk menghentikan permainan sementara	<i>Black Box</i>
13.	Tombol <i>Resume</i>	Memilih tombol <i>resume</i> untuk melanjutkan permainan	<i>Black Box</i>
14.	Tombol <i>Retry</i>	Memilih tombol <i>retry</i> untuk mengulang permainan	<i>Black Box</i>
15.	Tombol <i>Home</i>	Memilih tombol <i>home</i> untuk kembali ke menu utama	<i>Black Box</i>
16.	Tampilan Panel <i>Game Over</i>	Terdapat tampilan beberapa tombol dan tampilan <i>score</i> yang telah didapatkan	<i>Black Box</i>
17.	Tampilan Panel <i>Pohon</i>	Terdapat tampilan perolehan <i>berapa banyak mendapatkan sampah organik</i>	<i>Black Box</i>
18.	Animasi <i>Item Hutan</i>	Terdapat tampilan perolehan <i>berapa banyak mendapatkan pohon</i>	<i>Black Box</i>

6.1.2 Pengujian *Alpha*

Pengujian *Alpha* merupakan pengujian fungsional yang digunakan untuk menguji sistem yang telah dibuat dengan metode pengujian *BlackBox*. Daftar pengujian alpha *Game Go Green “Save Our Earth”* dapat dilihat pada tabel 6.2.

Tabel 6 2 Daftar Pengujian Alpha Game “Go Green “Save Our Earth”

No.	Aktifitas yang di Uji	Skenario Uji	Hasil yang Diharapkan	Hasil Uji
1.	Tampilan awal	Terdapat tampilan menu utama	Menu utama terdiri dari Menu <i>play</i> , <i>tutorial</i> , <i>setting</i> , <i>info</i> , dan <i>exit</i> .	Berhasil
2.	Menu <i>Play</i>	Memilih tombol <i>play</i> untuk memulai permainan	Ketika memilih tombol <i>play</i> akan keluar tampilan prolog	Berhasil
3.	Menu <i>Tutorial</i>	Memilih tombol <i>tutorial</i> untuk melihat cara bermain	Ketika memilih tombol <i>tutorial</i> maka akan keluar panel <i>tutorial</i> .	Berhasil
4.	Menu <i>Setting</i>	Memilih tombol	Ketika memilih	Berhasil

		pengaturan untuk mengatur musik	tombol pengaturan, akan ada animasi yang berupa tombol musik	
5.	Menu <i>Info</i>	Memilih tombol kredit untuk melihat <i>high score player</i>	Ketika memilih tombol kredit akan keluar panel yang berisi <i>high score player</i>	Berhasil
6.	Menu <i>Exit</i>	Memilih tombol <i>exit</i> untuk keluar dari aplikasi	Ketika memilih tombol <i>exit</i> maka akan keluar dari aplikasi	Berhasil
7.	Tombol Musik	Memilih tombol musik untuk <i>on/off</i> musik	Ketika memilih tombol musik maka semua musik <i>game</i> akan <i>on/off</i>	Berhasil
8.	Tampilan Edukasi	Terdapat tampilan Edukasi Materi dan Video	Edukasi berisi pembelajaran melalui majalah digital dan pembelajaran melalui Video	Berhasil
9.	Tombol <i>Next</i> , <i>Back</i> , dan <i>Skip</i> pada Edukasi	Memilih tombol <i>next</i> untuk melanjutkan cerita ke pembelajaran selanjutnya. Tombol <i>back</i> untuk ke Halaman menu sebelumnya, dan Tombol <i>skip</i> untuk langsung menuju ke <i>game stage</i>	Ketika memilih tombol <i>next</i> akan tampil gambar materi selanjutnya. Tombol <i>back</i> untuk ke materi sebelumnya, dan Tombol <i>skip</i> untuk langsung menuju ke <i>game stage</i>	Berhasil
10.	Tampilan Stage	Tampilan <i>stage</i> dari awal sampai akhir	Permainan akan dimulai dari awal sampai akhir.	Berhasil
12.	Tombol <i>Pause</i>	Memilih tombol <i>pause</i> untuk menghentikan permainan sementara	Ketika memilih tombol <i>pause</i> permainan akan berhenti sementara dan akan keluar panel <i>pause</i>	Berhasil
13.	Tombol <i>Resume</i>	Memilih tombol <i>resume</i> untuk melanjutkan permainan	Ketika memilih tombol <i>resume</i> , <i>player</i> akan melanjutkan permainan di tempat terakhir pada saat memilih tombol <i>pause</i>	Berhasil
14.	Tombol <i>Retry</i>	Memilih tombol	Ketika memilih	Berhasil

		<i>retry</i> untuk mengulang permainan	tombol <i>retry</i> , <i>player</i> akan mengulang permainan pada posisi <i>stage</i> tersebut	
15.	Tombol <i>Home</i>	Memilih tombol <i>home</i> untuk kembali ke menu utama	Ketika memilih tombol <i>home</i> , akan kembali ke tampilan menu utama	Berhasil
16.	Tampilan Panel <i>Game Over</i>	Terdapat tampilan <i>game over</i>	Ketika <i>player</i> kalah akan tampil panel <i>game over</i> yang menampilkan perolehan skor selama bermain	Berhasil
17.	Tampilan Panel <i>Pohon</i>	Terdapat tampilan dapat Pohon	Ketika <i>player</i> mendapatkan 1 sampah organik senjata, maka <i>player</i> mendapatkan 1 <i>Pohon</i>	Berhasil
18.	Tampilan Panel Hutan	Terdapat tampilan dapat Hutan	Ketika <i>player</i> mendapatkan 5 pohon organik, maka <i>player</i> mendapatkan 1 <i>Hutan</i>	Berhasil
20.	Tampilan Latar Belakang <i>stage</i>	Terdapat tampilan latar belakang <i>stage</i> , yang terdiri dari 2 level	Permainan dimulai dari: Level 1 yang berlatar belakang hutan saat siang hari Level 2 yang berlatar belakang kota.	Berhasil

6.1.3 Pengujian *Betha*

Pengujian *betha* merupakan pengujian yang dilakukan dengan tujuan untuk mengetahui bagaimana kualitas sistem yang telah dibuat. Dalam pengujian *betha* dilakukan terhadap responden atau calon pengguna sistem dengan menggunakan kuesioner atau angket. Kuesioner diberikan kepada 20 orang yang berusia 6 – 13 tahun. Kuesioner dibuat menggunakan skala jawaban 1 sampai 3. Daftar pertanyaan kuesioner akan dijelaskan pada Tabel 6.3.

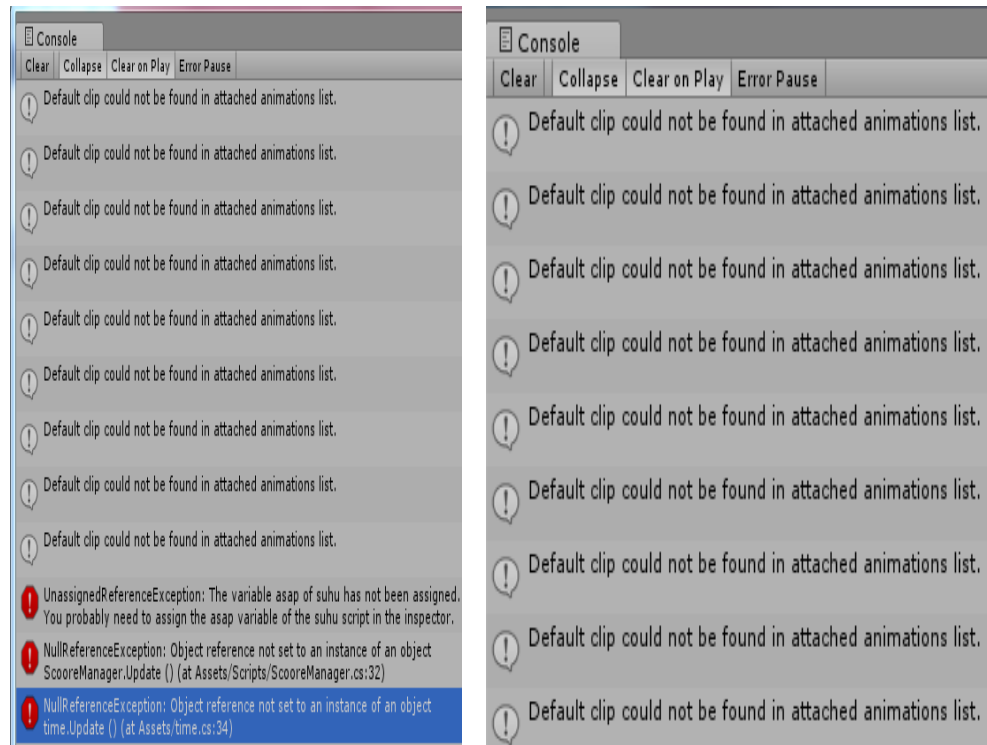
Tabel 6 3 Daftar Pertanyaan Kuesioner

No.	Pertanyaan	Skala Jawaban		
		1	2	3
1.	Bagaimana Tampilan <i>game</i> “Go Green Save Our Earth” secara umum?	Bagus	Cukup	Kurang
2.	Bagaimana tingkat kesulitan permainan dalam <i>game</i> “Go Green Save Our Earth”?	Sangat Sulit	Cukup	Terlalu Mudah
3.	Bagaimana edukasi tentang global warming dan efek rumah kaca di dalam <i>game</i> “Go Green Save Our Earth”?	Baik	Cukup	Kurang
4.	Apakah <i>game</i> “Go Green Save Our Earth” menarik untuk dimainkan?	Sangat Menarik	Cukup Menarik	Kurang Menarik

6.1.4 Hasil Pengujian Metode *Finite State Machine (FSM)*

Untuk menguji metode finite state machine ini berjalan sesuai yang diharapkan, maka dilakukan pengujian dengan cara membandingkan pengacakan pertama dan kedua tampilan stage serta array pada *Unity*. Perbandingan tampilan

pengacakan menggunakan metode finite state machine akan di jelaskan pada Tabel 6.4.

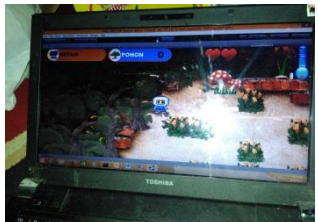



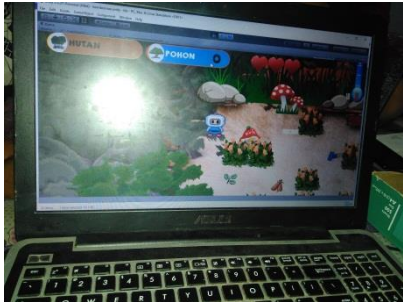

Gambar 6.1 Perbandingan Time Metode FSM

6.1.5 Hasil Pengujian Perangkat




Pengujian dilakukan dengan memainkan *Game* “Go Green Save Our Earth” di beberapa perangkat *desktop* yang berbeda. Daftar perangkat yang digunakan dalam uji coba dijelaskan pada tabel 6.5.

Tabel 6 4 Daftar perangkat uji coba

No.	Nama Perangkat	Spesifikasi	Hasil Tampilan
1.	Toshiba Satelit R845	<ul style="list-style-type: none"> • CPU Core i3 GHz & 2.10 GHz • 4 GB RAM • Windows 8 64 Bit • Intel HD Graphics 3000 	

2.	Dell inc	<ul style="list-style-type: none"> • Cpu Core i3 & 1.40 Ghz • 4 GB RAM • Windows 7 ultimate 32bit • <i>Intel HD Graphics 3000</i> 	
3.	Asus XZ55	<ul style="list-style-type: none"> • Core i5 – 5200U & 2.20 Ghz • 4 GB RAM • Windows 10 ,64 Bit • <i>Intel HD Graphics 3000</i> 	
4.	Acer Aspire 4752	<ul style="list-style-type: none"> • CPU Core i3 & 2.30Ghz • 4 GB RAM • Windows 7 Ultimate 32bit • <i>Intel HD Graphics 3000</i> 	

Tabel 6 5 Pengujian User

1.	Jihan Fara Cahya Alya	<ul style="list-style-type: none"> Siswa Kelas 4 Sd Umur 9 Tahun 	
2.	Feiby Aulia	<ul style="list-style-type: none"> Siswa Kelas 4 Sd Umur 10 Tahun 	
3.	M.Haniful Fikri	<ul style="list-style-type: none"> Siswa Kelas 4 Sd Umur 9 Tahun 	

6.2 Pembahasan

6.2.1 Hasil Pengujian Alpha

Berdasarkan hasil pengujian *Alpha* di tabel 6.2, diperoleh bahwa *Game* “Go Green Save Our Earth” ini secara fungsional mengeluarkan hasil sesuai dengan yang diharapkan.

6.2.2 Hasil Pengujian *Betha*

Hasil kuesioner yang telah dilakukan kepada 20 orang pada umur 9 tahun dapat dilihat pada tabel 6.6.

Tabel 6 6 Hasil Kuesioner Penilaian Game

No.	Nama Responden	Umur	JK	#1	#2	#3	#4
1.	Jihan Fara Cahya Alya	9	P	1	2	1	1
2.	Aulia	10	P	2	1	3	2
3.	M.Haniful Fikri	9	L	1	2	3	2
4.	Demas F.R	9	L	1	2	1	1
5.	Bagus Dian Kusuma	10	L	1	2	1	1
6.	Rizqy Putra Suryanto	9	L	1	1	2	1
7.	Feiby Aulia	10	P	2	2	2	1
8.	Nezya Okcta Hilda	9	P	2	2	2	1
9.	Kalya Hegar	9	P	1	2	2	1
10.	Nabilah Arum Nur	10	P	2	2	2	1
11.	Alvina Amalia Subhan	9	P	2	1	2	1
12.	Lintang Ayu Maharani	10	P	1	2	1	1
13.	Fellica Fissyari Selia	10	P	1	2	1	1
14.	Altan Naufal P.A	9	L	1	3	1	2
15.	Novalia Zahra A	9	P	1	2	1	1
16.	Hanung	10	L	1	1	1	1
17.	Wibawa Tara	10	L	1	1	1	1
18.	Shafira	9	P	1	2	1	1
19.	Abdu Robby	10	L	1	3	2	1
20.	Nadhief Sachio	9	L	1	2	1	2

Berdasarkan hasil kuesioner tersebut, dicari prosentase masing-masing jawaban dengan menggunakan rumus : $H = J/U \times 100\%$

Keterangan:

H = Hasil perhitungan

J = Banyaknya jawaban (skala jawaban) oleh Responden

U = Jumlah User

Berikut ini hasil perhitungan prosentase dari jawaban hasil kuesioner yang telah dilakukan terhadap 20 *user* yang dijadikan contoh penelitian *Game* “Go Green Save Our Earth”.

- b. Pertanyaan pertama, “Bagaimana tampilan *Game* “Go Green Save Our Earth” secara umum?”

Tabel 6 7 Hasil Pengujian Pertanyaan Pertama

Skala Jawaban	1	2	3
Responden	38	2	0
Hasil	95%	5%	0%

Berdasarkan hasil tabel 6.7, dapat dilihat bahwa dari 40 responden sebanyak 38 responden menjawab bagus dengan prosentase 95%, 2 responden menjawab cukup dengan prosentase 5% dan tidak ada responden menjawab kurang dengan prosentase 0%. Dengan hasil tersebut dapat disimpulkan bahwa tampilan Game “Go Green Save Our Earth” bagus.

- c. Pertanyaan kedua, “Bagaimana tingkat kesulitan permainan dalam *Game* “Go Green Save Our Earth”?”

Tabel 6 8 Hasil Pengujian Pertanyaan Kedua

Skala Jawaban	1	2	3
Responden	8	32	0
Hasil	20%	80%	0%

Berdasarkan hasil tabel 6.8, dapat dilihat bahwa dari 40 responden sebanyak 8 responden menjawab sangat sulit dengan prosentase 20%, 32 responden menjawab cukup dengan prosentase 80% dan tidak ada responden menjawab sangat mudah dengan prosentase 0%. Dengan hasil tersebut dapat disimpulkan bahwa tingkat kesulitan permainan dalam *game* “Go Green Save Our Earth” adalah cukup.

- d. Pertanyaan ketiga, “Bagaimana edukasi tentang Global Warming dan penyelamatan bumi di dalam *Game* “Go Green Save Our Earth”?”

Tabel 6 9 Hasil Pengujian Pertanyaan Ketiga

Skala Jawaban	1	2	3
Responden	36	3	1
Hasil	90%	7,5%	2,5%

Berdasarkan hasil tabel 6.9, dapat dilihat bahwa dari 40 responden sebanyak 34 responden menjawab baik dengan prosentase 90%, 3 responden menjawab cukup dengan prosentase 7,5%, dan 1 responden menjawab kurang dengan prosentase 2,5%. Dengan hasil tersebut dapat disimpulkan bahwa edukasi tentang Global Warming dan Penyelamatan bumi di dalam *game* baik.

- e. Pertanyaan keempat, “Apakah *game* “Go Green Save Our Earth” menarik untuk dimainkan?”

Tabel 6 10 Hasil Pengujian Pertanyaan Keempat

Skala Jawaban	1	2	3
Responden	24	16	0
Hasil	60%	40%	0%

Berdasarkan hasil tabel 6.10, dapat dilihat bahwa dari 40 responden sebanyak 24 responden menjawab sangat menarik dengan prosentase 60%, 16 responden menjawab setuju dengan prosentase 40%, dan tidak ada responden menjawab kurang menarik dengan prosentase 0%. Dengan hasil tersebut dapat disimpulkan bahwa *game* “Go Green Save Our Earth” sangat menarik untuk dimainkan..

BAB VII. PENUTUP

Bab VII menjelaskan tentang kesimpulan yang didapat pada saat pengerjaan skripsi melalui uji coba yang dilakukan dan analisa yang dilakukan dalam penelitiannya. Bab ini juga berisi saran yang bisa dilakukan untuk penelitian di masa yang akan datang.

7.1 Kesimpulan

Dari hasil pengujian aplikasi secara fungsional dan performa yang telah dilakukan, dapat ditarik kesimpulan bahwa *game* “Go Green Save Our Earth” memiliki tampilan *gameplay* yang menarik dan membuat proses pengetahuan tentang Global Warming dan penyelamatan bumi dengan sangat menyenangkan sehingga *user* dapat mengetahui bagaimana cara merawat bumi dengan jelas, hal ini dapat dilihat dari prosentasi jawaban setiap pengguna atau responden terhadap pertanyaan kuesioner. Metode pada game ini yaitu Finite State Machine (FSM) dapat digunakan untuk kondisi musuh yang telah di tentukan pada *game*.

7.2 Saran

Saran diberikan untuk pengembang aplikasi *game* edukasi dalam penelitian ini adalah aplikasi ini dapat dikembangkan menggunakan metode – metode lain, dan aplikasi ini dapat dibuat versi lain, yaitu aplikasi berbasis *IOS* sebagai media yang baru.

DAFTAR PUSTAKA

- [1].Usman, Nurdin.. *“Konteks Implementasi Berbasis Kurikulum”*, 2012
- [2].Setiawan, *“Perancangan Software Embedded System Berbasis FSM”*, 2006.
- [3].Marimin, *“Teknik dan Aplikasi Pengambilan keputusan Kriteria Majemuk”*, 2004.
- [4].Suparman, *“Mengenal Artificial Intelligence”*, Andi Offset, Yogyakarta, 2008.
- [5].Abdullah Dahlan dkk, *“Game Edukasi Berbasis Role Playing Dengan Metode Finite State Machine”*, Aceh : Universitas Malikussaleh Reuleut, 2015.
- [6].Rahadian, Miftah Fauzan dkk. 2016. *“Penerapan Metode Finite State Machine Pada Game “The Relationship”*. Samarinda : Universitas Mulawarman”, 2016.
- [7].Rostianingsih, Silvia dkk, *Game Simulasi Finite State Machine Untuk Pertanian dan Peternakan*. Surabaya : Universitas Kristen Petra, 2011.
- [8].Feisal, ferdian, *Pengembangan FSM Untuk Memodelkan Agen Dan Pergerakan Olahraga Futsal*. Bogor: Institute Pertanian Bogor, 2015.
- [9].Roedavan, Rickman, *Unity Tutorial Game Engine*. Bandung: Informatika Bandung, 2014.
- [10].Baidowi, Ardilas(2015, Sept 20). *Pengertian Dari Adobe Illustrator (AI)*.
[Online] Tersedia: <http://www.ardilas.com/2015/05/apakah-itu-definisi-pengertiandari-software-aplikasi-adobe-illustrator-adalah-merupakan.html>
- [11].Abdullah Dahlan, dkk, *Penerapan Metode Finite State Machine Pada Game “The Relationship”*.Samarinda, Kalimantan Timur. Jurnal Informatika Mulawarman, 2016.

LAMPIRAN

Lampiran 1 Source Code Program Sampah Anorganik

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class healthbar : MonoBehaviour {
    public static int jumlahkebakaran;
    public Text txtjumlahkebakaran;
    public GameObject panelgameover;
    public float max_Health = 5f;
    public float cur_Health = 0f;
    public GameObject healthBar;
    public Text persen;
    public suhu sh;
    public int hitungasap;
    public GameObject[] asap;
    public GameObject[] timee;
    public GameObject[] timemusuh;
    public GameObject[] keluarwaktu;
    public float seconds,seconds2,seconds3,seconds4,seconds5,seconds6,seconds7,seconds8,seconds9,seconds10,seconds11,seconds12,seconds13,seconds14,seconds15 = 0f + 0;
    public GameObject[] musuh;
    private IEnumerator coroutine;
    public AudioClip clip;
    private AudioSource source;
    // Use this for initialization
    void Start () {
        jumlahkebakaran = 0;
        txtjumlahkebakaran.text = "" + jumlahkebakaran;
        cur_Health = max_Health;
        max_Health = 5f;
        cur_Health = 0f;
        //InvokeRepeating ("decre", 1f, 1f);

        asap [1].SetActive (false);
        asap [2].SetActive (false);
        asap [3].SetActive (false);
        asap [4].SetActive (false);
        asap [5].SetActive (false);
        asap [6].SetActive (false);
        asap [7].SetActive (false);
        asap [8].SetActive (false);
        asap [9].SetActive (false);
        asap [10].SetActive (false);
        asap [11].SetActive (false);
        asap [12].SetActive (false);
        asap [13].SetActive (false);
        asap [14].SetActive (false);
        asap [15].SetActive (false);
    }
    void Awake(){
        source = GetComponent<AudioSource> ();
    }
}
```

```

public void decrees(){

    cur_Health += 1f;
    if (cur_Health <= 0) {
        cur_Health = 0;
    }
    if (cur_Health >= 5f) {
        cur_Health = 5f;
        CancelInvoke ("decrees");
    }

    persen.text = "" +System.Math.Round (cur_Health, 0);

    float calc_Health = cur_Health / max_Health;
    SetHealthBar (calc_Health);
}

public void nyawamusuh1(){
    seconds += Time.deltaTime;

    if (seconds > 60) {
        seconds = 0;
    }
    if (musuh[1].activeSelf == false) {
        asap [1].SetActive (false);
    }
    if (seconds >= 5) {
        seconds = 5;
        asap [1].SetActive (true);
        musuh [1].SetActive (false);
        timemusuh [1].SetActive (false);
        jumlahkebakaran += 1;
        txtjumlahkebakaran.text = "" + jumlahkebakaran;
        Debug.Log ("kebakaran :" + jumlahkebakaran);
    }
    Debug.Log("Detik : " + System.Math.Round(seconds,0));
}
//nyawa musuh 2
public void nyawamusuh2(){
    seconds2 += Time.deltaTime;

    if (seconds2 > 60) {
        seconds2 = 0;
    }
    if (musuh[2].activeSelf == false) {
        asap [2].SetActive (false);
    }
    if (seconds2 >= 5) {
        seconds2 = 5;
        asap [2].SetActive (true);
        musuh [2].SetActive (false);
        timemusuh [2].SetActive (false);
        jumlahkebakaran += 1;
        txtjumlahkebakaran.text = "" + jumlahkebakaran;
        Debug.Log ("kebakaran :" + jumlahkebakaran);
    }
}

```

```

        Debug.Log(" Detik : " + System.Math.Round(seconds,0));
    }
    //nyawa musuh 3
    public void nyawamusuh3(){
        seconds3 += Time.deltaTime;

        if (seconds3 > 60) {
            seconds3 = 0;
        }
        if (musuh[3].activeSelf == false) {
            asap [3].SetActive (false);
        }
        if (seconds3 >= 5) {
            seconds3 = 5;
            asap [3].SetActive (true);
            musuh [3].SetActive (false);
            timemusuh [3].SetActive (false);
            jumlahkebakaran += 1;
            txtjumlahkebakaran.text = "" + jumlahkebakaran;
            Debug.Log ("kebakaran :" + jumlahkebakaran);
        }
        Debug.Log(" Detik : " + System.Math.Round(seconds,0));
    }
    //nyawa musuh 4
    public void nyawamusuh4(){
        seconds4 += Time.deltaTime;

        if (seconds4 > 60) {
            seconds4 = 0;
        }
        if (musuh[4].activeSelf == false) {
            asap [4].SetActive (false);
        }
        if (seconds4 >= 5) {
            seconds4 = 5;
            asap [4].SetActive (true);
            musuh [4].SetActive (false);
            timemusuh [4].SetActive (false);
            jumlahkebakaran += 1;
            txtjumlahkebakaran.text = "" + jumlahkebakaran;
        }
        Debug.Log(" Detik : " + System.Math.Round(seconds,0));
    }
    //nyawa musuh 5
    public void nyawamusuh5(){
        seconds5 += Time.deltaTime;

        if (seconds5 > 60) {
            seconds5 = 0;
        }
        if (musuh[5].activeSelf == false) {
            asap [5].SetActive (false);
        }
        if (seconds5 >= 5) {
            seconds5 = 5;
            asap [5].SetActive (true);
            musuh [5].SetActive (false);
        }
    }

```

```

        timemusuh [5].SetActive (false);
        jumlahkebakaran += 1;
        txtjumlahkebakaran.text = "" + jumlahkebakaran;

    }
    Debug.Log(" Detik : " + System.Math.Round(seconds,0));
}
//nyawa musuh 6
public void nyawamusuh6(){
    seconds6 += Time.deltaTime;

    if (seconds6 > 60) {
        seconds6 = 0;
    }
    if (musuh[6].activeSelf == false) {
        asap [6].SetActive (false);
    }
    if (seconds6 >= 5) {
        seconds6 = 5;
        asap [6].SetActive (true);
    }
    Debug.Log(" Detik : " + System.Math.Round(seconds,0));
}
//nyawa musuh 7
public void nyawamusuh7(){
    seconds7 += Time.deltaTime;

    if (seconds7 > 60) {
        seconds7 = 0;
    }
    if (musuh[7].activeSelf == false) {
        asap [7].SetActive (false);
    }
    if (seconds7 >= 5) {
        seconds7 = 5;
        asap [7].SetActive (true);
    }

    Debug.Log(" Detik : " + System.Math.Round(seconds,0));
}
//nyawa musuh 8
public void nyawamusuh8(){
    seconds8 += Time.deltaTime;

    if (seconds8 > 60) {
        seconds8 = 0;
    }
    if (musuh[8].activeSelf == false) {
        asap [8].SetActive (false);
    }
    if (seconds8 >= 5) {
        seconds8 = 5;
        asap [8].SetActive (true);
    }
}

```

```

        Debug.Log(" Detik : " + System.Math.Round(seconds,0));
    }
    //nyawa musuh 9
    public void nyawamusuh9(){
        seconds8 += Time.deltaTime;

        if (seconds9 > 60) {
            seconds9 = 0;
        }
        if (musuh[9].activeSelf == false) {
            asap [9].SetActive (false);
        }
        if (seconds9 >= 5) {
            seconds9 = 5;
            asap [9].SetActive (true);
        }

        Debug.Log(" Detik : " + System.Math.Round(seconds,0));
    }
    //nyawa musuh 10
    public void nyawamusuh10(){
        seconds10 += Time.deltaTime;

        if (seconds10 > 60) {
            seconds10 = 0;
        }
        if (musuh[10].activeSelf == false) {
            asap [10].SetActive (false);
        }
        if (seconds10 >= 5) {
            seconds10 = 5;
            asap [10].SetActive (true);
        }

        Debug.Log(" Detik : " + System.Math.Round(seconds,0));
    }
    //nyawa musuh 11
    public void nyawamusuh11(){
        seconds11 += Time.deltaTime;

        if (seconds11 > 60) {
            seconds11 = 0;
        }
        if (musuh[11].activeSelf == false) {
            asap [11].SetActive (false);
        }
        if (seconds11 >= 5) {
            seconds11 = 5;
            asap [11].SetActive (true);
        }

        Debug.Log(" Detik : " + System.Math.Round(seconds,0));
    }
}

```

```

//nyawa musuh 12
public void nyawamusuh12(){
    seconds12 += Time.deltaTime;

    if (seconds12 > 60) {
        seconds12 = 0;
    }
    if (musuh[12].activeSelf == false) {
        asap [12].SetActive (false);
    }
    if (seconds12 >= 5) {
        seconds12 = 5;
        asap [12].SetActive (true);
    }

    Debug.Log(" Detik : " + System.Math.Round(seconds,0));
}
//nyawa musuh 13
public void nyawamusuh13(){
    seconds13 += Time.deltaTime;

    if (seconds13 > 60) {
        seconds13 = 0;
    }
    if (musuh[13].activeSelf == false) {
        asap [13].SetActive (false);
    }
    if (seconds13 >= 5) {
        seconds13 = 5;
        asap [13].SetActive (true);
    }

    Debug.Log(" Detik : " + System.Math.Round(seconds,0));
}
//nyawa musuh 14
public void nyawamusuh14(){
    seconds14 += Time.deltaTime;

    if (seconds14 > 60) {
        seconds14 = 0;
    }
    if (musuh[14].activeSelf == false) {
        asap [14].SetActive (false);
    }
    if (seconds14 >= 5) {
        seconds14 = 5;
        asap [14].SetActive (true);
    }

    Debug.Log(" Detik : " + System.Math.Round(seconds,0));
}
//nyawa musuh 15
public void nyawamusuh15(){
    seconds14 += Time.deltaTime;

```



```

        if (seconds15 > 60) {
            seconds15 = 0;
        }
        if (musuh[15].activeSelf == false) {
            asap [15].SetActive (false);
        }
        if (seconds15 >= 5) {
            seconds15 = 5;
            asap [15].SetActive (true);
        }

        Debug.Log(" Detik : " + System.Math.Round(seconds,0));
    }
    // Update is called once per frame
    void Update () {

        if (jumlahkebakaran >= 5) {
            panelgameover.SetActive (true);
            Time.timeScale = 0;
        }

        if(keluarwaktu[1].activeSelf == true){
            nyawamusuh1 ();
        }
        if(keluarwaktu[2].activeSelf == true){
            nyawamusuh2 ();
        }
        if(keluarwaktu[3].activeSelf == true){
            nyawamusuh3 ();
        }
        if(keluarwaktu[4].activeSelf == true){
            nyawamusuh4 ();
        }
        if(keluarwaktu[5].activeSelf == true){
            nyawamusuh5 ();
        }
        if(keluarwaktu[6].activeSelf == true){
            nyawamusuh6 ();
        }
        if(keluarwaktu[7].activeSelf == true){
            nyawamusuh7 ();
        }
        if(keluarwaktu[8].activeSelf == true){
            nyawamusuh8 ();
        }
        if(keluarwaktu[9].activeSelf == true){
            nyawamusuh9 ();
        }
    }

```

```

    }
    if(keluarwaktu[10].activeSelf == true){
        nyawamusuh10 ();
    }

    if(keluarwaktu[11].activeSelf == true){
        nyawamusuh11 ();
    }

    if(keluarwaktu[12].activeSelf == true){
        nyawamusuh12 ();
    }

    if(keluarwaktu[13].activeSelf == true){
        nyawamusuh13 ();
    }

    if(keluarwaktu[14].activeSelf == true){
        nyawamusuh14 ();
    }

    if(keluarwaktu[15].activeSelf == true){
        nyawamusuh15 ();
    }
}

public IEnumerator WaitAndPrint(float waitTime)
{
    while (true)
    {
        yield return new WaitForSeconds(waitTime);
    }
}

IEnumerator waktuu()
{
    yield return new WaitForSeconds(5);
    asap[1].SetActive (true);
}

void OnCollisionEnter2D(Collision2D coll) {
    //jarakmusuh1
    if (coll.gameObject.name == "block1" ) {
        timemusuh[1].SetActive (true);
        Destroy (coll.gameObject);
    }
    //ambilmusuh1
    if (coll.gameObject.name == "musuh1") {
        cur_Health -= 10f;
        sh.okee ();
        source.PlayOneShot (clip);
        timemusuh[1].SetActive (false);
        Destroy (coll.gameObject);
    }
}

```

```

    }
    //jarak musuh 2
    if (coll.gameObject.name == "block2" ) {
        timemusuh[2].SetActive (true);
        Destroy (coll.gameObject);
    }
    //ambilmusuh2
    if (coll.gameObject.name == "musuh2") {
        cur_Health -= 10f;
        sh.okee ();
        source.PlayOneShot (clip);
        timemusuh[2].SetActive (false);
        Destroy (coll.gameObject);

    }
    //jarak musuh 3
    if (coll.gameObject.name == "block3" ) {
        timemusuh[3].SetActive (true);
        Destroy (coll.gameObject);
    }
    //ambilmusuh3
    if (coll.gameObject.name == "musuh3") {
        cur_Health -= 10f;
        sh.okee ();
        source.PlayOneShot (clip);
        timemusuh[3].SetActive (false);
        Destroy (coll.gameObject);

    }
    //jarak musuh 4
    if (coll.gameObject.name == "block4" ) {
        timemusuh[4].SetActive (true);
        Destroy (coll.gameObject);
    }
    //ambilmusuh4
    if (coll.gameObject.name == "musuh4") {
        cur_Health -= 10f;
        sh.okee ();
        source.PlayOneShot (clip);
        timemusuh[4].SetActive (false);
        Destroy (coll.gameObject);

    }
    //jarak musuh 5
    if (coll.gameObject.name == "block5" ) {
        timemusuh[5].SetActive (true);
        Destroy (coll.gameObject);
    }
    //ambilmusuh5
    if (coll.gameObject.name == "musuh5") {
        cur_Health -= 10f;
        sh.okee ();
        source.PlayOneShot (clip);
        timemusuh[5].SetActive (false);
        Destroy (coll.gameObject);

    }
}

```

```

//jarak musuh 6
if (coll.gameObject.name == "block6" ) {
    timemusuh[6].SetActive (true);
    source.PlayOneShot (clip);
    Destroy (coll.gameObject);
}
//ambilmusuh6
if (coll.gameObject.name == "musuh6") {
    cur_Health -= 10f;
    sh.okee ();
    source.PlayOneShot (clip);
    timemusuh[6].SetActive (false);
    Destroy (coll.gameObject);

}
//jarak musuh 7
if (coll.gameObject.name == "block7" ) {
    timemusuh[7].SetActive (true);
    Destroy (coll.gameObject);
}
//ambilmusuh7
if (coll.gameObject.name == "musuh7") {
    cur_Health -= 10f;
    sh.okee ();
    source.PlayOneShot (clip);
    timemusuh[7].SetActive (false);
    Destroy (coll.gameObject);

}
//jarak musuh 8
if (coll.gameObject.name == "block8" ) {
    timemusuh[8].SetActive (true);
    Destroy (coll.gameObject);
}
//ambilmusuh8
if (coll.gameObject.name == "musuh8") {
    cur_Health -= 10f;
    sh.okee ();
    source.PlayOneShot (clip);
    timemusuh[8].SetActive (false);
    Destroy (coll.gameObject);

}
//jarak musuh 9
if (coll.gameObject.name == "block9" ) {
    timemusuh[9].SetActive (true);
    Destroy (coll.gameObject);
}
//ambilmusuh9
if (coll.gameObject.name == "musuh9") {
    cur_Health -= 10f;
    sh.okee ();
    source.PlayOneShot (clip);
    timemusuh[9].SetActive (false);
    Destroy (coll.gameObject);

}

```

```

//jarak musuh 10
if (coll.gameObject.name == "block10" ) {
    timemusuh[10].SetActive (true);
    Destroy (coll.gameObject);
}
//ambilmusuh10
if (coll.gameObject.name == "musuh10") {
    cur_Health -= 10f;
    sh.okee ();
    source.PlayOneShot (clip);
    timemusuh[10].SetActive (false);
    Destroy (coll.gameObject);

}
//jarak musuh 11
if (coll.gameObject.name == "block11" ) {
    timemusuh[11].SetActive (true);
    Destroy (coll.gameObject);
}
//ambilmusuh11
if (coll.gameObject.name == "musuh11") {
    cur_Health -= 10f;
    sh.okee ();
    source.PlayOneShot (clip);
    timemusuh[11].SetActive (false);
    Destroy (coll.gameObject);

}
//jarak musuh 12
if (coll.gameObject.name == "block12" ) {
    timemusuh[12].SetActive (true);
    Destroy (coll.gameObject);
}
//ambilmusuh12
if (coll.gameObject.name == "musuh12") {
    cur_Health -= 10f;
    sh.okee ();
    source.PlayOneShot (clip);
    timemusuh[12].SetActive (false);
    Destroy (coll.gameObject);

}
//jarak musuh 13
if (coll.gameObject.name == "block13" ) {
    timemusuh[13].SetActive (true);
    Destroy (coll.gameObject);
}
//ambilmusuh13
if (coll.gameObject.name == "musuh13") {
    cur_Health -= 10f;
    sh.okee ();
    source.PlayOneShot (clip);
    timemusuh[13].SetActive (false);
    Destroy (coll.gameObject);

}
//jarak musuh 14

```

```

        if (coll.gameObject.name == "block14" ) {
            timemusuh[14].SetActive (true);
            Destroy (coll.gameObject);
        }
        //ambil musuh 14
        if (coll.gameObject.name == "musuh14") {
            cur_Health -= 10f;
            sh.okee ();
            source.PlayOneShot (clip);
            timemusuh[14].SetActive (false);
            Destroy (coll.gameObject);

        }
        //jarak musuh 15
        if (coll.gameObject.name == "block15" ) {
            timemusuh[15].SetActive (true);
            Destroy (coll.gameObject);
        }
        //ambil musuh 15
        if (coll.gameObject.name == "musuh15") {
            cur_Health -= 10f;
            sh.okee ();
            source.PlayOneShot (clip);
            timemusuh[15].SetActive (false);
            Destroy (coll.gameObject);

        }
        CancelInvoke ("decre");
    }

    public void SetHealthBar(float myHealth)
    {
        healthBar.transform.localScale = new Vector3(myHealth,healthBar.
transform.localScale.y, healthBar.transform.localScale.z);

    }
}

```

Lampiran 2 Source Code Program Suhu

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class suhu : MonoBehaviour {
    public float max_Health = 100f;
    public float cur_Health = 0f;
    public GameObject healthBar;
    public Text persen;
}

```

```

public GameObject nyawa1;
public GameObject nyawa2;
public GameObject nyawa3;
public static int count;
public GameObject panelgameover;
public AudioClip clip;
private AudioSource source;
public ScooreManager level2;
public GameObject asap;
public Text txtnyawa;
// Use this for initialization
void Start () {
    cur_Health = max_Health;
    cur_Health = 0f;
    InvokeRepeating ("decre", 1f, 1f);
    asap.SetActive (false);
}

void Awake(){
    source = GetComponent<AudioSource>();
}
public void okee(){

    cur_Health -= 7f;
    InvokeRepeating ("decre", 0f, 0f);

}
public void kurangnyawa(){
    cur_Health += 3f;
    if (cur_Health >= 100 ) {
        asap.SetActive (true);
    }
}
public void decre(){

    cur_Health += 3f;
    if (cur_Health <= 0) {
        cur_Health = 0;
    }

    if (cur_Health >= 100f) {
        cur_Health = 100f;
        count += 1;
        nyawa1.SetActive (false);
        txtnyawa.text = "" + 2;
        if (cur_Health == 100f) {
            cur_Health = 0f;
            source.PlayOneShot (clip);
        }
        if (count == 2) {
            nyawa2.SetActive (false);
            source.PlayOneShot (clip);
            txtnyawa.text = "" + 1;
        }
    }
}

```

```

        if (count == 3) {
            nyawa3.SetActive (false);
            source.PlayOneShot (clip);
            panelgameover.SetActive (true);
            txtnyawa.text = "" + 0;
        }
        if (count >= 4) {
            cur_Health = 100f;
            CancelInvoke ("decreases");
        }
    }
    if (cur_Health >= 100 ) {
        asap.SetActive (true);
    }
    persen.text = "" +System.Math.Round (cur_Health, 0);

    float calc_Health = cur_Health / max_Health;
    SetHealthBar (calc_Health);
}

// Update is called once per frame
void Update () {

}

public void SetHealthBar(float myHealth)
{
    healthBar.transform.localScale = new Vector3(myHealth,healthBar.
transform.localScale.y, healthBar.transform.localScale.z);

}
}

```


Lampiran 3 Soal Kuisisioner

PERTANYAAN KUESIONER				
No.	Pertanyaan	Skala Jawaban		
		1	2	3
1.	Menurut anda bagaimana dengan tampilan desain <i>game</i> edukasi "Go Green Save Our Earth" ?	Bagus ✓	Cukup	Kurang
2.	Bagaimana penggunaan fitur-fitur yang ada di <i>game</i> ini, mudah dimainkan ?	Sangat Sulit	Cukup ✓	Mudah
3.	Bagaimana dengan fitur-fitur aplikasi ini, sudah menarik ?	Sangat Menarik ✓	Cukup	Kurang
4.	Dalam Pemahaman <i>game</i> yang telah disampaikan, Anda merasa terbantu akan aplikasi edukasi ini ?	Sangat Terbantu ✓	Cukup	Tidak Terbantu
5.	Aplikasi <i>game</i> ini layak disebar luaskan ?	Sangat Layak ✓	Layak	Belum

Nama : Jihan Fara Cahya Alya

Umur : 9thn

Jenis Kelamin : Perempuan

Saran dan Kritik :

Cahayanya harus diterangkan

Lampiran 4 Profil Penulis

BIODATA MAHASISWA



Nama Lengkap	: Azis Fauzan Muhadzdzib
Nomor Induk Mahasiswa	: 1341180081
Jurusan	: Teknologi Informasi
Program Studi	: Teknik Informatika
Jenis Kelamin	: Laki-laki
Tempat, tanggal lahir	: Malang, 20 Januari 1995
Alamat asal	: Jalan Tuntang No.1 RT. 07 RW. 07 Kelurahan Bunulrejo, Kecamatan Blimbing, Kota Malang
No. Telepon	: 082285263760
Agama	: Islam

Riwayat Pendidikan

2001 – 2007	: SDN Arjosari 1 Malang
2007 – 2010	: SMPN 16 Malang
2010 – 2013	: SMKN 9 Malang
2013 – 2017	: Politeknik Negeri Malang, Program Studi Teknik Informatika, Jurusan Teknologi Informasi.