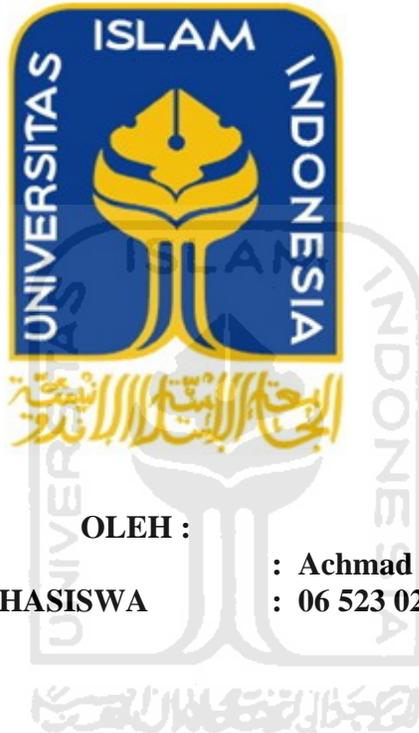


**RANCANG BANGUN GAME LABIRIN 3D MENGGUNAKAN UNITY**

**TUGAS AKHIR**

**Diajukan sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana  
Jurusan Teknik Informatika**



**OLEH :**

**NAMA : Achmad Zacky**  
**NO MAHASISWA : 06 523 022**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2011**

# LEMBAR PENGESAHAN DOSEN PEMBIMBING

## RANCANG BANGUN GAME LABIRIN 3D MENGGUNAKAN UNITY

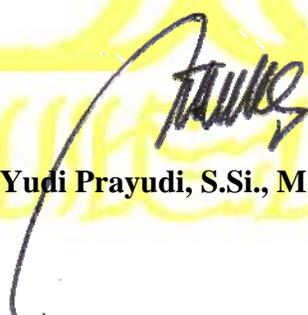
### TUGAS AKHIR

OLEH :

NAMA : Achmad Zacky  
NO MAHASISWA : 06 523 022

Yogyakarta, 14 Februari 2010

Pembimbing,



Yudi Prayudi, S.Si., M.Kom.

# LEMBAR PENGESAHAN DOSEN PENGUJI

## RANCANG BANGUN GAME LABIRIN 3D MENGGUNAKAN UNITY

### TUGAS AKHIR

OLEH :

NAMA : Achmad Zacky  
NO MAHASISWA : 06 523 022

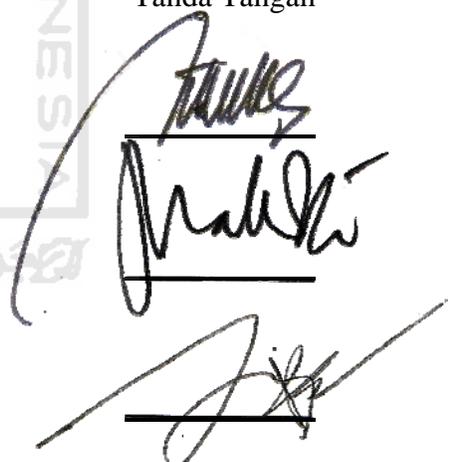
Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat  
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika Fakultas  
Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Maret 2011

Tim Penguji,

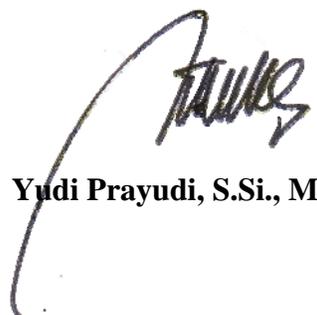
1. Yudi Prayudi, S.Si., M.Kom.
2. Affan Mahtarami S.Kom., M.T.
3. Irving Vitra Papatungan S.T., M.Sc.

Tanda Tangan



Mengetahui,

Ketua Jurusan Teknik Informatika  
Universitas Islam Indoneisa



Yudi Prayudi, S.Si., M.Kom.

## HALAMAN PERSEMBAHAN

*Skripsi ini saya persembahkan untuk  
Ibuku tercinta, Ayahku tersayang, dan semua kakak-kakakku*



## MOTTO

*"...Man Jadda wajada : Siapa yang bersungguh sungguh akan berhasil."*

*(Negeri 5 Menara)*



## KATA PENGANTAR



*Assalamu'alaikum wr.wb.*

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT, yang telah memberikan rahmat, hidayah serta karunia-Nya, sehingga laporan Tugas Akhir dapat penulis selesaikan. Tak lupa shalawat serta salam kami haturkan kepada junjungan kita Nabi Muhammad S.A.W, yang telah memberi uswatun khasanah bagi umat manusia.

Tugas Akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di jurusan Teknik Informatika Universitas Islam Indonesia.

Tugas Akhir yang penulis laksanakan adalah membuat Perancangan Game Labirin 3d Menggunakan Unity. Pada kesempatan ini penulis mengucapkan terima kasih yang sebesar– besarnya kepada :

1. Ibu, Ayah, dan keluarga tercinta.
2. Bapak Gumbolo Hadi Susanto, Ir., M.Sc, selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Yudi Prayudi, S.Si., M.Kom, selaku ketua Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
4. Bapak Yudi Prayudi, S.Si., M.Kom, selaku dosen pembimbing Tugas Akhir, yang telah memberikan pengarahan dan bimbingan selama pelaksanaan Tugas Akhir dan penulisan laporan.
5. Seluruh staf pengajar FTI UII, khususnya dosen-dosen jurusan Teknik Informatika yang telah memberikan bekal ilmu.
6. Seluruh teman-teman seperjuangan angkatan 2006 Teknik Informatika yang telah memberikan keceriaan dan semangat tanpa henti kepada penulis.
7. Dia yang jauh, yang selalu menunggu dengan penuh kesetiaan.
8. Serta semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dari awal hingga akhir.

Penulis menyadari bahwa laporan ini masih belum sempurna, karena keterbatasan kemampuan dan pengalaman. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun untuk membantu penulis di masa yang akan datang.

Akhir kata penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

*Wassalamu'alaikum Wr Wb.*

Yogyakarta, 14 maret 2010

Achmad Zacky



## SARI

*Game 3D* merupakan *game* yang menarik untuk dimainkan sebagai penghilang stress dan juga media ketangkasan otak. Labirin adalah sebuah teka-teki yang berupa rute perjalanan yang penuh dengan lika liku sehingga membutuhkan logika untuk menyelesaikannya. Akan sangat menarik sekali apa bila konsep labirin ini di terapkan dalam sebuah *game 3d* .

Untuk mengatasi pesolatan tersebut, maka dirancanglah *game* yang mengambil konsep labirin dengan memnggunakan Unity. *game* tersebut dibuat degan tampilan 3d. Dengan begitu pemain akan merasakan permaian yang mengasah logika dan juga menyenangkan. *Game* tersebut dibuat menggunakan Unity sebagai *game engine*, 3dmax sebagai *modeling software*, dan illustrator sebagai *software* untuk membuat GUI.

Hasilnya adalah *game 3d* yang menerapkan konsep labirin di setiap arena permainannya.

Kata kunci : UNITY, Game 3D, Labirin, Maze



## TAKARIR

<i>Story board</i>	alur cerita yang terdapat dalam game.
<i>Single player</i>	hanya dapat dimainkan oleh satu orang pemain.
<i>Gameplay</i>	permainan.
<i>Player</i>	pemain dalam suatu permainan.
<i>Ruleplay</i>	aturan – aturan yang dibuat untuk memberikan batasan dalam game.
<i>Level game</i>	sebuah tahapan situasi dalam game.
<i>Intro</i>	awalan.
<i>Loading</i>	pengisian.
<i>Icon</i>	lambang.
<i>Game won</i>	permainan menang.
<i>Game complete</i>	permainan lengkap.
<i>Game over</i>	permainan usai.



## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
HALAMAN PERSEMBAHAN .....	iv
HALAMAN MOTTO.....	v
KATA PENGANTAR .....	vi
SARI .....	viii
TAKARIR.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xiv
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Manfaat .....	2
1.6 Metode penelitian.....	2
1.7 Sistematika Penulisan .....	3
<b>BAB II LANDASAN TEORI .....</b>	<b>5</b>
2.1 Game.....	5
2.1.1 Sejarah Singkat .....	5
2.1.2 Definisi Game .....	5
2.1.3 Teori Pembuatan Game.....	7
2.1.4 Elemen Game.....	8
2.1.5 Game Dev .....	9
2.2 Labirin.....	11
<b>BAB III METODOLOGI.....</b>	<b>13</b>

3.1 Analisis Kebutuhan.....	13
3.2 Hasil Analisis.....	13
3.2.1 Analisi Kebutuh Masukan.....	13
3.2.2 Analisi Kebutuh Pengeluaran.....	14
3.2.3 Analisi Kebutuhan Proses.....	14
3.2.4 Analisi Kebutuhan Antar Muka.....	14
3.2.5 Analisi Kebutuh Perangkat Lunak.....	14
3.2.6 Analisi Kebutuh Perangkat Keras.....	15
3.3 Perancangan.....	16
3.3.1 Hasil Perancangan.....	16
3.4 <i>Game Board</i> .....	16
3.5 Perancangan Level Permainan.....	19
3.5.1 Arena Level 1.....	19
3.5.2 Arena Level 2.....	20
3.5.3 Arena Level 3.....	20
3.5.4 Arena Level 4.....	21
3.6 Perancangan Aturan Permainan.....	21
3.7 Perancangan <i>Game Object</i> .....	23
3.7.1 Karakter Utama.....	23
3.7.2 Karakter Musuh/Hambatan.....	23
3.8 Perancangan Antar Muka.....	25
3.8.1 Halaman Menu.....	25
3.8.2 Halaman Menu Mulai permainan.....	26
3.8.3 Halaman Menu <i>Pause</i> .....	26
3.8.4 Halaman <i>Gameplay</i> .....	27
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>28</b>
4.1 Implementasi Perangkat Lunak.....	28
4.1.1 Batasan Implementasi.....	28
4.1.2 Implementasi Prosedural.....	29
4.2 Hasil.....	29

4.3.1 Halaman Menu Utama .....	30
4.3.1 Halaman Pilih level.....	30
4.3.1 Halaman <i>Game Over</i> .....	31
4.3.1 Halaman <i>Game Pause</i> .....	31
4.3.1 Halaman Tamat <i>Game</i> .....	32
4.3.1 Halaman level 1 .....	32
4.3.1 Halaman Level 2.....	33
4.3.1 Halaman Level 3.....	33
4.3.1 Halaman Level 4.....	34
4.3.1 Tampilan <i>Game Object</i> .....	34
4.3 Pengujian Perangkat Lunak .....	34
4.3.1 Pengujian Tombol.....	35
4.3.1.1 Pengujian Tombol start dan exit.....	35
4.3.1.2 pengujian tombol pilih level .....	35
4.3.2 Pengujian <i>Game Play</i> .....	37
4.3.2.1 <i>Charakter Control</i> .....	37
4.3.2.2 Instruksi <i>Game</i> .....	38
4.3.2.2.1 Kondisi mendapatkan kristal.....	38
4.3.2.2.2 Kondisi Terkena Hambatan .....	39
4.3.2.2.3 Kondisi Menghancurkan Hambatan .....	40
4.3.2.2.4 Kondisi Menemukan Finish.....	41
4.4 Langkah Pembuatan.....	42
4.4.1 Pemodelan <i>Game Object</i> .....	42
4.4.1.1 <i>Character</i> .....	43
4.4.1.2 <i>Arena</i> .....	45
4.4.2 GUI .....	46
4.4.3 Importing.....	48
4.4.4 Scripting.....	52
4.4 Evaluasi.....	54
<b>BAB V KESIMPULAN DAN SARAN</b> .....	<b>56</b>
5.1 KESIMPULAN.....	56

5.2 SARAN.....	556
DAFTAR PUSTAKA.....	57
LAMPIRAN.....	

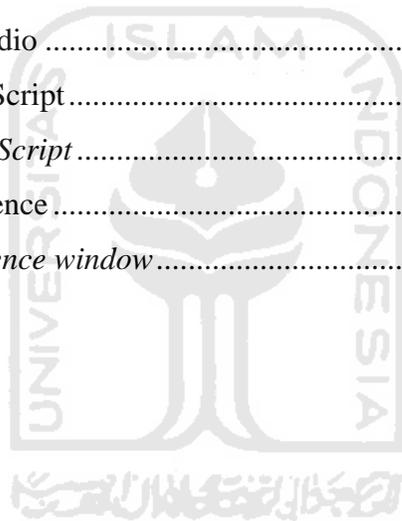


## DAFTAR GAMBAR

GAMBAR 2.1 Labirin .....	12
GAMBAR 3.1 Karater Kontrol.....	17
GAMBAR 3.2 Karater Mendapatkan Kristal .....	17
GAMBAR 3.3 Karakter Menghancurkan Hambatan.....	18
GAMBAR 3.4 Karakter Mendapatkan Titik Akhir. ....	18
GAMBAR 3.5 Arena Level 1 .....	19
GAMBAR 3.6 Arena Level 2 .....	20
GAMBAR 3.7 Arena Level 3 .....	20
GAMBAR 3.8 Arena Level .....	21
GAMBAR 3.9 Karakter .....	23
GAMBAR 3.10 Ranjau.....	23
GAMBAR 3.11 Penembak. ....	24
GAMBAR 3.12 Bomjatuh .....	24
GAMBAR 3.13 Musuh.....	25
GAMBAR 3.14 Halaman Menu Utama.....	25
GAMBAR 3.15 Menu Mulai Permainan.....	26
GAMBAR 3.16 Menu Pause .....	26
GAMBAR 3.17 <i>Gameplay</i> .....	27
GAMBAR 4.1 Halaman Menu Utama.....	30
GAMBAR 4.2 Halaman Pilih Level.....	30
GAMBAR 4.3 Halaman Game Over .....	31
GAMBAR 4.4 <i>Game Pause</i> .....	31
GAMBAR 4.5 Halaman Tamat <i>Game</i> .....	32
GAMBAR 4.6 <i>Screenshot Level1</i> .....	32
GAMBAR 4.7 <i>Screenshot Level2</i> .....	33
GAMBAR 4.8 <i>Screenshot Level3</i> .....	33
GAMBAR 4.9 <i>Screenshot Level4</i> .....	34

GAMBAR 4.10 <i>Character</i> .....	34
GAMBAR 4.11 Ranjau.....	34
GAMBAR 4.12 Boom .....	35
GAMBAR 4.13 Pelontar.....	35
GAMBAR 4.14 Musuh.....	35
GAMBAR 4.15 <i>Finish</i> .....	35
GAMBAR 4.16 Pengujian Tombol <i>Start</i> .....	36
GAMBAR 4.17 Tombol Pilih <i>Level</i> .....	37
GAMBAR 4.18 Tombol Arah .....	37
GAMBAR 4.19 Gambar <i>Character</i> Menembak.....	38
GAMBAR 4.20 Kondisi Mendapatkan Kristal.....	38
GAMBAR 4.21 Kondisi kristal Normal .....	39
GAMBAR 4.22 Kondisi Kristal <i>Get 1</i> .....	39
GAMBAR 4.23 Kondisi Terkena Ranjau.....	39
GAMBAR 4.24 Kondisi Normal .....	40
GAMBAR 4.25 Kondisi Berkurang .....	40
GAMBAR 4.26 <i>Character</i> Terjatuh.....	40
GAMBAR 4.27 Ranjau Ketika Meledak .....	41
GAMBAR 4.28 Musuh Meledak .....	41
GAMBAR 4.29 <i>Charackter</i> Berada di <i>Finish</i> .....	42
GAMBAR 4.30 <i>Tool Box</i> .....	43
GAMBAR 4.31 <i>Capsule</i> .....	43
GAMBAR 4.32 <i>editable poly</i> .....	43
GAMBAR 4.33 Object seletch dibentuk .....	44
GAMBAR 4.34 Mata.....	44
GAMBAR 4.35 Senjata .....	44
GAMBAR 4.36 Bentuk Akhir .....	45
GAMBAR 4.37 Denah.....	45
GAMBAR 4.38 Denah Dalam Bentuk 3d .....	45
GAMBAR 4.39 Lingkaran.....	46
GAMBAR 4.40 Setengah Lingkaran.....	46

GAMBAR 4.41 Setelah di Bentuk.....	46
GAMBAR 4.42 Setelah diCerminkan .....	46
GAMBAR 4.43 Setelah di Gabungkan.....	47
GAMBAR 4.44 Setelah diwarnai .....	47
GAMBAR 4.45 Nyawa Dalam 3 Kondisi .....	47
GAMBAR 4.46 Create New Project.....	48
GAMBAR 4.47 <i>Unity Interface</i> .....	48
GAMBAR 4.48 Memasukan arena permainan.....	49
GAMBAR 4.49 <i>Generate Collider</i> .....	50
GAMBAR 4.50 <i>Create New Material</i> .....	50
GAMBAR 4.51 PemberianMaterial .....	51
GAMBAR 4.52 Memasukan Audio .....	51
GAMBAR 4.53 Membuat Java Script.....	51
GAMBAR 4.54 <i>Window Editor Script</i> .....	52
GAMBAR 4.55 Scripting Reference.....	53
GAMBAR 4.56 <i>Scripting Reference window</i> .....	53



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Permainan merupakan salah satu cara yang dari dulu digunakan oleh manusia untuk menghilangkan stress dan kejenuhan. Dengan kata lain permainan berguna untuk refreshing dan menghilangkan ketegangan setelah melakukan rutinitas. Karena sifat dari permainan itu menyenangkan, seringkali permainan juga di gunakan sebagai media yang tidak membosankan jika diselipkan dengan suatu pembelajaran bagi penggunanya.

Belajar sambil bermain, adalah suatu keadaan yang sering diterapkan dalam sebuah konsep dalam sebuah permainan, labirin contohnya. Labirin merupakan permainan klasik yang tidak pernah hilang penerapannya sampai sekarang. Labirin merupakan permainan yang dimana seorang pemain harus melewati teka teki yang berbentuk jalan dan menemukan jalan keluarnya. Selain menyenangkan, permainan labirin juga dapat melatih otak manusia untuk mengingat dan memecahkan masalah.

Seiring dengan perkembangan zaman, permainan yang tadinya dimainkan secara nyata sekarang sudah mulai banyak diterapkan dalam *game* komputer. Teknologi komputer yang semakin pesat membuat perkembangan *game* komputer semakin bervariasi. *Game* 2 dimensi yang dirasa kurang mampu menampilkan visualisasi mendekati realita kini semakin ditinggalkan. *Game* 3d adalah salah satu bukti nyata perkembangan teknologi komputer yang diterapkan dalam *game*. Dengan kemampuan visual yang hampir mendekati realita, maka *game* 3d menjadi *game* yang paling diminati saat ini. Karena *game* 3d banyak diminati. Hal ini membuat *game* 3d terus dikembangkan untuk menghasilkan visual yang mendekati realita.

*Game engine* pun semakin berkembang dan membantu dalam pembuatan *game*. Unity adalah salah satu *game engine* yang saat ini sedang berkembang.

Dengan Kemampuan yang bisa berjalan di beberapa *platform*, membuat *game engine* ini menjadi salah satu pilihan terbanyak saat ini oleh devoloper *game*.

### 1.2. Rumusan masalah

Berdasarkan latar belakang masalah yang telah dikemukakan sebelumnya, dapat dibuat rumusan masalah sebagai berikut: Bagaimana mengatasi masalah visual yang terdapat pada game 2 dimensi labirin dengan cara membuat game labirin 3 dimensi dengan menggunakan unity.

### 1.3. Batasan Masalah

1. *Game* ini akan menampilkan bentuk 3 dimensi dari karakter, hambatan, dan arena permainan.
2. *Game* ini mengambil konsep dari permainan labirin
3. *Game* ini memiliki 4 level

### 1.4. Tujuan Penelitian

Tujuan utama penelitian tugas akhir ini adalah

1. Membuat game 3d yang mengambil konsep labirin
2. Membuat game 3d dengan menggunakan game engine unity

### 1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Memberikan alternatif hiburan dalam teknologi komputer
2. Mengasah kemampuan logika otak
3. Memberikan wawasan baru tentang *game engine* Unity

### 1.6. Metodologi Penelitian

Metode penelitian adalah langkah langkah yang digunakan dalam pengerjaan tugas akhir ini. Langkah - Langkah yang digunakan adalah sebagai berikut :

#### a. Studi Pustaka

Pengumpulan data dilakukan dengan cara mengumpulkan referensi dan literatur baik dari dalam buku atau kasus – kasus yang ada, yang mendukung dalam pembuatan *game* yang akan dibuat.

b. Analisa kebutuhan

Tahapan ini digunakan untuk menganalisis kebutuhan dari sistem yang meliputi masukan dan keluaran *game* 3D, *software*, *hardware* dan analisis *game*.

c. Perancangan pembuatan *Game*

Dalam tahapan ini *game* akan didesain dari sistem sesuai dengan konsep yang akan dikembangkan meliputi alur, dan hasil akhir.

d. Implementasi

Merupakan tahap penerapan prosedur sistem sesuai dengan hasil analisa dan desain *game* yang telah di implementasikan.

e. Hasil

Pada tahapan ini dilakukan untuk membahas hasil sistem yang telah dibangun.

### 1.7. Sistematika Penulisan

Untuk memudahkan dalam memahami laporan tugas akhir, dikemukakan sistematika penulisan agar menjadi satu kesatuan yang utuh. Adapun penulisan laporan tugas akhir ini adalah sebagai berikut.

## BAB I PENDAHULUAN

Bab ini berisi pembahasan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, metodologi pengembangan aplikasi dan sistematika penulisan.

## BAB II LANDASAN TEORI

Bab ini membahas tentang dasar teori yang berfungsi sebagai sumber referensi dalam membangun aplikasi ini. Dalam bab ini juga dibahas tentang konsep labirin yang dijadikan konsep

dasar dalam *game* ini, serta tentang *software-software* yang di gunakan dalam membangun aplikasi *game* ini

### **BAB III METODOLOGI**

Berisi analisis kebutuhan perangkat lunak dan perangkat keras yang diperlukan dalam membangun *game* ini. Serta berisi perancangan sistem seperti, perancangan antar muka, perancangan aturan permainan, perancangan design karakter, level, dan rintangan dalam *game* ini.

### **BAB IV HASIL DAN PEMBAHASAN**

Berisi hasil dari perancangan *game* yang telah di bangun. Analisis hasil sistem yang mencakup pengujian terhadap *game* ini.

### **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran yang merupakan rangkuman dari hasil analisis kinerja sistem yang telah diuraikan pada bab- bab sebelumnya, serta saran-saran untuk perbaikan aplikasi dimasa mendatang.

## BAB II

### LANDASAN TEORI

#### 2.1 *Game*

##### 2.1.1 Sejarah Singkat

*Game* telah dimainkan oleh manusia selama beribu-ribu tahun lamanya dan ini merupakan hal yang wajar bagi semua kultur budaya di dunia. Sepanjang sejarah di bumi, manusia telah menggunakan tongkat untuk menggambar *game board* yang simpel di permukaan tanah, kemudian membuat beberapa aturan untuk menyusun batu atau *obyek* lain untuk kemudian dimainkan. Sekitar 5000 tahun yang lalu, manusia mulai membuat *game board* yang lebih permanen, mereka menggunakan kayu atau lempengan tanah liat. Senet adalah salah satu *game* pertama di dunia yang berasal dari mesir. Seperti *game* awal lainnya, senet mempunyai makna religius. Orang-orang mesir percaya bahwa gambar-gambar yang berada pada kotak permainan melambangkan perjalanan hidup jiwa manusia setelah meninggal.

Beberapa *game board* yang tertua dikembangkan dari metode ramalan, atau menceritakan masa depan. Ada juga *game* yang memerlukan kecermatan yang tinggi seperti *game* strategi. Lebih dari 3000 tahun yang lalu, di china telah dikembangkan sebuah *game* strategi yang berasal dari metode ramalan, pemain melemparkan/menempatkan lempengan putih dan hitam pada garis dalam papan permainan dimana garis atau simbol tersebut mempunyai arti tersendiri.[HEN05]

##### 2.1.2 Definisi *Game*

Seiring dengan kemajuan zaman, pengertian *game* tidak hanya diartikan dalam bahasa indonesia sebagai permainan, tapi sekarang *game* lebih lazim digunakan hanya untuk istilah permainan yang menggunakan teknologi, seperti permainan komputer atau permainan yang menggunakan konsol tertentu.

Jika menurut wikipedia, *game* adalah aktivitas terstruktur atau semi terstruktur, yang biasanya dilakukan untuk bersenang-senang dan kadang juga digunakan

sebagai alat pembelajaran. Sedangkan menurut Cris Crawford *game* adalah gambar atau objek 2d,3d yang pergerakannya diatur oleh pemain.

menurut cris crawford, *game* di bagi menjadi 4 jenis, yaitu:[FE105]

1. *Board Games*, merupakan permainan yang media bermainnya menggunakan papan seperti tenis meja, catur dan sebagainya.
2. *Card Game*, permainan yang menggunakan kartu sebagai media bermainnya.
3. *Athletic Games*, permainan yang diperuntukkan bagi anak anak misalnya permainan petak umpet.
4. *Computer Games*, permainan yang menggunakan komputer sebagai media bermainnya untuk computer games sendiri di bagi menjadi 2 bagian, yaitu: skill an action games, dan strategy games

*a. Skill and Action Games*

Merupakan jenis permainan yang real time play. Dalam permainan ini sangat mementingkan koordinasi antara mata, tangan dan fikiran untuk beraksi dengan cepat dan tepat. Skill and action games dibagi menjadi empat, yaitu sport game(tennis, bola voli) maze game(pac-man, snake) combat game(star raider, star wars) dan race games(road race, dog daze).

*b. Strategy games*

Merupakan permainan yang menitik beratkan pada permainan yang tidak realtima play tetapi pada pemikiran yang tepat dan pemain bisanya menyelesaikan permainan dalam waktu yang cukup lama. *Game* ini juga tidak memerlukan skill an action untuk menyelesaikan suatu permainan.

### 2.1.3 Teori Pembuatan Game

Menurut Crish Crawford sebagai seorang yang sangat berperan dalam perkembangan *game* di dunia mengungkapkan, dalam membuat *game* memerlukan langkah-langkah sebagai berikut:[TGO]

#### 1. Menentukan tipe *game*

Dalam membuat *game*, terlebih dahulu harus menentukan tipe *game* apa yang akan dibuat. Apakah *game* ini mau diarahkan *game* action atau ke *game* strategi.

#### 2. Definiskan model permainan

Dalam langkah ini, seorang *game desingner* harus mendeskripsikan modelnya permainan dalam sebuah cerita. Langkah ini biasanya akan mengakitkannya ide *game* yang tadinya sederhana menjadi berkembang.

#### 3. Mendefinisikan *game worldsnya*

*Game* worlds adalah elemen elemen penting yang harus ada dalam sebuah *game* antar lain:

- *Game Board*

Ini mirip seperti *story board* dalam sebuah film. Tapi ini untuk *game*.

- *Isntruksi* permainan

Berisi perintah yang harus dijalankan oleh pemain.

- Informasi

Semisal jumlah tenaga, atau jumlah amunisi.

- Penghargaan

Jika didalam sebuah permainan tidak ada unsur penghargaan terhadap pemain, ini tentu akan membuat *game* tidak menarik dan terasa sangat sia sia untuk dimainkan. Dengan adanya penghargaan, tentu ini akan memotivasi pemain dalam memainkan *game* ini.

- Variasi

Game yang statis, akan membuat jenuh pemain. Ini tentu bertolak belakang dengan tujuan *game* dibuat. Variasi dalam permainan akan membuat *game* tidak membosankan. Variasi juga tidak boleh berlebihan, karena bisa menimbulkan kinerja otak yang berlebihan sehingga malah dapat menimbulkan *stress*.

- Tingkat kesulitan

Ini bisa disamakan dengan level. Hampir mirip dengan variasi, tingkat kesulitan membuat pemain akan menjadi termotivasi dan merasakan kepuasan tersendiri ketika berhasil melewati level demi level.

4. Pastikan permainan bisa dimainkan

Permainan yang tidak mungkin diselesaikan akan membuat program permainan tidak dapat dimainkan.

5. Rancang program sebaikmungkin

Gunakan teknik pemograman yang sesuai dalam membuatnya dan buat program yang mudah dimodifikasi dan dikembangkan.

6. Pengujian program

Program yang selesai harus di uji. pertama, program diuji oleh perancangnya sendiri(alpha test) untuk menentukan kesalahan logika yang mungkin terjadi. Kedua, program diuji oleh orang lain(beta test) untuk mengetahui kesalahan logika yang tidak terlihat oleh perancangnya sehingga memberi masukan saran atau ide dalam pengembangan program permainan.

#### 2.1.4 Elemen *Game*

Di dalam sebuah *game* terdapat beberapa elemen elemen yang akan membentuk sebuah *game* itu sendiri, yaitu:[FE105]

1. *Rules*

dalam sebuah *game* tentu harus memiliki *rules*/aturan. aturan inilah yang

menentukan aksi dan gerakan pemain dalam sebuah *game*

2. *Victory Condition* dan *Lose Condition*

Karena adanya *victory condition*(kondisi kemenangan), maka *game* juga memiliki *lose condition*(kondisi kekalahan). Kadang kondisi kekalahan ini bersifat implisit, yaitu jika pemain tidak mendapatkan kondisi kemenangan maka dinyatakan kalah, sedangkan kondisi yang eksplisit misalnya tokoh telah mati karena kehabisan nyawa.

3. *Setting*

Sebuah *game* mengambil tempat pada suatu dunia atau seting tertentu. Contohnya pada permainan bola yang bersetting pada sebuah lapangan dengan batasan batasan tertentu

4. *Interaction Model*

Yaitu cara pemain berinteraksi dengan *game* dan melakukan aksi untuk menghadapi tantangan dari *game* tersebut.

5. *Perspective*

Yaitu menjelaskan bagaimana sudut pandang pemain melihat *gameworld* dari suatu *game* pada layar

6. *Role*

Yaitu tokoh yang dimainkan oleh pemain dalam suatu *game*. Dengan adanya peran ini maka pemain akan lebih mudah untuk memahami tujuan apa yang sebenarnya ingin dicapai dan aturan apa yang dimainkan

7. *Mode*

Pemain selalu ingin mencapai atau menyelesaikan hal yang sama dengan cara yang sama pula. Namun ada juga *game* yang memiliki *mode* yang nyata, yaitu di mana *game* playnya berubah dari satu *mode* ke *mode* yang lainya

8. *Structure*

Hubungan antara *mode* dan aturan menentukan kapan dan mengapa *game* berubah secara bersama sama membentuk struktur permainan.

9. *Realism*

*Game* yang menggambarkan sebuah dunia, bahkan mungkin dunia yang

khayal. Sebuah *game* yang menerapkan akal sehat dan logika pada aturan permainannya dapat dikatakan sebagai *game* yang realistis

#### 10. *Story*

Beberapa *game* memiliki alur cerita yang *linear* atau hanya satu jalan cerita dan bersifat *non* interaktif, namun ada juga yang bersifat interaktif dimana cerita akan berbeda atau bercabang tergantung pada pilihan atau tindakan yang diambil. Sehingga cerita akhir(ending) yang didapat akan berbeda pula dengan *mutiple ending*

### 2.1.5 *Game Dev*

*Game Dev* merupakan pengembangan rancangan konsep yang digunakan untuk membangun suatu rancangan *game*. Dalam pengembangan *game* memiliki beberapa unsur yaitu sebagai berikut:[ROG03]

#### a. *Game Difficulty Balancing*

*Game Difficulty Balancing* juga dikenal sebagai penyesuai kesulitan dinamis atau penyeimbang permainan dinamis, adalah proses otomatis pengubah parameter skenario dan perilaku dalam sebuah *game* secara *real-time* berdasarkan kemampuan pemain. Hal ini dilakukan dalam rangka untuk menghindari pengguna menjadi bosan (jika permainan terlalu mudah) atau frustrasi (jika terlalu keras). Tujuan dari menyeimbangkan kesulitan dinamis ini adalah untuk menjaga pengguna agar tetap tertarik dari awal sampai akhir dan untuk memberikan tingkat tantangan bagi pengguna.

#### b. *Rule Play*

*Rule play* adalah aturan – aturan yang dibuat untuk memberikan batasan – batasan dalam *game*. Hal ini memberi persepsi pada pengguna untuk mengikuti aturan yang ada dalam *game*.

## 2.2 Labirin

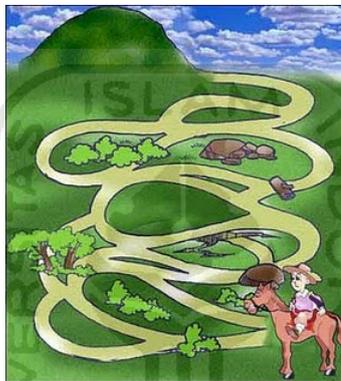
Labirin (maze) adalah permainan yang sudah tidak asing di telinga kita. Labirin adalah jaringan jalan yang rumit dan berliku-liku. Sejak zaman dahulu, labirin telah digunakan dalam berbagai kepentingan, mulai dari proteksi keamanan hingga hiburan. Pada umumnya, labirin dibuat untuk tujuan hiburan. Dalam kehidupan nyata, labirin dapat ditemukan pada susunan jalan kecil atau gang-gang di kawasan perumahan. Sangat sulit bila seseorang yang asing dengan daerah tersebut untuk mencari jalan. Bila seseorang mengetahui metode untuk keluar dari sebuah labirin, maka mereka dapat dengan mudah mengatasi kesulitan yang dirasakan.

Ada berbagai jenis metode pencarian jalan keluar untuk masalah labirin ini. Salah satu metode pencari jalan keluar yang sederhana adalah metode wall follower. Metode ini juga dikenal dengan aturan tangan kanan (right-hand rule) atau aturan tangan kiri (left hand rule). Metode ini merupakan kombinasi dari algoritma runut-balik (backtracking) dan algoritma greedy. Metode ini akan mencari jalan sesuai dengan dinding labirin, baik itu ke kiri maupun ke kanan. Makalah ini akan mengulas tentang penggunaan metode wall follower untuk menemukan jalan keluar dalam sebuah labirin.

Labirin atau maze adalah sebuah puzzle dalam bentuk percabangan jalan yang kompleks dan memiliki banyak jalan buntu. Tujuan permainan ini adalah pemain harus menemukan jalan keluar dari sebuah pintu masuk ke satu atau lebih pintu keluar. Bisa juga kondisi pemain menang yaitu ketika dia mencapai suatu titik di dalam labirin tersebut.

Labirin dalam dunia nyata banyak dibuat di taman atau ruangan-ruangan dengan pembatas berupa pagar tanaman, tembok atau pagar. Ukurannya bervariasi, tergantung ukuran ruangan atau taman tersebut. Labirin ini biasanya memang dirancang untuk menjadi sebuah atraksi permainan (misalnya rumah kaca) atau hanya sebagai hiasan saja. Selain itu banyak labirin yang terbentuk secara “tidak sengaja”. Contohnya jalan-jalan kecil atau gang-gang yang terbentuk diantara rumah-rumah pada kawasan pemukiman. Labirin-labirin ini secara tidak langsung “menyesatkan” orang asing yang masuk ke dalamnya.

Pada umumnya pembuatan labirin hanya untuk hiburan belaka. Namun, banyak bangunan yang menerapkan labirin sebagai salah satu sistem keamanan agar orang yang tidak berkepentingan atau tidak dikenal sulit untuk masuk ke dalam bangunan. Labirin untuk permainan biasanya dicetak dalam sebuah kertas untuk diselesaikan oleh pemain. Permainan dilakukan dengan cara menuliskan jalan yang telah ditempuh menggunakan pensil atau hanya dengan menunjuk jalannya menggunakan jari. Labirin terbagi menjadi beberapa kategori sesuai jenisnya, yaitu Labirin 2 dimensi, 3 dimensi, bentuk segitiga, sigma, dan masih banyak lagi. [WAH03]



**Gambar 2. 1** Labirin

## **BAB III**

### **METODOLOGI**

#### **3.1 Analisis Kebutuhan**

Analisis merupakan identifikasi terhadap hal-hal yang berkaitan dengan detail atau struktur perancangan *game* ini. Dapat dibayangkan sebagai pemecahan sistem yang utuh menjadi bagian-bagian penyusunnya untuk mengetahui permasalahan yang ada. Analisis diperlukan untuk perancangan *game* yang akan dibuat dan berhubungan dengan hasil yang ingin dicapai oleh *game* itu sendiri.

*Game* yang dirancang adalah *game* yang mengambil konsep permainan labirin. dengan tampilan 3 dimensi dan interface sebagai informasi dalam permainan.

#### **3.2 Hasil Analisis**

Berdasarkan analisis yang telah dilakukan maka diketahui apa saja yang dibutuhkan dalam membangun *game* ini. Hasil yang didapat dari analisis terhadap perancangan *game* ini, ada beberapa hal yang perlu diperhatikan. Membuat *game play* yang menyenangkan karena ini sebuah *game*, bagian tersebut tidak bisa sembarang karena sangat berpengaruh terhadap ketertarikan pemain dalam memainkan *game* ini. Selain itu berdasarkan analisis yang telah dilakukan maka diketahui apa saja yang menjadi masukan sistem, keluaran sistem, antarmuka yang diinginkan dan metode yang digunakan sistem, serta kebutuhan perangkat lunak dari sistem tersebut.

##### **3.2.1 Analisa Kebutuhan Masukan**

Dalam *game* ini user tidak memberi masukan yang berupa informasi. *User* hanya memberi masukan berupa kontrol terhadap karakter yang dimainkan. Pengontrolan dilakukan menggunakan keyboar dan mouse. *Keyboard* mengontrol pergerakan kemana karater berjalan sedangkan *mouse* hanya digunakan untuk mengklik menu pilihan yang ada dalam interface dalam *game* tersebut.

### 3.2.2 Analisi Kebutuh Pengeluaran

Output yang dihasilkan dari *game* ini berupa:

1. Informasi *live* pemain
2. Informasi waktu permainan
3. Informasi *score*
4. *Sound* efek

### 3.2.3 Analisi Kebutuhan Proses

Proses yang dibutuhkan untuk merancang *game* ini sebagai berikut:

1. Aplikasi dapat melakukan *action* dari instruksi control yang diberikan oleh pemain
2. Musuh dari setiap level akan memiliki kecerdasan buatan yang mengakibatkan musuh akan menembaki pemain utama
3. Pemain dan musuh bisa mengalami kondisi kalah dan kondisi menang

### 3.2.4 Analisi Kebutuhan Antar Muka

Atar muka atau yang biasa disebut *interface* merupakan bagian yang penting dalam melakukan perancangan sebuah *system*. *Interface* akan memberikan informasi kepada pemain apa saja yang terjadi di dalam *game* ini. *interface* di *game* ini mempunyai point-point sebagai berikut:

1. Menginformasikan waktu permainan.
2. Menginformasikan misi permainan.
3. Menginformasikan kesempatan dalam permainan.
4. Menampilkan tombol- tombol yang akan menginteraksikan pemain dengan *game* ini.

### 3.2.5 Analisi Kebutuh Perangkat Lunak

Perangkat lunak merupakan alat yang sangat penting dalam pembuatan suatu aplikasi, pemilihan perangkat lunak yang tepat guna tentu akan

memudahkan penyelesaian aplikasi ini. Adapun perangkat lunak yang digunakan dalam pembuatan aplikasi ini antara lain

### 1. Microsoft windows 7

Microsoft windows 7 ini merupakan sistem operasi yang digunakan dalam merancang *game* ini. Windows 7 dipilih karena ini merupakan versi yang bisa menjalankan perangkat lunak Unity di atas sistem operasi ini

### 2. Unity 3.1

Merupakan *software* yang sangat penting dalam perancangan *game* ini. Software ini merupakan *game engine* yang memegang peranan penting dalam membentuk interaktifitas dalam sebuah *game*. *Software* ini hanya menangani *scripting* dan tidak dianjurkan melakukan design 3d. *Script* yang di gunakan dalam *software* ini adalah javascript dan C.

### 3. Adobe illustrator

Adobe illustrator merupakan perangkat lunak yang digunakan untuk mendesain gambar dengan format *vector*. Perangkat lunak ini digunakan untuk membuat tampilan *interface* yang berupa gambar 2d. Bagian yang dibuat menggunakan Adobe Illustrator ini adalah *power* dari pemain dan beberapa informasi untuk pemain.

### 4. Autodesk 3D MAX

Autodesk 3D MAX merupakan *software* yang biasa digunakan untuk merancang bangunan 3d dan biasa digunakan oleh para arsitek untuk mendesign suatu bangun ruang. *Software* ini juga biasa digunakan untuk membuat film animasi. Dalam merancang *game* ini, *software* ini digunakan untuk mendesign arena permainan, dan mendesign karakter pemain dan juga musuh.

#### 3.2.6 Analisa Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam menjalankan *game* ini antar lain:

1. Prosesor inter core2duo
2. RAM 1 giga
3. *VGA card*
4. *Keyboard*
5. *Mouse*
6. *Spiker*
7. Monitor

### **3.3 Perancangan**

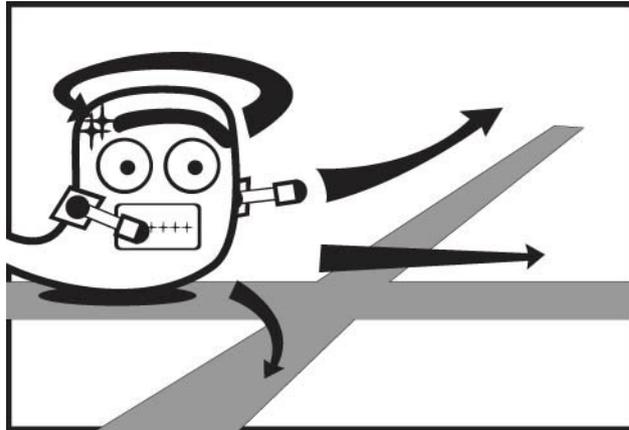
#### **3.3.1 Hasil Perancangan**

Dari hasil analisis yang telah dilakukan, dibutuhkan perancangan aplikasi yang baik meliputi *input*, proses dan output sehingga menjadi sebuah aplikasi yang baik. Untuk mencapai aplikasi yang diinginkan maka perancangan aplikasi dibagi kembali menjadi beberapa bagian, meliputi :

- a. Perancangan Game board
- b. Perancangan arena permainan
- c. Perancangna aturan permainan
- d. Perancangan karakter pemain dan musuh
- e. Perancangan *Interface*.

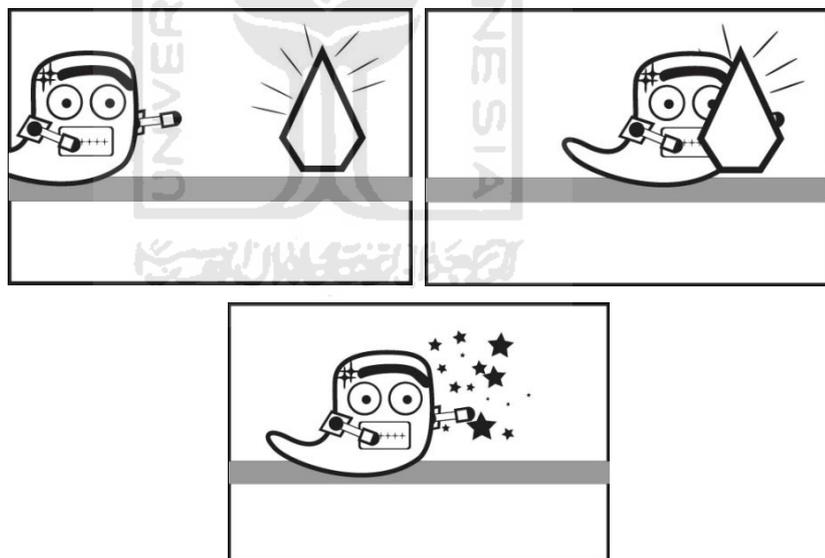
#### **3.4 Game Board**

*Game board* adalah alur cerita yang terdapat dalam sebuah *game*, yang didalamnya terdapat informasi-informasi yang berkaitan dengan misi dalam *game*. Sehingga dengan adanya *Game board* dalam sebuah *game* pengguna mengerti apa yang harus dilakukan dalam memainkan *game*. Berikut adalah *Game board* dari *game* ini



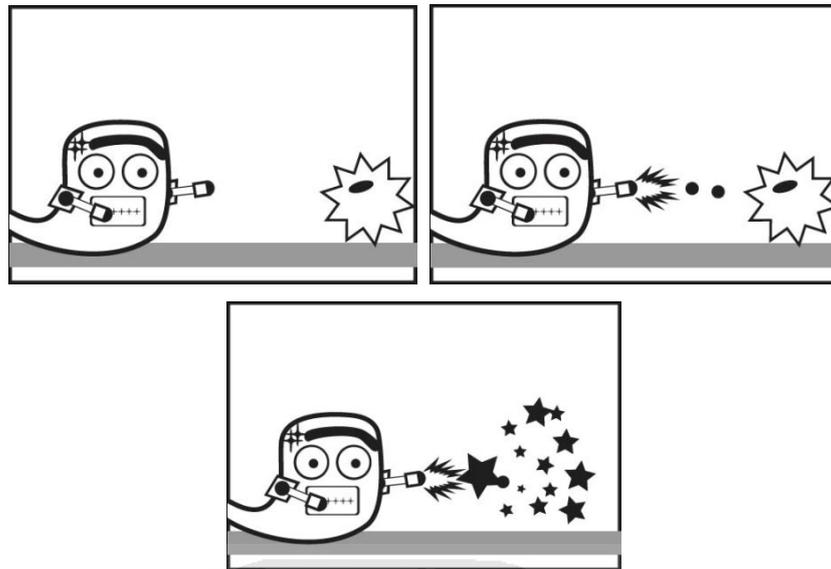
**Gambar 3.1** Karater Kontrol

*Game board* di atas menggambarkan karakter sedang menelusuri arena. Karakter bisa melakukan pergerakan berbelok, hingga berputar. Karakter hanya tidak bisa meloncat.



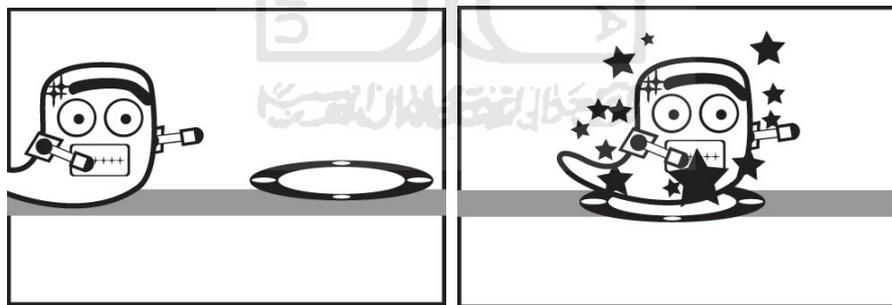
**Gambar 3.2** Karater Mendapatkan Kristal

*Game board* di atas menjelaskan ketika karakter menemukan sebuah kristal untuk diambil. Mengambil kristal tersebut dengan cara mendekatkannya. Setelah kristal tersebut bersentuhan dengan karakter, ini mengakibatkan jumlah perolehan kristal bertambah, dan kristal akan menghilang.



**Gambar3.3** Karakter Menghancurkan Hambatan

*Game board* di atas menjelaskan ketika karakter berhadapan dengan hambatan. Karakter tersebut bisa menghancurkan hambatan dengan ditembak. Ketika hambatan terkena tembakan dari karakter, maka hambatan tersebut akan hancur dan menghilang.



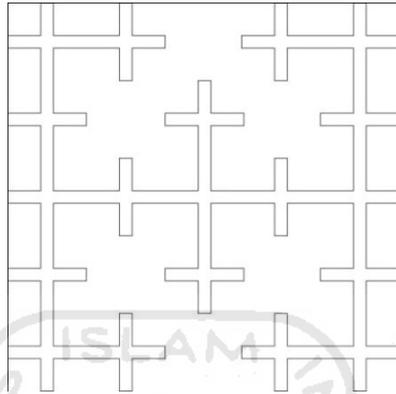
**Gambar 3.4** Karakter Mendapatkan Titik Akhir

*Game board* di atas menjelaskan ketika karater menemukan tujuan akhir dari tiap level dari game ini. Untuk melanjutkan level berikutnya, karater harus berada di atas titik tersebut. Ketika karakter berada di atas titik tersebut, maka akan keluar partikel yang menggambarkan terjadinya sebuah teleportasi dari satu lokasi ke lokasi yang lain.



### 3.5.2 Arena Level 2

Level 2 tidak jauh berbeda dengan level satu. Hanya saja bentuknya lebih divariasikan. Berikut gambar dari level 2

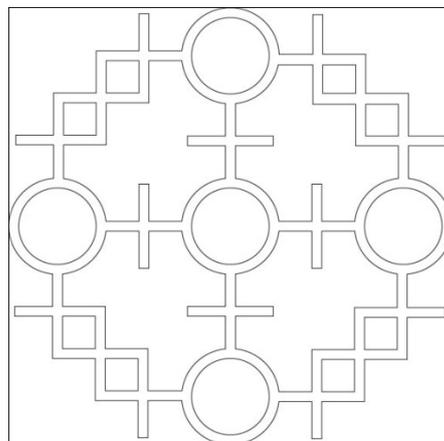


**Gambar 3.6** Arena Level 2

Kesulitan dari level 2 ini adalah hambatan ranjau dan di tambah dengan hambatan boom yang jatuh dari atas. Tentu ini membuat pemain harus lebih hati-hati dalam mengontrol karakter

### 3.5.3 Arena Level 3

level 3 menggabungkan bentuk lingkaran dengan bentuk persegi. Maka jadilah denah sebagai berikut.

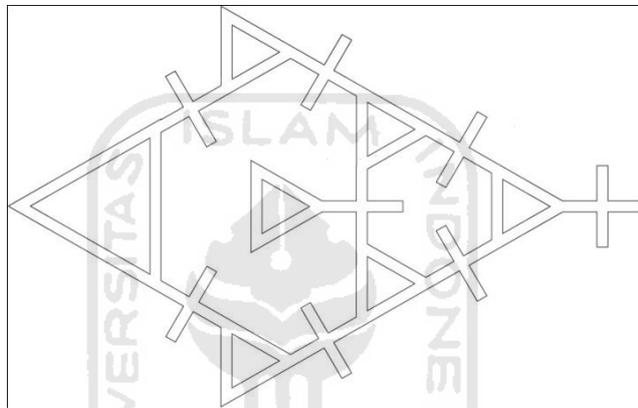


**Gambar 3.7** Arena Level 3

Kesulitan dari level 3 ini adalah hambatan ranjau ditambah dengan sebuah pelontar peluru yang selalu mengikuti kemana pemain mengarahkan karakter untuk berjalan. Hambatan tersebut semakin susah untuk dihindari.

#### 3.5.4 Arena Level 4

Arena 4 dibuat dari bentuk segi tiga yang dimodifikasi dengan beberapa bentuk bangunan yang lain.



**Gambar 3.8** Arena Level 4

Kesulitan dari level 4 ini adalah hambatan berupa monter yang bisa berjalan sambil menembak. Ini membuat level ini lebih susah dibanding dengan level sebelumnya karena akan banyak peluru yang berjalan.

### 3.6 Perancangan Aturan Permainan

Dalam permainan tentu harus memiliki aturan yang membuat permainan ini mempunyai persyaratan untuk mencapai titik akhir atau yang bisa disebut dengan goal dalam sebuah permainan. Berikut aturan permainan dalam *game* ini.

1. *Player* harus mencapai tujuan

Karena ini *game* yang mengambil konsep dari labirin yang dimana aturan permainan labirin itu sendiri harus menemukan

tujuan akhir. Begitu juga dengan *game* ini, harus mencapai tujuan yang ditentukan oleh game.

2. Berpacu dengan waktu.

Jika hanya mencari tujuan tanpa batasan waktu, *game* ini tentu akan tidak menarik. Maka dari itu, *game* ini mempunyai waktu pencarian jalan yang tidak boleh dilewati oleh pemain. Jika pemain terlalu lama dalam mencari jalan, permainan selesai dan pemain dianggap kalah.

3. Mengumpulkan kristal sebelum finish

Pemain harus mengumpulkan kristal dengan jumlah yang ditentukan di setiap arena sebagai syarat mencapai finish. Pemain tidak bisa melewati satu arena jika kristal yang di kumpulkan masih di bawah jumlah yang diperintahkan dari permainan ini.

4. Memiliki 3 kesempatan

Pemain hanya memiliki 3 kesempatan untuk jatuh dari arena. Dan pemain hanya memiliki 3 kesempatan untuk terkena musuh atau senjata musuh. Jika lebih dari itu, permainan dianggap selesai dan pemain dianggap kalah

5. Kondisi menang

kondisi menang menang diperoleh jika pemain bisa menemukan 10 kristal dan menemukan tujuan akhir

6. Kondisi kalah

a) Kondisi kalah terjadi jika pemain tidak bisa mengumpulkan kristal dan titik akhir sebelum waktu yang ditentukan habis.

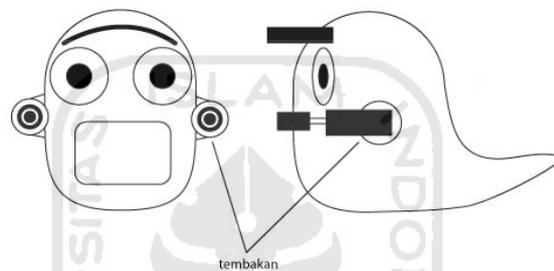
b) Pemain kehabisan nyawa.

### 3.7 Perancangan *Game Object*

Dalam sebuah permainan tentu ada pemain yang memainkan karakter utama. Dan ada juga karakter musuh yang mejadi tantangan pemain dalam menyelesaikan permainan ini. Berikut pembahasan perancangan karakter utama dan musuh/hambatan

#### 3.7.1 Karakter Utama

Karakter dari pemain utama ini terinspirasi dari cacing terdiri dari 3 bagian, yaitu kepala, badan, dan buntut. Berikut design karater utama



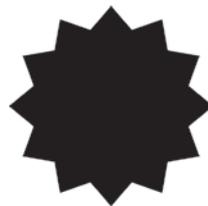
Gambar 3.9 Karakter

#### 3.7.2 Karakter Musuh/Hambatan

Karakter musuh disini bersifat menghambat/mengahalang halangin pemain untuk menelusuri arena permainan dalam *game* ini. Ada beberapa design karater musuh dalam *game* ini, tentu dengan fungsi yang berbeda.

##### 1. Ranjau

Bersifat tidak menyerang, tapi jika terkena oleh pemain akan mengurangi satu nyawa yang dimiliki oleh pemain. Hanya bisa dihancurkan dengan ditembak



Gambar 3.10 Ranjau

## 2. Musuh Penembak

Musuh ini tidak bergerak kemana mana, hanya saja musuh yang satu ini menembakan peluru ke arah pemain. Jika pemain terkena tembakan tersebut efeknya sama seperti terkena ranjau. Musuh yang satu ini hanya bisa dihancurnya dengan di tembak. Berikut gambar dari musuh penembak.



**Gambar 3.11** Penembak.

## 3. *Boom* Jatuh

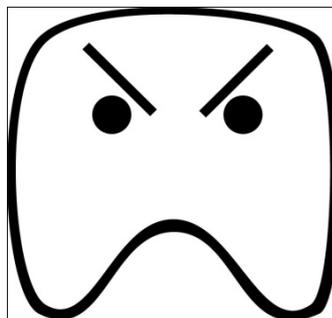
Hambatan yang satu ini bersifat tiba tiba. Datang dari atas, dan bila terkena dengan pemain akan berefek sama dengan hambatan hambatan sebelumnya. Berikut gambar dari bom jatuh.



**Gambar 3.12** Bomjatuh

#### 4. Musuh

Musuh ini memiliki kemampuan menembak dan berjalan. Untuk menghancurkan musuh ini adalah dengan menembak



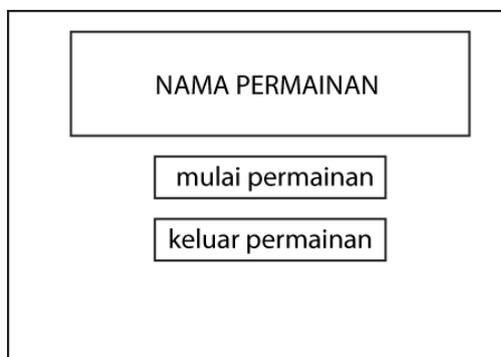
**Gambar 3.13** Musuh

### 3.8 Perancangan Antar Muka

Pada pembuatan sistem ini desain interface sangat penting. Dalam pembuatan desain ada beberapa hal yang harus diperhatikan, yang terpenting adalah membuat sebuah tampilan yang *user friendly* dan sesuai dengan segmen penggunaannya.

#### 3.8.1 Halaman Menu

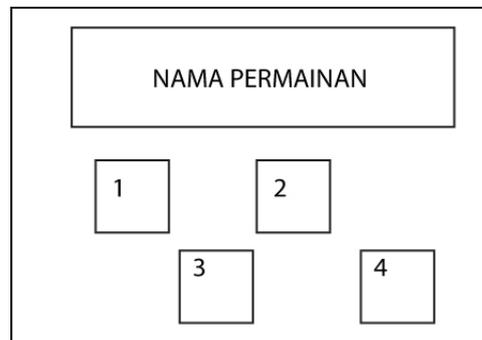
Gambar berikut merupakan gambar desain tampilan untuk halaman menu utama Halaman ini merupakan halaman utama dari aplikasi ini.



**Gambar 3.14** Halaman Menu Utama

### 3.8.2 Halaman Menu Mulai permainan

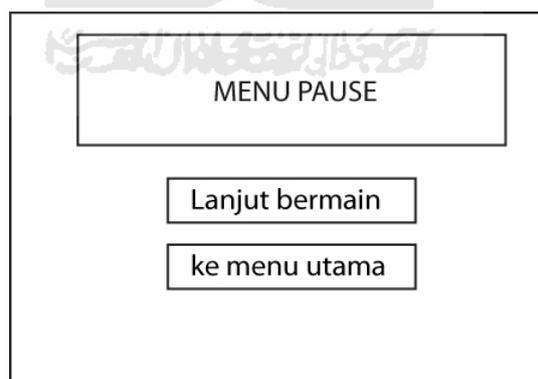
Halaman ini memberikan pilihan level yang akan dimainkan, berikut design dari halaman menu mulai permainan.



**Gambar 3.15** Menu Mulai Permainan.

### 3.8.3 Halaman Menu Pause

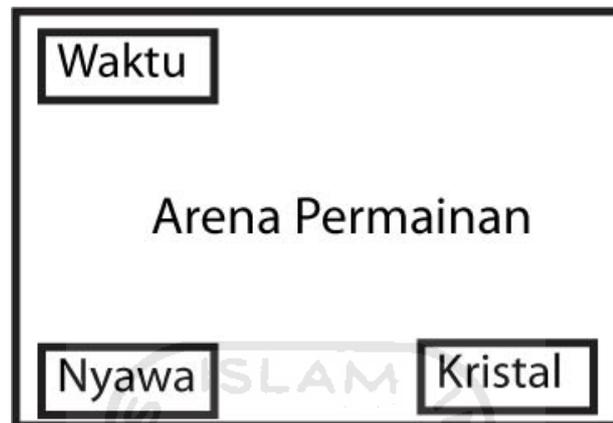
Halaman pause ini hanya tampil ketika pemain menghentikan permainan untuk sementara. Ada 2 menu pilihan yang akan tampil ketika permainan dihentikan. Lanjut memainkan, atau kembali kemenu utama. Berikut design tampilan dari menu *pause*.



**Gambar 3.16** Menu Pause

### 3.8.4 Halaman *Gameplay*

Halaman ini menjelaskan informasi apa saja yang ditampilkan ketika permainan dijalankan.



Gambar 3.17 *Gameplay*

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi Perangkat Lunak

Bagian ini adalah suatu bagian dimana aplikasi yang telah dirancang akan dibahas implementasinya. Dengan adanya pembahasan aplikasi, maka akan diketahui apakah aplikasi yang telah dihasilkan sesuai dengan perancangan atau tidak.

##### 4.1.1 Batasan Implementasi

Dalam implementasi penyelesaian tugas akhir, Rancang Bangun Game Labirin Menggunakan Unity, dibuat menggunakan *software* dan *hardware*, sebagai berikut :

###### A. SOFTWARE

1. UNITY 3.1

*Software* ini digunakan untuk membangun keseluruhan game ini.

2. 3D MAX

*Software* ini digunakan untuk mendesign semua objek 3d dalam game ini.

3. Adobe Illustrator

*Software* ini digunakan untuk mendesign semua tampilan 2d yang nantinya akan digunakan untuk dalam game ini.

###### B. Hardware

Aplikasi ini menggunakan beberapa *software* grafis dan *hardware* yang digunakan untuk membangun aplikasi tersebut sehingga dibutuhkan *hardware* yang memenuhi dalam pembuatan aplikasi ini, antara lain :

- a. Processor AMD Venom X2 2.2 GHz.
- b. RAM(Random Acces Memory) 2GB DDR2.
- c. Harddisk 80GB
- d. DVDR

- e. VGA ATI RADEON HD 3200 GRAPHICS
- f. Keyboard
- g. Mouse
- h. Monitor dengan resolusi 1024 x 768 pixel.
- i. Speaker.

#### 4.1.1 Implementasi Prosedural

Implementasi prosedural adalah penerapan perancangan program yang dibuat dengan *script* sehingga menjadi sebuah aplikasi yang lengkap. Dalam Perancangan Game ini menggunakan text editor yang terdapat didalam program Unity itu sendiri.

Bahasa yang bisa digunakan dalam *game engine* Unity adalah javascript dan C. tapi dalam perancangan game ini hanya menggunakan javascript. Ini dikarenakan *javascript* adalah bahasa yang usianya lebih muda di banding dengan bahasa C, dan pemahamannya pun lebih mudah.

#### 4.2 Hasil

Hasil dari perancangan *Game* Labirin menggunakan Unity terdapat beberapa halaman *user interface*, dan pembahasan *game object* yaitu:

1. Halaman menu utama, dimana terdapat menu utama yang akan digunakan untuk memulai permainan.
2. Halaman kedua adalah halaman pilih level.
3. Halaman ketiga adalah halaman ketika *gameover*.
4. Halaman keempat, "*pause game*", dimana ketika *user* menghentikan permainan untuk sementara waktu.
5. Halaman kelima adalah halaman ketika menyelesaikan permainan hingga akhir.
6. Halaman ke enam adalah halaman tampilan *game* dari tiap *level* nya
7. Tampilan *game object*, berisi tampilan *game object* yang terdapat dalam *game* ini

#### 4.2.1 Halaman Menu Utama

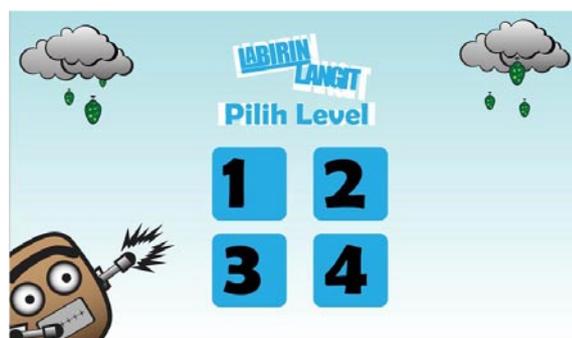
Tampilan halaman menu utama merupakan halaman utama yang terdapat dua menu utama di awal permainan. Menu yang terdapat dalam halaman ini adalah "*start*" sebagai perintah untuk memulai *game* ini, dan "*exit*" sebagai perintah untuk keluar dari *game* ini. Pada halaman ini terdapat background untuk membuat *game* ini menjadi lebih hidup



Gambar 4.1 Halaman Menu Utama

#### 4.2.2 Halaman Pilih level

Tampilan halaman Pilih *Level* adalah halaman dimana pemain bisa memilih levelnya. Pemilihan *level* disini sesuai dengan sejauh mana *game* ini dimainkan. Jika dimulai dari awal, maka *level* yang tersedia hanya *level* satu, sedangkan *level* yang lain masih terkunci.



Gambar 4.2 Halaman Pilih Level

### 4.2.3 Halaman *Game Over*

Pada halaman ini tidak berisi menu pilihan. Halaman ini hanya berisi gambar menandakan si pemain dalam kondisi kalah. Halaman hanya akan tampil beberapa detik. Setelah itu halaman ini akan melanjutkan kembali ke halaman utama



Gambar 0.1 Halaman *Game Over*

### 4.2.3 Halaman *Game Pause*

Halaman ini akan muncul di tengah tengah permainan ketika pemain ingin menghentikan permainan untuk sementara waktu



Gambar 0.4 *Game Pause*

#### 4.2.4 Halaman Tamat *Game*

Halaman ini akan muncul pemain berhasil mejalani dan melewati semua *level* yang terdapat dalam *game* ini.



Gambar 0.5 Halaman Tamat *Game*

#### 4.2.5 Halaman level 1

Pada *level* 1 rintangan untuk yang terdapat dalam *level* ini masih mudah, hanya sebuah ranjau yang bergerak sesuai pola yang sudah di tentukan.



Gambar 0.6 Screenshot *Level* 1

#### 4.2.6 Halaman Level 2

Pada *level 2* tingkat kesulitan sudah lebih tinggi dari *level* sebelumnya. Selain adanya ranjau seperti pada *level 1*. Di *level 2* terdapat rintangan tambahan berupa *boom* yang jatuh dari atas secara acak.



Gambar 0.7 Screenshot Level 2

#### 4.2.7 Halaman Level 3

Pada *level 3* terdapat rintangan berupa objek yang bisa menembaki pemain.



Gambar 0.8 Screenshot Level 3

#### 4.2.8 Halaman Level 4

Pada *level* 4 terdapat hambatan yang dapat bergerak sesuai pola sambil menembaki peluru yang tak terarah.



**Gambar 0.2** Screenshot Level 4

#### 4.2.9 Tampilan *Game Object*

berikut adalah bentuk bentuk *game object* yang terdapat game ini

a. *Character*



**Gambar 4.10** *Character*

b. Ranjau



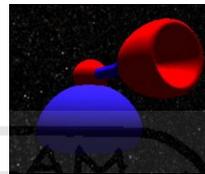
**Gambar 4.11** Ranjau

c. *Boom*



**Gambar 4.12** *Boom*

d. Pelontar



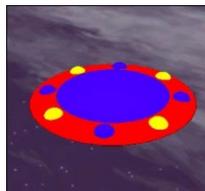
**Gambar 4.13** Pelontar

e. Musuh



**Gambar 4.14** Musuh

f. *Finish*



**Gambar 4.15** *Finish*

### 4.3 Pengujian Perangkat Lunak

Pengujian dan analisa ini merupakan bagian penting dalam pembuatan aplikasi ini. Dalam pengujian dan analisa ini akan ditemukan kesalahan-kesalahan sistem selama pembuatan, yang nantinya akan menjadi evaluasi bagi *programmer*.

Pengujian ini dilakukan dengan menggunakan *Black box testing* yaitu pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada *output* yang dihasilkan yang merespon input yang dipilih dan kondisi eksekusi serta untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu. Berikut pengujian dan analisa yang dilakukan

#### 4.3.1 Pengujian Tombol

Pengujian ini adalah pengujian yang dilakukan terhadap tombol tombol yang terdapat dalam *GUI* game ini.

##### 4.3.1.1 Pengujian Tombol start dan exit

Tombol *start game* membesar jika *pointer* di arahkan ke atas tombol *start*. tombol *start* akan mengantarkan pemain ketampilan pilih level. Begitu juga dengan tombol *exit* yang akan membesar ketika *pointer* mouse di arahkan keatas tombol *exit*. Tombol *exit* berisi perintah untuk keluar dari permainan ini.



Gambar 0.16 Pengujian Tombol Start

##### 4.3.1.2 pengujian tombol pilih level

Tombol pilih *level* akan muncul setelah pemain menekan tombol *start* pada tampilan utama *game* ini. Tombol pilih *level* memiliki efek membesar jika *mouse* *pointer* di arahkan keatas tombol ini. Tombol pilih *level* akan mengantarkan pemain ke tampilan *level* yang di pilih. Pada awal permainan

tombol yang aktif hanya tombol untuk level 1. Pemain harus menamatkan *level* satu terlebih dahulu jika ingin mengaktifkan tombol *level* 2.



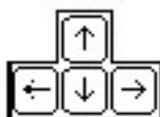
Gambar 0.17 Tombol Pilih Level

#### 4.3.2 Pengujian *Game Play*

Pengujian *game play* adalah pengujian yang khusus untuk menguji *game play* permainan ini. Mulai dari *character control*, hingga berbagai kondisi dalam *game* ini

##### 4.3.2.1 Karakter Control

Untuk mengontrol *character* dalam *game* ini, pemain bisa menggunakan tombol 'W' untuk maju, tombol 'S' untuk mundur, tombol 'A' untuk belok kiri, dan tombol 'D' untuk belok kanan. Pemain juga bisa menggunakan tombol arah untuk mengatur semua pergerakan itu. Untuk menembakan peluru, pemain bisa menggunakan tombol 'SPACE'



Gambar 0.18 Tombol Arah



**Gambar 0.19** Gambar *Character* Menembak

#### 4.3.2.2 Instruksi *Game*

Pengujian instruksi *game* adalah pengujian dimana pemain sedang berada dalam kondisi tidak menjalankan atau menjalankan instruksi *game*. Hal tersebut akan mengakibatkan kondisi kondisi sebagai tanda instruksi tersebut sudah di jalankan atau belum. Antara lain kondisi *character* ketika mendapatkan kristal, kondisi *character* ketika terkena musuh atau hambatan, kondisi jatuh, dan ketika kondisi pemain menemukan *portal* untuk menuju level berikutnya.

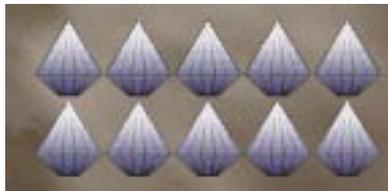
##### 4.3.2.2.2 Kondisi mendapatkan kristal

Kondisi ini adalah kondisi ketika *character* mendapatkan kristal. Kristal akan menghilang dan berubah menjadi partikel seperti cahaya yang beterbangan.



**Gambar 0.20** Kondisi Mendapatkan Kristal

Setelah mendapatkan kristal, tampil *GUI* kristal akan mengubah kondisinya yang menginformasikan bahwa pemain sudah mendapatkan kristalnya.



**Gambar 0.21** Kondisi kristal Normal

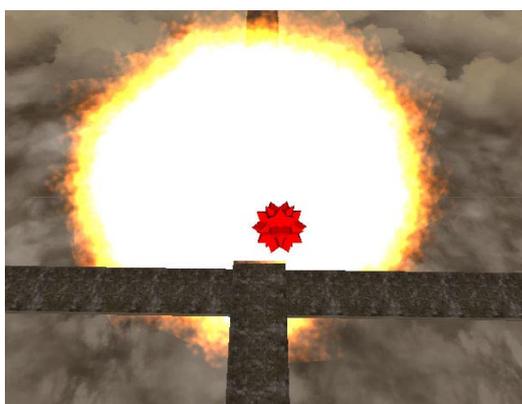


**Gambar 0.22** Kondisi Kristal *Get 1*

#### 4.3.2.2.3 Kondisi Terkena Hambatan

Kondisi ini terjadi ketika *character* tidak bisa menghindari dari hambatan yang terdapat dalam setiap levelnya. Seperti terkena ranjau, terkena musuh, jatuh dari arena, dan terkena moom yang jatuh secara acak dari atas arena permainan.

Ketika *character* terkena ranjau, *character* akan meledak dan akan kembali lagi ke posisi awal. Berikut gambar ketika *character* meledak



**Gambar 0.23** Kondisi Terkena Ranjau

Ketika karakter terkena ranjau maka ini akan mengurangi nyawa dari *character* itu. Kondisi awal karakter memiliki 3 nyawa.

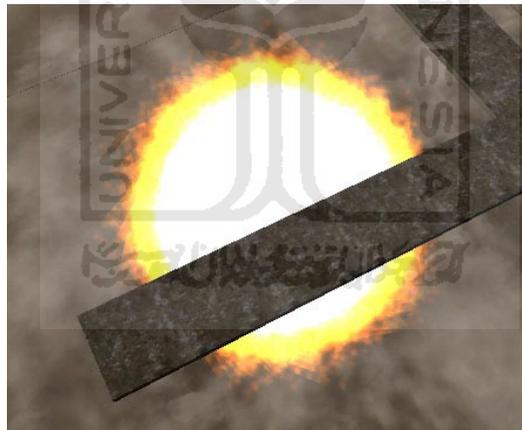


**Gambar 0.24** Kondisi Normal



**Gambar 0.25** Kondisi Berkurang

Kondisi yang sama akan terjadi juga apa bila *character* tidak bisa menjaga posisi di dalam jalan yang membuat *character* terjatuh kebawah, dan ketika *character* terkena peluru dari musuh.



**Gambar 0.26** Character Terjatuh

#### 4.3.2.2.4 Kondisi Menghancurkan Hambatan

Semua hambatan dari game ini bisa dihancurkan jika di tembak, kecuali hambatan *boom* yang akan terus jatuh secara acak. Gambar berikut adalah gambar ketika semua hambatan dalam *game* ini terkena peluru dari *character*



**Gambar 0.27** Ranjau Ketika Meledak



**Gambar 0.28** Musuh Meledak

#### **4.3.2.2.5 Kondisi Menemukan Finish**

Setelah semua kristal terkumpul semua, maka portal *finish* akan muncul secara otomatis. *Ending* dari setiap *level* adalah ketika pemain berada di atas portal tersebut. Dengan begitu akan mengantarkan pemain ke *level* berikutnya.



**Gambar 0.29** *Charackter Berada di Finish*

#### **4.4 Langkah Pembuatan**

Dalam pembuatan *game* ini menggunakan 3 *software*, yaitu 3Dmax, adobe illustrator dan unity. Yang dimana ketiga *software* tersebut mempunyai pembagian kerja masing-masing. 3d max digunakan untuk semua pemodelan dalam *game* ini. Illustrator digunakan untuk semua pemodelan 2 dimensi di dalam *game* ini. Sedangkan Unity digunakan untuk memberikan interaksi kepada setiap objek yang sudah dibuat oleh kedua *software* tersebut.

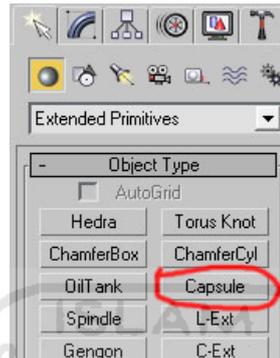
##### **4.4.1 Pemodelan Game Object**

Seperti yang disebutkan sebelumnya, semua *game* object 3d dimensi didalam *game* ini dibuat menggunakan *software* 3d max. Versi yang digunakan penulis adalah Autodesk 3Ds max 2009 yang dimana dalam vesri terbaru ini sudah memiliki beberapa kelebihan dari versi sebelumnya.

Dalam pemodelan suatu objek 3d sebaiknya sebelumnya dibuat terlebih dulu garis garis besar bentuk apa yang akan dibuat. Hal ini berguna untuk memudahkan dalam pembuatan model *game object* tersebut. Garis besar pemodelan setiap *game* object dalam *game* ini sudah di sampaikan penulis di dalam bab sebelumnya. Berikut langkah-langkah pemodelan *game* objek yang terdapat dalam *game* ini.

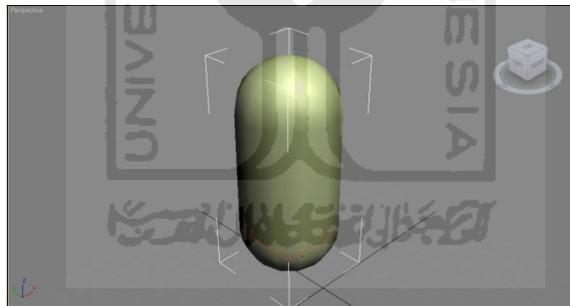
#### 4.4.1.1 Character

Langkah pertama untuk membuat *charater* yang sudah dirancang sebelumnya adalah membuat *capsule* terlebih dahulu. Membuat *capsule* dengan cara mengklik *toolbox capsule* yang ada di *create geometry* di kolom *extended primitives*.



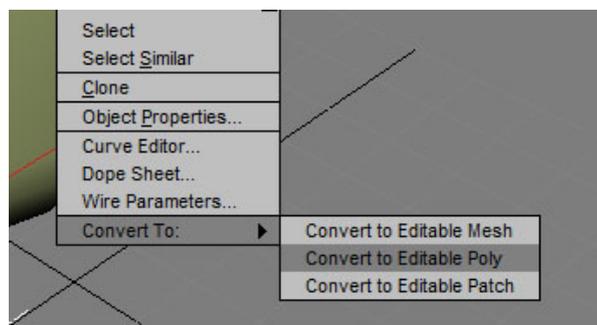
**Gambar 4.30** *Tool Box*

Klik dan geser di *workspace* untuk membuat *capsule*. maka jadilah seperti gambar berikut.



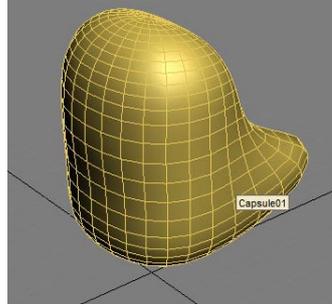
**Gambar 4.31** *Capsule*

Setelah *capsule* terbentuk, klik kanan pada *object* dan pilih *convert to editable poly* untuk melakukan pengeditan secara lebih bebas.



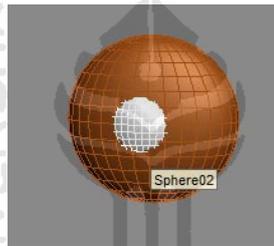
**Gambar 4.32** *editable poly*

Gunakan semua *tools* yang ada di *edit poly tools* sehingga objek terbentuk hingga sedemikian rupa



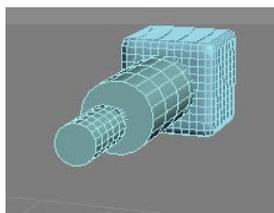
**Gambar 4.33** Object setelah dibentuk

Tambahkan mata pada *body character* yang sudah dibuat sebelumnya. Membuat mata cukup menggunakan dua *sphere* yang di pipihkan sehingga terbentuk sedemikian rupa



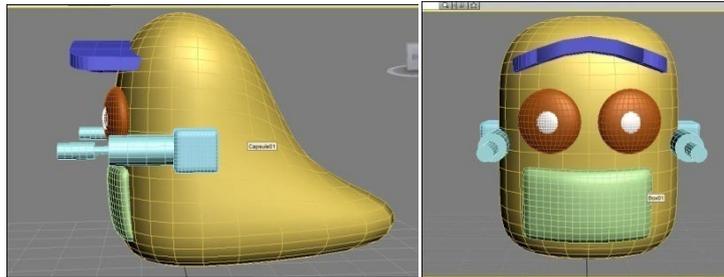
**Gambar 4.34** Mata

Untuk membentuk senjata yang pada character, gunakan object *square* dan *cylinder* yang digabungkan sehingga terbentuk sedemikian rupa



**Gambar 4.35** Senjata

Atur tata letak semua *object* tersebut sehingga terbentuk *character* secara sempurna.

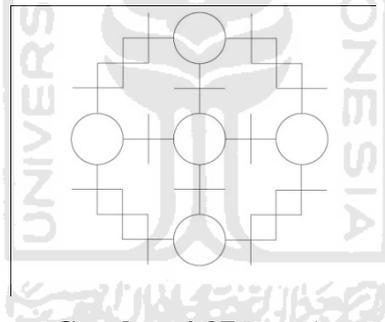


**Gambar 4.36** Bentuk Akhir

Langkah langkah dalam pembuatan character juga diterapkan dalam pemodelan game object ranjau, musuh, dan boom

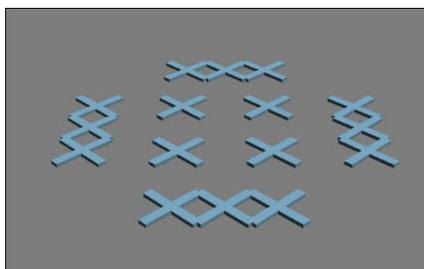
#### 4.4.1.2 Arena

Langkah pertama untuk membentuk arena adalah dengan membuat denahnya terlebih dahulu dengan menggunakan adobe illustrator. Gunakan *pen tool* dan bentuklah gambat seperti berikut.



**Gambar 4.37** Denah

Export gambar tersebut ke format DFX(*AutoCAD Interchange File*) dengan cara memilih *file menu*, dan *export*. file sudah di *export* tersebut di *import* ke 3d max untuk di jadikan bentuk 3dimensi.



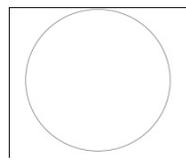
**Gambar 4.38** Denah Dalam Bentuk 3d

Pembuatan semua level yang ada di *game* ini menggunakan langkah yang sama.

#### 4.4.2 GUI

Semua GUI yang terdapat *game* ini dibuat dengan menggunakan adobe illustrator. Untuk membuat informasi nyawa *Character* kepada pemain dibutuhkan gui yang meberikan simbol yang mengartikan nyawa, semisal gambar hati. Berikut langkah langkah Membuat GUI

1. Buatlah sebuah lingkaran dengan mengunakan *ellipse tool* yang terdapat pada *toolbox*



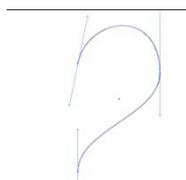
**Gambar 4.39** Lingkaran

2. Hapus *anchor point* sebelah kiri pada lingkaran sehingga terbentuk menjadi setengah lingkaran



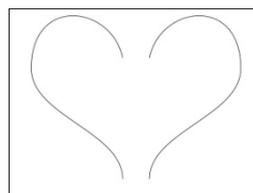
**Gambar 4.40** Setengah Lingkaran

3. Dengan *direct selection tool*, ubah semua shape yang hingga terbentuk seperti berikut



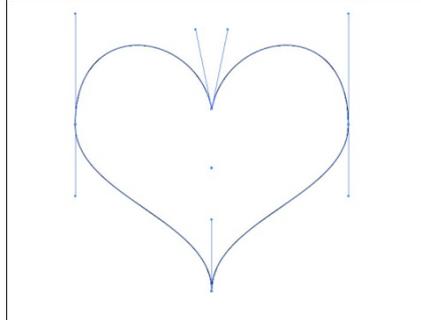
**Gambar 4.41** Setelah di Bentuk

4. *Copy* objek tersebut dan gunakan *reflect tool* untuk membentuk objek yang berlawanan dengan objek sebelumnya



**Gambar 4.42** Setelah di Cerminkan

5. Gabungkan kedua objek tersebut dengan menggunakan *direct selection tool* dan dan hubungkan semua *point* sehingga menjadi satu objek yang utuh



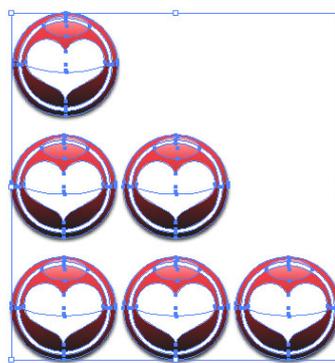
**Gambar 4.43** Setelah di Gabungkan

6. Tambah gambar lingkarang dan beri warna pada objek tersebut. Gunakan *gradient tool* untuk memberikan efek gradasi pada setiap warna yang akan di masukan ke objek tersebut. Hasil akhir dari GUI nyawa bisa dilihat digambar berikut.



**Gambar 4.44** Setelah diwarnai

7. Karena nyawa yang diberika kepada *character* ada 3, maka *copy* objek tersebut hingga terbentuk 3 objek yang sama dan *export* gambar terserbut untuk 3 kondisi. Kondisi nyawa penuh, kondisi nyawa tersisa 2, dan kondisi nyawa tersisa satu.



**Gambar 4.45** Nyawa Dalam 3 Kondisi

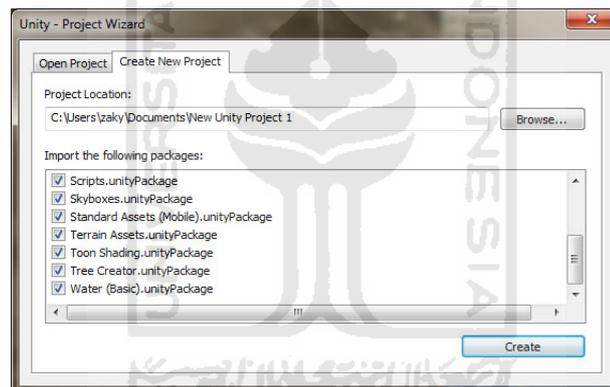
8. *Export* setiap kondisi dalam keadaan terpisah dengan format PNG

Langkah ini juga digunakan untuk pembuatan GUI yang lainnya, hanya saja model kreasi nya berbeda.

#### 4.4.3 Importing

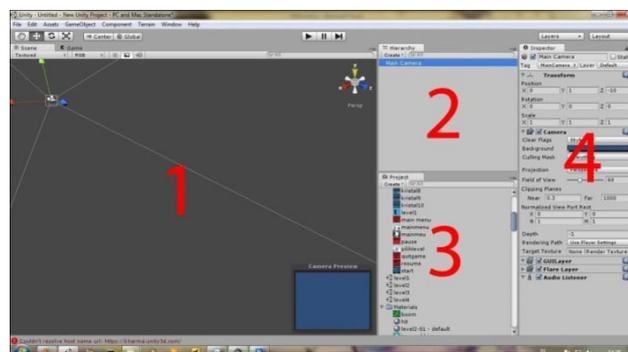
Pada bagian ini akan membahas bagaimana menggabungkan semua file 3d yan sudah dibuat agar bisa saling berintraksi dan menjadi satu kesatuan *game* menggunakan unity.

Jalankan unity dan buat project baru. Centang semua paket yang ada untuk memudahkan dalam pengerjaan project *game* tersebut. Kegunaan paket tersebut agar kita bisa menggunakan semua assets standar bawaan yang di tawarkan langsung dari *software* Unity.



**Gambar 4.46** Create New Project

setelah membuat projet yang baru, maka akan muncul tampilan *workspace* unity secara keseluruhan. Berikut penjelasan dasar dari *user interface* unity

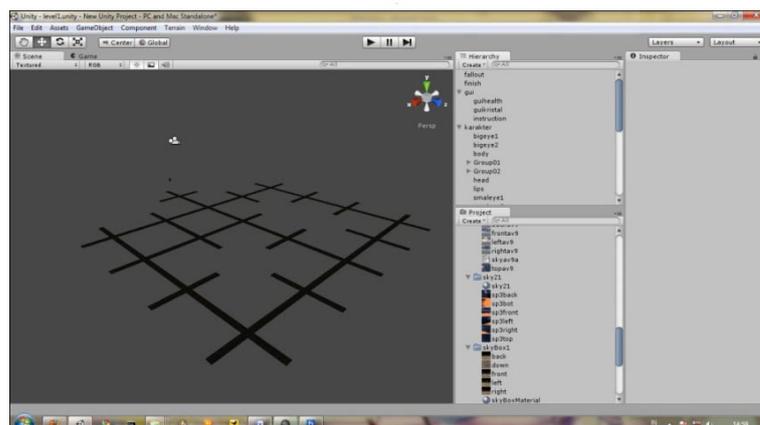


**Gambar 4.47** Unity Interface

1. Adalah bagian dimana kita membangun ruang 3 dimensi untuk game yang akan di rancang. Mulai dari *character*, latar permainan, hingga musik. Semuap peletakan akan di atur di bagian ini.
2. Adalah kolom yang menginformasikan apa saja *game object* yang ada di dalam kolom nomer satu. Hal ini berguna untuk mengkordinir object yang tidak sedikit.
3. Kolom ketika adalah implementasi folder project yang dibuat.
4. Berisi semua perintah yang terdapat di dalam *object* yang dipilih. Seperti perintah berupa *script*, animasi, dan *object propertis*.

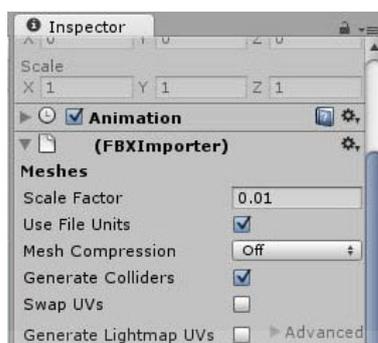
Masukan semua *file object* sudah dibuat sebelumnya kedalam *folder* dimana tempat *project unity* digunakan. Untuk mengetahui dimana *folder* tersebut. klik menu *assete -> show in explorer*. Dengan otomatis akan membuka jendela *explorer* yang langsung menuju *diretory unity project*. Masuk ke *folder assets* dan buat *folder* tersendiri untuk meletakan semua *object 3d* yang sudah dibuat. Dengan begitu, secara otomatis unity akan meingport langsung semua *file* yang telah dimasukan.

Untuk membuat arena permainan pada setiap *level*. Masukan *object arena* yang sudah dibuat kedalam kolom *scene*.



**Gambar 4.48** Memasukan arena permainan

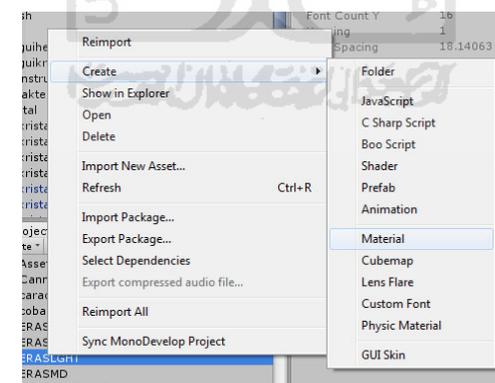
Agar semua objek 3d yang dimasukan menjadi objek yang padat, maka centang *generate collider* yang terdapat pada kolom *inspecture* objek. *Colidder* berfungsi agar objek menjadi benda yang tidak dapat ditembus oleh karakter.



**Gambar 4.49** *Generate Collider*

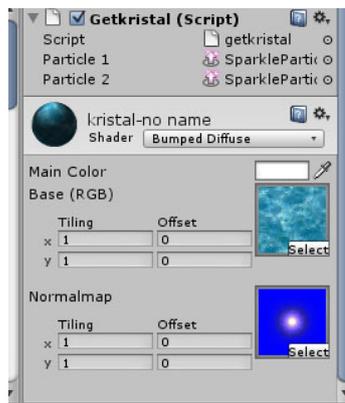
Pemberian material pada setiap objek tidak bisa langsung mengikuti apa yang sudah dibuat menggunakan 3d max. Diperlukan pemberian material ulang agar material objek sesuai dengan apa yang diinginkan.

klik kana pada kolom *asesst* dan pilih *new* material untuk membuat material baru



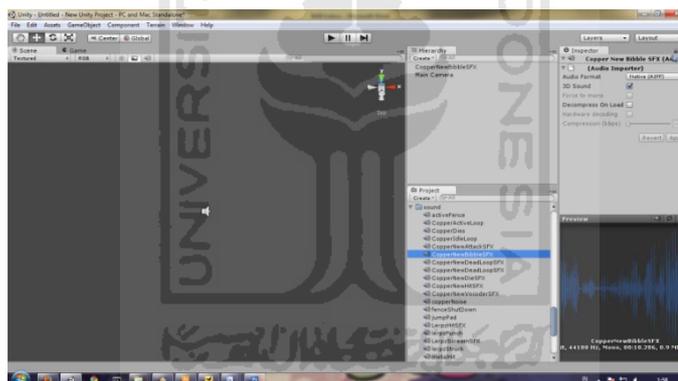
**Gambar 4.50** *Create New Material*

Aktifkan objek yang ingin diberi material. Geser material ke kolom *inspecture* object tersebut. Secara otomatis objek tersebut akan terhubung dengan material tersebut. Perubahan apa pun yang terjadi di material tersebut dengan secara otomatis akan mengubah material obejct tersebut.



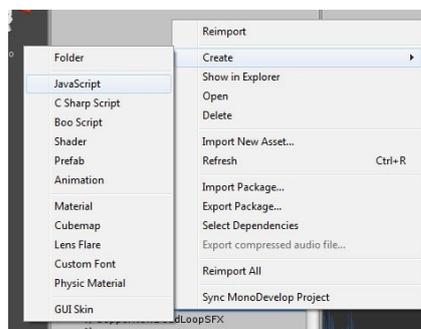
**Gambar 4.51** PemberianMaterial

Untuk memasukan audio kedalam arena permainan, cukup dengan cara menggeser file audio yang ada didalam kolom asset kedalam kolom scene. Atur tata letak sesuai dengan keinginan.



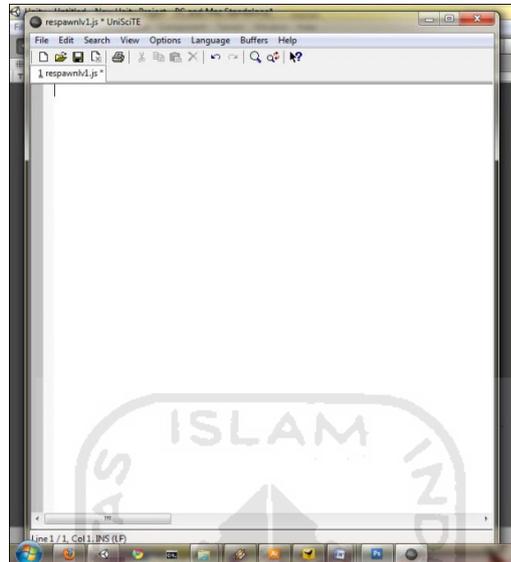
**Gambar 4.52** Memasukan Audio

Untuk membuat script, Klik kanan pada kolom *asesst* dan pilih *create java script*.



**Gambar 5.53** Membuat Java Script

Beri nama pada *java script* yang baru dibuat. klik 2 kali untuk melakukan pengeditan *java script*.

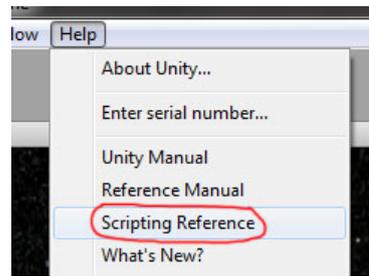


**Gambar 5.54** *Window Editor Script*

Untuk memasukan semua script yang sudah dibuat. aktifkan objek yang ingin diberi script tersebut. Geser script dari kolom project ke kolom *inspecture* objek tersebut. Dengan secara otomatis, objek tersebut akan melakukan semua proses yang sudah dinstruksikan melalui script yang dibuat.

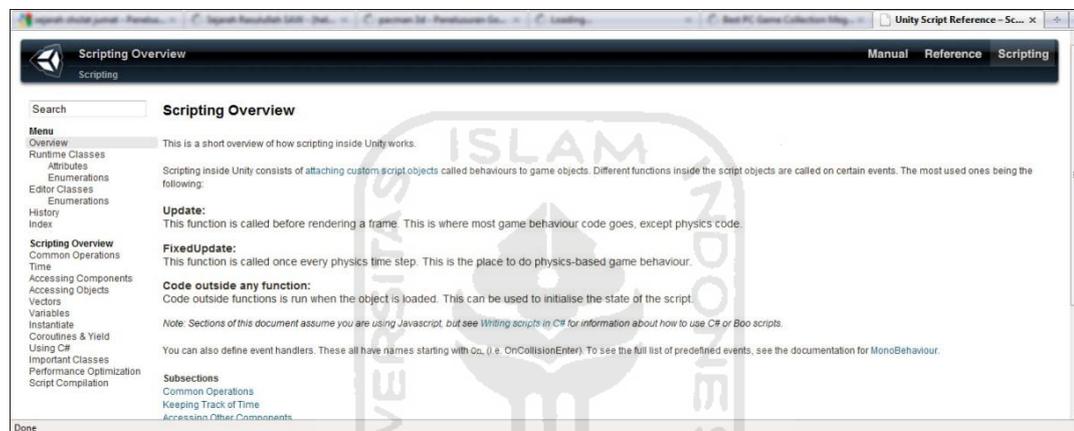
#### 4.4.4 Scripting

*Scripting* digunakan untuk membuat semua objek yang sudah dibuat sebelumnya agar bisa saling berintraksi dengan *user*. Sehingga menjadi suatu kesatuan yang bisa dimainkan. dalam membuat script di unity ini menggunakan aturan tata bahasa yang digunakan dalam bahasa pemograman java. untuk melihat *syntak* dan *variabel* apa saja yang terintergrasi di unity dengan cara klik menu *help* kemudian *script refrence*.



Gambar 5.55 Scripting Reference

langkah tersebut digunakan untuk membuka jendela *refrensi script* yang terintegrasi dengan unity



Gambar 5.56 Scripting Reference window

Berikut contoh salah satu script yang digunakan dalam game ini. Script ini digunakan untuk pergerakan karakter.

```

var speed : float = 3.0

var rotateSpeed : float = 3.0

var gravity : float = 20.0; private var moveDirection :
Vector3 = Vector3.zero;

function Update () {

var controller : CharacterController =
GetComponent(CharacterController);

// berputar

```

```

transform.Rotate(0, Input.GetAxis ("Horizontal") *
rotateSpeed, 0);

// maju/mundur

var forward : Vector3 =
transform.TransformDirection(Vector3.forward);

var curSpeed : float = speed * Input.GetAxis ("Vertical");

controller.SimpleMove(forward * curSpeed);

// grafitasi

moveDirection.y -= gravity * Time.deltaTime;
}

```

## 4.5 Evaluasi

Evaluasi dilakukan untuk mencari kekurangan dan kelebihan dalam game ini. Evaluasi dilakukan dengan cara melakukan perbandingan dengan *game* yang dijadikan referensi utama dalam pembuatan *game* ini. Referensi utama *game* ini adalah *game* Pacman Maniac yang sama sama memiliki visual 3d. *Game* ini di beri nama labirin langit. Berikut hasil dari evaluasi *game* ini.

### 4.5.1 Kekurangan

- a) Tidak memiliki persyaratan untuk menyerang.

Game ini tidak memiliki persyaratan tertentu untuk melakukan penyerangan terhadap musuh. Tidak seperti game pacman yang harus mengambil satu item berupa buah yang bisa membuat karakter dari game pacman tersebut bisa memakan semua monster yang menghalangi misi dari karakter tersebut. Dalam *game* ini, karakter bisa menembak tanpa sebuah persyaratan. Hal ini membuat permainan menjadi kurang seru, karena pemain bisa langsung menembak tanpa menghadapi kesulitan terlebih dahulu.

b) Tidak memiliki radar

Tidak memiliki radar membuat permainan ini terlalu susah dimainkan. Radar disini berguna sebagai perkiraan dimana letak objek-objek yang harus di ambil.

#### 4.5.2 Kelebihan

a) Hanya mengambil 10 kristal.

Tidak seperti *game* pacman yang harus mengambil banyak item sebagai penunjuk jalan. *Game* Labirin Langit ini hanya mengambil 10 kristal yang letaknya tidak beraturan. Ini membuat pemain tidak bisa menebak di mana letak kerystal berikutnya dan ini membuat *game* lebih sulit dimainkan dari *game* pacman.

b) Lebih menantang

Dengan mengabil latar diluar angkasa, dan dengan adanya aturan pemain yang harus menjaga karater tetap berada di arena permainan. Permainan ini menjadi lebih menantang. Karena pemain harus mengatur pegerakan karater agar tidak jatuh dari arena. Hal ini tidak begitu mudah, karena hambatan yang selalu bergerak mendekati charater. Jadi pemain memiliki dua tantangan, menghidar dari hambatan, dan menjaga charater tetap berada di arena permainan.

c) *Game* lebih membuat pemain berpikir

*Game* pacman 3d memiliki sudut pandang dari atas karater dan menampilkan seluruh arena permaian. Hal ini tentu membuat pemain sudah bisa melihat dimana monter berada dan sudah bisa memikirkan langkah langkah untuk menghindari monster tersebut. Berbeda dengan *game* labirin langit. *game* ini hanya menampilkan sebagian kecil arena yang terlihat di dekat karakter *game* ini berada. Ini tentu membuat pemain tidak bisa melihat di mana hambatan dari *game* ini berada, dan dimana kristal yang harus di kumpulkan berada. Hal ini tentu membuat *game* ini lebih sulit dimainkan dari *game* pacman 3d tersebut.

## BAB IV

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah melalui tahapan pengujian dan hasil, maka penulis dapat mengambil beberapa kesimpulan sebagai berikut :

- a. Labirin adalah bentuk arena yang cocok untuk game sederhana tapi cukup menantang. Ini dikarenakan labirin adalah suatu teka teki yang harus dipecahkan dengan logika.
- b. *Game* ini hanya mengutamakan *game play*.
- c. Unity masih memiliki cacat dalam compile *game* ke dalam format .EXE yang mengakibatkan suara terdengar lebih kecil.

#### 5.2 Saran

Setelah melihat hasil kesimpulan diatas, berikut ini adalah saran yang mungkin berguna dan dapat dipakai pengembang aplikasi ini, yaitu :

- a. 3d max adalah *software* berbayar, begitu juga dengan adobe illustrator. Hanya unity *software* yang gratis dalam pembuatan unity ini. Untuk kedepannya mungkin bisa menggunakan semua *software* gratis seperti blander atau sketchup untuk pemodelan 3dnya, dan inkscape untuk tapilan 2d nya
- b. Diberikan informasi radar yang bisa membuat pemain memperkirakan dimana letak finish dari game ini. Radar tidak berisi minimap, tapi hanya berisi jarak antara pemain dan tujuan akhir.
- c. Memberikan animasi terhadap *character* utama ketika menembak dan berjalan, sehingga tidak seperti benda mati yang bergerak
- d. Memberi persyaratan untuk bisa menembak, sehingga pemain tidak langsung bisa menembak sebelum sebuah persyaratan itu di penuhi. Ini akan menambah tantangan dalam game ini. Contohnya seperti *game* Mario Bros yang harus mengambil jamur terlebih dahulu baru bisa menembak musuh

**DAFTAR PUSTAKA**

- [FEI05] Feil,John.Beginning Game level design.  
<http://courseptr.com/resource.htm>  
Diakses terakhir 12 agustus 2010
- [HEN05] Henry,Samuel.Panduan Praktis Membuat Game  
3D.Yogyakarta:Graham Ilmu,2005.
- [TGO] <http://www.thegamecreators.com/>  
Diakses terakhir 20 oktober 2010
- [ROG03] Roger E,Pedersen. Game Design Foundations,  
texas,2003.
- [ROB03] Roberts, Steve.Animasi karakter 3D:Bayumedia,2006.
- [WAH03] <http://wahw33d.blogspot.com/2010/02/rahasia-tentang-labirin.html#ixzz1Clp4WCdI>.  
Diakses terkahir tanggal 2 februari

