

SKRIPSI
PENERAPAN ALGORITMA RANDOM WALK DALAM MEMBUAT
LEVEL PADA GAME RPG SECARA PROSEDURAL



DISUSUN OLEH :

BAYU BORNEO

DBC 115 041

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA

2022

**PENERAPAN ALGORITMA RANDOM WALK DALAM MEMBUAT LEVEL PADA
GAME RPG SECARA PROSEDURAL**

SKRIPSI

Sebagai salah satu syarat untuk menyelesaikan Program Strata-I pada Jurusan Teknik
Informatika Fakultas Teknik Universitas Palangka Raya

Oleh

BAYU BORNEO
DBC 115 041

Telah dipertahankan didepan tim penguji, pada :

Hari/Tanggal : Selasa, 11 Juli 2022

Waktu : 15.30-16.30 WIB

1. **DEDDY RONALDO, S.T., M.T**
NIP. 19801226 200812 1 002 (Ketua)
2. **AGUS SEHATMAN SARAGIH, S.T., MENG.**
NIP. 19850818 201212 1 003 (Anggota)
3. **ARIESTA LESTARI, S. KOM., M.CS., PHD**
NIP. 19800322 200501 2 004 (Anggota)

Mengetahui :

Fakultas Teknik
Universitas Palangka Raya
Dekan,

Ir. WALUYO NUSWANTORO, M.T.
NIP. 19681119 199302 1 001

Jurusan / Program Studi Teknik Informatika
Fakultas Teknik Universitas Palangka Raya
Ketua Jurusan,


ABERTUN SAGIT SAHAY, S.T., M.Eng
NIP. 19751212 200312 1 002

**PENERAPAN ALGORITMA RANDOM WALK DALAM MEMBUAT
LEVEL PADA GAME RPG SECARA PROSEDURAL**

SKRIPSI


Sebagai salah satu syarat menyelesaikan Program Strata - I
pada Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya

OLEH :


BAYU BORNEO
NIM. DBC 115 041

Disetujui untuk diajukan dalam Seminar Akhir Skripsi,

Pembimbing I


DEDDY RONALDO, S.T., M.T.
NIP. 198012262008121002

Pembimbing II


AGUS SEHATMAN SARAGIH, S.T., M.ENG.
NIP. 198508182012121003

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2022**

PERNYATAAN

Dengan ini saya menyatakan dengan sebenar - benarnya bahwa dalam Skripsi ini tidak terdapat karya ilmiah yang pernah diajukan untuk memperoleh gelar kesarjanaan disuatu Perguruan Tinggi, serta tidak terdapat karya ilmiah atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam Skripsi ini dan disebutkan dalam Tinjauan Pustaka.

Palangka Raya, 19 Juli 2022



Bayu Borneo
DBC 115 041

RIWAYAT PENYUSUN

Data Diri

Nama : BAYU BORNEO
NIM : DBC 115 041
Fakultas : Teknik
Jurusan/Program Studi : Teknik Informatika
Jenjang : Strata 1 (S-1)
Jenis Kelamin : Laki - Laki
Tempat, Tanggal Lahir : Karang Sari, 15 Mei 1997
Agama : Islam
Status Dalam Keluarga : Anak
Anak Ke : Satu (1)
Alamat : Jl. Ulin no 26
No Telepon/HP : 0858 4501 0726



Data Orang Tua

Nama Ayah : Jamroji
Pekerjaan Ayah : Petani
Nama Ibu : Suwarsih
Pekerjaan Ibu : Pedagang
Alamat Orang Tua : Karang sari, Parenggean
No.HP Orang Tua : 0858 2086 1905

Riwayat Pendidikan

SD : SDN 1 Karang Sari (Tahun Lulus 2009)
SMP : SMPN 3 Parenggean (Tahun Lulus 2012)
SMA : SMAN 3 Sampit (Tahun Lulus 2015)

HALAMAN PERSEMBAHAN

Dengan hati yang penuh dengan sukacita,
saya panjatkan segala puji dan syukur dengan setulus hati saya kepada:

ALLAH SWT

Kupersembahkan karya Skripsi ini untuk :

Ibu dan Bapak saya yang tercinta

Yang selalu mendoakan dan memberikan dukungan sepenuhnya

Yang tidak pernah habis-habisnya memberikan semangat dan motivasi

Yang selalu memberikan pengertian atas segala permasalahan saya

Dan tidak pernah gagal untuk memberikan motivasi dan nasihat untuk tetap maju

Sahabat-sahabat dan keluarga besar di Fakultas Teknik

Almamater tercinta Universitas Palangka Raya

Yang selalu membantu dan memberikan pengalaman yang berharga

Selama perjalanan menempuh studi saya.

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Allah Swt. atas ridanya saya dapat menyelesaikan penyusunan skripsi ini. Adapun judul skripsi yang saya ajukan adalah “*Penerapan Algoritma Random Walk Dalam Membuat Level Pada Game Rpg Secara Prosedural*”

Skripsi ini diajukan untuk memenuhi syarat kelulusan mata kuliah Skripsi di Fakultas Jurusan/Prodi Teknik Informatika, Fakultas Teknik, pada Universitas Palangka Raya. Tidak dapat disangkal bahwa butuh usaha yang keras dalam penyelesaian pengerjaan skripsi ini. Namun, karya ini tidak akan selesai tanpa orang-orang tercinta di sekeliling saya yang mendukung dan membantu. Terima kasih saya sampaikan kepada:

1. Bapak Abertun Sagit Sahay, ST., M.Eng. selaku Ketua jurusan/Program Studi Teknik Informatika Fakultas Teknik pada Universitas Palangka Raya
2. Bapak Deddy Ronaldo, S.T., M.T dan Bapak Agus Sehatman Saragih, S.T., M.Eng. selaku Dosen Pembimbing I dan Dosen Pembimbing II skripsi, yang senan tiasa membantu dan selalu memberikan kesempatan kepada penulis untuk memperbaiki diri.
3. Ibu Ariesta Lestari, S. Kom., M.Cs., Ph.D yang menguji dengan sabar dan memberi kesempatan pada penulis untuk menyelesaikan skripsi.
4. Rekan – rekan Mahasiswa angkatan 2015 yang selalu mevotesi dan memberi semangat pada saat penulis kehilangan motivasi dalam proses pembuatan skripsi.

Semoga segala kebaikan dan pertolongan semuanya mendapat berkah dari Allah Swt. dan akhirnya saya menyadari bahwa skripsi ini masih jauh dari kata sempurna, karena keterbatasan ilmu yang saya miliki. Untuk itu saya dengan kerendahan hati mengharapkan saran dan kritik yang sifatnya membangun dari semua pihak demi membangun laporan penelitian ini.

Palangka Raya, 20 Juli 2022



BAYU BORNEO

DBC 115 041

PENERAPAN ALGORITMA RANDOM WALK DALAM MEMBUAT LEVEL PADA GAME RPG SECARA PROSEDURAL

Bayu Borneo (DBC 115 025)

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangka Raya
Universitas Palangka Raya Tunjang Nyaho, Jl. Yos Sudarso Palangka Raya 73113.

Email : bayuborneo97@gmail.com

ABSTRAK

Videogame adalah permainan dimana pemain berinteraksi menggunakan sebuah perangkat pengendali untuk menghasilkan umpan balik tampilan video. Dengan semakin meningkatnya tenaga komputasi yang juga berpengaruh dengan meningkatnya kompleksitas *videogame*, developer dapat mengimplementasikan beberapa teknik koding dalam mengembangkan suatu *videogame*. Salah satunya berupa *Procedural Content Generation*, dimana sebuah konten pada *videogame* digenerasikan menggunakan komputer berdasarkan parameter awal yang diberikan. Penerapan konsep ini telah dilihat di berbagai genre permainan, seperti *Role-Playing Game* (RPG) dimana pemain memerankan sebuah karakter dalam menjelajahi dunia yang dikembangkan dalam suatu naratif. Salah satu algoritma kandidat dalam penerapan PCG adalah *random walk*, dikarenakan kemampuan algoritma untuk membuat level dengan menyimpan lokasi mana saja yang dilewati *random walk*

Subjek penelitian berupa implementasi *random walk* pada proses pembuatan level permainan pada game RPG. Pembuatan level dimulai dengan membuat walker pada tengah level, kemudian walker akan mengambil arah atas, kanan, bawah atau kiri secara random. Setelah *walker* memilih arah *walker* akan melangkah dan setiap langkah yang dilakukan oleh walker akan disimpan dalam sebuah array. Setiap 5 langkah walker akan mengambil arah baru dan membuat sebuah ruangan. Kemudian setelah *walker* selesai melakukan langkah, letak pemain, objektif dan musuh akan di kalkulasikan untuk membuat level. Tahap perancangan game dilaksanakan dengan menggunakan metodologi *Luther* sebagai acuan. Analisis dan desain dilakukan dengan menjabarkan keperluan sistem dan pemodelan diagram *UML*. Pengkodean dan implementasi dilakukan menggunakan software Godot Engine, sementara pengujian dilakukan menggunakan metode *sampling*, dan pengujian oleh relawan.

Berdasarkan *sampling* yang dilakukan dari 9 generasi, semua level memiliki objektif yang bisa dicapai dan tiap level bertambah sulit dengan bertambahnya musuh. Pengujian oleh relawan menunjukkan bahwa semua level yang digenerasi menggunakan algoritma ini dapat diselesaikan oleh relawan tanpa ada masalah.

Kata-Kata Kunci : *Game RPG, Procedural Content Generation, Algoritma random walk, Metodologi luther*

IMPLEMENT OF RANDOM WALK ALGORITHM IN GENERATING LEVEL IN RPG GAME PROCEDURALLY

Bayu Borneo (DBC 115 025)

Department of Informatics, Faculty of Engineering, University of Palangka Raya,
Campus of UPR Tunjung Nyaho Yos Sudarso Street Palangka Raya 73112

Email : bayuborneo97@gmail.com

ABSTRAK

Videogames are games in which players interact using a control device to produce video display feedback. By using a workforce that is also influential with videogame comparisons, developers can implement several coding techniques in developing a videogame. One of them is in the form of Procedural Content Creation, where a videogame content is generated using a computer based on the initial parameters given. The application of this concept has been seen in various game genres, such as Role-Playing Game (RPG) where a character plays in exploring a world developed in a narrative. One of the candidate algorithms in the application of PCG is a random walk, due to the ability of the algorithm to create levels by storing any locations that the random walk passes through

The research subject is the implementation of random walk in the process of making game levels in RPG games. The creation of the level begins by making a walker at the middle level, then the walker will take up, right, down, or left directions at random. After the walker chooses the direction the walker will step and every step made by the walker will be stored in an array. Every 5 steps the walker will take a new direction and create a room. Then after the walker finishes making a move, the player's location, objectives and enemies will be calculated to create a level. The game design stage is carried out using the Luther methodology as a reference. Analysis and design is done by describing the system requirements and modeling UML diagrams. The coding and implementation was done using Godot Engine software, while testing was done using the sampling method, and testing was done by volunteers.

Based on sampling done from 9 generations, all levels have achievable objects and each level gets more difficult as enemies increase. Tests by volunteers show that all levels generated using this algorithm can be completed by volunteers without any problems.

Kata-Kata Kunci : *RPG game, Procedural Content Generation, random walk algorithm, luther Methodology*

DAFTAR ISI

COVER.....	i
LEMBAR PENGESAHAN.....	ii
LEMBAR PERSETUJUAN.....	iii
LEMBAR PERNYATAAN.....	iv
RIWAYAT PENYUSUN.....	v
HALAMAN PERSEMBAHAN.....	vi
KATA PENGANTAR.....	vii
ABSTRAK.....	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan	3
1.5. Manfaat	3
1.6. SISTEMATIKA PENULISAN	3
BAB II LANDASAN TEORI.....	6
2.1. TINJAU PUSTAKA	6
2.1.1. A Short Introduction To Procedural Content Generation Algorithm For Video <i>Games</i> (Nicolas A. Barriga, 2018).....	6

2.1.2.	An Analog History Of Procedural Content Generation (Gillian Smith, 2015)	6
2.1.3.	Procedural Generation : An Algorithmic Analysis Of Video <i>Game</i> Design And <i>Level</i> Creation (Logan Bond, 2017).....	7
2.1.4.	A Video <i>Game</i> Level Analysis Model Proposal (Samuel Almeida, Ana Veloso, Óscar Mealha, Licínio Roque, 2012).....	7
2.1.5.	Exit Probability and First Passage Time of a Lazy Pearson Walker: Scaling Behaviour (Muktish Acharyya, 2015).....	8
2.2.	TEORI PENDUKUNG	8
2.2.1.	Level	8
2.2.2.	Role Playing <i>Game</i> (RPG).....	8
2.2.3.	Survival Horror	9
2.2.4.	Random walk.....	10
2.2.5.	Aseprite	11
2.2.6.	Godot Engine.....	12
2.2.7.	UMLet.....	12
2.2.8.	Bahasa Pemrograman GDscript	13
2.2.9.	Flowchart	13
2.2.10.	Unified Modeling Language (UML)	14
BAB III METODE PENELITIAN		21
3.1.	Metodologi Pengembangan Perangkat Lunak.....	21
3.2.	Concept	22
3.2.1.	Deskripsi Project	22
3.3.	Design	23
3.3.1.	Gameplay	23
3.3.2.	Flowchart	25
3.3.3.	UML.....	26
3.3.4.	Desain Algoritma.....	35

3.3.5.	Desain Antarmuka	35
3.3.6.	Desain Level.....	38
3.3.7.	Material Collecting	38
BAB IV HASIL DAN PEMBAHASAN.....		43
4.1.	<i>Assembly</i>	43
4.1.1.	Scene Main Menu	43
4.1.2.	Scene Opening.....	43
4.1.3.	Scene level tutorial	44
4.1.4.	Scene World	45
4.1.5.	Scene Finale	46
4.1.6.	Menu Mati.....	47
4.1.7.	Scene Tamat	47
4.2.	Pengujian Aplikasi.....	48
4.2.1.	Pengujian <i>Game</i>	48
4.2.2.	Pengujian Algoritma.....	54
4.3.	Distribusi.....	63
BAB V KESIMPULAN DAN SARAN		64
DAFTAR PUSTAKA.....		65

DAFTAR TABEL

Tabel 2.1 Simbol-Simbol Flowchart.....	13
Tabel 2.2 Simbol-Simbol Use Case Diagram	16
Tabel 2.3 Bagian Suatu Kelas (Dennis, Wixom & Roth, 2012)	17
Tabel 2.4 Simbol-Simbol Activity Diagram	20
Tabel 3.1 Deskripsi Project	22
Tabel 3.2 Tombol keyboard	24
Tabel 3.3 Tabel Definisi Use Case	27
Tabel 3.4 Tabel Definisi Aktor.....	28
Tabel 3.5 Asset	39
Tabel 4.1 Perbandingan Parameter	46
Tabel 4.2 Level 1 Sample 1	54
Tabel 4.3 Level 1 Sample 2.....	55
Tabel 4.4 Level 1 Sample 3.....	56
Tabel 4.5 Level 2 Sample 1	57
Tabel 4.6 Level 2 Sample 2.....	58
Tabel 4.7 Level 2 Sample 3.....	59
Tabel 4.8 Level 3 Sample 1	60
Tabel 4.9 Level 3 Sample 2.....	61
Tabel 4.10 Level 3 Sample 3.....	62

DAFTAR GAMBAR

Gambar 2.1 Dark Wood oleh Acid Wizard Studio	9
Gambar 2.2 1D integer	10
Gambar 2.4 Interface Godot	12
Gambar 2.5 Interface UMLet	13
Gambar 2.4 Simbol-Simbol Relasi Antar Kelas	19
Gambar 3.1 Metodologi Luther (Sutopo 2003)	21
Gambar 3.2 Flowchart <i>Game</i>	25
Gambar 3.3 Use Case Diagram	27
Gambar 3.4 Activity Diagram	29
Gambar 3.5 Scene dan node	30
Gambar 3.6 Class Diagram Main Menu	31
Gambar 3.7 Class Diagram Lobby(tutorial)	32
Gambar 3.8 Class Diagram World(procedural level)	33
Gambar 3.9 Class Diagram Finale	34
Gambar 3.10 Class Diagram Tamat	34
Gambar 3.11 Antarmuka HUD	36
Gambar 3.12 Antarmuka Menu Utama	37
Gambar 3.13 Antarmuka Pesan Mati	37
Gambar 3.14 Antarmuka Akhir <i>Game</i>	38
Gambar 3.15 Antarmuka Dialog	38
Gambar 4.1 Scene Main Menu	43
Gambar 4.2 Scene Opening	44
Gambar 4.3 Tutorial	44
Gambar 4.4 Level tutorial	44
Gambar 4.3 Scene world level 1 (Kiri), Level 2 (Kanan), dan Level 3 (Bawah) ..	45
Gambar 4.4 lawang	45
Gambar 4.5 Scene Finale	47
Gambar 4.6 Menu Mati	47
Gambar 4.7 Scene Tamat	48
Gambar 4.8 Diagram Pemahaman Cara Bermain	49
Gambar 4.9 Diagram Kejelasan Cerita	49

Gambar 4.10 Diagram Kejelasan Cerita Game	50
Gambar 4.11 Diagram Tingkat Kesusahan	50
Gambar 4.12 Diagram Tantangan Level	51
Gambar 4.13 Diagram Kualitas Grafis dan Style Pada Game	51
Gambar 4.14 Diagram Kenyamanan Kontrol Character.....	52
Gambar 4.15 Diagram Kepuasan <i>Keybind</i>	52
Gambar 4.16 Diagram Penyelesaian Level	53
Gambar 4.17 Diagram Peletakan Musuh	53
Gambar 4.18 Level 1 Sample 1	54
Gambar 4.19 Level 1 Sample 2	55
Gambar 4.20 Level 1 Sample 3	56
Gambar 4.21 Level 2 Sample 1	57
Gambar 4.22 Level 2 Sample 2	58
Gambar 4.23 Level 2 Sample 3	59
Gambar 4.24 Level 3 Sample 1	60
Gambar 4.25 Level 3 Sample 2	61
Gambar 4.26 Level 3 Sample 3	62

BAB I

PENDAHULUAN

1.1.Latar Belakang

Video Game merupakan salah satu medium hiburan yang perkembangannya dapat dianggap berpegangan tangan dengan perkembangan teknologi, khususnya dalam teknologi komputasi. Dengan semakin meningkatnya tenaga komputasi dalam sebuah komputer, kompleksitas dari *video game* semakin juga meningkat, sehingga melahirkan beragam jenis *game* dengan yang sekarang dapat dikategorikan dalam beberapa genre berdasarkan karakteristik *game* tertentu. Salah satunya berupa genre RPG, dimana pemain mengendalikan sebuah karakter dalam sebuah setting fiktif.

Dengan tenaga komputasi yang semakin meningkat, *developer game* juga diberikan suatu kebebasan untuk mengimplementasikan beragam teknik koding. Salah satu teknik ini berupa implementasi *procedural content generation*, dimana konten spesifik dalam *game* akan dihasilkan secara otomatis berdasarkan parameter tertentu yang ditentukan oleh *developer*, biasanya dalam bentuk *level game* yang berubah setiap kali pemain memulai *game* baru. Jika dilakukan dengan baik, hal ini dapat meningkatkan nilai *replayability* dari sebuah *game*, sementara mengurangi beban di tangan *developer* dalam membentuk konten *game*.

Berdasarkan paparan diatas, akan dibuat sebuah konsep *game RPG* dengan *procedural content generation* yang diimplementasikan menggunakan algoritma *Random walk* dengan tujuan untuk menjelajahi kelayakan algoritma *Random walk* dalam menghasilkan konten *level* prosedural. Berdasarkan permasalahan tersebut maka diangkat sebuah materi Tugas Akhir dengan judul : **“Penerapan *Random walk* dalam Membuat *Level Game RPG* Secara Prosedural”**, dengan judul *game* berupa **“Lawang”**.

1.2.Rumusan Masalah

Adapun rumusan masalah yang dapat ditarik berdasarkan latar belakang yang telah diuraikan adalah bagaimana cara menerapkan algoritma *random walk* sehingga dapat menghasilkan *level* sebuah *game* yang fungsional dan dapat dimainkan.

1.3.Batasan Masalah

Dalam pembuatan *game* “Lawang” akan memiliki batasan masalah sebagai berikut :

1. *Game* akan didesain untuk dimainkan secara *single pemain*.
2. Pemain ditujukan untuk remaja
3. Konsep *game* yang akan dibuat dalam bentuk *game* 2D dengan tipe *game RPG* dengan gaya sudut pandangan *Top Down/Bird’s Eye View*.
4. *Game* akan dibuat menggunakan Godot Engine dengan bahasa pemrograman GDscript, sementara aset-aset 2D yang digunakan akan dibuat menggunakan perangkat lunak Acesprite.
5. *Game* akan terdiri atas 1 *Dungeon*.
6. *Dungeon* terdiri atas 5 level, dimana level pertama adalah level *tutorial*, 3 tingkat pertama akan di-generate secara prosedural oleh komputer dan 1 tingkat terakhir berupa ruangan akhir yang tidak di-generate secara prosedural.
7. *Level* yang di-generate dengan algoritma *random walk* akan memiliki bentuk dan tampilan seperti ruangan bawah tanah.
8. Setiap tingkat pada *dungeon* terdiri atas beberapa musuh yang jumlahnya telah ditentukan.
9. Untuk dapat melanjutkan ke tingkat selanjutnya, pemain harus Menemukan Lawang.
10. Setelah menempuh 3 tingkat pertama, pemain akan memasuki *level* akhir, dimana pemain akan memenangkan *game* jika berhasil mengalahkan bos di *level* ini

1.4. Tujuan

Tujuan dari pembuatan *game* “Lawang” ini adalah untuk menjelajahi penggunaan algoritma *Random walk* dalam menghasilkan sebuah *level game* RPG.

1.5. Manfaat

Adapun manfaat yang diberikan dalam pembuatan *game* “Lawang” ini adalah sebagai berikut:

1. Memberikan sebuah implementasi untuk pembuatan *level game* RPG menggunakan algoritma *Random walk*.
2. Mengurangi beban pembuatan *level* dari tangan *developer*, dengan mengotomatisasi proses dengan bantuan algoritma.
3. Sebagai bahan pembelajaran awal untuk mendalami penerapan konsep algoritma *procedural content generation* dalam media video *game*.

1.6. SISTEMATIKA PENULISAN

Penulisan laporan ini akan disusun dalam enam bab utama yang terdiri atas Pendahuluan, Landasan Teori, Metode Penelitian, Hasil dan Pembahasan, dan Kesimpulan dan Saran. Berikut adalah susunan sistematika penulisan laporan:

1. BAB I PENDAHULUAN

Bab I Pendahuluan akan terdiri atas latar belakang masalah, rumusan masalah, batasan masalah, tujuan, sistematika penulisan laporan, dan jadwal pelaksanaan kegiatan tugas akhir.

2. BAB II LANDASAN TEORI

Bab II Landasan Teori akan membahas teori-teori dasar dan tinjauan pustaka yang akan digunakan sebagai basis penelitian tugas akhir ini.

3. BAB III METODOLOGI PENELITIAN

Bab III Landasan Teori akan membahas tentang metodologi yang digunakan dan proses analisis dan desain aplikasi yang akan dikerjakan.

4. BAB IV HASIL DAN PEMBAHASAN

Bab IV Hasil dan Pembahasan berisikan analisis dan evaluasi terhadap hasil penelitian tugas akhir dan aplikasi yang telah dihasilkan.

5. BAB V KESIMPULAN DAN SARAN

Bab V Kesimpulan dan Saran berisikan kesimpulan dan saran berdasarkan hasil dan pembahasan dari penelitian tugas akhir yang telah dilaksanakan.



1.5 Jadwal Kegiatan

Tabel 1.1 Jadwal Kegiatan Skripsi

Rencana Kegiatan	Bulan dan Minggu, Tahun 2021 - 2022								
	September	Oktober	November	Desember	Januari	Februari	Maret	April	Mei
Penyusunan dan Pengumpulan Proposal									
Seminar Proposal									
Analisis dan Desain									
Implementasi dan Testing									
Pembuatan Laporan Hasil									
Seminar Hasil									
Seminar Akhir									

BAB II LANDASAN TEORI

2.1. TINJAU PUSTAKA

2.1.1. A Short Introduction To Procedural Content Generation Algorithm For Video Games (Nicolas A. Barriga, 2018)

Algoritma PCG dapat dikelompokkan menjadi tiga kategori, yaitu metode tradisional yang mencakup penggunaan pseudo random generator seperti *seed*, *fractal* dan *noise* dalam pembuatan konten, metode *search-based* dimana algoritma berperan sebagai penghasil konten sekaligus pengevaluasi kelayakan konten, dan metode *machine learning* dimana konten dihasilkan berdasarkan hasil prediksi dari dataset yang telah disediakan.

2.1.2. An Analog History Of Procedural Content Generation (Gillian Smith, 2015)

Procedural Content Generation adalah dimana sebuah konten dihasilkan melalui sebuah perantara *agent* dengan mengikuti suatu prosedur atau langkah yang telah ditentukan. Dalam konteks video *game*, PCG adalah penggunaan sebuah algoritma formal untuk menghasilkan sebuah konten *game* yang biasanya dilakukan secara manual oleh manusia (Gillian Smith, 2015).

Menurut Gillian Smith, awal implementasi PCG dapat ditemukan pada *game* analog terdahulu, dimana implementasi PCG dilakukan dengan motivasi untuk menambahkan nilai pengulangan suatu *game*, membantu proses kreativitas pemain, sebagai sarana berekspresi, menjamin konten tanpa bias dan adil, dan menggantikan manusia dalam menghasilkan konten.

2.1.3. Procedural Generation : An Algorithmic Analysis Of Video Game Design And Level Creation (Logan Bond, 2017)

Fungsi dari procedural generation adalah untuk menghasilkan sebuah data secara algoritmik, yang biasanya perlu dilakukan secara manual. Pengimplementasian procedural generation dapat menggeserkan beban pekerjaan dari individu ke algoritma, sehingga dapat mengizinkan lebih banyak konten untuk dapat dihasilkan. Procedural generation sering digunakan untuk menghasilkan berbagai komponen dalam video game, hingga seseorang dapat membuat game yang menggunakan algoritma untuk menghasilkan keseluruhan kontennya. Game yang menggunakan procedural generation sering kali digunakan dalam pembuatan peta, barang, karakter, grafis dan banyak hal lain.

2.1.4. A Video Game Level Analysis Model Proposal (Samuel Almeida, Ana Veloso, Óscar Mealha, Licínio Roque, 2012)

Level design adalah sebuah bagian yang penting dalam process pengembangan game yang menghasilkan sebuah stage dimana semua aksi terjadi. Pada sebuah tim pengembangan game saat membuat game (dan level-level pada game), hasil akhirnya adalah dimana pemain dapat menjelajahi level sepenuhnya. Tetapi hal ini tidak selalu terjadi. Di beberapa kejadian, 25% dari level game diabaikan oleh pemain. Sementara cara bermain pemain yang berbeda dapat mempengaruhi seberapa banyak pemain memanfaatkan level, dan hal ini tergantung dari objektif dari game tersebut; Dapat dibayangkan bagi pemain untuk menyelesaikan game tanpa berinteraksi dengan bagian dari elemen game (scene, objek, karakter) yang beberapa diantaranya penting untuk aliran gameplay yang sebenarnya.

2.1.5. Exit Probability and First Passage Time of a Lazy Pearson Walker: Scaling Behaviour (Muktish Acharyya, 2015)

Random walk adalah masalah, yang dipelajari secara luas dalam matematika, statistik dan fisika untuk menganalisis berbagai fenomena alam. Sebagai contoh, di fisika statistik, proses polimerisasi, difusi, mikropartikel dll adalah contoh fenomena klasik yang menarik perhatian para peneliti beberapa decade terakhir. Dasar mekanisme dari fenomena ini dijelaskan oleh random walk dalam berbagai bentuk. Berbagai macam random walk dipelajari di titik kordinat di dimensi yang berbeda oleh simulasi computer.

2.2. TEORI PENDUKUNG

2.2.1. Level

Menurut Jesse Schell (2014) Dalam *game* video atau komputer, sebuah *level* (disebut juga misi, tingkat, zona, maupun banyak istilah lainnya) adalah ruang terbatas yang dapat dijelajahi pemain dalam mencapai tujuan tertentu. Banyak *game* video memiliki rangkaian *level* yang semakin lama semakin sulit sehingga dapat dinikmati pemain dengan tingkat kemampuan yang berbeda-beda. Setiap *level* baru juga dapat memberikan konten baru atau tantangan baru sehingga pemain tidak bosan.

2.2.2. Role Playing Game (RPG)

Menurut Wandah Wibawanto(2020), Sesuai dengan struktur kata yaitu role playing atau bermain peran, maka definisi sederhana dari RPG adalah permainan di mana pemain menjalankan peran karakter dalam sebuah cerita fiksi. Pemain bertugas untuk memerankan peran-peran yang ditentukan oleh narasi permainan, serta dituntut untuk melalui proses pengambilan keputusan terstruktur terkait pengembangan karakter. Peran yang dilakukan

oleh pemain ditentukan oleh sistem aturan yang didefinisikan secara spesifik oleh pengembang permainan.

2.2.3. Survival Horor

Menurut Melodie Griffin (2019), Survival Horor Video Game (SHVG) adalah salah satu genre yang tumbuh paling cepat untuk pengembang video game independen. Dengan elemen sinematik yang mengerikan dan karakteristik menakutkan, video game survival horror menciptakan pengalaman yang menakutkan bagi pengguna dalam lingkungan maya.

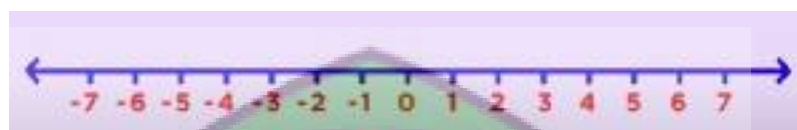
Seperti semua video game, pemain mengharapkan pengalaman yang memuaskan, menantang, dan menghibur; tetapi tidak seperti kebanyakan genre game, SHVG harus mendesain dengan hati-hati narasi untuk menakut-nakuti, menantang, dan menggairahkan para pemainnya. Untuk mencapai trifecta ini, sebagian besar SHVG dibangun untuk membuat pengguna merasa "terhibur" hanya dengan bertahan sepanjang permainan. Keseimbangan horor dan hiburan dalam lingkungan virtual menghasilkan imersi pemain.



Gambar 2.1 Dark Wood oleh Acid Wizard Studio

2.2.4. Random walk

Menurut Feng Xia (2019). Random walk adalah sebuah objek matematis, dikenal sebagai proses acak (stokastik), yang menggambarkan sebuah jalur yang terdiri dari serangkaian langkah acak berturut-turut dalam suatu ruang matematis seperti bilangan-bilangan bulat. Contoh dasar dari Random walk adalah pergerakan pada 1 dimensi integer.



Gambar 2.2 1D integer

Dimana probabilitas kita bergerak kekanan adalah P dan probabilitas kita bergerak ke kiri adalah $1-p$ apabila $P = \frac{1}{2}$ maka random walk ini disebut non bias random walk. Pada non bias random walk kita dapat mencari posisi kita setelah X coin flip dengan rumus berikut.

$$S_n = \sum_{k=1}^n X_k,$$

Atau, $S_n = X_1 + X_2 + X_3 \dots X_n$

Dimana

S_n = posisi setelah n coin flip

X = coin flip dimana probabilitas nilai 1 adalah $\frac{1}{2}$ dan -1 adalah $\frac{1}{2}$

apabila nilai $P > \frac{1}{2}$ maka kita cenderung bergerak kekanan. dan apabila $P < \frac{1}{2}$ kita akan bergerak kekiri. Hal ini disebut dengan bias random walk.

contoh apabila $P = \frac{3}{4}$ maka $P = \frac{3}{4}$ dan $1-P = \frac{1}{4}$. Jadi, kita akan 3 kali lebih cenderung bergerak kekanan dibandingkan kekiri. Kita dapat menentukan rata-rata probabilitas lokasi kita setelah X coin flip apabila dengan rumus $2(P-1)$. Jadi apabila $p = \frac{3}{4}$ maka $2(\frac{3}{4}-1) = \frac{1}{2}$. Yang berarti apabila ada 10 coin flip. Kita diharapkan berada pada posisi 5 disebelah kanan.

Sedangkan pada 2 dimensi kesempatan bergerak ke 1 arah spesifik adalah $\frac{1}{4}$ pada masing-masing arah yaitu $x,y,-x,-y$.

2.2.5. Aseprite

Aseprite adalah aplikasi pengolah gambar yang diperuntukan untuk membuat animasi 2d dan sprite *Pixel Art* yang diperuntukan untuk *game* atau kebutuhan grafik lainnya.



Gambar 2.3 Interface Aseprite

2.2.6. Godot Engine

Godot adalah sebuah Integrated Development Environment (IDE) yang digunakan untuk mengembangkan sebuah *game* 2D dan 3D lintas platform. Godot dapat didapatkan dengan gratis karena dirilis di bawah lisensi MIT. Godot pada awalnya dikembangkan oleh pengembang perangkat lunak Argentina, Juan Linietsky dan Ariel Manzur untuk beberapa perusahaan di Amerika Latin sebelum rilis publik



Gambar 2.4 Interface Godot

2.2.7. UMLet

UMLet adalah alat UML berbasis open source Java yang dirancang untuk mengajarkan UML atau membuat diagram UML.



Gambar 2.5 Interface UMLet


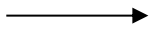

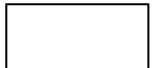
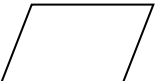
2.2.8. Bahasa Pemrograman GDscript

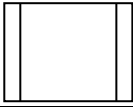
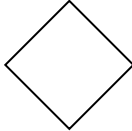
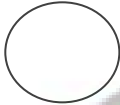

GDScript adalah bahasa pemrograman tingkat tinggi yang diketik secara dinamis yang digunakan untuk membuat konten. GDScript menggunakan sintaks yang mirip dengan Python (blok indent-based dan banyak kata kunci yang serupa). Tujuannya adalah untuk dioptimalkan dan terintegrasi erat dengan Godot Engine, yang memungkinkan fleksibilitas yang besar dalam pembuatan dan integrasi konten.

2.2.9. Flowchart

Flowchart atau diagram alir adalah sebuah jenis diagram yang merepresentasikan suatu algoritma, alur kerja atau proses, dengan cara menunjukkan setiap langkah dalam berbagai bentuk kotak, dan urutan langkahnya menggunakan tanda panah. Flowchart digunakan untuk menganalisa, mendesain, mendokumentasikan, atau mengelola suatu proses atau program.

Tabel 2.1 Simbol-Simbol Flowchart

Simbol	Nama	Fungsi
	Terminator.	Permulaan atau akhir program.
	Garis Alir (Flow Line).	Arah aliran program.
	Preparation.	Proses inialisasi atau pemberian harga awal.
	Proses.	Proses perhitungan atau proses pengolahan.
	Input/Output Data.	Proses input atau output data, parameter, informasi.

	Predefined Process (Sub Program).	Permulaan sub program atau proses menjalankan sub program.
	Decision.	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
	On Page Connector.	Penghubung bagian-bagian flowchart yang berada pada satu halaman.
	Off Page Connector.	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda.

2.2.10. Unified Modeling Language (UML)

UML yang merupakan singkatan dari Unified Modeling Language adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML telah menjadi bahasa standar dalam industri untuk memvisualisasikan, merancang dan mendokumentasi sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti & Wahono, 2003).

A. Use Case Diagram

Sebuah use case menggambarkan sekumpulan aktivitas yang dilakukan untuk menghasilkan suatu hasil keluaran. Setiap use case menjelaskan bagaimana pengguna luar dapat memicu suatu 'kejadian' dimana sistem perlu






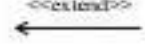
menanggapinya (Dennis, Wixom & Roth, 2012). Diagram use case dapat digunakan untuk mendeskripsikan bagaimana suatu sistem akan terlihat di mata pengguna. .

Untuk menghasilkan diagram use case yang mudah dibaca dan dipahami, Ambler (2005) menyarankan beberapa panduan dalam pembuatan diagram use case, yaitu:

1. Mulai nama use case dengan kata kerja yang kuat.
2. Berikan nama use case sesuai dengan terminologi bidang kajian.
3. Susun use case sesuai dengan urutan perlakuan.
4. Letakan aktor utama pada kiri atas diagram.
5. Gambarkan aktor di luar tepi diagram.
6. Asosiasikan setiap aktor dengan satu atau lebih use case.
7. Namakan aktor sesuai perannya, bukan gelar pekerjaannya.
8. Gunakan <<system>> untuk menandakan aktor dari sistem.
9. Jangan perbolehkan interaksi antar aktor.
10. Tambahkan aktor “*Time*” untuk memulai kegiatan yang periodik.
11. Gunakan simbol asosiasi jika aktor muncul pada logika use case.
12. Hindari menggunakan tanda panah untuk relasi aktor-use case.

13. Gunakan <<include>> ketika diketahui pasti kapan use case dikerjakan.
14. Gunakan <<extend>> ketika pengerjaan use case memerlukan beberapa langkah use case lain.
15. Gunakan <<generalized>> untuk use case dimana satu kondisi dapat menghasilkan logika use case yang berbeda.
16. Hindari menggunakan relasi asosiasi use case pada dua atau lebih tingkat.
17. Letakan use case yang meng-include di sebelah kanan.
18. Letakan use case yang ter-extend dibawah use case induk.

Tabel 2.2 Simbol-Simbol Use Case Diagram

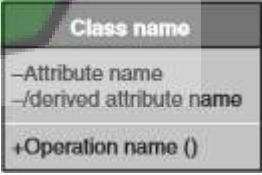
Simbol	Keterangan
	Aktor : Mewakili perorangan, sistem yang lain, atau alat ketika berkomunikasi dengan use case
	Use case : Abstraksi dan interaksi antara sistem dan aktor
	Association : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case sebenarnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

B. Class Diagram

Class diagram pada UML adalah suatu diagram struktur statis yang menggambarkan suatu struktur dari sebuah sistem dengan cara memperlihatkan kelas-kelas yang dimiliki oleh sebuah sistem, beserta atribut, operasi (atau method), dan relasi yang dimilikinya antara objek lainnya.

Kelas adalah balok pembangun utama dari sebuah diagram kelas. yang menyimpan dan mengelola informasi dalam sebuah sistem (Dennis, Wixom & Roth, 2012). Berikut adalah bentuk dasar kelas dalam sebuah diagram kelas.

Tabel 2.3 Bagian Suatu Kelas (Dennis, Wixom & Roth, 2012)

Istilah dan Definisi	Simbol
<p style="text-align: center;">Kelas</p> <ul style="list-style-type: none"> • Merepresentasikan seseorang, tempat, atau sesuatu yang harus didata oleh sistem. • Memiliki nama yang ditulis tebal dan menengah pada bagian atas. • Memiliki daftar atribut pada bagian tengah. • Memiliki daftar operasi yang dapat dilakukan pada bagian bawah. • Tidak secara langsung menampilkan seluruh operasi yang tersedia bagi seluruh kelas. 	 <p style="text-align: center;"> Class name -Attribute name -/derived attribute name +Operation name () </p>
<p style="text-align: center;">Atribut</p>	<p style="text-align: center;">Nama atribut</p> <p style="text-align: center;">/Nama atribut turunan</p>

<ul style="list-style-type: none"> • Merepresentasikan properti yang menggambarkan kondisi suatu objek. • Dapat menurunkan dari atribut lain, ditunjukkan dengan garis miring sebelum nama atribut. 	
<p style="text-align: center;">Method</p> <ul style="list-style-type: none"> • Merepresentasikan aksi atau fungsi yang dapat dilakukan kelas. • Dapat diklasifikasikan sebagai konstruktor, <i>query</i>, atau operasi <i>update</i>. <p>Diikuti tanda kurung yang dapat berisi parameter atau informasi yang diperlukan untuk menjalani operasi.</p>	<p style="text-align: center;">Nama operasi ()</p>

Pada class diagram juga digambarkan bagaimana interaksi hubungan antar kelas dalam sebuah konstruksi perangkat lunak seperti hubungan asosiasi, inheritance, generalisasi, agregasi, dan komposisi.

1. Asosiasi

Menggambarkan kelas yang memiliki atribut berupa kelas lain, atau kelas yang harus mengetahui keberadaan kelas lain.

2. Inheritance

Menggambarkan hubungan hirarkis antar kelas. Kelas dapat diturunkan dari kelas lain dan mewarisi semua atribut dan method kelas asalnya, dan menambahkan

fungsionalitas baru, sehingga ia disebut anak dari kelas yang diwarisinya

3. Generalisasi

Kebalikan dari inheritance, menggambarkan suatu hubungan dimana suatu kelas memiliki atribut yang sama dari dua atau lebih kelas lainnya.

4. Agregasi

Menggambarkan hubungan antar kelas yang memiliki relasi “has-a”.

5. Komposisi

Menggambarkan hubungan antar kelas yang memiliki relasi “part-of”.




Gambar 2.4 Simbol-Simbol Relasi Antar Kelas

C. Activity Class Diagram

Activity diagram digunakan untuk menggambarkan rangkaian aliran dari aktivitas, dan berguna untuk

mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.

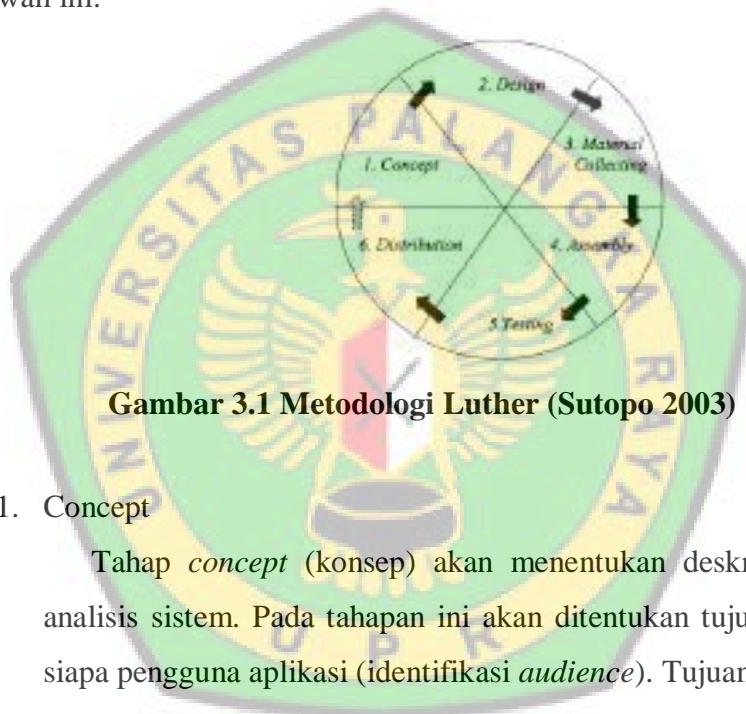
Tabel 2.4 Simbol-Simbol Activity Diagram

Keterangan	Simbol
Titik Awal atau permulaan.	●
Titik Akhir atau akhir dari aktivitas.	●
Aktivitas, atau aktivitas yang dilakukan oleh aktor.	
Decision, atau pilihan untuk mengambil keputusan.	
Arah tanda panah alur proses.	

BAB III METODE PENELITIAN

3.1. Metodologi Pengembangan Perangkat Lunak

Metode penelitian yang akan digunakan dalam pengembangan *game* “Lawang” ini adalah Metodologi Luther. Menurut Sutopo, metodologi Pengembangan multimedia terdiri dari 6 tahapan, yaitu *concept*, *design*, *material collecting*, *assembly*, *testing* dan *distribution* seperti gambar di bawah ini:



Gambar 3.1 Metodologi Luther (Sutopo 2003)

1. Concept

Tahap *concept* (konsep) akan menentukan deskripsi proyek dan analisis sistem. Pada tahapan ini akan ditentukan tujuan aplikasi dan siapa pengguna aplikasi (identifikasi *audience*). Tujuan aplikasi adalah untuk menyembunyikan informasi yang telah dienkripsi kedalam sebuah gambar dan target penggunanya adalah pengguna *smartphoneandroid*. Sedangkan pada analisis sistem akan dibuat diagram alir(*flowchart*) aplikasi yang dibuat yang menjelaskan proses bisnis aplikasi steganografi ini.

2. Design

Design (perancangan) adalah tahap membuat spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan material/bahan untuk program.

3. *Material Collecting*

Material Collecting adalah tahap dimana pengumpulan bahan-bahan seperti gambar, foto, animasi, video, audio dan lain-lain. Bahan-bahan yang mendukung dalam pembuatan aplikasi ini di peroleh dari internet dan bahan-bahan yang didesain oleh penulis selaku pengembang aplikasi.

4. *Assembly*

Tahap *assembly* (pembuatan) adalah tahap dimana semua objek atau bahan multimedia dibuat. Pembuatan aplikasi didasarkan pada tahap design.

5. *Testing*

Dilakukan setelah selesai tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan dilihat apakah ada kesalahan atau tidak. Tahap ini disebut juga sebagai tahap pengujian alpha (*alpha test*) dimana pengujian dilakukan oleh pembuat atau lingkungan pembuatnya sendiri.

6. *Distribution*

Tahapan dimana aplikasi disimpan dalam suatu media penyimpanan. Pada tahap ini jika media penyimpanan tidak cukup untuk menampung aplikasinya, maka dilakukan kompresi terhadap aplikasi tersebut.

3.2. Concept

3.2.1. Deskripsi Project

Tabel 3.1 Deskripsi Project

1	Nama Project	“Penerapan Algoritma Random Walk Dalam Membuat Level Pada <i>Game</i> Rpg Secara Prosedural ”
2	Target Pengguna	Semua umur/Umum
3	Tujuan Project	Membuat <i>game</i> RPG dengan penerapan algoritma random walk dalam membuat level secara procedural
4	Macam Aplikasi	<ol style="list-style-type: none"> 1. Aplikasi <i>game</i> dibuat dengan Godot Engine. 2. Aplikasi <i>game</i> ini memiliki beberapa level yang akan dilalui pemain yang akan dibuat oleh algoritma random walk secara procedural. 3. Aplikasi <i>game</i> yang dibuat berbasis desktop dengan tampilan 2D.
5	Spesifikasi umum	<ol style="list-style-type: none"> 1. AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz 2. 8,00 GB RAM (5,94 GB usable) 3. Windows 10 Home Single Language 4. HDD 1 TB

3.3.Design

Jenis *game* dalam “Lawang” adalah RPG dengan perspektif *Top Down view* Dimana pemain menggunakan senjata dalam perspektif dari atas atau untuk menyelesaikan Level

3.3.1. Gameplay

Gameplay merupakan gambaran cara bermain pemain dalam *game* seperti melakukan pergerakan dan tindakan lain untuk mencapai tujuan dari sebuah *game*.

Tabel 3.2 Tombol keyboard

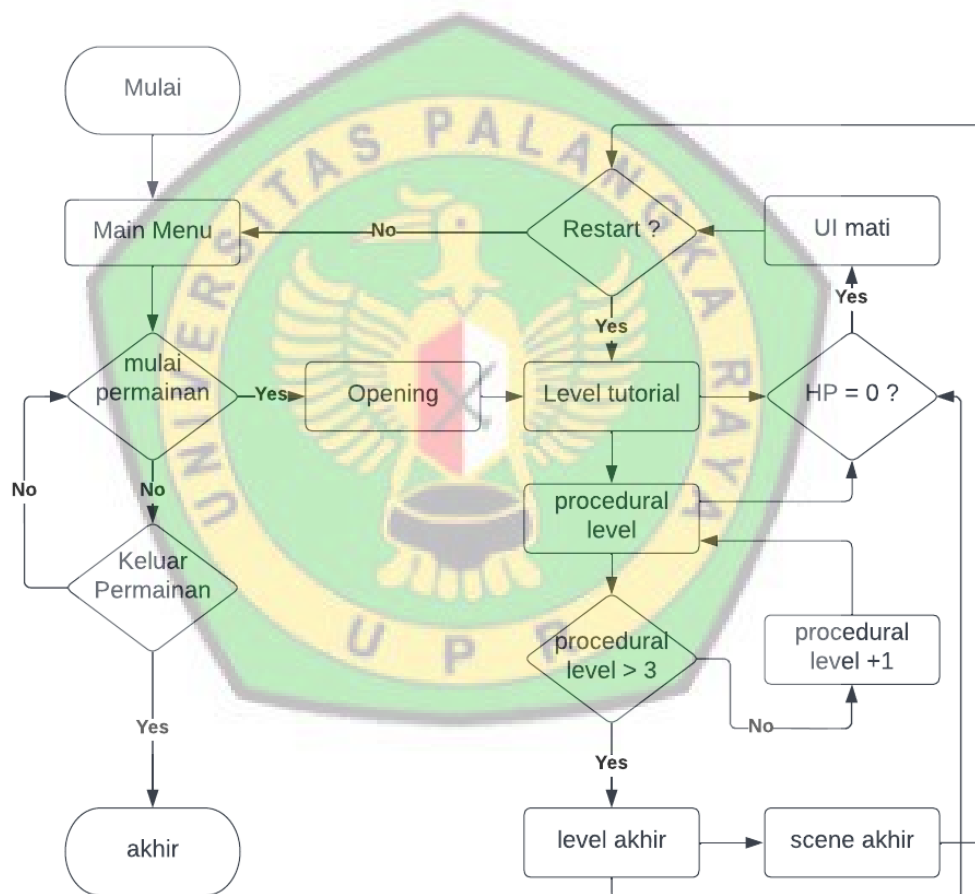
Keyboad	Fungsi
A	Bergerak ke kiri
D	Bergerak ke kanan
S	Bergerak kebawah / memilih menu keatas
W	Bergerak keatas / memilih menu kebawah
R	Reset level
Esc	Keluar dari menu mati
T	Menuju level berikutnya
J	Memilih pilihan dan menyalakan jimat

Seperti pada tabel diatas aksi diatas berdasarkan *input* dari keyboard. Untuk menggerakan pemain dapat menggunakan tombol keyboard A, S, W dan D. Untuk mengganti pilihan menu pemain dapat menggunakan tombol Keyboard S atau W. pemain dapat mengulang level dengan menekan tombol R dan menggunakan tombol T untuk melewati level tanpa harus melewati rintangan. Kemudian tombol Esc pada keyboard berfungsi untuk kembali ke menu utama apabila pemain mati. pemain dapat menggunakan tombol K untuk menyalakan jimat dan melawan monster.

Pemain memiliki HP atau jumlah *hit point* sebanyak 100, apabila pemain terkena serangan musuh maka HP pemain akan turun, HP pemain dapat pulih kembali apabila pemain tidak terkena serangan oleh monster. bila HP pemain mencapai 0 maka pemain

akan mati. Pemain juga memiliki jimat energi yang memiliki energi sebanyak 100, yang akan berkurang apabila pemain menggunakan jimat. Jimat energy akan pulih kembali apabila pemain tidak menggunakan jimat. Namun, bila pemain menghabiskan jimat energi sekaligus, maka pemain tidak dapat menggunakan jimat sampai jimat terisi 100%.

3.3.2. Flowchart



Gambar 3.2 Flowchart Game

Ketika pemain memulai *game* pemain akan masuk kedalam main menu, pada scene ini pemain dapat memulai *game* atau keluar dari *game*. Bila pemain memilih keluar *game* maka *game* akan

tertutup, bila memilih mulai *game* pemain akan masuk kedalam Scene opening untuk melihat cerita pada *game* kemudian pemain akan dikirim secara otomatis pada level tutorial *game*, setelah memasuki lawang pertama pada level tutorial pemain akan dimasukan level procedural, setiap kali pemain berhasil menyelesaikan level melalui lawang, level akan dicek, apakah pemain telah melewati 3 level. Apabila pemain belum melalui 3 level procedural, pemain akan dikirim kembali ke level procedural dengan tingkat kesusahan yang bertambah. Apabila pemain telah melalui 3 level maka pemain akan dikirim ke level terakhir. Setelah menyelesaikan level terakhir pemain akan dikirim ke scene terakhir. Pada scene terakhir pemain dapat memilih kembali kemain menu atau memulai *game* lagi.

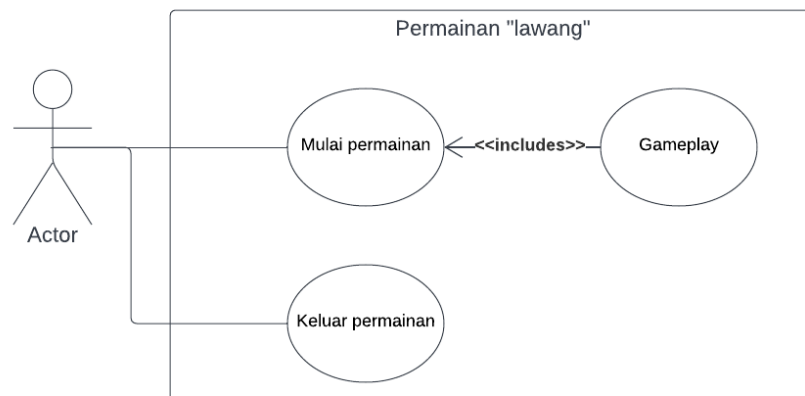
Pada tiap scene level, HP pemain akan terus di cek. Apabila HP pemain kurang dari atau sama dengan 0 maka pemain akan dianggap kalah dan masuk kedalam ui mati, disini pemain dapat memilih untuk mengulang *game* ataupun kembali ke main menu.

3.3.3. UML

Unified Modeling Language digunakan untuk menggambarkan system yang dibuat. Berikut adalah diagram-diagram yang digunakan.

a) Use Case Diagram

Diagram use case berikut memodelkan aktivitas yang ada dalam *game* yang akan dibuat.



Gambar 3.3 Use Case Diagram

b) Deskripsi Use Case

Deskripsi Use Case menggambarkan aktivitas-aktivitas yang dapat dilakukan oleh aktor dalam sebuah sistem, kondisi awal, proses, dan kondisi akhir. Berikut adalah deskripsi use case yang ada dalam sistem.

Tabel 3.3 Tabel Definisi Use Case

No	Use Case	Deskripsi
1	Mulai <i>Game</i>	<i>Pre-condition</i> : Pemain ingin memulai <i>game</i> baru <i>Post-condition</i> : Pemain memulai <i>game</i> baru
2	<i>Gameplay</i>	<i>Pre-condition</i> : Pemain masuk kedalam level <i>game</i> <i>Post-condition</i> : Pemain menyelesaikan level <i>game</i>
3	Keluar <i>Game</i>	<i>Pre-condition</i> : Pemain ingin keluar <i>game</i> <i>Post-condition</i> : Pemain keluar permainan

c) Definisi Aktor

Aktor merupakan entitas eksternal yang terlibat dalam berbagai kegiatan yang dimiliki sistem. Berikut adalah deskripsi aktor yang.

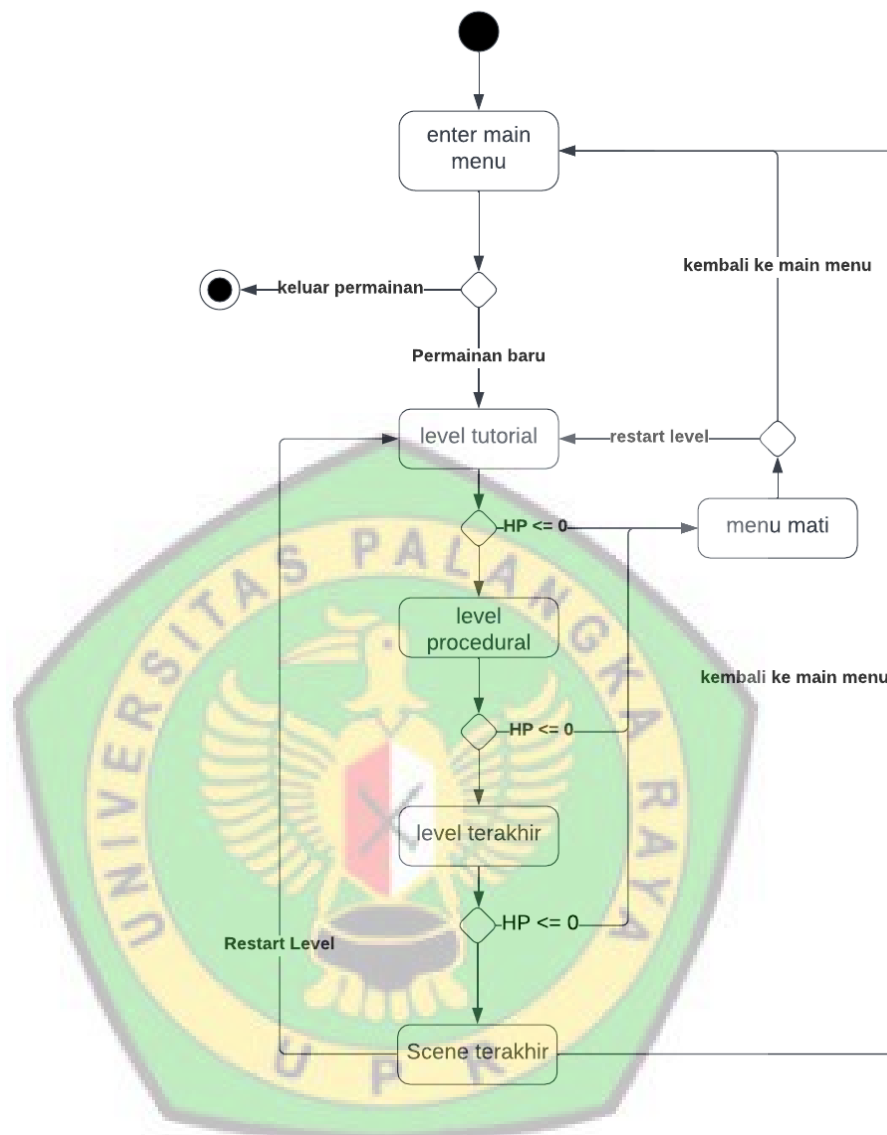
Tabel 3.4 Tabel Definisi Aktor

Aktor	Deskripsi
Pemain	Aktor yang akan memainkan <i>game</i> yang dibuat beserta fitur-fitur <i>game</i> yang tersedia.

d) Activity Diagram

Diagram Aktivitas digunakan untuk menggambarkan alur kerja yang dilalui ketika menjalani suatu proses. Alur kerja dalam aplikasi yang dibuat akan digambarkan dalam sebuah diagram aktivitas. Berikut adalah diagram aktivitas pada aplikasi.





Gambar 3.4 Activity Diagram

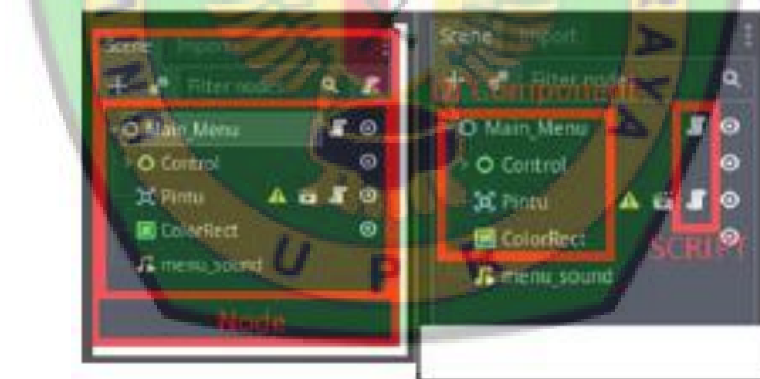
e) Class Diagram

Diagram kelas menunjukkan struktur dari sistem yang akan dibuat dalam segi pendefinisian kelas-kelas yang membangun seluruh sistem. Dikarenakan struktur objek pada *game* engine Godot bersifat sedikit berbeda dari struktur kelas tradisional pada pemrograman berorientasi objek, maka diagram kelas dalam konteks

pembuatan *game* ini akan digunakan untuk menampilkan relasi antar Node yang ada pada scene di dalam *game* yang akan dibuat, dengan upaya untuk mempersingkat dan memperjelas.

Node adalah sebuah kelas bawaan Godot yang menjadi basis dari seluruh entitas *game*. Serupa dengan struktur kelas tradisional, Node mempunyai peran dan karakteristik yang mempengaruhi jalannya aplikasi, dengan perbedaan dimana selain dapat memuat script atau koding, Node juga dapat memuat komponen-komponen tambahan Godot seperti Audio, Mesh, UI, Navigation, dll.

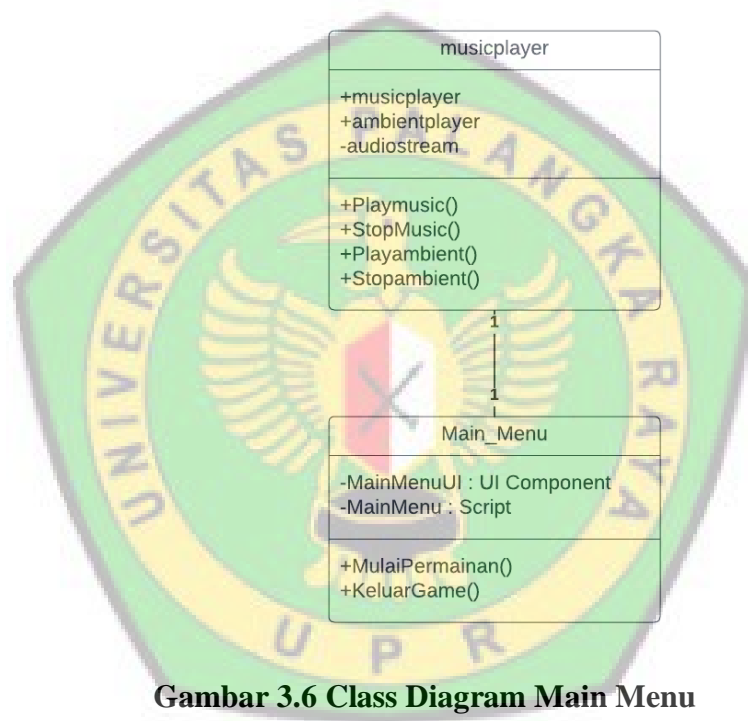
Scene adalah tempat dimana node disatukan dan dihubungkan atau lebih singkatnya scene terbuat dari kumpulan node. Sebuah scene memiliki parent Node dan child node.



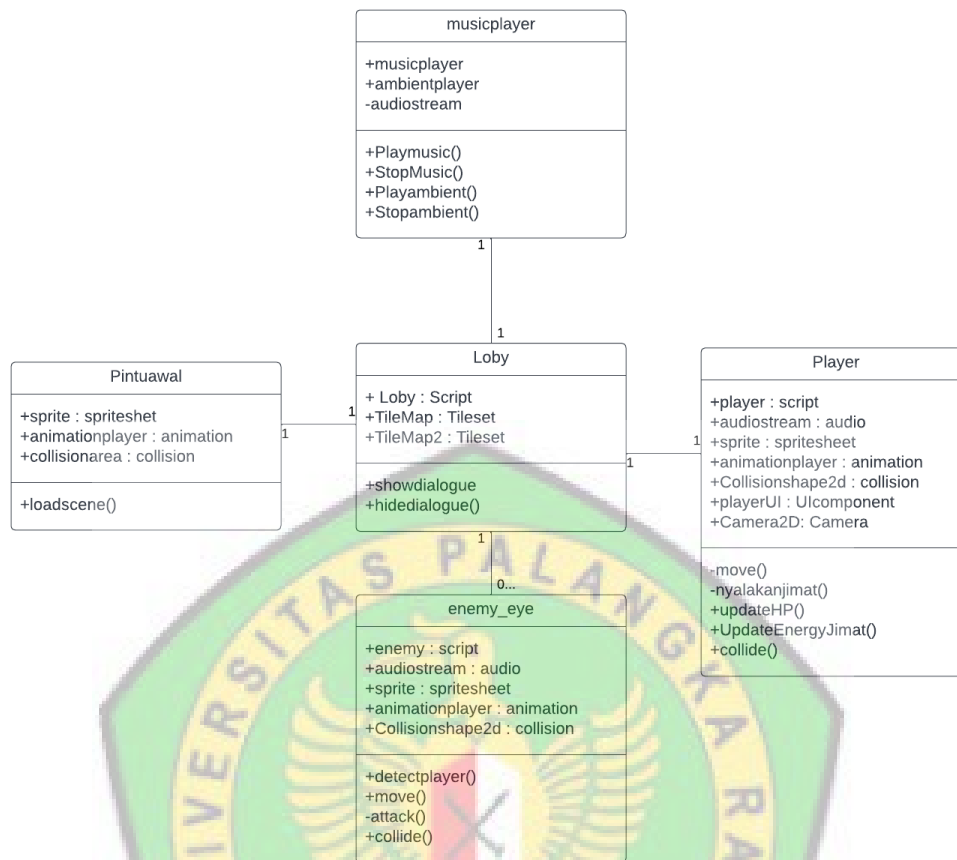
Main_Menu
-MainMenuUI : UI Component -MainMenu : Script
+MulaiPermainan() +KeluarGame()

Gambar 3.5 Scene dan node

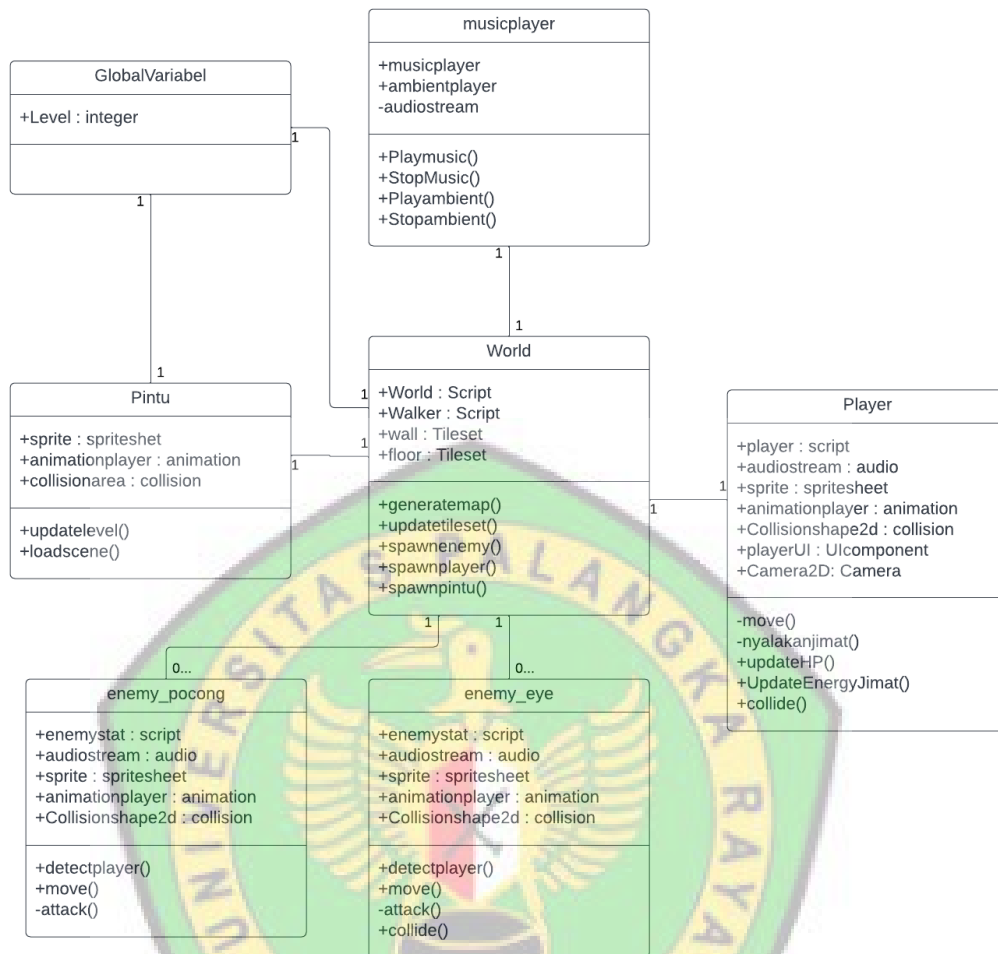
Untuk pembuatan diagram kelas akan dibagikan menjadi beberapa bagian, yaitu struktur kelas diagram yang digunakan dalam Scene Main Menu, struktur kelas diagram yang digunakan pada Scene loby (level tutorial), Scene world (Level procedural), scene level finale dan scene Tamat(ending) Berikut adalah perancangan diagram kelas yang akan digunakan dalam proses implementasi pengkodean *game*.



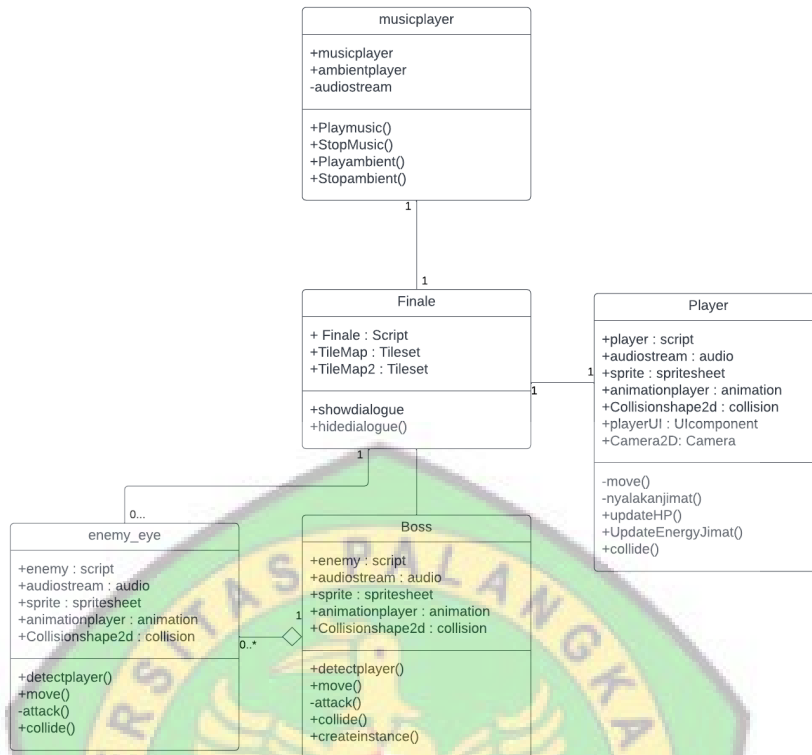
Gambar 3.6 Class Diagram Main Menu



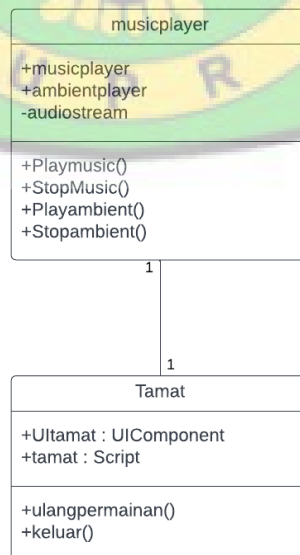
Gambar 3.7 Class Diagram Loby(tutorial)



Gambar 3.8 Class Diagram World(procedural level)



Gambar 3.9 Class Diagram Finale



Gambar 3.10 Class Diagram Tamat

3.3.4. Desain Algoritma

Desain algoritma yang akan dipakai adalah Algoritma Random walk 2 dimensi. Yang dimana tiap langkah pada $x, -x, y, -y$ memiliki probabilitas yang sama yaitu $\frac{1}{4}$. pada Random walk Godot disederhanakan dengan memilih arah pada vector2. arah kanan adalah *VECTOR.RIGHT* yang nilai vektornya $((x+1),(y+0))$,arah kiri adalah *VECTOR.LEFT* yang nilai vektornya $((x-1),(y+0))$, arah atas adalah *VECTOR.UP* yang nilai vektornya $((x+0),(y+1))$ dan arah bawah adalah *VECTOR.DOWN* yang nilai vektornya $((x+0),(y-1))$.

Pada saat algoritma ini dipanggil, algoritma akan mulai melangkah pada posisi tengah border yang berukuran 1280 x 720. Yang tiap langkahnya dikali 32 pixel sama dengan pixel snap pada tileset godot. Langkah pertama disimpan sebagai ruangan character pemain. Kemudi Walker akan melangkah sebanyak 50 langkah (pada level pertama) tiap coordinate langkahnya akan disimpan. Pada saat melakukan langkah, tiap 5 langkah walker akan berganti arah, arah akan diambil secara random dari 4 arah VECTOR2. Pada saat berganti arah algoritma akan membuat ruangan yang ruangan ini akan disimpan, kemudian walker akan mulai langkah baru dari kanan atas ruangan.

Setelah seluruh langkah selesai maka algoritma akan mengkalkulasikan ruangan terjauh dari pemain, yang akan dijadikan ruangan objectif untuk menuju level berikutnya. Kemudian musuh akan di *spawn* pada ruangan selain pada ruangan pemain.

Langkah terakhir yang akan dilakukan algoritma adalah menghapus tileset dinding untuk memberi ruangan untuk permainan.

3.3.5. Desain Antarmuka

Desain antarmuka akan dibagi menjadi 2 yaitu HUD (*Heads Up Display*) tertempel di pov pemain yang berisi info dari pemain, Interface ini akan selalu ditampilkan saat pemain sedang bermain di level dan interaktif interface seperti menu untuk berganti scene.

Berikut daftar rancang interface yang akan dibuat:

a) Antarmuka HUD

Antarmuka HUD akan ditampilkan selama pemain berada pada level antarmuka ini berisi info tentang berapa energy jimat yang dimiliki pemain dan berapa banyak HP atau *Hit Point* yang dimiliki pemain



Gambar 3.11 Antarmuka HUD

b) Desain antarmuka main menu

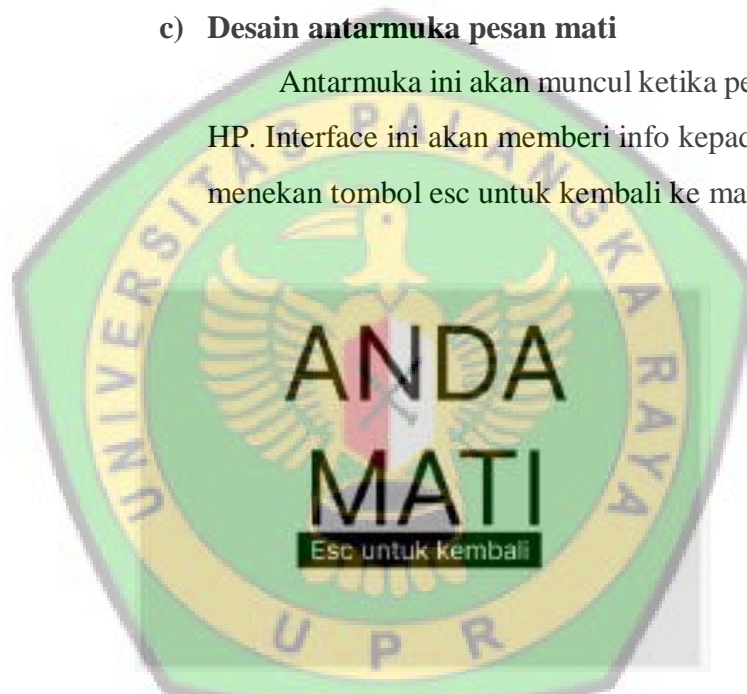
Antarmuka ini adalah antarmuka pertama yang akan dilihat pemain ketika membuka *game* ini. Antarmuka ini memiliki 2 pilihan, tombol untuk memulai *game* dan tombol keluar dari *game*.



Gambar 3.12 Antarmuka Menu Utama

c) Desain antarmuka pesan mati

Antarmuka ini akan muncul ketika pemain kehabisan HP. Interface ini akan memberi info kepada pemain untuk menekan tombol esc untuk kembali ke main menu.



Gambar 3.13 Antarmuka Pesan Mati

d) Desain antarmuka akhir *game*

Antarmuka akhir *game* akan muncul apabila pemain menyelesaikan *game* yang menyediakan pilihan untuk keluar ke main menu atau mengulang *game*.



Gambar 3.14 Antarmuka Akhir Game

e) **Desain antarmuka dialog**

Antarmuka ini akan menampilkan dialog yang menjelaskan atau menceitakan sesuatu kepada pemain



Gambar 3.15 Antarmuka Dialog

3.3.6. Desain Level

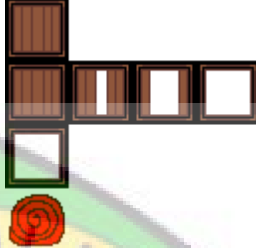





Untuk menyelesaikan *game*, pemain harus melewati 4 level. 1 level tutorial, 3 level utama yang dibuat secara procedural menggunakan algoritma random walk dan 1 level terakhir yang dimana pemain harus melawan bos.








3.3.7. Material Collecting

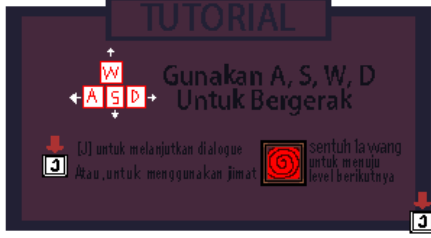


a) *Assets.*

Assets berisikan *material-material* yang nantinya akan dibuat ke dalam *game*. Asset meliputi gambar 2d, Music, suara yang dipakai untuk membuat *game*.

Tabel 3.5 Asset

NO.	Asset	Sumber
1	 <p>Lawang</p>	<i>Selfmade</i> dengan Asesprite
2	 <p>UI HP</p>	<i>Selfmade</i> dengan Asesprite
3	 <p>UI Jimat</p>	<i>Selfmade</i> dengan Asesprite
4	 <p>Tileset <i>Floor</i></p>	<i>Selfmade</i> dengan Asesprite
5	 <p>UI <i>Wall</i></p>	<i>Selfmade</i> dengan Asesprite
6	 <p>Efek <i>Damage</i></p>	<i>Selfmade</i> dengan Asesprite

7	 <p>Efek <i>Teleport</i></p>	<i>Selfmade</i> dengan Asesprite
8	 <p>Ui Press J</p>	<i>Selfmade</i> dengan Asesprite
9	 <p>Peluru</p>	<i>Selfmade</i> dengan Asesprite
10	 <p>Pocong</p>	<i>Selfmade</i> dengan Asesprite
11	 <p>Iblis Mata</p>	<i>Selfmade</i> dengan Asesprite
12	 <p>Aya</p>	<i>Selfmade</i> dengan Asesprite
13	 <p>Jae</p>	<i>Selfmade</i> dengan Asesprite

14	 <p>Tutorial</p>	Selfmade dengan Asesprite
15	 <p>Ending</p>	Selfmade dengan Asesprite
16	 <p>Opening Cutscene</p>	Selfmade dengan Asesprite
17	Sound effect	https://sfxr.me
18	Music	https://www.youtube.com/channel/UCDOVASYNDROMEYoficial

b) Cerita

Game ini dibuat dengan karakter utama yaitu seorang paranormal bernama Jae. Ia ditugaskan untuk menyelidiki penyebab terbukanya gerbang dimensi yang diberi nama Lawang. Selain itu ia juga mendengar bahwa gerbang ini ada sangkut pautnya dengan kehilangan Aya, adik Jae. Maka dari itu ia memaksakan dirinya untuk memasuki dimensi Lawang yang gelap dan dipenuhi monster. dengan bantuan dari jimat saktinya, yang dapat mengusir monster, ia memasuki lawang terdalam demi mencari petunjuk hilangnya Aya.



BAB IV

HASIL DAN PEMBAHASAN

4.1.Assembly

Tahap *assembly* (pembuatan) adalah tahap dimana semua objek atau bahan multimedia dibuat. Pembuatan aplikasi didasarkan pada tahap design.

4.1.1. Scene Main Menu

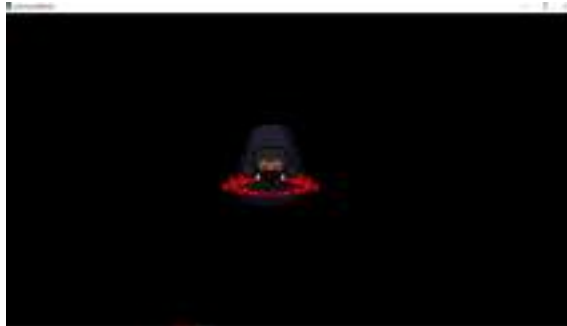


Gambar 4.1 Scene Main Menu

Main menu adalah scene paling pertama di temui pemain ketika membuka *game*. Main menu memiliki 2 buah menu yang bisa disorot menggunakan huruf “W,A,S,D” atau tombol panah pada keyboard. Untuk memilih menu pemain bisa menggunakan tombol “Enter” pada keyboard, *game* akan tertutup bila pemain memilih tombol keluar *game*. Apabila pemain memilih tombol mulai *game* maka pemain akan di pindahkan ke level tutorial.

4.1.2. Scene Opening

Pada scene ini pemain akan diberikan narasi dan animasi tentang cerita dari game lawang. Pemain juga dapat melewati scene ini dengan menekan tombol enter atau menunggu hingga scenenya selesai.



Gambar 4.2 Scene Opening

4.1.3. Scene level tutorial



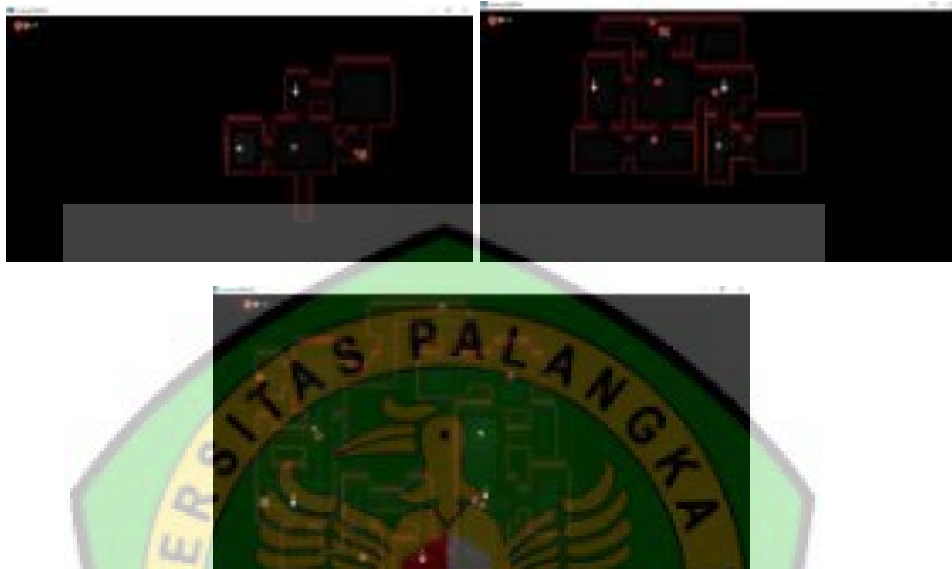
Gambar 4.3 Tutorial

Gambar 4.4 Level tutorial

Scene Loby (Level tutorial) adalah scene yang pertama dimainkan oleh *game*, fungsi dari scene ini adalah untuk memperkenalkan cara bermain dan memperkenalkan musuh yang akan dihadapi oleh pemain. Pada scene ini

4.1.4. Scene World

Scene world adalah scene yang akan degenerate menggunakan random walk untuk 3 level utama yang harus dilewati pemain. Sebelum scene terakhir melawan boss.



Gambar 4.3 Scene world level 1 (Kiri), Level 2 (Kanan), dan Level 3 (Bawah)

Pemain akan di letakan di posisi tengah pada level 1, dan level akan di generate pertama kali pada posisi ini. Untuk menuju level berikutnya pemain harus menemukan lawang sebuah pintu berbentuk kotak yang apabila didekati akan terbuka dan memperlihatkan spiral lawang.



Gambar 4.4 lawang

Dalam pembuatannya, Level 1, Level 2 dan Level 3 menggunakan metode generasi yang sama. Yang membedakan diantaranya adalah nilai-nilai parameter yang ditentukan pada awal pembuatan masing-masing scene. Berikut adalah perbedaannya.

Tabel 4.1 Perbandingan Parameter

Parameter	Level 1	Level 2	Level 3
Luas border	38 x 21	38 x 21	38 x 21
Seed	Random	Random	Random
Walker Step	50	100	300
Max enemy 1	3	6	18
Max enemy 2	1	2	6
Max sebelum belok	5 step	5 step	5 step

Berdasarkan tabel diatas, kita dapat melihat semakin tinggi level yang dilalui pemain, semakin tinggi pula tingkat kesusahan yang harus dilalui pemain, hal ini dapat dilihat dari besarnya wilayah yang harus pemain lalui untuk mencari Lawang, dan juga banyak dari jumlah musuh yang menghalangi pemain dalam mencari Lawang.

4.1.5. Scene Finale

Pada scene ini pemain akan berhadapan dengan boss dari game ini. Boss ini mempunyai 2 pola serangan yaitu serangan bola api dan sihir pemanggil makhluk kegelapan. Sihir bola api memiliki 80% kesempatan untuk dilakukan sedangkan sihir pemanggil makhluk kegelapan hanya memiliki 20% kesempatan untuk dilakukan.



Gambar 4.5 Scene Finale

4.1.6. Menu Mati

Ketika HP pemain kurang dari atau sama dengan 0 maka karakter pemain akan mati. Apabila ini terjadi pemain akan diminta untuk menekan tombol “esc” untuk kembali ke main menu.



Gambar 4.6 Menu Mati

4.1.7. Scene Tamat

Scene ini akan tampil apabila pemain berhasil mengalahkan boss terakhir. Pada scene ini pemain dapat mengulang permainan atau kembali ke main menu



Gambar 4.7 Scene Tamat

4.2. Pengujian Aplikasi

Pada tahapan ini akan dilakukan pengujian terhadap berbagai fungsionalitas aplikasi untuk menentukan kelayakan aplikasi untuk digunakan oleh pengguna dan juga kelayakan algoritma untuk membuat level. Hasil dari pengujian yang dilakukan akan menghasilkan kesimpulan yang dapat digunakan untuk pengembangan aplikasi selanjutnya.

4.2.1. Pengujian *Game*

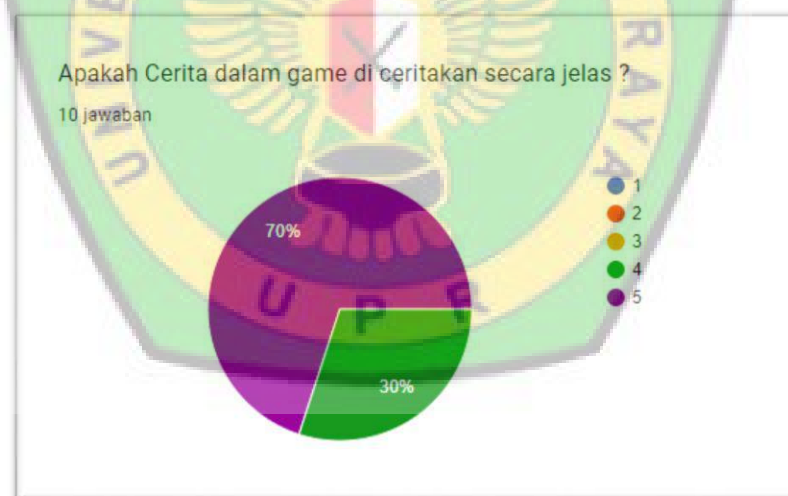
Pengujian game dilakukan dengan meminta relawan untuk mencoba game yang telah dibuat dan memberikan penilaian. Penilaian yang diminta dari relawan adalah tingkat kesenangan dalam bermain, kejelasan cerita dalam game, tingkat kesulitan game, kualitas kontrol, dan kualitas grafik game.

Relawan terdiri dari beberapa teman penulis serta mahasiswa Universitas Palangka Raya yang bersedia untuk memainkan game ini. Jumlah relawan yang memainkan game ini ada sebanyak 10 orang



Gambar 4.8 Diagram Pemahaman Cara Bermain

Berdasarkan gambar di atas, hampir semua relawan menilai bahwa cara memainkan game ini cukup mudah dipahami. Hal ini dapat dilihat dari 90% memilih nilai diatas 3.



Gambar 4.9 Diagram Kejelasan Cerita

Relawan menilai bahwa cerita pada game ini sudah cukup jelas untuk dipahami. Hal ini dapat dilihat dari 100% memilih nilai diatas 3. Hal ini juga ditunjukkan dengan gambar dibawah, dimana relawan menjelaskan isi cerita dari game ini, dan hampir semua dari relawan

menjelaskan bahwa cerita dalam game ini adalah tentang karakter utama yang sedang mencari adiknya.



Gambar 4.10 Diagram Kejelasan Cerita Game

Kejelasan cerita ini juga ditunjukkan dengan gambar di atas, dimana relawan menjelaskan isi cerita dari game ini, dan hampir semua dari relawan menjelaskan bahwa cerita dalam game ini adalah tentang karakter utama yang sedang mencari adiknya.



Gambar 4.11 Diagram Tingkat Kesusahan

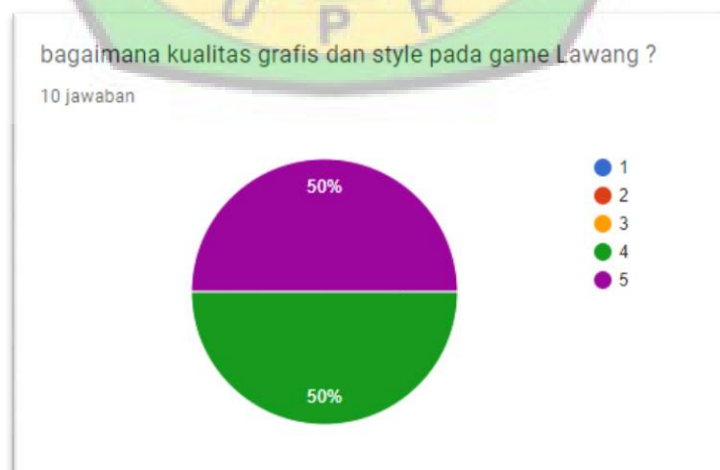
Pada gambar di atas kita dapat melihat bahwa tiap relawan memiliki pengalaman yang bervariasi pada tingkat kesusahan pada

game lawan, namun sebagian besar merasakan bahwa game Lawang tidak terlalu susah dan juga tidak terlalu mudah



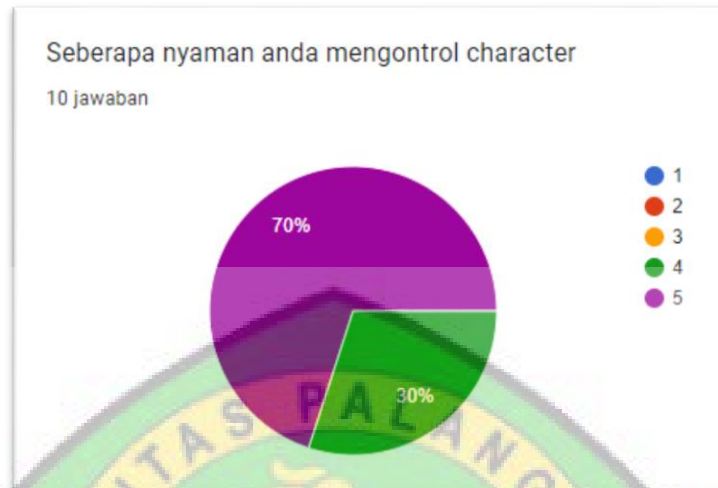
Gambar 4.12 Diagram Tantangan Level

Berdasarkan gambar diatas, menurut relawan, perubahan tiap level pada game sangat bervariasi, hal ini terjadi karena level pada game Lawang yang digenerasi secara random. 60 % menyatakan bahwa game cukup menantang, 30% menyatakan biasa-biasa saja dan 10% menyatakan bahwa level terlalu mudah.



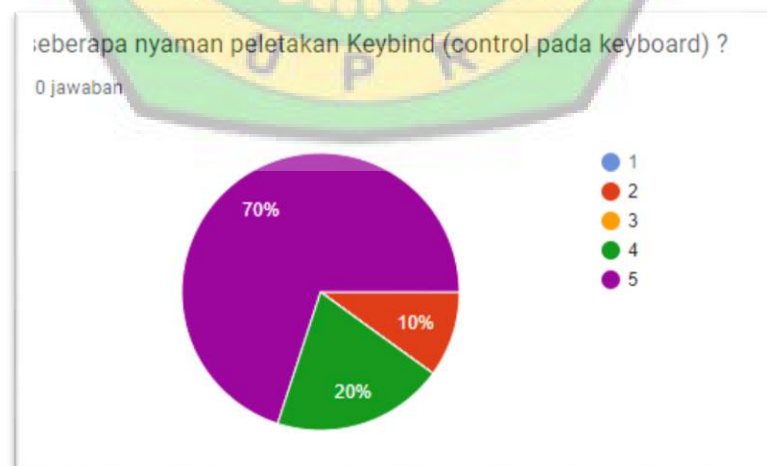
Gambar 4.13 Diagram Kualitas Grafis dan Style Pada Game

Berdasarkan gambar diatas, menurut relawan, grafis yang ada telah memadahi game Lawang.



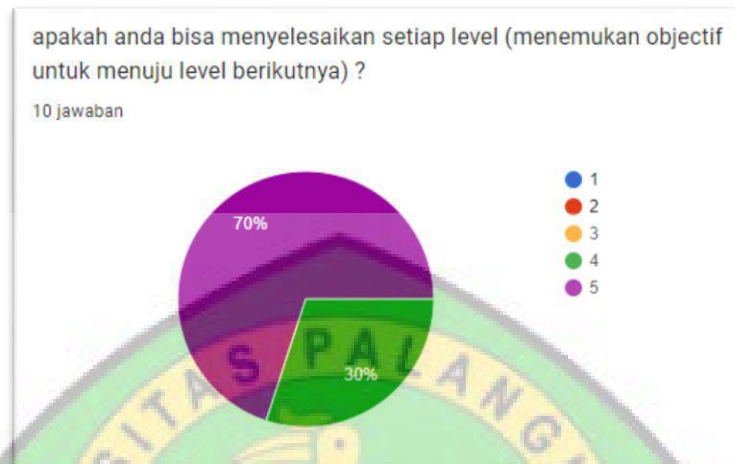
Gambar 4.14 Diagram Kenyamanan Kontrol Character

Menurut relawan, berdasarkan diagram diatas, karakter terasa nyaman untuk dikontrol. Hal ini dapat dilihat dari 100% relawan yang memberi nilai diatas 3.



Gambar 4.15 Diagram Kepuasan *Keybind*

Menurut relawan, berdasarkan diagram diatas, Peletakan keybind sudah tepat. Hal ini dapat dilihat dari 90% relawan yang memberi nilai diatas 3.



Gambar 4.16 Diagram Penyelesaian Level

Berdasarkan gambar diagram diatas, semua relawan dapat menemukan objektif pada tiap level untuk menuju level berikutnya.



Gambar 4.17 Diagram Peletakan Musuh

Berdasarkan gambar diagram diatas, Relawan merasa sudah puas dengan peletakan musuh yang dilakukan oleh algoritma map generating *random walk*.

4.2.2. Pengujian Algoritma

Pengujian algoritma akan dilakukan dengan metode sampling. Hal yang akan di sampling adalah level yang dibuat oleh algoritma *random walk*. Apakah level yang digenerate oleh algoritma ini layak digunakan untuk membuat level pada game ini.

Kelayakan level dapat dilihat dari beberapa elemen yaitu, adanya wilayah terbatas yang bisa dijelajahi pemain, objektif yang dapat dicapai pemain dan tingkat kesulitan yang bervariasi seperti yang dijelaskan oleh Jesse Schell (2014)



Gambar 4.18 Level 1 Sample 1

Berikut parameter dari level 1 sample 1:

Tabel 4.2 Level 1 Sample 1

Parameter	Nilai
Luas border	38 x 21

Seed	Random
Walker Step	50
enemy 1	3
enemy 2	1
Max sebelum belok	5 step
objektif	Dapat dituju



Gambar 4.19 Level 1 Sample 2

Berikut parameter dari level 1 sample 2:

Tabel 4.3 Level 1 Sample 2

Parameter	Nilai
Luas border	38 x 21
Seed	Random
Walker Step	50
enemy 1	2
enemy 2	1

Max sebelum belok	5 step
objektif	Dapat dituju



Gambar 4.20 Level 1 Sample 3

Berikut parameter dari level 1 sample 3:

Tabel 4.4 Level 1 Sample 3

Parameter	Nilai
Luas border	38 x 21
Seed	Random
Walker Step	50
enemy 1	3
enemy 2	1
Max sebelum belok	5 step
objektif	Dapat dituju



Gambar 4.21 Level 2 Sample 1

Berikut parameter dari level 2 sample 1:

Tabel 4.5 Level 2 Sample 1

Parameter	Level 2
Luas border	38 x 21
Seed	Random
Walker Step	100
Max enemy 1	5
Max enemy 2	1
Max sebelum belok	5 step
objektif	Dapat dituju



Gambar 4.22 Level 2 Sample 2

Berikut parameter dari level 2 sample 2:

Tabel 4.6 Level 2 Sample 2

Parameter	Level 2
Luas border	38 x 21
Seed	Random
Walker Step	100
Max enemy 1	4
Max enemy 2	2
Max sebelum belok	5 step
objektif	Dapat dituju



Gambar 4.23 Level 2 Sample 3

Berikut parameter dari level 2 sample 3:

Tabel 4.7 Level 2 Sample 3

Parameter	Level 2
Luas border	38 x 21
Seed	Random
Walker Step	100
Max enemy 1	5
Max enemy 2	2
Max sebelum belok	5 step
objektif	Dapat dituju



Gambar 4.24 Level 3 Sample 1

Berikut parameter dari level 3 sample 1:

Tabel 4.8 Level 3 Sample 1

Parameter	Level 2
Luas border	38 x 21
Seed	Random
Walker Step	300
Max enemy 1	7
Max enemy 2	3
Max sebelum belok	5 step
objektif	Dapat dituju



Gambar 4.25 Level 3 Sample 2

Berikut parameter dari level 3 sample 2:

Tabel 4.9 Level 3 Sample 2

Parameter	Level 2
Luas border	38 x 21
Seed	Random
Walker Step	300
Max enemy 1	5
Max enemy 2	2
Max sebelum belok	5 step
objektif	Dapat dituju



Gambar 4.26 Level 3 Sample 3

Berikut parameter dari level 3 sample 3:

Tabel 4.10 Level 3 Sample 3

Parameter	Level 2
Luas border	38 x 21
Seed	Random
Walker Step	300
Max enemy 1	5
Max enemy 2	2
Max sebelum belok	5 step
objektif	Dapat dituju

Berdasarkan dari 3 generasi pada tiap level pada game Lawang, dapat disimpulkan bahwa:

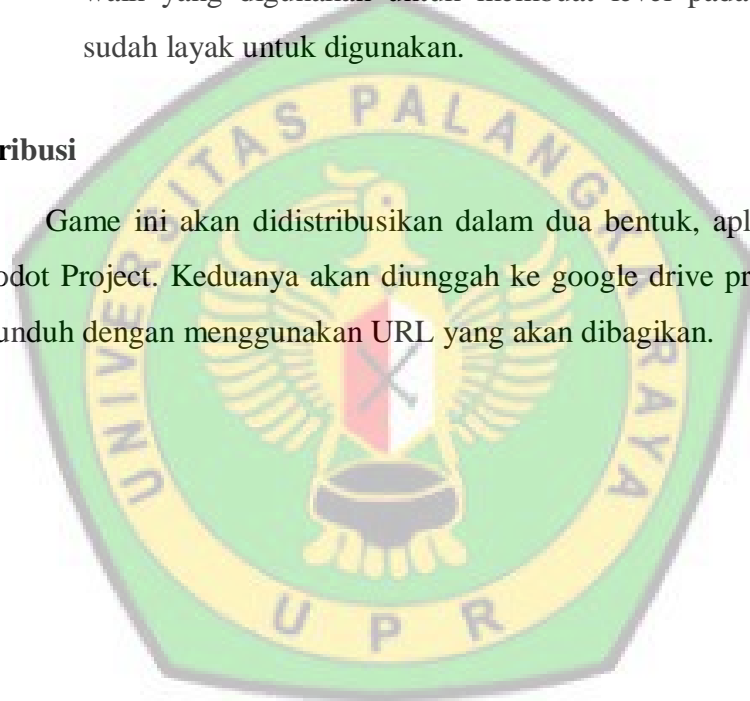
1. Algoritma berhasil membuat wilayah pada level permainan yang dapat dijelajah pemain.

2. Algoritma berhasil membuat objectif yang dapat dituju atau diakses pemain untuk menuju level berikutnya
3. Tingkat kesulitan pada tiap level bervariasi berdasarkan jumlah musuh dan besar wilayah level yang harus dijelajahi
4. Algoritma berhasil membuat Bentuk wilayah yang bervariasi pada semua level.

Berdasarkan kesimpulan tersebut, Maka algoritma random walk yang digunakan untuk membuat level pada game Lawang sudah layak untuk digunakan.

4.3.Distribusi

Game ini akan didistribusikan dalam dua bentuk, aplikasi game dan Godot Project. Keduanya akan diunggah ke google drive pribadi dan dapat diunduh dengan menggunakan URL yang akan dibagikan.



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan apa yang telah didapat dari bab-bab sebelumnya, maka kesimpulan yang didapat dari penyusunan skripsi “Penerapan Algoritma Random Walk Dalam Membuat Level Pada Game Rpg Secara Prosedural” sebagai berikut

1. Dengan Menggunakan procedural level generator, pembuat game dapat di mudahkan dalam pembuatan level pada game
2. Berdasarkan variasi level dan aksesibilitas pada objektif, Algoritma *random walk* yang diterapkan telah layak digunakan untuk membuat level secara procedural.
3. Random walk pada Godot disederhanakan dengan memilih arah pada vector2. Misalnya arah kanan maka vektornya $((x+1),(y+0))$ dan arah kiri $((x+0),(y+1))$. Namun hal ini membuat random walk tidak bisa mengarah diagonal.

5.2. Saran

Berikut adalah saran dari penulis bagi yang tertarik ingin mengembangkan penelitian ini:

1. disarankan untuk menambah menambah jenis musuh dan tampilan level agar pemain tidak cepat merasa bosan
2. Menambah variable dalam pembuatan
3. memperbaiki aspek game yang dinilai kurang, seperti AI pada musuh dan tantangan pada game.

DAFTAR PUSTAKA

- Acharyya, Muktish.2015. Exit Probability and First Passage Time of a Lazy Pearson Walker: Scaling Behaviour. Department of Physics. Calcutta, India: Presidency University.
- Almeida, Samuel. Dkk .2012. A Video *Game* Level Analysis Model Proposal. Retrieved from https://www.researchgate.net/publication/230770594_A_Video_Game_Level_Analysis_Model_Proposal.
- Ambler, Scott W. 2005. *The elements of UML 2.0 style*. Cambridge U.K. New York: Cambridge University Press.
- Barriga, A. Nicolas. 2018. A Short Introduction to Procedural Content Generation Algorithm for Video *Games*. *International Journal on Artificial Intelligence Tools*. Retrieved from https://www.researchgate.net/publication/331717755_A_Short_Introduction_to_Procedural_Content_Generation_Algorithms_for_Videogames
- Bond, Logan.2017. *Procedural Generation: An Algorithmic Analysis of Video Game Design and Level Creation. Honors Theses*. 249. Retrieved from http://scholarlycommons.obu.edu/honors_theses/249.
- Dennis, Alan, Barbara H. Wixom, and Roberta M. Roth. 2012. *System analysis and design*. Hoboken, NJ: John Wiley.
- Dharwiyanti, Sri., Wahono, R. S. 2003. Pengantar Unified Modeling Language (UML). 1-13.
- ECMA International. 2017. *C# Language Specification. ECMA-334*. Retrieved from <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-334.pdf>.
- Linietsky, Juan. Manzur, Arie, dkk. GDscript basics. https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_basics.html. (diakses 15 juni 2012).
- Pressman, RS. 2010. *Software Engineering: A Practitioner's Approach*. New York, America: McGraw-Hill Higher Education.
- Schell, Jesse. 2014. *The Art of Game Design: A book of lenses*. Pittsburgh, Pennsylvania, USA: CRC Press.

Smith, Gillian. 2015. An Analog History of Procedural Content Generation. FDG. Retrieved from <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-334.pdf>

Griffin, Melodie. 2019. Evaluating Player Immersion In Survival Horror Video Game Design. Muncie, Indiana. Ball State University.

Wibawanto, Wandah. 2020. Game Edukasi Rpg (Role Playing Game). LPPM UNNES.

Xia, Feng, dkk. 2019. Random Walks: A Review of Algorithms and Applications. Retrieved from https://www.researchgate.net/publication/337501163_Random_Walks_A_Review_of_Algorithms_and_Applications/citation/download

