

**Implementasi Procedural Level Generation pada Aplikasi
Game Pyramid Exploration**

Tugas Akhir

**Diajukan untuk Memenuhi Salah Satu Persyaratan Mencapai Derajat
Sarjana Komputer**



Dibuat Oleh:

IMMIEL NICHOLAS KITING

15 07 08215

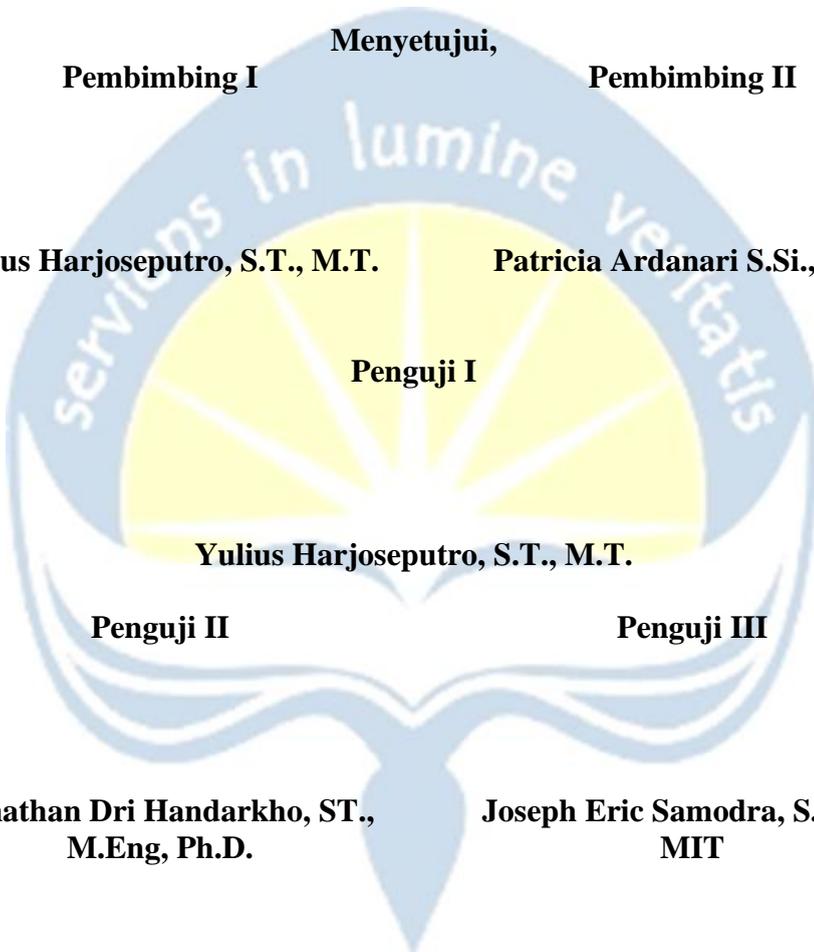
**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
2020**

LEMBAR PENGESAHAN

Implementasi Procedural Level Generation pada Aplikasi Game Pyramid
Exploration

Yogyakarta, 26 Januari 2021

Immiel Nicholas Kiting
15 07 08215



Menyetujui,
Pembimbing I Pembimbing II

Yulius Harjoseputro, S.T., M.T. Patricia Ardanari S.Si., M.T.

Penguji I

Yulius Harjoseputro, S.T., M.T.

Penguji II Penguji III

Yonathan Dri Handarkho, ST., M.Eng, Ph.D. Joseph Eric Samodra, S.Kom., MIT

Mengetahui,
Dekan Fakultas Teknologi Industri

Dr. A. Teguh Siswanto

HALAMAN PENGESAHAN

Tugas Akhir Berjudul

IMPLEMENTASI PROCEDURAL LEVEL GENERATION PADA APLIKASI GAME PYRAMID EXPLORATION

yang disusun oleh

Immiel Nicholas Kiting

150708215

dinyatakan telah memenuhi syarat pada tanggal 27 Januari 2021

		Keterangan
Dosen Pembimbing 1	: Yulius Harjoseputro, S.T., M.T.	Telah Menyetujui
Dosen Pembimbing 2	: Patricia Ardanari, S.Si.,M.T.	Telah Menyetujui
Tim Penguji		
Penguji 1	: Yulius Harjoseputro, S.T., M.T.	Telah Menyetujui
Penguji 2	: Yonathan Dri Handarkho, ST., M.Eng, Ph.D.	Telah Menyetujui
Penguji 3	: Joseph Eric Samodra, S.Kom., MIT	Telah Menyetujui

Yogyakarta, 27 Januari 2021

Universitas Atma Jaya Yogyakarta

Teknologi Industri

Dekan

ttd.

Dr. A. Teguh Siswanto, M.Sc.

PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama Lengkap : Immiel Nicholas Kiting
NPM : 15 07 08215
Program Studi : Informatika
Fakultas : Teknologi Industri
Judul Penelitian : Implementasi Procedural Level Generation pada
Aplikasi Game Pyramid Exploration

Menyatakan dengan ini:

1. Tugas Akhir ini adalah benar tidak merupakan salinan sebagian atau keseluruhan dari karya penelitian lain.
2. Memberikan kepada Universitas Atma Jaya Yogyakarta atas penelitian ini, berupa Hak untuk menyimpan, mengelola, mendistribusikan, dan menampilkan hasil penelitian selama tetap mencantumkan nama penulis.
3. Bersedia menanggung secara pribadi segala bentuk tuntutan hukum atas pelanggaran Hak Cipta dalam pembuatan Tugas Akhir ini.

Demikianlah pernyataan ini dibuat dan dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 1 Desember 2020

Yang menyatakan,

Immiel Nicholas Kiting

15 07 08215

HALAMAN PERSEMBAHAN

Semua akan indah pada waktu-Nya



KATA PENGANTAR

Puji dan syukur penulis haturkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan pembuatan tugas akhir “Implementasi Procedural Level Generation pada Aplikasi Game Pyramid Exploration” ini dengan baik.

Penulisan tugas akhir ini bertujuan untuk memenuhi salah satu syarat untuk mencapai derajat sarjana Komputer dari Program Studi Informatika, Fakultas Teknologi Industri di Universitas Atma Jaya Yogyakarta.

Penulis menyadari bahwa dalam pembuatan tugas akhir ini penulis telah mendapatkan bantuan, bimbingan, dan dorongan dari banyak pihak. Untuk itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yesus Kristus yang selalu membimbing dalam iman-Nya, memberikan berkat-Nya, dan menyertai penulis selalu.
2. Bapak Dr. A. Teguh Siswanto, selaku Dekan Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.
3. Bapak Yulius Harjoseputro, selaku dosen pembimbing I yang telah membimbing dan memberikan masukan serta motivasi kepada penulis untuk menyelesaikan tugas akhir ini.
4. Ibu Patricia Ardanari, selaku dosen pembimbing II yang telah membimbing dan memberikan masukan serta motivasi kepada penulis untuk menyelesaikan tugas akhir ini.

Demikian laporan tugas akhir ini dibuat, dan penulis mengucapkan terima kasih kepada semua pihak. Semoga laporan ini dapat bermanfaat bagi pembaca.

Yogyakarta, 1 Desember 2020

Immiel Nicholas Kiting

15 07 08215

DAFTAR ISI

Implementasi Procedural Level Generation pada Aplikasi Game Pyramid Exploration.....	1
LEMBAR PENGESAHAN	2
PERNYATAAN ORISINALITAS & PUBLIKASI ILMIAH.....	4
HALAMAN PERSEMBAHAN	5
KATA PENGANTAR	6
DAFTAR ISI.....	6
DAFTAR GAMBAR	9
DAFTAR TABEL.....	13
INTISARI.....	14
BAB I PENDAHULUAN.....	14
1.1. Latar Belakang	14
1.2. Rumusan Masalah	16
1.3. Batasan Masalah.....	16
1.4. Tujuan Penelitian	16
1.5. Metode Penelitian.....	16
BAB II TINJAUAN PUSTAKA.....	20
BAB III LANDASAN TEORI.....	27
BAB IV ANALISIS DAN PERANCANGAN GAME	31
4.1. Deskripsi Game.....	31
4.1.1. Konsep Game	31
4.1.2. Aliran/Genre Game	31
4.1.3. Target Pengguna Game	32
4.1.4. Ringkasan Alur Game	32
4.1.5. Look and Feel Game	32
4.2. Gameplay dan Mekanika.....	33

4.2.1.	Gameplay	33
4.2.2.	Mekanika.....	33
4.2.3.	Opsi Game.....	34
4.3.	Cerita, Dunia dan Karakter Game	35
4.3.1.	Cerita dan Narasi.....	35
4.3.2.	Dunia.....	35
4.3.3.	Karakter.....	35
4.4.	Level.....	35
4.4.1.	Filosofi Desain Sistem Procedural Generator	38
4.4.2.	Desain Sistem Procedural Generation Secara Modular	41
4.5.	Antarmuka.....	42
4.6.	Kecerdasan Buatan.....	43
4.7.	Kebutuhan Teknis	44
4.8.	Game Art.....	45
BAB V IMPLEMENTASI DAN PENGUJIAN GAME		47
5.1.	Implementasi Level.....	47
5.1.1.	Implementasi Sistem Procedural Generator.....	47
5.1.2.	Module Random Generator.....	47
5.1.3.	Random Generator Ruang.....	47
5.1.4.	Random Generator Koridor.....	52
5.1.5.	Module Processor.....	59
5.1.6.	Module Converter	63
5.1.7.	Module Exporter	64
5.1.8.	Penggunaan Library dalam Game.....	64
5.1.9.	Implementasi Penambahan Data Pada Game.....	66

5.1.10.	Implementasi Konstruksi Level	70
5.2.	Implementasi Gameplay dan Mekanika	93
5.3.	Implementasi Kecerdasan Buatan	94
5.4.	Pengujian Game	95
5.4.1.	Pengujian Main Menu	95
5.4.2.	Pengujian Bagian Awal Game	96
5.4.3.	Pengujian UI Game	98
5.4.4.	Pengujian Turret dan Kondisi Pemain	100
5.4.5.	Pengujian Menyelesaikan Level	101
5.4.6.	Ringkasan Pengujian Game	103
5.5.	Pengujian User	103
5.5.1.	Hasil Pengujian User dalam bentuk Chart	105
5.5.2.	Hasil Pengujian User	109
BAB VI PENUTUP		112
6.1.	Kesimpulan	112
6.2.	Saran.....	113
DAFTAR PUSTAKA		115

DAFTAR GAMBAR

Gambar 1: Perbedaan pembuatan level secara manual vs menggunakan sistem procedural generation.....	28
Gambar 2: Gambar Windhelm, salah satu lokasi di Skyrim.....	30
Gambar 3: Ilustrasi bagaimana level tersebut dibuat, dari ringkasan instruksi.....	37
Gambar 4: Ilustrasi keseimbangan kompleksitas level vs kecepatan pembuatan level.....	40
Gambar 5: Ilustrasi alur sistem Procedural Generator secara Modular.....	41
Gambar 6: Model Gun Turret yang dapat menyerang pemain.....	45
Gambar 7: Model turret sensor yang menghidupkan Gun Turret.....	45
Gambar 8: Model jalan keluar pemain ke level selanjutnya.....	46
Gambar 9: Gambar lebih dekat, jalan keluar pemain.....	46
Gambar 10: Ilustrasi bagaimana Box-Muller bekerja, berdasarkan dari penjelasan diatas.....	48
Gambar 11: Rumus Box-Muller Transform.....	49
Gambar 12: Pseudocode fungsi Box-Muller Transform.....	49
Gambar 13: Pseudocode fungsi untuk mendapatkan point acak di dalam lingkaran.....	50
Gambar 14: Pseudocode fungsi yang membuat ruangan secara acak.....	51
Gambar 15: Ilustrasi Flip Algorithm.....	53
Gambar 16: Ilustrasi algoritma Incremental.....	55
Gambar 17: Ilustrasi algoritma Sweepshull.....	57
Gambar 18: Ilustrasi fungsi RemoveOverlapping.....	58
Gambar 19: Ilustrasi pembuatan koridor dari koneksi ruangan yang dibuat.....	59
Gambar 20: Ilustrasi pembuatan koridor, dengan sudut derajat ditampilkan.....	59
Gambar 21: Ilustrasi ruangan; bagian transparent adalah data palsu, dan bagian non-transparent adalah data asli.....	60
Gambar 22: Ilustrasi penggunaan Flocking: Separation pada data ruangan.....	61
Gambar 23: Ilustrasi perubahan representasi data dari point linear ke grid.....	63

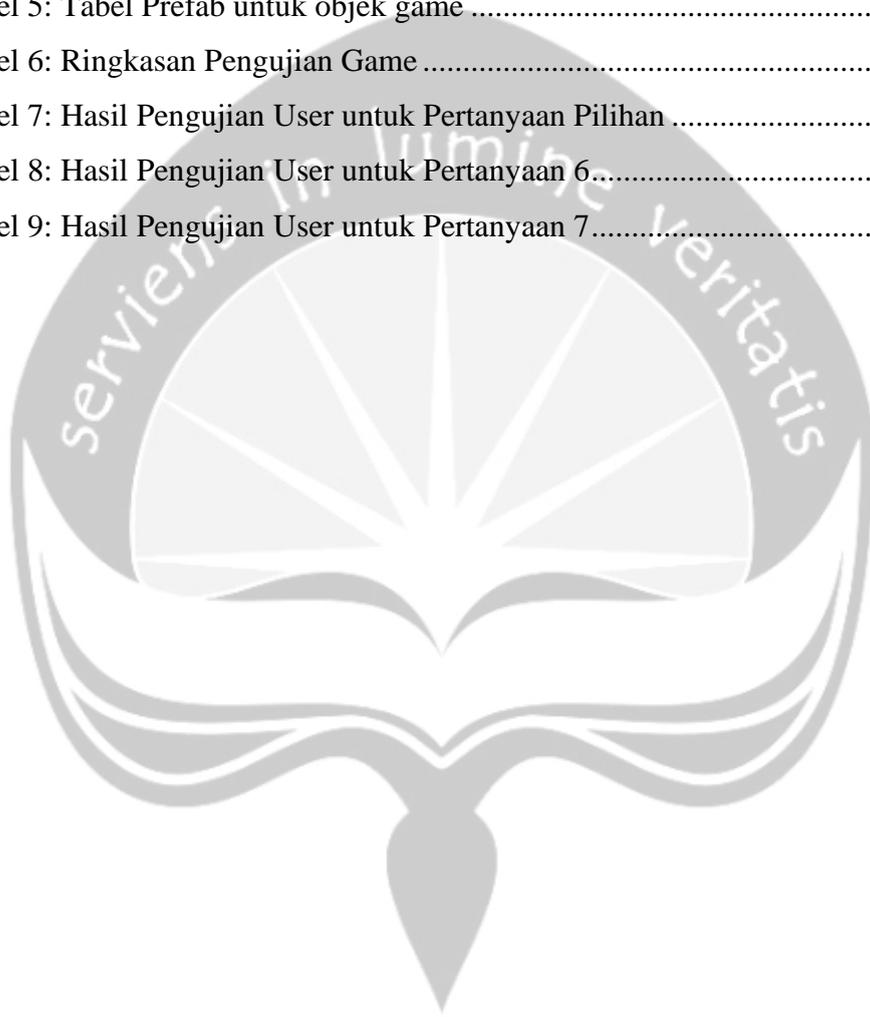
Gambar 24: Ilustrasi bagaimana data koridor dan ruangan digunakan sebagai referensi untuk membangun data grid.....	64
Gambar 25: Ilustrasi penggunaan library di dalam game	65
Gambar 26: Pseudocode algoritma untuk mendapatkan lokasi valid Gun Turret. 66	
Gambar 27: Pseudocode algoritma untuk mendapatkan lokasi valid Sensor Turret.	67
Gambar 28: Pseudocode untuk membuat lokasi mulai dan keluar pemain	69
Gambar 29: Pseudocode fungsi untuk mendapat blok dalam sebuah ruangan	69
Gambar 30: Model koridor pendek.....	72
Gambar 31: Model pojok koridor.....	72
Gambar 32: Model pojok koridor dengan pintu kiri.....	73
Gambar 33: Model pojok koridor dengan pintu kanan.....	73
Gambar 34: Model pojok koridor dengan dua pintu.....	74
Gambar 35: Model jalan buntu koridor.....	74
Gambar 36: Model jalan buntu koridor dengan pintu di sebelah kanan.....	75
Gambar 37: Model jalan buntu koridor dengan pintu disebelah kiri.....	75
Gambar 38: Model jalan buntu koridor dengan dua pintu.....	76
Gambar 39: Model jalan keluar koridor.....	76
Gambar 40: Model jalan keluar koridor dengan pintu sebelah kiri.....	77
Gambar 41: Model jalan keluar koridor dengan pintu sebelah kanan.....	78
Gambar 42: Model jalan keluar koridor dengan pintu di kiri dan kanan.....	79
Gambar 43: Model koridor biasa.....	80
Gambar 44: Model koridor biasa dengan pintu di sebelah kiri.....	80
Gambar 45: Model koridor biasa dengan pintu di sebelah kanan.....	81
Gambar 46: Model koridor biasa dengan pintu di sebelah kiri dan kanan.....	81
Gambar 47: Model koridor empat arah.....	82
Gambar 48: Model koridor tiga arah.....	82
Gambar 49: Model koridor tiga arah dengan pintu.....	83
Gambar 50: Model ruangan tanpa tembok.....	84
Gambar 51: Model pojok ruangan.....	84
Gambar 52: Model pojok ruangan, dengan pintu di sebelah kiri.....	85

Gambar 53: Model pojok ruangan, dengan pintu di sebelah kanan.....	85
Gambar 54: Model pojok ruangan dengan dua pintu.....	86
Gambar 55: Model tembok ruangan dengan pintu.....	86
Gambar 56: Model tembok ruangan	87
Gambar 57: Prefab pemain.....	88
Gambar 58: Model senter.....	88
Gambar 59: Model Gun Turret	89
Gambar 60: Model Sensor Turret	90
Gambar 61: Model peluru	91
Gambar 62: Model eskalator untuk maju ke level selanjutnya.....	91
Gambar 63: Pseudocode untuk membuat dunia game	92
Gambar 64: Main Menu Game	95
Gambar 65: Gambar game ketika tidak merespon.....	96
Gambar 66: Gambar tampilan Area Map dalam game	97
Gambar 67: Tampilan info pada game.....	97
Gambar 68: Tampilan game ketika pemain masuk ke level	98
Gambar 69: Tampilan game dalam keadaan pause.....	99
Gambar 70: Tampilan game. Pemain sedang melihat Scanner Turret yang sedang mencari pemain.	100
Gambar 71: Gambar game. Menunjukkan ruangan yang memiliki banyak pintu.	100
Gambar 72: Gambar game. Karakter pemain mati ditembaki turret.....	101
Gambar 73: Gambar game. Jalan keluar pemain dalam bentuk Elevator.	102
Gambar 74: Chart jawaban pertanyaan 1	105
Gambar 75: Chart jawaban pertanyaan 2	106
Gambar 76: Chart jawaban pertanyaan 3	106
Gambar 77: Chart jawaban pertanyaan 4	107
Gambar 78: Chart jawaban pertanyaan 5	107
Gambar 79: Hasil jawaban opsional pada pertanyaan 6	108
Gambar 80: Hasil jawaban lisan pada pertanyaan 7	109



DAFTAR TABEL

Tabel 1: Perbandingan Pustaka	26
Tabel 2: Koneksi antar empat point yang berbeda di gambar.....	54
Tabel 3: Tabel Prefab untuk Koridor	72
Tabel 4: Tabel Prefab untuk Ruangan.....	84
Tabel 5: Tabel Prefab untuk objek game	88
Tabel 6: Ringkasan Pengujian Game	103
Tabel 7: Hasil Pengujian User untuk Pertanyaan Pilihan	109
Tabel 8: Hasil Pengujian User untuk Pertanyaan 6.....	110
Tabel 9: Hasil Pengujian User untuk Pertanyaan 7.....	111



INTISARI

Implementasi Procedural Level Generation pada Aplikasi Game Pyramid Exploration

Intisari

Immiel Nicholas Kiting

15 07 08215

Semakin maju teknologi yang ada, semakin besar aplikasi yang bisa dibuat dan digunakan. Hal ini tidaklah berbeda dengan video game. Dengan majunya teknologi, video game berkembang menjadi semakin besar dan ambisius. Salah satu perkembangan yang terbuat adalah pembuatan dunia video game secara prosedural, dimana dunia video game dibuat secara otomatis oleh komputer, bukan secara manual oleh developer. Disini, kita meneliti bagaimana kita bisa membuat sistem procedural generation untuk membuat sebuah level dalam game.

Penelitian ini berisi desain dan implementasi sistem procedural generation, dimana implementasi dibuat untuk game engine Unity, dengan bahasa pemrograman C#. Kita melihat bagaimana kita bisa membuat sistem procedural generation yang dapat digunakan dalam proyek game pyramid exploration, dan membuat sistem tersebut mudah untuk dikembangkan lebih lanjut.

Hasil dari penelitian ini adalah informasi dan implementasi sistem procedural generation yang bersifat modular, dan dapat digunakan di proyek game Unity lain-nya. Ada juga implementasi game yang menggunakan sistem procedural generation, yang dapat digunakan sebagai referensi untuk penggunaan sistem procedural generation.

Kata Kunci: procedural generation, video game, level, unity, c#

Dosen Pembimbing I : Yulius Harjoseputro, S.T., M.T.

Dosen Pembimbing II : Patricia Ardanari S.Si., M.T.

Jadwal Sidang Tugas Akhir : xxx

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sistem procedural generation adalah sistem yang membuat content secara otomatis, dimana sistem membantu developer dan artist dalam pembuatan asset dan content dalam game[1]. Karena pembuatan asset dan content dilakukan secara otomatis, maka developer dan artist tidak memiliki banyak beban dalam melakukan pembuatan asset dan content untuk game yang dibuat. Waktu dan sumber daya yang dibutuhkan oleh developer dan artist untuk membuat asset menjadi sedikit, sehingga mereka dapat melakukan perihal lain dalam game, seperti pengkodean game dan pengembangan style visual game.

Sistem procedural generation sering digunakan dalam pembuatan game open world, dimana dunia game tersebut sangatlah besar. Dalam pembuatan asset dan content, sistem ini biasanya ada sebagai aplikasi tersendiri, karena mereka dibuat dengan maksud untuk membuat asset dan content tersendiri. Salah satu aplikasi populer yang digunakan untuk membuat asset dan content secara procedural adalah Houdini, dibuat oleh SideFX[2].

Tetapi, sistem procedural generation tidak hanya digunakan untuk membantu artist dan developer untuk membuat asset dan content. Sistem juga bisa digunakan secara live dalam game. Salah satu contoh populer adalah sebuah game bernama No Man's Sky[3]. Game ini adalah game space exploration, dimana galaxy, planet, dan segala flora dan fauna dibuat secara procedural[4]. Artist hanya perlu membuat asset dasar yang kemudian akan diproses oleh procedural generation engine, yang dibuat oleh developer.

Ada juga beberapa contoh lain; Minecraft[5], dimana dunia 3D adalah block yang hampir tak memiliki batas[6]. Terraria[7] juga ada sistem procedural generation, karena dunia yang pemain eksplorasi dibuat secara acak setiap kali pemain memulai game yang baru[8]. Contoh lain adalah Left 4 Dead 2[9], dimana cuaca, musuh dan jalan main berubah berdasarkan performa pemain di dalam

game[10].

Kita bisa lihat bahwa ada beberapa game yang menggunakan sistem procedural generation, dengan berbagai konteks yang berbeda. Penggunaan aplikasi yang memiliki sistem procedural generation untuk membuat asset dan content sangatlah berguna, tetapi hal yang menarik adalah penggunaan sistem procedural generation langsung di dalam game tersebut.

Kebanyakan game yang menggunakan sistem procedural generation sangatlah populer. Minecraft, salah satu game yang menggunakan sistem procedural generation dalam game mereka, memiliki 180 juta pengguna pada bulan May 2019[11]. Terraria terhitung berhasil menjual 30 juta unit pada bulan April 2020[12].

Jika game yang menggunakan sistem procedural generation dalam game sangatlah populer, kenapa tidak banyak game yang menggunakan sistem tersebut? Hal ini dikarenakan sistem procedural generation dalam game yang baik sangatlah lama dan sulit untuk dibuat.

Salah satu kasus besar untuk hal ini adalah No Man's Sky. Pada waktu pengembangan game, Hello Games membuat terlalu banyak menjanjikan hal dalam game tersebut dan gagal melaksanakannya ketika game tersebut dirilis. Akibat, game tersebut dikritik sebagai game yang sangat tidak memuaskan pada tahun itu. Tetapi, meski game tersebut sudah dirilis dan menghancurkan reputasi studio mereka, Hello Games terus melanjutkan game tersebut. Karena game tersebut terus dilanjutkan, sekarang, No Man's Sky bisa dibilang sebagai game yang baik[13].

Dari kasus tersebut, dari sisi teknis, kita bisa mengambil asumsi bahwa membuat sistem procedural generation dalam game yang baik itu memakan waktu yang lama. Apalagi jika sistem procedural generation tersebut sangatlah kompleks, seperti yang Hello Games gunakan untuk No Man's Sky. Itulah kenapa tidak banyak game studio yang mengimplementasikan sistem procedural generation di dalam game mereka.

Jadi bagaimana kita bisa melakukan implementasi sistem procedural generation dalam game dengan baik? Bagian ini adalah bagian yang banyak dibahas dalam dokumen ini. Penulis akan mendesain dan mengimplementasi sebuah game yang memiliki sistem procedural generation.

Untuk ini, penulis akan membuat game eksplorasi piramid yang memiliki sistem procedural generation, dimana sistem tersebut akan membuat level yang pemain akan mainkan di dalam game itu. Kita lihat apa yang dibutuhkan, dikorbankan dan mekanik yang dapat diimplementasi dalam game tersebut.

1.2. Rumusan Masalah

Berdasarkan latar belakang sebelumnya, maka rumusan masalah dalam penelitian ini adalah bagaimana kita mengimplementasi procedural level generation pada aplikasi game pyramid exploration.

1.3. Batasan Masalah

Penelitian ini memiliki fokus ke procedural level generation. Karena itu, penelitian tidak terlalu fokus ke bagian game, seperti cerita, musuh, dan lain-lain. Bagian tersebut ada dan didokumentasikan, tetapi yang menjadi diskusinya adalah bagaimana procedural level generation yang telah dibuat bisa digunakan untuk game tersebut. Batasan yang mencangkup projek yang dibuat adalah;

1. Penelitian ini membahas algoritma procedural generation yang digunakan.
2. Penelitian ini juga membahas bagaimana algoritma tersebut di implementasikan.
3. Penelitian ini juga menginformasikan bagaimana pembaca dapat menggunakan procedural generation untuk game mereka.
4. Ada disertakan demo implementasi yang bisa dilihat sebagai contoh, seandainya mereka ingin mengimplementasi procedural generation di game mereka sendiri.
5. Sistem operasi yang akan digunakan untuk demo adalah Windows 10. Game engine yang digunakan untuk membuat demo adalah Unity.

1.4. Tujuan Penelitian

Tujuan pada penelitian ini adalah membangun procedural level generation pada aplikasi game pyramid exploration.

1.5. Metode Penelitian

Adapun langkah-langkah yang harus dideskripsikan sebelum melakukan

penelitian ini sepenuhnya. Berikut adalah langkah-langkah tersebut;

1. Analisa Sistem Procedural Generation yang dibutuhkan.

Pada langkah ini, penulis mendeskripsikan skenario apa yang membutuhkan sistem procedural generation. Penulis juga melakukan analisa dan rasionalisasi keputusan yang dipilih untuk menggunakan sistem procedural generation pada game yang ingin dibuat. Langkah ini menghasilkan desain sistem secara dasar, yang akan dikembangkan di langkah selanjutnya.

Informasi yang didapatkan pada langkah ini akan ditempatkan pada bagian Analisis dan Perancangan Game, di bagian Level. Hal ini dikarenakan informasi yang ada mencakup pembuatan level dan segala perihal level desain.

2. Mendesain dan Mengimplementasikan Sistem Procedural Generation yang akan digunakan.

Dari hasil analisa tersebut, penulis mengumpulkan konklusi dan rasional keputusan desain sistem procedural generation. Setelah itu, penulis melakukan implementasi sistem procedural generation tersebut. Bagian ini juga melakukan analisa tentang bagaimana pengguna dapat menggunakan sistem procedural generation yang didesain dan diimplementasikan tersebut.

Untuk langkah ini, informasi yang ada akan diletakkan di bagian Analisis dan Perancangan Game, di bagian Level.

3. Mendesain dan Mengimplementasikan Game yang akan menggunakan Sistem Procedural Generation tersebut.

Berdasarkan dari hasil analisa sebelumnya, penulis mendesain dan membuat game yang menggunakan sistem procedural generation tersebut. Informasi langkah ini terdapat di bagian Analisis dan Perancangan Game, di bagian Level dan Kecerdasan Buatan. Bagian ini juga mencakup informasi tentang bagaimana game tersebut menggunakan library yang dibuat.

Aset gambar yang ada dapat dilihat pada bagian Analisis dan

Perancangan Game, di bagian Game Art. Ketika implementasi game, semua informasi akan didokumentasikan di bagian Implementasi dan Pengujian Game, di Implementasi Gameplay dan Mekanika, dan Implementasi Level.

4. **Pengujian secara Ekstensif pada Sistem Procedural Generation dan Game yang sudah di implementasikan.**

Langkah ini adalah langkah pengujian game yang sudah dibuat, dan memastikan bahwa semua implementasi yang dilakukan sesuai dengan rencana yang dilakukan sebelumnya. Semua informasi pengujian game akan didokumentasikan pada bagian Implementasi dan Pengujian Game, di bagian Pengujian Game.

5. **Pengujian User.**

Setelah semua sudah selesai, langkah terakhir adalah menguji game dilapangan. Pada bagian ini, penulis merilis game tersebut ke publik, dimana game akan dimainkan oleh masyarakat publik. Dalam kasus ini, game akan dirilis ke situs game distributor itch.io[14] secara gratis, dan user secara opsional dapat memberikan feedback tentang game yang mereka download dan mainkan melalui Google Form[15]. Feedback yang diberikan oleh user akan didokumentasikan pada bagian Implementasi dan Pengujian Game, di bagian Pengujian Game, pada bagian Pengujian User.

1.6. Sistematika Penulisan

Ada sistematika penulisan yang digunakan oleh penulis, untuk membantu dalam penulisan laporan atau skripsi ini. Berikut adalah penjelasan pendek untuk semua bab yang ada;

1. BAB 1 Pendahuluan: Berisi penjelasan dasar projek, seperti Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Masalah dan Metode Penelitian.
2. BAB 2 Tinjauan Pustaka: Berisi ringkasan dan analisa projek atau dokumen yang telah dibuat oleh peneliti lain, dimana projek tersebut hampir sama dengan projek ini.
3. BAB 3 Landasan Teori: Berisi informasi dasar yang akannya menjadi teori awal dalam melaksanakan projek yang dibuat.
4. BAB 4 Analisis dan Perancangan Game: Bagian ini berisi informasi projek game yang dibuat, seperti gameplay, mekanik, cerita, dan lain-lain. Informasi tersebut ada sebagai pengarahan dalam membuat projek game.
5. BAB 5 Implementasi dan Pengujian Game: Bagian ini berisi tentang implementasi yang dilakukan. Implementasi yang dibahas adalah gameplay dan mekanik, level, AI dan pengujian game.
6. BAB 6: Penutup: Bagian ini adalah bagian akhir dari skripsi. Bagian ini berisi kesimpulan projek dan saran yang ditulis oleh penulis, untuk pembaca.

BAB II

TINJAUAN PUSTAKA

Hampir semua game menggunakan sistem procedural content generation. Tetapi, tidak semua game menggunakan procedural generation dalam game mereka. Berikut adalah penelitian-penelitian yang sudah dilakukan sebelumnya, yang akan dijadikan acuan dasar untuk penelitian ini. Penelitian ini memiliki fokus ke teknologi procedural level generation yang digunakan, jadi dasar yang akan diambil untuk menjadi acuan adalah bagaimana teknologi procedural level generation digunakan dan bagaimana sistem tersebut bekerja.

Linden dkk. mempublikasikan survey penelitian tentang metode-metode procedural generation yang digunakan game untuk membuat konten pada tahun 2014[1]. Dalam penelitian tersebut, dibahas beberapa metode dan langkah yang harus diperhatikan. Yang pertama adalah membuat representasi model abstrak sebuah level. Metode ini digunakan untuk membuat model awal dari level yang ingin digunakan. Yang kedua adalah mengkonstruksi level abstrak dari model yang telah dibuat dari langkah sebelumnya. Disini, model abstrak sebelumnya diproses lagi menjadi data yang dapat digunakan untuk mengkonstruksi level dalam game. Yang terakhir adalah mengkonstruksi level berdasarkan data dari sebelumnya. Linden dkk. mengatakan bahwa ada banyak cara untuk membuat sistem procedural dungeon generation. Seperti apapun pendekatan penyelesaian masalah ini, seberapa spesifik dungeon yang ingin dibuat secara prosedural, tidak ada metode procedural dungeon generator yang terlihat menonjol dari satu sama lain. Hal ini dikarenakan begitu kompleks masalah pembuatan dungeon secara prosedural, dan setiap masalah memiliki solusi yang berbeda. Dalam penelitian mereka, ada banyak cara untuk membuat sistem procedural generation, tetapi setiap cara yang ada untuk menyelesaikan masalah memberikan hasil yang berbeda. Dari sini, penulis mengambil konklusi bahwa tidak ada tidak ada solusi masalah pembuatan procedural generator yang benar. Solusi yang dianggap benar adalah solusi yang terlihat baik untuk digunakan, dan tidak menimbulkan masalah ketika ingin meningkatkan kemampuan atau kecepatan sistem yang dibuat.

Selanjutnya, ada penelitian lain yang membahas implementasi dan penggunaan procedural content dalam game, yang dilakukan oleh Yannakakis. Topik pada penelitian tersebut tentang penggunaan procedural content generation sebagai konten game itu sendiri[16]. Hal tersebut memiliki arti yakni, procedural generation digunakan untuk membuat level secara live, dimana pemain dapat menggunakan generator untuk membuat level yang mereka dapat mainkan, atau level dibuat secara otomatis tanpa interferensi pemain. Algoritma tersebut digunakan untuk membuat level di Super Mario Bros secara prosedural. Yannakis dkk. menggunakan dua sistem procedural generator, dimana kedua sistem tersebut bersifat berlapis. Sistem pertama adalah sistem yang membuat level yang akan digunakan oleh pemain, dan sistem kedua membuat dan mengkonfigurasi bagaimana sistem pertama membuat level tersebut. Kedua sistem memiliki desain yang berbeda. Sistem pertama berisi agen-agen yang membuat level yang pemain akan mainkan. Agen-agen tersebut berisi algoritma dan logika untuk membuat dan menempatkan platform dan tiles pada level tersebut. Algoritma dan logika di agen-agen tersebut mengikuti aturan Conway's "Game of Life" untuk membuat level yang pemain dapat selesaikan dengan baik. Sistem kedua membuat konfigurasi yang akan digunakan oleh sistem pertama. Bagian ini membantu sistem pertama untuk membuat level yang baik untuk dimainkan pemain, dimana sistem memiliki perilaku untuk melakukan pengecekan kualitas level berdasarkan input yang diberikan. Sistem ini melakukan pengecekan dengan menggunakan Ribon Baumtargen's Mario-playing agent, dimana itu adalah sebuah AI yang akan menyelesaikan level tersebut. AI tersebut akan memberikan feedback kepada sistem tentang level yang sudah dimainkan, dan sistem akan menggunakan informasi tersebut untuk membuat level yang lebih bagus.

Ada juga penelitian lain yang dilakukan tentang membuat level yang seimbang untuk game 3D paintball[17]. Penelitian ini berisi algoritma search-based procedural content generator yang dapat digunakan untuk membuat level atau map. Algoritma ini memiliki fokus untuk membuat level yang digunakan untuk tactical shooter game, jadi didesain untuk membuat level yang seimbang. Dalam penelitian ini, seimbang berarti tidak adanya area yang memberikan keuntungan untuk kedua tim yang melawan satu sama lain. Sistem ini menggunakan sistem AI untuk

membuat level tersebut. Sistem berisi formula dan algoritma yang menentukan lokasi-lokasi penting dalam sebuah level, seperti flanking route, defensiveness dan dispersion location. Lara-Carbera dkk. juga membuat implementasi eksperimen untuk membuktikan formula dan algoritma tersebut. Dari penelitian yang dilakukan oleh Lara-Carbera dkk., mereka mengambil konklusi bahwa membuat sistem procedural generation untuk membuat level yang bisa disesuaikan sesuai keinginan sangatlah mungkin. Mereka memberikan nota bahwa level yang dibuat bisa dibuat menjadi terlihat lebih baik, seperti membuat level menjadi lebih dinamis atau mengubah bagaimana level terlihat.

Selain itu, ada penelitian lain yang juga dilakukan mengenai procedural generation. Penelitian tersebut berupa analisis dan adaptasi procedural dungeon generation[18]. Penelitian ini menganalisa bagaimana procedural content generation digunakan didalam game. Pembuatan “dungeon” secara prosedural dilakukan sebagai contoh penggunaan algoritma yang dianalisa. Analisa adaptasi dilakukan untuk game proyek 2D dan 3D. Untuk proyek 2D, peneliti menganalisa lima algoritma yang dapat digunakan untuk proyek yang ada. Kemudian peneliti akannya mencoba menggunakan algoritma tersebut untuk proyek 3D, menggunakan game engine Unreal Engine 4[19]. Sistem ini membuat sebuah level 3D yang berisi ruangan dan koridor yang pemain dapat eksplorasi. Pertama, sistem pertama membuat area tersebut dalam bentuk 2D. Sistem membuat ruangan-ruangan yang akan digunakan oleh pemain. Kemudian, sistem membuat koridor-koridor yang ada dengan menyambungkan ruangan satu sama lain dan memilih ruangan yang ingin digunakan. Setelah informasi tersebut ada, sistem akan membuat level 3D berdasarkan informasi yang dibuat. Baron dkk. mengambil konklusi bahwa sistem procedural content generator sangatlah berguna untuk game. Level yang dibuat bisa diperbesar dan diperkecil sesuai dengan keinginan pengguna. Untuk kedepannya, penelitian tentang peletakkan musuh, pemain, dan lain-lain dapat dilakukan berdasarkan sistem ini.

Penelitian yang sama juga dilakukan dalam pembuatan 2D side-view platformer game, dimana level dibuat secara prosedural[20]. Penelitian ini memiliki fokus ke masalah teknis dan kreativitas dalam pembuatan level secara prosedural. Dari penelitian tersebut, ada beberapa spesifikasi yang harus dipenuhi ketika

membuat level secara prosedural; level harus bisa diselesaikan oleh pemain, tingkat kesulitan harus menjadi faktor dalam level yang dibuat, dan hasil desain harus berjalan dengan gameplay dengan benar. Penelitian ini dibuat untuk menyelesaikan masalah kebutuhan asset pada proses pembuatan game. Salah satu masalah tersebut adalah pembuatan sebuah level dalam pembuatan game. Pada penelitian ini, solusi yang diberikan adalah sistem procedural generation yang dapat membuat level secara prosedural. Dalam pembuatan level, pertama mereka memilih tipe level yang ingin dibuat. Ada beberapa tipe level yang mereka bisa buat, seperti tipe underwater, pyramid, factory, dan lain-lain. Setiap tipe level yang ada memiliki aturan procedural generation yang ada. Aturan tersebut ada untuk memastikan bahwa level yang dibuat tersebut sama dengan tipe level yang diinginkan. Dari aturan tersebut, implementasi dibuat dan dipastikan level yang dibuat mengikuti aturan tersebut. Validasi dilakukan oleh pemain, yang kemudian memberikan nilai tentang level-level yang telah mereka mainkan. Nilai-nilai tersebut digunakan untuk memberbaharui sistem procedural generation yang digunakan untuk membuat level tersebut.

Peneliti lain juga melakukan penelitian yang sama, dan merilis dokumen penelitian penggunaan dan implementasi procedural generation untuk membuat dunia game yang besar[21]. Penelitian ini berfokus kepada masalah yang ada ketika menggunakan procedural generation untuk membuat dunia game yang besar, dan solusi yang ada untuk menyelesaikan masalah-masalah tersebut. Masalah pertama adalah kurangnya memori yang ada untuk membuat konten yang banyak. Dalam pembuatan dunia game yang besar, tentu dibutuhkan ukuran memori yang besar untuk menyimpan data tersebut, sebelum dijadikan dunia yang dapat dimainkan oleh pemain. Penelitian ini membahas bagaimana cara meminimalisasikan penggunaan cpu dan memori dalam pembuatan dunia game yang besar. Masalah kedua adalah kebutuhan membuat struktur untuk struktur besar yang sudah ada, tanpa harus membuat struktur besar itu sendiri dalam waktu yang sama. Penelitian ini berisi desain, implementasi dan evaluasi general prototype framework yang dapat digunakan untuk membuat game world yang besar. Sebagai contoh kasus yang akan diselesaikan oleh solusinya, Carpenter membuat sistem procedural generation yang membuat sebuah galaxy, penuh dengan bintang dan planet-planet

yang berisi bangunan, gunung, dan lain-lain. Sistem tersebut berisi formula dan algoritma yang membuat asset-asset tanpa campur tangan manusia – selain konfigurasi yang diberikan. Dari situ, Carpenter melakukan analisa performa pada setiap algoritma yang digunakan. Dari analisa performa mereka, mereka berhasil hanya menggunakan memory kurang dari 1kB, dimana setiap bintang memiliki paling tidak 5 planet yang ada. Secara normal, pembuatan sebuah galaxy dimana semua bintang dan planet dihitung sebelumnya akan memakan memori sebanyak 102TB. Sebagai konklusi, Carpenter mengatakan bahwa sistem procedural generation sangatlah populer di industri game. Hal ini terjadi karena ada banyak kebutuhan untuk dunia game yang bagus dan mendetil. Meski begitu, dalam sistem procedural generation, ada masalah dengan tidak adanya standar struktur dan logika untuk sistem procedural generation besar. Penelitian yang dilakukan oleh Carpenter berguna untuk menyelesaikan masalah tersebut, dengan mendesain dan mengimplementasi sistem procedural generation untuk membuat dunia game yang sangat besar.

Ada penelitian yang dilakukan oleh peneliti lain yang melakukan penelitian tentang pembuatan dungeon secara otomatis dalam game komputer[22]. Penelitian ini tidak sepenuhnya masuk ke sistem procedural generation, tetapi penelitian ini membantu teori yang dibutuhkan untuk membuat sistem procedural generation. Peneliti Adams menggunakan teknik graph grammars untuk membuat level pada sebuah game. Dengan kolaborasi dengan Infogrames, mereka membuat sebuah sistem yang dapat membuat level secara otomatis. Sistem yang dibuat oleh peneliti Adams bekerja dengan membuat point bernama “node”, dan menyambungkan node tersebut dengan node lainnya. Menggunakan berbagai aturan pada koneksi node yang dibuat, mereka dapat membuat jalan antar area yang ada, dimana area tersebut adalah node, dan jalan adalah koneksi antar node yang dibuat. Penelitian ini juga melihat beberapa faktor dalam pembuatan level, seperti tingkat kesulitan dan masalah “pointless area”. Untuk tingkat kesulitan, mereka menyebutkan beberapa faktor yang dapat dimanipulasi untuk meningkatkan atau mengurangi tingkat kesulitan; jumlah musuh, health pack, puzzle dan posisi mereka dapat diubah sesuai dengan tingkat kesulitan. Masalah “pointless area” muncul ketika sebuah area tidak memiliki jalan kemanapun. Salah satu cara mudah untuk mengatasi masalah ini

adalah menaruh item seperti health pack, ammo dan lain-lain pada akhir area, untuk membuat pemain mau masuk ke area tersebut. Sesudah semua masalah diatasi, data node yang ada diproses lagi di dalam sistem untuk membuat sebuah level yang dapat digunakan oleh game. Peneliti Adams mengambil konklusi bahwa proyek sukses dilakukan. Hasil dari penelitian ini adalah metode yang baik untuk membuat struktur sebuah level dengan baik dan benar.

Dan untuk yang terakhir, terdapat penelitian lain yang melakukan penelitian yang sama, yang melakukan penelitian video game yang menggunakan procedural generation[23]. Penelitian ini meneliti bagaimana game tersebut mengimplementasikan procedural generation, dan bagaimana sistem tersebut digunakan dengan baik dalam konteks game tersebut. Penelitian ini juga berisi konsep workflow yang menggunakan procedural generation secara modular. Implementasi konsep ini juga dibuat, menggunakan Unity[24] sebagai game engine, dan digunakan untuk membuat contoh aplikasi. Contoh aplikasi itu dibuat untuk memberikan contoh apa yang mampu dilakukan oleh sistem tersebut. Hasil dari penelitian ini adalah sebuah plugin untuk game engine Unity, dimana plugin tersebut dapat membuat dunia procedural generation dengan baik. Sistem procedural generation yang dibuat bersifat modular, dimana pengguna dapat mengubah dan menggunakan sistem sesuai kebutuhan mereka.

Berdasarkan dari penelitian yang dilakukan sebelumnya oleh peneliti-peneliti lain, maka penulis dalam hal ini akan melakukan penelitian mengenai implementasi procedural level generation pada aplikasi game pyramid exploration.

Tabel 1: Perbandingan Pustaka

Peneliti	[17] Lara-Cabrera dkk.	[23] Freiberg	[16] Kerssemakers dkk.	[18] Baron	Kiting (2020)*
Metode	Pembuatan level 3D shooter secara prosedural.	Pembuatan level berdasarkan parameter yang diberikan (framework).	Pembuatan level untuk Super Mario secara prosedural.	Analisa beberapa algoritma procedural level generation 2D, dan eksperimentasi penggunaan algoritma yang sama untuk 3D.	Pembuatan game dimana level dibuat secara procedural.
Platform	PC, Mobile, Console	PC, Mobile, Console	PC	PC	PC
Konten	Presentasi dan dokumentasi search-based procedural content generator algorithm.	Berisi penjelasan dan contoh game yang menggunakan Procedural Content Generation. Juga konsep PCG modular.	Implementasi procedural level generator untuk Super Mario.	Analisa dan implementasi procedural content di game. Lima algoritma procedural content digunakan di UE4.	Game yang menggunakan procedural level generation.
Hasil	Game Demo	Unity Plugin (Framework)	Super Mario Game Demo	Algorithm, Game Demo	Algorithm, Library dan Game

*) Penelitian yang dilakukan.

BAB III

LANDASAN TEORI

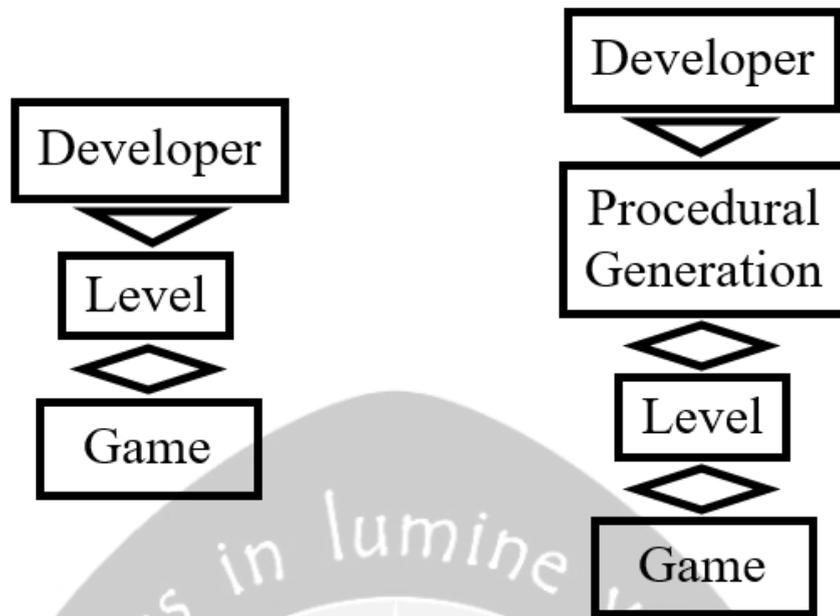
Pada bagian ini, akan dibahas beberapa teori yang mendukung dalam penelitian ini.

1. Procedural Content Generation

Procedural Content Generation didefinisikan sebagai sistem yang membuat konten atau asset yang dibuat secara otomatis, tanpa banyak campur tangan dari developer atau artist. Dalam konteks ini, prosedural berarti dilakukan menggunakan algoritma, dimana algoritma tersebut membuat sebuah konten atau asset dalam game[23]. Dalam game ini, Procedural Content Generation, atau juga bisa disebut Procedural Generation, digunakan untuk membuat dungeon area 3D, yang akan menjadi ruangan yang di eksplorasi oleh pemain.

Dalam konteks penelitian ini, penulis ingin membuat game eksplorasi pyramid, dimana setiap level yang dimainkan oleh pemain selalu berbeda. Penulis akan membuat sistem Procedural Generation yang dapat membuat level secara otomatis, menggunakan asset dasar yang dibuat oleh penulis.

Berbeda dengan membuat level secara manual, dimana developer harus membuat level untuk dimainkan oleh pemain di dalam game, sistem procedural generation dapat langsung membuat level secara otomatis. Developer hanya perlu memberikan instruksi sekali pada sistem procedural generation, dan sistem tersebut dapat terus membuat level yang berbeda, setiap kali game meminta level yang baru. Perbedaan ini diilustrasikan pada Gambar 1.



- Level dibuat manual
- Variasi tergantung jumlah level yang dibuat.
- Memakan waktu banyak jika membuat banyak level.
- Level dibuat otomatis
- Selalu acak, jadi selalu berbeda.
- Hanya perlu memasukkan instruksi pendek dan level langsung dibuat.

Gambar 1: Perbedaan pembuatan level secara manual vs menggunakan sistem procedural generation

Secara garis besar, sistem procedural generation yang akan dibuat hanya akan membuat informasi struktur level secara otomatis, ketika diminta oleh game, menggunakan konfigurasi dan algoritma yang ditentukan oleh developer. Game akan menggunakan informasi struktur level tersebut untuk membuat level yang dapat dimainkan oleh pemain. Level akan dikonstruksi oleh game menggunakan asset dasar yang dibuat oleh artist dan developer.

Sistem procedural generation hanya membuat informasi struktur level. Hal ini dikarenakan sistem yang dibuat akannya terpisah dari game. Sistem tersebut akan dibuat dalam bentuk library, dan kemudian digunakan oleh game dengan meng-import library tersebut. Sistem procedural generation dan game dibuat terpisah, karena sistem procedural generation yang dibuat terlalu besar untuk satu proyek, dan penulis ingin sistem procedural generation dapat digunakan di proyek lain.

Dalam penelitian ini, sistem procedural generation akan dibuat secara modular. Pertama, penulis akan melihat bagaimana level akan dibuat dalam game. Dari situ, penulis akan mendesain sistem procedural generation yang akan membuat informasi struktur level.

Sistem procedural generation berisi module-module berbeda, dimana setiap module memiliki implementasi algoritma berbeda yang dapat dikombinasikan untuk membuat data informasi struktur level. Sistem dibuat seperti ini untuk memudahkan pengembangan sistem procedural generation yang sudah dibuat.

Ketika sistem procedural generation ingin digunakan di dalam game, developer hanya perlu meng-import library ke dalam game. Di dalam game, developer hanya perlu memanggil fungsi yang sudah dibuat di dalam module-module yang sudah dibuat. Menggunakan sistem ini, game dapat meminta informasi struktur level dari library, dan menggunakan informasi tersebut untuk membuat level.

2. Game

Dalam konteks ini, game adalah video game. Video game adalah game elektronik, dimana pemain memberikan input berdasarkan apa yang terdapat di layar monitor[25]. Game yang akan dibuat adalah permainan eksplorasi piramid. Sebagai pemain, mereka harus melakukan eksplorasi piramid dan mencari jalan keluar dari piramid tersebut. Pemain hanya memiliki lampu cahaya senter untuk menerangi tempat pemain. Tema dari game ini adalah stealth dan exploration game.

3. Exploration Game

Exploration didefinisikan sebagai eksplorasi area yang sebelumnya tidak diketahui[26]. Exploration game adalah game dimana pemain dapat mengeksplorasi dunia game dengan bebas, dimana pemain bereksplorasi untuk menemukan area yang mereka belum pernah temui sebelumnya. Game ini memenuhi spesifikasi tersebut.

Salah satu contoh game eksplorasi yang ada adalah Skyrim[27] dan Fallout 3[28]. Kedua game tersebut memiliki tema eksplorasi, dimana pemain dapat melakukan eksplorasi di dunia yang dibuat oleh artist dan developer Bethesda Studios.



Gambar 2: Gambar Windhelm, salah satu lokasi di Skyrim.[29]

Game ini memiliki tema yang mirip, tapi dengan skala yang sangat kecil. Pemain harus mencari jalan keluar dari level tersebut. Setiap level yang pemain temui selalu berbeda, dan pemain bebas melakukan eksplorasi di dalam piramid.

4. Unity Game Engine

Unity adalah general-purpose game engine yang digunakan untuk membuat game. Game engine ini berguna untuk membuat game[24]. Unity digunakan karena game engine tersebut bersifat user-friendly. Scripting di Unity menggunakan C#, jadi ada kemudahan dalam penulisan kode. Semua hal yang dibutuhkan sudah ada di Unity, termasuk render engine, sound engine, dan lain-lain. Dengan begitu, developer dan *artist* tidak perlu bersusah-payah membuat game engine yang baru.

5. Steam

Steam adalah distributor video game yang memiliki fokus ke PC gaming[30]. Sebagai distributor, layanan yang diberikan adalah tempat diskusi, review dan beli video game. Game developer dapat menjual game mereka di Steam. Jika game sudah dibuat, jika situasi dan kualitas layak, maka game bisa dipublikasikan ke Steam untuk dimainkan oleh publik.

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, penulis menyimpulkan bahwa sistem procedural level generation telah berhasil implementasikan pada game eksplorasi piramid. Sistem procedural generation berhasil dibuat menggunakan berbagai algoritma dan formula, yang kemudian digunakan dalam game eksplorasi pyramid untuk membuat level secara acak. Sistem bekerja dengan baik dalam game, hal ini dibuktikan dengan kuesioner ke 30 responden yang penulis lakukan. Dari hasil kuesioner tersebut, penulis mengambil konklusi bahwa 17.3% user sangat senang dengan game, dan 44.7% user merasa senang dengan game eksplorasi piramid.

6.2. Saran

Ada juga saran dari penulis untuk pengembangan yang dapat dilakukan pada penelitian yang sudah dilakukan. Saran pengembangan tersebut diharapkan dapat membantu peneliti lain yang ingin melanjutkan penelitian yang sudah dilakukan oleh penulis.

1. Dalam pembuatan level di game, mungkin ada beberapa langkah yang bisa di tambahkan atau lewati, untuk memberikan hasil yang terlihat lebih baik. Sebagai contoh, game bisa menggunakan data kumpulan point koordinat untuk membuat level, daripada data grid dua dimensi. Perubahan ini membutuhkan perubahan drastis tentang bagaimana level dibuat, karena pembuatan level tidak lagi seperti membangun balok-balok.
2. Ada juga perubahan lain yang bisa digunakan untuk mengembangkan sistem procedural generation ini. Kita bisa menggunakan Gabriel Graph untuk membuat koneksi antar ruangan, yang akan digunakan untuk membuat koridor.
3. Selain itu, kita juga bisa mengembangkan apa yang sudah ada. Kita bisa menambahkan axis lokasi ruangan, dimana ruangan bisa di atas atau dibawah ruangan lain. Bagian ini dapat memperbesar lokasi level yang pemain dapat eksplorasi, dan membuat level menjadi lebih kompleks. Atau kita bisa menambah labirin di dalam ruangan yang sudah ada, untuk membuat eksplorasi lebih menyenangkan dan menegangkan.
4. Variasi ruangan bisa ditambahkan. Salah satu feedback yang diberikan user adalah “[...] game memiliki potensial. Game hanya membutuhkan lebih banyak konten dan variasi di labirin.” Sebagai pengembangan game ini, bisa ditambahkan variasi dalam level, seperti ruangan berisi air, koridor jebakan, dan lain-lain.
5. Ketika game diuji oleh publik, banyak feedback dari user tentang menambahkan konten dalam game. Konten bisa apa saja, seperti sistem puzzle, musuh yang berbeda, mekanik, dan lain-lain. Penulis merekomendasikan untuk menambah sistem puzzle, karena puzzle membutuhkan upaya yang besar ketika di implementasikan dalam level yang dibuat secara prosedural.

6. Bisa juga ditambahkan musuh pada game, seperti monster yang berjalan di dalam piramid. Dalam implementasi ini, musuh yang ada hanyalah turret yang mendeteksi, menghidupkan turret lain, dan menyerang pemain. Dalam level yang dibuat secara procedural, bagaimana musuh dapat berjalan dan berinteraksi di dalam level? Bagaimana cara membuat interaksi tersebut menarik? Penelitian ini dapat membantu implementasi musuh pada game yang menggunakan sistem procedural generation.



DAFTAR PUSTAKA

- [1] R. Van Der Linden, R. Lopes, and R. Bidarra, "Procedural generation of dungeons," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 1, pp. 78–89, 2014.
- [2] SideFX, "Houdini - SideFX." [Online]. Available: <https://www.sidefx.com/products/houdini/>. [Accessed: 09-Sep-2020].
- [3] Hello Games, "No Man's Sky." Hello Games.
- [4] H. Alexandra, "A Look At How No Man's Sky's Procedural Generation Works," *Kotaku*, 2016. [Online]. Available: <https://kotaku.com/a-look-at-how-no-mans-skys-procedural-generation-works-1787928446>. [Accessed: 09-Sep-2020].
- [5] Mojang, "Minecraft." [Online]. Available: <https://www.minecraft.net/>.
- [6] J. Peel, "Just how big is a Minecraft world? Big, as it turns out.," *PCGamesN*, 2015. [Online]. Available: <https://www.pcgamesn.com/minecraft/just-how-big-minecraft-world-big-it-turns-out>. [Accessed: 09-Sep-2020].
- [7] "Terraria," *Re-Logic*. [Online]. Available: <https://terraria.org/>. [Accessed: 09-Sep-2020].
- [8] T. M. Shea, "Terraria Review - The magical world of Terraria is bursting with terrific content and surprises to keep you invested for a long time.," *GameSpot*, 2014. [Online]. Available: <https://www.gamespot.com/reviews/terraria-review/1900-6316247/>. [Accessed: 09-Sep-2020].
- [9] Valve, "Left 4 Dead 2." Valve, 2009.
- [10] J. Walker, "Left 4 Dead 2: Exclusive RPS Hands-On Preview," *Rock Paper Shotgun*, 2009. [Online]. Available: <https://www.rockpapershotgun.com/2009/06/01/left-4-dead-2-exclusive-rps-preview/>. [Accessed: 09-Sep-2020].
- [11] PCGamesN, "How many Minecraft players are there?," *PCGamesN*, 2020. [Online]. Available: <https://www.pcgamesn.com/minecraft/minecraft-player-count>. [Accessed: 09-Sep-2020].
- [12] M. Handrahan, "Terraria reaches 30m units sold," *gamesindustry.biz*, 2020. [Online]. Available: <https://www.gamesindustry.biz/articles/2020-04-01->

- terraria-reaches-30m-units-sold. [Accessed: 09-Sep-2020].
- [13] J. Santos, “‘No Man’s Sky’: Widely Criticized Game Recovers, Successfully Improves,” *International Business Times*, 2019. [Online]. Available: <https://www.ibtimes.com/no-mans-sky-widely-criticized-game-recovers-successfully-improves-2784554>. [Accessed: 09-Sep-2020].
- [14] “itch.io.” [Online]. Available: <https://itch.io/>. [Accessed: 12-Dec-2020].
- [15] Google, “Google Forms.” [Online]. Available: <https://www.google.com/forms/about/>. [Accessed: 12-Dec-2020].
- [16] M. Kerssemakers, J. Tuxen, J. Togelius, and G. N. Yannakakis, “A procedural procedural level generator generator,” *2012 IEEE Conf. Comput. Intell. Games, CIG 2012*, vol. 1, pp. 335–341, 2012.
- [17] R. Lara-Cabrera, V. Rodríguez-Fernández, J. Paz-Sedano, and D. Camacho, “Procedural generation of balanced levels for a 3D paintball game,” *CEUR Workshop Proc.*, vol. 1957, no. 1, pp. 43–55, 2017.
- [18] J. Baron, “Procedural dungeon generation analysis and adaptation,” *Proc. SouthEast Conf. ACMSE 2017*, vol. 1, pp. 168–171, 2017.
- [19] Epic Games, “Unreal Engine 4.” [Online]. Available: <https://www.unrealengine.com/>.
- [20] A. C. D’íaz, “Procedural generation applied to a video game level design,” Barcelona School of Informatics, 2015.
- [21] E. Carpenter, “Procedural Generation of Large Scale Gameworlds,” University of Dublin, 2011.
- [22] D. Adams, “Automatic Generation of Dungeons for Computer Games,” p. 60, 2002.
- [23] S. Freiberg, “Procedural Generation of Content in Video Games,” Hamburg University of Applied Science, 2016.
- [24] Unity Technologies, “Unity.” [Online]. Available: <https://unity3d.com/unity>.
- [25] “Video Game - Definition,” *Merriam-Webster*. [Online]. Available: [https://www.merriam-webster.com/dictionary/video game](https://www.merriam-webster.com/dictionary/video%20game).
- [26] “Exploration - Definition,” *Oxford Lexico*. [Online]. Available: <https://www.lexico.com/en/definition/exploration>.

- [27] Bethesda Game Studios, “Elder of Scrolls V: Skyrim Special Edition,” 2016. [Online]. Available: https://store.steampowered.com/app/489830/The_Elder_Scrolls_V_Skyrim_Special_Edition/?curator_clanid=35501445.
- [28] Bethesda Game Studios, “Fallout 3,” 2008. [Online]. Available: https://store.steampowered.com/app/22300/Fallout_3/.
- [29] “Windhelm (Skyrim).” [Online]. Available: [https://elderscrolls.fandom.com/wiki/Windhelm_\(Skyrim\)](https://elderscrolls.fandom.com/wiki/Windhelm_(Skyrim)).
- [30] Valve, “Steam - About.” [Online]. Available: <https://store.steampowered.com/about/>.
- [31] “blender.org - Home of the Blender project - Free and Open 3D Creation Software.” [Online]. Available: <https://www.blender.org/>. [Accessed: 12-Nov-2019].
- [32] “Paint.NET.” [Online]. Available: <https://www.getpaint.net/>.
- [33] G. E. P. Box and M. E. Muller, “A Note on the Generation of Random Normal Deviates,” *Ann. Math. Stat.*, vol. 29, no. 2, pp. 610–611, 1958.
- [34] J. Loera, J. Rambau, and F. Santos, *Triangulations. Structures for algorithms and applications*, vol. 25. 2010.
- [35] L. J. Guibas, D. E. Knuth, and M. Sharir, “Randomized incremental construction of Delaunay and Voronoi diagrams,” *Algorithmica*, vol. 7, no. 1, pp. 381–413, 1992.
- [36] D. Sinclair, “S-hull: a fast radial sweep-hull routine for Delaunay triangulation,” no. d, pp. 1–7, 2016.