

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Game atau permainan adalah sebuah aktifitas yang interaktif, dimana satu atau lebih pemain mengikuti aturan yang membatasi perilaku mereka, memberlakukan konflik artifisial yang berakhir dengan hasil yang dapat diukur (Salen & Zimmerman, 2003). Menurut kamus Cambridge, “*video game* atau *computer game* merupakan permainan yang dimainkan pada komputer, yang dimana gambar muncul pada layar dapat dikendalikan dengan menekan tombol pada *keyboard* atau menggerakkan pengontrol atau konsol” (Cambridge English Dictionary, n.d.).

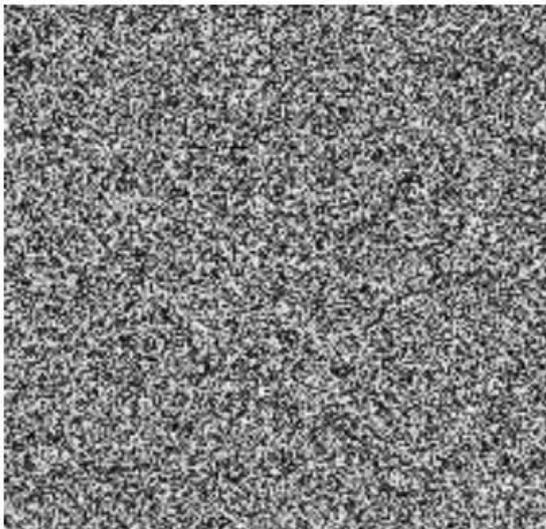
Game development adalah *software development* dimana membuat suatu software yang mempunyai seni, *audio*, cerita atau simulasi, dan *gameplay* yang interaktif (Bethke, 2003). Dalam membangun sebuah *game*, *game developer* akan membuat sebuah *terrain* atau level atau yang biasa disebut area permainan untuk menampilkan sebuah cerita, situasi, kondisi, dan suasana yang ada dalam permainan tersebut agar dapat menarik pemain untuk merasakan situasi yang ada dalam *game* yang telah dibuat.

Pemandangan alam atau bahkan dunia yang beragam disekitar kita selalu menjadi sorotan umat manusia. Berbagai upaya telah dilakukan *developer* untuk menggambarkan fenomena dunia sebagai area bermain dalam *game*. 3D *game* yang modern, simulasi, atau bahkan animasi seringkali diperlukan dalam membuat area yang sangat detail untuk memvisualisasikan pemandangan dalam sebuah *game* (Kříž, Spring 2019).

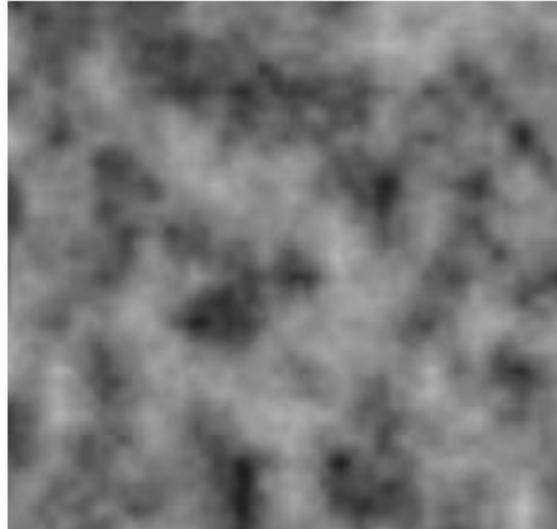
Tujuan utamanya adalah untuk membuat area permainan atau *terrain* yang realistis secara visual, dimana memiliki kriteria yang realistis seperti : tinggi, lekukan pada lereng dan retakan pada area permainan. Melakukan hal – hal ini secara manual sangatlah lama, mahal dan membuat file memiliki ukuran yang besar. Semuanya masalah ini dapat diminimalkan dengan menggunakan *Procedural Content Generation*, dimana merupakan suatu tren yang jelas dalam *game* dan simulasi untuk menghasilkan dunia sebanyak mungkin (Gollent, Landscape Creation and Rendering in REDEngine 3, 2014).

Algoritma Perlin Noise merupakan salah satu teknik dari *Procedural Content Generation* yang sering digunakan untuk setiap tahap dari generasi suatu *terrain* atau area permainan. Teknik ini dipilih karena layak diterapkan dalam kurun waktu tugas akhir ini dan waktu implementasi sangat penting dalam pengembangan *game*. Terdapat banyak jalan untuk menghasilkan suatu *terrain* secara procedural, namun banyak dari algoritma tersebut yang sangat rumit sehingga seringkali memakan waktu implementasi yang lama (Olsson & Frank, 2017).

Perlin Noise menghasilkan *noise* dengan perilaku properti yang dapat diimplementasikan lebih alami dan saling berelasi. *Noise* ini memiliki bentuk alami, membuat algoritma ini sangat berguna pada saat menghasilkan *terrain* atau area permainan yang terlihat natural dimana biasanya digunakan pada *game* petualangan, *sandbox game*, dan *open world game*.



Gambar 1.1 Contoh struktur Random Noise



Gambar 1.2 Contoh struktur Perlin Noise

Gambar 1.1 dan 1.2 menunjukkan bahwa perbandingan struktur *behavior* antara *Random Noise* (Gambar 1.1) dengan Perlin Noise (Gambar 1.2) yang sangatlah berbeda. Dimana *Random Noise* memiliki masalah, yaitu tidak halus, dan tidak saling berelasi. Namun tidak dengan Perlin Noise, dimana sangatlah halus dan setiap *noise* saling berelasi. Sehingga membuat Perlin Noise sering digunakan untuk membuat tekstur (2D) acak, sampai dengan *Procedural Terrain Generation* (2D/3D) dalam pembuatan *game*.

1.2 Rumusan Masalah

Bagaimana melakukan implementasi algoritma *Perlin Noise* pada *game* berjenis *Adventure* dengan subjenis *Open World* dan berfitur *Sandbox*, dengan judul “*Let’s Mining*” pada platform *Personal Computer*.

1.3 Tujuan dan Manfaat

Membuat Implementasi *LET’S MINING* dengan *Procedural Terrain Generation* menggunakan Algoritma *Perlin Noise* dengan platform *Personal Computer* menggunakan *Unity Game Engine*. Serta sebagai sarana hiburan bagi pengguna untuk mengisi waktu luang.

1.4 Ruang Lingkup

Dalam pembuatan Tugas Akhir yang terarah dan terperinci agar mendapatkan hasil yang spesifik, maka sistem yang dirancang dibatasi ruang lingkup sebagai berikut :

1. Sistem dikembangkan dengan metode *Agile Development*.
2. Sistem dikembangkan dengan bantuan dari *AI Perlin Noise*.
3. Sistem dirancang agar user dapat memainkan *game* ini tanpa adanya penurunan dari *FPS (Frame Per Second)* dengan menggunakan *library Threading* yang ada dalam *game engine Unity*.
4. Mempunyai beberapa bioma (*biomes*) yang berbeda-beda.

1.5 State Of The Art

State of the art adalah analisa penelitian yang sudah dilakukan sebelumnya. Untuk mengetahui sejauh mana penelitian sebelumnya dilakukan dan menjadi acuan dari penelitian sekarang.

Tabel 1.1 State of The Art untuk Implementasi game pada TA ini

No.	Judul (DOI/URN)	Ringkasan	Perbedaan
1.	<p><i>Improving Perlin Noise</i> 10.1145/566654.566636</p>	<p>Artikel tersebut membahas tentang perubahan yang dilakukan oleh Ken Perlin pada noise temuannya untuk membuat tekstur yang dihasilkan lebih <i>smooth</i> dan alami.</p>	<p>Perbedaan artikel tersebut dengan penelitian ini adalah implementasi dari adaptasi <i>Improved Perlin Noise</i> pada <i>terrain</i> 3D menggunakan library dari <i>Unity Game Engine</i>.</p>
2.	<p><i>Procedural city generation using Perlin noise</i> urn:nbn:se:bth-14855</p>	<p>Artikel tersebut membahas tentang penggunaan Perlin Noise untuk mensimulasi generasikan kota secara virtual agar sesuai dengan kota pada dunia nyata secara <i>random</i>.</p>	<p>Perbedaan artikel tersebut dengan penelitian ini adalah Implementasi pada <i>game Let's Mining</i> mengenerasikan bioma yang dibuat menyerupai bentuk alami dari alam secara <i>random</i>.</p>
3.	<p><i>Multithreading Handbook for Simulation Developers</i></p>	<p>Artikel yang diterbitkan pada jurnal <i>roadtomr</i> yang diterbitkan tahun 2020 tersebut berisi tentang pengenalan dan cara</p>	<p>Perbedaan artikel tersebut dengan penelitian ini adalah mengadaptasi sistem <i>Multi-threading</i> pada <i>game Let's</i></p>

		<p>untuk implementasi sistem <i>Multi-threading</i> dalam <i>Unity Game Engine</i>.</p>	<p><i>Mining</i> namun dengan menggunakan <i>Threading system</i> pada <i>library</i> dari <i>Dot Net 5.0</i>.</p>
4.	<p><i>Multi – Fractal Terrain Generation</i></p>	<p>Artikel yang diterbitkan pada jurnal <i>Masaryk University</i> (via <i>researchgate.net</i>) yang diterbitkan pada tahun 2019 tersebut berisi tentang analisis dari berbagai macam interpolasi untuk membangun <i>noise</i> dan perbandingan pada masing – masing <i>noise</i> yang telah ditemukan.</p>	<p>Perbedaan artikel tersebut dengan penelitian ini adalah menggunakan salah satu <i>noise system</i> yang telah ada yaitu menggunakan Perlin Noise.</p>
5.	<p><i>Application of the Perlin Noise Algorithm as a Track Generator in the Endless Runner Genre Game</i> (10.1088/1742-6596/1255/1/012064)</p>	<p>Artikel tersebut berisi tentang pembuatan game dengan <i>genre Endless Runner</i> dan menggunakan Algoritma Perlin Noise sebagai dasarnya.</p>	<p>Perbedaan artikel tersebut dengan penelitian ini adalah <i>genre</i> dari pada game ini yaitu Adventure dengan fitur <i>Open Word</i> dan <i>sandbox</i>.</p>

6.	<i>Teaching Scientific Concepts using a Virtual World - Minecraft</i>	Artikel yang diterbitkan pada jurnal <i>Robert Morris University</i> (via <i>researchgate.net</i>) yang diterbitkan tahun 2012 tersebut berisi tentang pembelajaran Ilmu Pengetahuan Alam yang ada pada dunia nyata dan disimulasikan pada dunia game.	Perbedaan artikel tersebut dengan penelitian ini adalah sebagai referensi bioma dan grafitasi untuk mengembangkan game ini.
----	---	---	---

1.6 Sistematika Penulisan

Secara garis besar penulisan laporan Tugas Akhir ini meliputi beberapa bab diantaranya adalah:

BAB 1 PENDAHULUAN

Pada bab ini berisi tentang latar belakang, rumusan masalah, tujuan, ruang lingkup, metodologi penelitian, *state of the art* dan sistematika penulisan laporan Tugas Akhir.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini berisi tentang teori dasar yang berhubungan dengan judul Tugas Akhir seperti algoritma dari Perlin Noise, model *Agile*, dan lain-lain.

BAB 3 METODE

Berisi tentang analisis kebutuhan sistem fungsional, *non* fungsional dan perancangan sistem.

BAB 4 PEMBAHASAN

Pada bab ini berisi tentang implementasi dari analisis dan desain yang telah dibuat dengan menggunakan bahasa pemrograman *C#* dan pengujian sistem.

BAB 5 PENUTUP

Pada bab ini berisi tentang kesimpulan dari Tugas Akhir yang telah dibuat dan saran terhadap aplikasi yang telah dibuat.