

## **BAB II**

### **TINJAUAN PUSTAKA**

Dalam pembuatan game, terdapat beberapa penelitian yang dilakukan untuk membuat game utuhnya. Tetapi ada juga yang memberikan hasil akhir berupa alat untuk membantu proses pengembangan gamenya. Dalam tinjauan pustaka ini, penulis akan menjabarkan beberapa penelitian yang dilakukan oleh peneliti terdahulu yang berkaitan dengan permasalahan yang ingin diselesaikan oleh penulis, maupun yang berkaitan dengan penggunaan metode *procedural generation* sendiri serta pengembangan game.

Pengamatan pertama merupakan pengamatan yang dilakukan oleh Albert. Pada pengamatannya, Albert ingin membuat game RPG berbasis dekstop menggunakan *Unreal Engine Blueprint*. *Blueprint* sendiri merupakan sarana pengganti *coding* yang lebih umum dikenal dengan istilah *visual scripting*. Pada bagian akhir penelitiannya, Albert sudah berhasil membuat game RPG berbasis dekstop dengan menggunakan *Unreal Engine*. Namun, Albert menemui permasalahan dimana *world map* / latar tempat yang diberikan masih terlalu kecil. Dengan begitu, Albert memberikan usulan untuk merancang *world map* / level yang lebih besar dari sebelumnya [6]. Berdasarkan usulan tersebut, penulis menemukan solusi untuk membuat sistem pembuatan level otomatis yang bukan hanya bisa lebih besar, tetapi fleksibel sehingga *developer* bisa dengan mudah mengatur ukuran dari *mapnya*.

Pengamatan selanjutnya merupakan pengamatan yang dilakukan oleh Mungky beserta teman-temannya. Pengamatan yang mereka lakukan mencakup tentang pengaruh *unreal engine* dalam perkembangan dunia game. Pada pengamatannya, mereka juga membahas tentang perbedaan *unreal engine*

4 dengan *unreal engine 5*. Mereka juga mencakup beberapa bagus yang disediakan dalam *Unreal Engine 5*. Namun, *unreal engine 5*, memiliki kelemahan yaitu memiliki grafis yang sangat baik sehingga tidak direkomendasikan untuk game yang ditujukan untuk platform *mobile* [7]. Berdasarkan pengamatannya, maka untuk penelitian yang akan dilakukan oleh penulis, akan menggunakan *unreal engine 4* demi mendapatkan sistem yang lebih ringan sehingga dapat digunakan oleh berbagai platform. Namun, algoritmanya tetap akan bisa diterapkan dalam *unreal engine 5*.

Pengamatan selanjutnya dilakukan oleh Mochamad Yusuf beserta temannya. Pengamatan yang dilakukan yaitu dengan membuat game yang berjudul “The Meet” yang merupakan game *first person* yang memiliki *genre horror*. Hasil dari pengamatannya, game berhasil dibuat sesuai dengan ekspektasi para *developernya*. Mereka juga memberikan beberapa saran, dimana salah satu dari saran tersebut yaitu untuk menambahkan jumlah *level* dalam game yang mereka buat [8]. Untuk game *horror* seperti itu, variasi pada *level* juga akan mempengaruhi bagaimana pengalaman pemain saat memainkan gamenya.

Pengamatan selanjutnya merupakan pengamatan yang dilakukan oleh Immiel. Pengamatan ini dilakukan untuk mengimplementasi *procedural level generation* untuk aplikasi game *pyramid exploration*. Pengamatan ini dilakukan menggunakan *game engine Unity* dengan bahasa pemrograman *c#*. Objek yang dibuat menggunakan metode *procedural generation*nya merupakan objek ruangan. Sedangkan, hasil akhir dari pengamatan ini berupa sebuah game yang *levelnya* digenerasi secara otomatis [4].

Pengamatan selanjutnya, diamati oleh Muhammad Hanif Atthariq beserta temannya. Mereka juga mengamati untuk *procedural content generation* yang digunakan menggunakan *game engine unity*. Bahasa pemrograman yang digunakan yaitu bahasa pemrograman *C#*. Objek yang dibuat dengan metode ini yaitu sebuah ruangan dua dimensi yang serupa dengan *dungeon*. Hasil akhir dari pengamatan ini yaitu sebuah game dengan *map* yang terdiri atas kumpulan

ruangan yang terbuat secara otomatis, dengan musuh dan beberapa komponen didalamnya [9].

Pengamatan selanjutnya merupakan pengamatan yang diamati oleh Hansen Krisna Putra dan juga Liliana. Mereka berdua mengamati tentang penerapan *procedural content generation* untuk generasi *level* pada game mereka yang berjudul *Mythical Maze*. Pengamatan mereka dilakukan menggunakan *game engine unity* dan menggunakan bahasa pemrograman C#. Objek yang dibuat adalah kumpulan ruangan yang terkoneksi satu sama lain secara acak. Hasil akhir dari pengamatan mereka berupa sebuah game yang bisa memilih tingkat kesulitannya, dimana tingkat kesulitan tersebut akan menentukan bagaimana *level* akan digenerate. Selain itu, gamenya juga bisa menggenerate latar tempat berupa dua dimensi dan juga tiga dimensi [10].

Pengamatan selanjutnya, dilakukan oleh William Hamilton. Pengamatannya mencakup tentang pembuatan game *level* tiga dimensi dengan arsitektur interior menggunakan metode *procedural generation*. Pengamatan yang dilakukan William menggunakan *game engine unreal engine 4* dengan bahasa pemrograman C++. Objek yang ia buat dalam pengamatannya yaitu sebuah bangunan dengan interior yang digenerate secara otomatis. Hasil akhir dari penelitiannya, ia berhasil membuat suatu alat yang bisa membuat bangunan dengan interior secara otomatis [11].

Pengamatan selanjutnya merupakan pengamatan yang diamati oleh Gilbert Nwankwo dan temannya. Mereka mengamati tentang pembuatan desain level dan tingkat kesulitan yang dinamis dalam game dua dimensi menggunakan *procedural generation*. Pengamatan mereka dilakukan dengan *game engine unity* dan bahasa pemrograman C#. Objek yang mereka buat yaitu pembuatan level dengan komponen-komponennya yang terbuat otomatis. Pada hasil akhir pengamatannya, mereka membuat sebuah game dengan desain level / latar tempatnya yang dinamis beserta tingkat kesulitannya juga [12].

Penelitian yang dilakukan oleh penulis, memiliki beberapa perbedaan dengan penelitian yang sudah disebutkan sebelumnya. Dalam penelitian yang dilakukan penulis, untuk menggenerate level, *procedural generation* juga

dikombinasikan dengan algoritma *random walk* yang dimodifikasi untuk menggenerasi jalur. Selain itu, kombinasi genre yang dipilih oleh penulis juga berbeda dengan kombinasi genre dari penelitian lainnya dimana *genre* yang dipilih yaitu *puzzle* dan kasual. Selain itu, jenis level serta cara bermain dari game yang dikembangkan penulis juga berbeda. Perbandingan penelitian yang akan dilakukan dengan peneliti terdahulu, dapat dilihat pada tabel berikut:



<b>Pembanding</b>	<b>Immiel Nicholas Kiting [4]</b>	<b>Muhammad Hanif Atthariq, dkk [9]</b>	<b>Hansen Krisna Putra dan Liliana [10]</b>	<b>William Hamilton [11]</b>	<b>Gilbert Nwankwo, dkk [12]</b>	<b>Yotam Niki Rahardja (Penulis)</b>
Metode	PG (Procedural Generation)	PG & Game Development Life Cycle	PG:Random Room Placement & Random Point Connect	PG: Room Generation	PG	PG & Random Walk Algorithm
<i>Game Engine</i>	<i>Unity</i>	<i>Unity</i>	<i>Unity</i>	<i>Unreal Engine 4</i>	<i>Unity</i>	<i>Unreal Engine 4</i>
<i>Genre</i>	<i>Puzzle, Exploration</i>	<i>Rogue-Like</i>	<i>Puzzle</i>	<i>- (Keluaran bukan berupa game)</i>	<i>Strategy</i>	<i>Puzzle, Casual</i>
Dimensi	3D	2D	2D / 3D	3D	2D	3D

Bahasa Pemrograman	<i>C#</i>	<i>C#</i>	<i>C#</i>	<i>C++</i>	<i>C#</i>	<i>C++</i>
Output Penelitian	<i>Game 3D dan Library</i>	<i>Game 2D</i>	<i>Game 2D dan 3D</i>	<i>Level Generator</i>	<i>Game 2D</i>	<i>Game 3D</i>

*Tabel 1.1 Tabel Perbandingan Penelitian*

