

# Implementasi Generate Map dan Pemunculan Objek secara Acak pada Game 3D menggunakan Bahasa C# dan Metode Perlin Noise di Unity: Studi Kasus pada Game Development

Renita Selviana<sup>1)</sup>, Dauw Bastha Fiastat Lugata<sup>2)</sup>, Alimin<sup>3)</sup>

Fakultas Teknologi Informasi, Institut Teknologi dan Bisnis Yadika, Pasuruan, Indonesia

email: [renita.selvi@itbyadika.ac.id](mailto:renita.selvi@itbyadika.ac.id), [ata040602@gmail.com](mailto:ata040602@gmail.com), [alimin@itbyadika.ac.id](mailto:alimin@itbyadika.ac.id)

## Abstrak

*This research explores the implementation of procedural content generation (PCG) techniques using C# programming language and Perlin noise method in Unity to generate map and random object spawning in 3D games. The main objective of this research is to develop PCG techniques that can be applied to game development. The Perlin noise algorithm is used to generate noise on the map surface, creating natural effects in 3D games. In this research, a C# script code has been implemented in Unity that can generate maps and spawn random objects in the game. The research results demonstrate that the implementation of map generation and random object spawning techniques can enhance gameplay quality and provide a more immersive playing experience for users. This research aims to contribute to game development on the Unity platform as well as the development of map generation and random object spawning techniques in 3D games using Perlin noise method.*

**Keywords :** Map generation, Random object spawning, Perlin noise, Unity, C#, Procedural Content Generation, Game Development.

## 1. Pendahuluan

Game development merupakan salah satu bidang yang terus berkembang dan menjadi semakin menarik bagi para pengembang dan pengguna game. Salah satu aspek penting dalam game development adalah pembuatan lingkungan atau environment yang dapat memberikan pengalaman bermain yang menyenangkan dan menarik bagi pemain. Pembuatan lingkungan dalam game development dapat dilakukan secara manual, namun hal tersebut dapat memakan waktu yang sangat lama dan membutuhkan sumber daya yang cukup besar. Oleh karena itu, para pengembang game menggunakan teknik Procedural Content Generation (PCG) untuk menghasilkan lingkungan dalam game secara otomatis.

Salah satu teknik PCG yang populer digunakan dalam pembuatan lingkungan game adalah algoritma Perlin Noise [1]. Algoritma ini dapat menghasilkan nilai-nilai acak yang dapat digunakan untuk membuat tampilan visual yang realistis pada game. Perlin Noise juga dapat digunakan untuk menghasilkan map atau peta secara otomatis, sehingga dapat mempercepat proses pengembangan game.

Dalam studi kasus yang dilakukan oleh mereka menggunakan algoritma Perlin Noise untuk

membuat map pada game membangun kerajaan [1]. Selain itu, melakukan analisis perbandingan antara Perlin Noise dan Simplex Noise untuk penciptaan permukaan daratan pada pembuatan game [2]. Mereka juga menggunakan algoritma Perlin Noise untuk melakukan Procedural Content Generation pada game world exploration sandbox [3]. Sedangkan mereka mengimplementasikan algoritma Perlin Noise untuk membuat Island Generator pada game open world [4].

Dalam penelitian ini, saya akan mengimplementasikan teknik PCG menggunakan algoritma Perlin Noise pada game 3D dengan menggunakan bahasa pemrograman C# dan Unity. Saya akan membuat sistem generasi map secara otomatis menggunakan algoritma Perlin Noise, serta memunculkan objek secara acak pada map yang telah dihasilkan. Tujuan dari penelitian ini adalah untuk memberikan pengalaman bermain yang lebih menarik dan realistis bagi pemain game.

## 2. Kajian Teori

### 2.1 Perlin Noise

Pembuatan peta pada game merupakan proses pembuatan peta menggunakan algoritma tertentu yang menghasilkan peta yang realistis dan menarik [1]. Algoritma yang umumnya

digunakan dalam pembuatan peta dapat bervariasi, seperti Cellular Automata, Diamond Square, atau Simplex Noise. Namun, salah satu algoritma yang populer digunakan dalam pembuatan peta pada game adalah algoritma Perlin Noise [2].

Algoritma Perlin Noise pertama kali diperkenalkan oleh Ken Perlin pada tahun 1985 [3]. Algoritma ini digunakan untuk menghasilkan nilai-nilai acak yang dapat digunakan untuk membuat tampilan visual yang realistis pada game. Algoritma Perlin Noise bekerja dengan cara menghasilkan pola acak yang memiliki konsistensi dan kehalusan yang dapat digunakan untuk membuat permukaan seperti air, tanah, atau awan pada game. Algoritma ini juga dapat digunakan untuk membuat peta secara otomatis dengan menghasilkan nilai-nilai acak yang diinterpretasikan sebagai elevasi pada permukaan peta.

Algoritma Perlin Noise sangat populer digunakan dalam pembuatan lingkungan game dan dapat digunakan untuk membuat lingkungan yang realistis dan menarik bagi pemain game. Selain itu, algoritma Perlin Noise juga dapat digunakan untuk membuat map atau peta secara otomatis, sehingga dapat mempercepat proses pengembangan game. Algoritma Perlin Noise juga dapat digunakan bersamaan dengan teknik PCG (Procedural Content Generation) untuk menghasilkan lingkungan game yang variatif dan tidak terduga.

## 2.2 Procedural Content Generation

Procedural generation adalah teknik yang digunakan dalam pengembangan game untuk membuat konten seperti lanskap, peta, dan level game secara dinamis dan efisien. Teknik Procedural Content Generation (PCG) digunakan untuk mengurangi beban kerja tim pengembangan. PCG melibatkan penggunaan algoritma untuk generasi konten game dengan kontribusi manusia yang terbatas atau tidak ada. [7]. Salah satu algoritma yang paling umum digunakan dalam procedural generation adalah Perlin noise, yang merupakan jenis gradient noise.

Beberapa penelitian telah dilakukan untuk menjelajahi penggunaan Perlin noise dalam pengembangan game. Penggunaan Perlin noise untuk menghasilkan peta dalam game membangun kerajaan [1]. Perbandingan penggunaan Perlin noise dan simplex noise untuk membuat permukaan tanah dalam game [2]. Pengeksplorasi penggunaan Perlin noise dalam menghasilkan konten untuk game

eksplorasi dunia pasir [3]. mempergunakan Perlin noise untuk menghasilkan pulau-pulau dalam game open-world [4]. Membelajari penggunaan procedural generation dengan batasan untuk menciptakan lingkungan dan medan alam [4].

Penelitian lain telah menjelajahi penggunaan Perlin noise dalam genre game tertentu fokus pada procedural generation dalam game dungeon [5], sementara menggunakan Perlin noise untuk menghasilkan trek untuk game endless runner [8]. menggunakan Perlin noise dan peta pengaruh untuk menciptakan generasi prosedural untuk game tower defense [6].

Selain menjelajahi penggunaan Perlin noise, beberapa penelitian juga menyelidiki teknik untuk meningkatkan efektivitasnya. mempelajari frekuensi filtering spektrum untuk Perlin noise [9], sementara penelitian yang lain menjelajahi penggunaan simplex noise untuk menciptakan peta alam yang natural di Unity Engine [10].

Secara keseluruhan, penelitian-penelitian ini menunjukkan keluwesan dan kegunaan Perlin noise dalam pengembangan game, dan upaya-upaya berkelanjutan untuk meningkatkan efisiensi dan efektivitasnya.

## 3. Metode Penelitian

Metode penelitian yang akan digunakan dalam penelitian ini adalah sebagai berikut :

### 3.1. Studi Literatur

Melakukan kajian teori tentang algoritma Perlin Noise, procedural content generation, pembuatan map, spawning objek secara acak pada game 3D, dll. Kemudian hasil kajian teori tersebut akan digunakan sebagai acuan dalam penelitian ini.

### 3.2. Pengumpulan Data Assets Gratis dan Perangkat Lunak

Memilih perangkat lunak yang diperlukan dan digunakan dalam penelitian berupa software seperti Unity, Visual Studio, dll. Sedangkan data yang digunakan adalah data assets gratis yang berhubungan dengan penelitian ini.

### 3.3. Analisis

Melakukan analisis terhadap kajian teori yang telah didapatkan untuk menentukan metode yang akan diimplementasikan dalam penelitian ini dalam menghasilkan map dan memunculkan objek secara acak pada game 3D menggunakan

bahasa pemrograman C# dan metode Perlin Noise di Unity.

### 3.4. Perancangan Sistem

Membuat rancangan sistem map generation dan spawning objek secara acak pada penelitian ini berdasarkan hasil analisis dengan menggunakan Unity, Visual Studio dan bahasa pemrograman C#.

### 3.5. Implementasi Sistem

Melakukan implementasi sistem map generation dan spawning objek secara acak pada Unity dengan menulis source code menggunakan bahasa pemrograman C#.

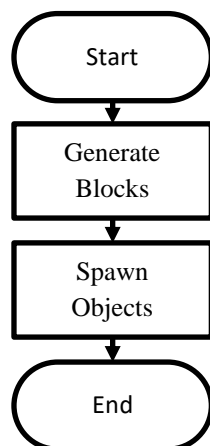
### 3.6. Evaluasi Sistem

Melakukan evaluasi sistem map generation dan spawning objek secara acak yang telah dibangun untuk menilai kelebihan dan kekurangan dari sistem yang dibangun serta memberikan saran perbaikan untuk pengembangan sistem selanjutnya.

## 4. Hasil Uji Coba Dan Pembahasan

### 4.1. Analisis Desain Program Berdasarkan Flowchart

Flowchart dibawah ini menggambarkan bahwa script GenerateWorld akan menjalankan method GenerateBlocks dan SpawnObjects saat dijalankan, dan tidak akan menjalankan method RefreshWorld. Method GenerateBlocks akan menghasilkan dunia 3D yang terdiri dari blok-blok, sementara method SpawnObjects akan men-spawn objek secara acak di dalam dunia tersebut. Kedua method ini akan dijalankan saat script GenerateWorld dijalankan.



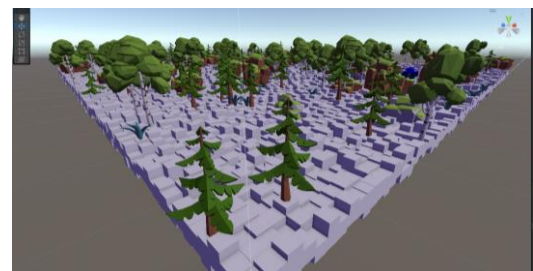
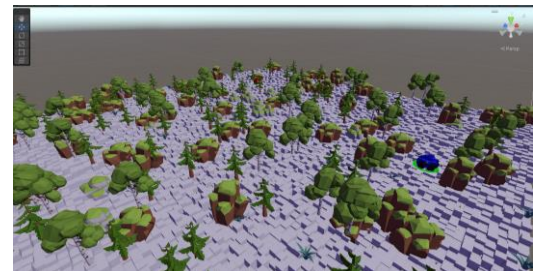
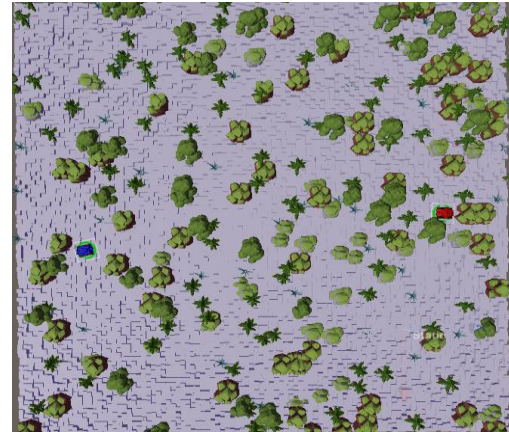
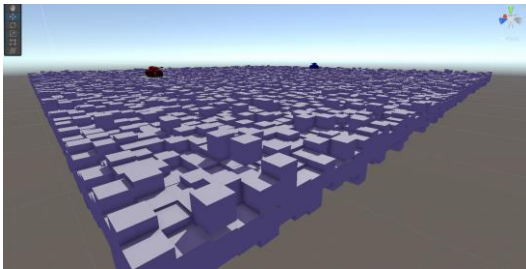
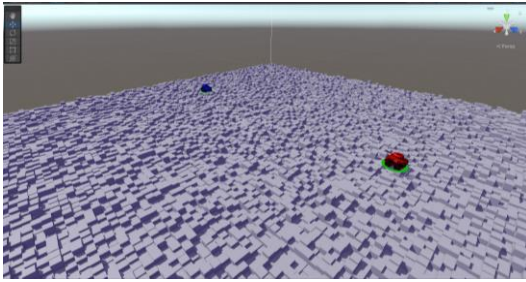
- a) Start: Program dimulai.
- b) GenerateBlocks(): Fungsi ini akan membangun dunia 3D dengan blok-blok yang dihitung menggunakan looping untuk setiap blok dalam arah X dan Z. Setiap blok akan diberi posisi dan diinstansiasi sebagai GameObject dan dimasukkan ke dalam dunia. Hasil akhirnya adalah daftar dari posisi blok yang tersimpan dalam List.
- c) SpawnObjects(): Fungsi ini akan men-spawn objek secara acak dalam dunia dengan menggunakan array dari GameObject yang telah diinisialisasi sebelumnya. Untuk setiap GameObject, 10 instance akan dibuat dengan posisi yang dihitung secara acak berdasarkan daftar posisi blok yang tersedia. Objek-objek tersebut akan diinstansiasi sebagai GameObject dan dimasukkan ke dalam daftar objek yang telah di-spawn.
- d) End: Program berakhir.

### 4.2. Implementasi Generate Map menggunakan Perlin Noise

Untuk mengimplementasikan generasi peta menggunakan Perlin Noise, saya menggunakan Unity sebagai platform pengembangan dan C# sebagai bahasa pemrograman. Berikut adalah langkah-langkah yang saya lakukan dalam proses implementasi:

1. Membuat terrain pada scene Unity.
2. Menambahkan script C# baru dengan nama GenerateWorld.cs.
3. Menulis kode pada script tersebut untuk menghasilkan peta berdasarkan Perlin Noise.

Setelah menambahkan script saya ke terrain dan mengisi X dan Z nya 100, saya mendapatkan peta yang dihasilkan secara otomatis berdasarkan Perlin Noise. Berikut adalah contoh hasil peta yang dihasilkan:



#### 4.3. Implementasi Pemunculan Objek secara Acak

Setelah berhasil menghasilkan peta, saya melanjutkan dengan implementasi pemunculan objek secara acak pada peta. Saya menggunakan script C# yang sama dengan nama function `SpawnObjects()`. Berikut adalah langkah-langkah yang saya lakukan:

1. Menambahkan Empty GameObject pada scene Unity dan menambahkan script function `SpawnObjects()`.
2. Menulis kode pada script tersebut untuk memunculkan objek secara acak pada peta.

Dalam script saya menggunakan array `objectToSpawn()` untuk menyimpan objek yang ingin saya munculkan secara acak pada peta. Saya juga menentukan jumlah objek yang ingin dihasilkan melalui variabel `spawnedObjects`. Kemudian, saya menggunakan metode `SpawnObjects()` untuk memunculkan objek secara acak pada peta. Setelah menambahkan function script ini, saya berhasil memunculkan objek secara acak pada peta. Berikut adalah contoh hasil objek yang dimunculkan secara acak:

#### 5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa implementasi teknik Procedural Content Generation (PCG) menggunakan bahasa pemrograman C# dan metode Perlin Noise di Unity dapat meningkatkan kualitas gameplay dan memberikan pengalaman bermain yang lebih mendalam bagi pengguna.

Teknik generasi map dan pemunculan objek secara acak yang digunakan dalam teknik PCG dapat menciptakan variasi dan kejutan dalam gameplay, sehingga membuat pengguna semakin tertarik dalam bermain game. Selain itu, implementasi teknik PCG juga dapat mempercepat proses pengembangan game karena konten yang dihasilkan secara otomatis.

Dengan demikian, hasil penelitian ini dapat memberikan kontribusi pada pengembangan game di platform Unity serta pengembangan teknik generasi

map dan pemunculan objek secara acak pada game 3D dengan menggunakan metode Perlin Noise. Oleh karena itu, penggunaan teknik PCG diharapkan dapat meningkatkan kualitas game dan memberikan pengalaman bermain yang lebih baik bagi pengguna. pendahuluan serta saran-saran untuk penelitian lanjutan jika diperlukan sesuai dengan hasil penelitian

## 6. Daftar Pustaka

- [1] Hertanto Wijaya, J. G., Andjarwirawan, J., & Adipranata, R. (2019). Game Membangun Kerajaan dengan Procedural Generated Map Menggunakan Perlin Noise. *Jurnal Infra*; Vol 7, No 1 (2019); 228-231. <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/8076>
- [2] Wijaya, W., & Rahman, A. (2018). Analisis Perbandingan Perlin Noise Dan Simplex Noise Untuk Penciptaan Permukaan Daratan Pada Pembuatan Game. *Konferensi Nasional Sistem Informasi (KNSI) 2018*; 2018: Konferensi Nasional Sistem Informasi (KNSI) 2018. <http://jurnal.atmaluhur.ac.id/index.php/knsi2018/article/view/383>
- [3] Ramadhan, D. A., & Indriyanti, A. D. (2022). Procedural Content Generation pada Game World Exploration Sandbox Menggunakan Algoritma Perlin Noise. *Journal of Informatics and Computer Science (JINACS)*; Vol 4 No 01 (2022); 86-91 ; 2686-2220 ; 10.26740/Jinacs.V4n01. <https://ejournal.unesa.ac.id/index.php/jinacs/article/view/48693>
- [4] Azzmi, N., Husniah, L., & Kholimi, A. S. (2020). Island Generator pada Game Open world Menggunakan Algoritma Perlin noise. *Jurnal Repositor*; Vol. 2 No. 7 (2020): Juli 2020; 965-976 ; *Jurnal Repositor*; Vol 2 No 7 (2020): Juli 2020; 965-976 ; 2716-1382 ; 2714-7975 ; 10.22219/Repositor.V2i7. <https://repositor.umm.ac.id/index.php/repositor/article/view/601>
- [5] Shen Zhenyuan. (2022). Procedural Generation in Games: Focusing on Dungeons. *SHS Web of Conferences*, 144, 02005. <https://doi.org/10.1051/shsconf/202214402005>
- [6] Budiono, M., Liliana, L., & Juwiantho, H. (2022). Meningkatkan Kesulitan Serangan Enemy Dengan Menambahkan Influence Map Pada Metode A\* Pada Procedural Generated Tower Defense Game. *Jurnal Infra*; Vol 10, No 1 (2022); 210-216. <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/12043>
- [7] de Lima, E. S., Feijó, B., & Furtado, A. L. (2019, October). Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning. In *SBGames* (pp. 144-153). [http://www.icad.puc-rio.br/~logtell/papers/Edirlei\\_SBGames\\_2019.pdf](http://www.icad.puc-rio.br/~logtell/papers/Edirlei_SBGames_2019.pdf)
- [8] Ginting, A. K., Sari, K., Fadhilah, C., Yusra, R. N., Hartama, D., & Zarlis, M. (2019). Application of the Perlin Noise Algorithm as a Track Generator in the Endless Runner Genre Game. *Journal of Physics: Conference Series*, 1255(1), 012064. <https://doi.org/10.1088/1742-6596/1255/1/012064>
- [9] Bennett, M. (2018). Frequency Spectra Filtering for Perlin Noise. Bennett , M 2018 , ' Frequency Spectra Filtering for Perlin Noise ' , *The Computer Games Journal* , Vol. 8 , Pp. 1-12 . <https://doi.org/10.1007/S40869-018-0074-7>. <https://pure.solent.ac.uk/en/publications/315cf7bf-d962-421e-9136-957c4ba6999e>
- [10] Tjiandra, C. V., Adipranata, R., & Dewi, L. P. (2021). Pemanfaatan Simplex Noise Untuk Menghasilkan Map Yang Natural dengan Unity Engine. *Jurnal Infra*; Vol 9, No 1 (2021); 71-77. <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/10924>