

Artikel Penelitian (Teknik Informatika)

Analisis Algoritma Wave Function Collapse sebagai Metode Prosedural Generator untuk Pembuatan Medan Permainan Digital

Farhan Febryan^{*}, Lukman Abdurrahman

Fakultas Rekayasa Industri, Sistem Informasi, Telkom University, Bandung, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 02 Desember 2025
Revisi Akhir: 13 Februari 2026
Diterbitkan Online: 02 April 2026

KATA KUNCI

Game Development
Procedural Generation
Wave Function Collapse

KORESPONDENSI

Phone: +62 851-5607-2879
E-mail: paangpemi@gmail.com

A B S T R A K

Pengembangan aplikasi permainan dimaksudkan untuk menghibur dan memberikan edukasi kepada para pemain dan penggunanya, tetapi tahap pembuatan aplikasi permainan tidak mudah dan cenderung rumit, biaya yang besar serta waktu yang banyak sangat dibutuhkan dalam pembuatan aplikasi permainan, ini sering sekali menjadi kendala utama dalam pembuatan aplikasi permainan hingga menyebabkan penundaan dalam pengembangan aplikasi permainan. Konsep dari algoritma Wave Function Collapse dibuat untuk memangkas biaya dan waktu dalam pengembangan aplikasi permainan, algoritma ini merupakan algoritma Content Generator yang dimaksudkan sebagai alat bantu untuk membuat atau menciptakan suatu konten yang dibutuhkan didalam aplikasi permainan secara procedural, baik itu level, desain, bahkan item dan objek game itu sendiri. Setelah penelitian dilakukan, didapati bahwa algoritma ini cocok untuk membangun dunia yang tidak terlalu besar, dalam kata lain setidaknya dalam cakupan 50 x 50 grid, jika dilakukan dengan lebih, maka akan memakan waktu serta membutuhkan memori yang besar.

PENDAHULUAN

Dalam membuat sebuah aplikasi permainan, terkadang para pengembang dihadapkan dengan sulitnya membuat medan permainan yang begitu luas, membuat medan permainan secara manual sangat memakan waktu dan biaya yang besar, sehingga sering kali waktu pembuatan permainan membutuhkan waktu yang sangat lama, didasari hal ini penulis akan membahas tentang sebuah algoritma Wave Function Collapse yang merupakan prosedural generator untuk membuat medan permainan secara prosedural.

Dalam konteks komputasi, algoritma WFC bekerja dengan memanipulasi himpunan kemungkinan (*domain*) pada setiap sel dalam *grid*. Algoritma ini secara iteratif memilih sel dengan jumlah kemungkinan opsi paling sedikit (entropi terendah), menetapkan satu nilai pasti pada sel tersebut, dan mempropagasi batasan aturan penyambungan (*adjacency rules*) ke sel-sel tetangga. Proses ini memastikan bahwa setiap sel yang ditempatkan selalu valid secara kontekstual dengan sel di sekitarnya, seperti aturan 'sisi jalan harus tersambung dengan jalan', Wave Function Collapse dapat menetapkan superposisi dari berbagai nilai untuk posisi, momentum, dan sifat-sifat lainnya [1].

Metode ini menempati posisi sebagai *Procedural Content Generation* (PCG), PCG merupakan suatu konsep dimana suatu konten bisa dibuat secara dinamis dan tidak statis. Konsep ini bertujuan untuk membuat suatu konten yang dimana user atau kreator tidak perlu mengulang pekerjaan yang sama terus menerus, konsep ini juga sudah diadopsi secara luas baik untuk pengembangan aplikasi permainan, arsitek bangunan sampai 3D artist. PCG sendiri memiliki banyak algoritma, dan masing-masing memiliki kelebihannya. Meski PCG bisa digunakan secara dinamis untuk membuat lingkungan atau environment, sebaiknya menggunakan algoritma PCG lain untuk hal kompleks seperti pepohonan atau bebatuan [2]. PCG

juga kerap digunakan untuk AI, PCG dimanfaatkan untuk membuat simulasi lingkungan agar AI bisa mempelajari situasi tertentu untuk lingkungan tertentu [3]. Meski banyak studi yang mengkaji tentang PCG dalam pembuatan game, tidak semua menjelaskan bagaimana PCG itu bisa memberikan dampak terhadap alur permainan dan pengalaman bermain [4].

Pengembangan game sendiri memiliki kesulitan yang kompleks, harus memperhatikan lingkungan yang tersedia, perilaku dari entitas yang dinamis atau statis, interaksi antara pengguna dan dunia, serta manajemen situasi yang bisa memakan waktu sangat lama dan biaya produksi yang besar, namun model tradisional sering gagal menjamin efektivitas antarmuka manusia-mesin (human-machine interfaces) dan pengalaman pengguna [5].

Demi memangkas waktu dan biaya yang besar itu, Wave Function Collapse digunakan untuk memberikan efektifitas yang lebih baik, baik secara waktu dan biaya. Dengan menggunakan algoritma ini, para pengembang bisa menciptakan content baru, baik untuk lingkungan dunia, model 3D atau yang lain sesuai dengan ketentuan yang diterapkan oleh pengguna.

Tujuan utama penelitian ini adalah mengembangkan dan menguji implementasi metode akses *Pixel Array* pada algoritma *Wave Function Collapse* untuk mengatasi permasalahan logika penyambungan ubin yang sering ditemui pada metode berbasis *noise*. Penelitian ini juga bertujuan membuktikan secara empiris hubungan antara peningkatan dimensi *grid* dengan lonjakan durasi komputasi akibat fenomena jalan buntu (*deadlock*) dalam proses generasi prosedural.

TINJAUAN PUSTAKA

Procedural Content Generation

Procedural Content Generation adalah sebuah konsep dimana suatu sistem berjalan dengan dinamis dan otomatis, dengan memberikan beberapa ketentuan dan sifat, maka sebuah konten bisa berjalan dengan otomatis dan dinamis dimana konten bisa semakin besar, semakin kompleks atau semakin kecil tanpa kehilangan sifat aslinya. Tujuan utamanya adalah untuk menciptakan objek yang diinginkan secara otomatis dengan input minimal, sehingga menghasilkan konten game atau lingkungan virtual dalam jumlah besar secara otomatis [6]. Dengan memanfaatkan konsep ini, maka lingkungan dunia tidak akan pernah sama dan akan selalu berbeda dan memberikan pengalaman yang lebih baik, jika lingkungan tetap sama terus menerus, akan membuat kejenuhan pemain karena tidak mendapatkan hal baru dan harus merasakan hal yang sama terus menerus, ini memiliki nilai replayability yang rendah [7].

Wave Function Collapse

WFC adalah algoritma PCG yang dirancang untuk menghasilkan tekstur atau konten digital yang menyerupai input (peta atau tekstur) yang disediakan [8]. WFC adalah teknik PCG baru yang dikembangkan untuk mengatasi keterbatasan yang diamati pada metodologi PCG sebelumnya, terutama keterbatasan desain praktis.

Pixel Array

Pixel Array memiliki arti umum yaitu sebuah array yang mengandung piksel piksel yang mana masing-masing piksel tersebut memiliki suatu value tersendiri seperti warna. Sedikit berbeda dengan penjelasan diatas, pengembangan tidak akan menggunakan informasi warna dalam suatu piksel, tetapi metode digunakan untuk menentukan suatu posisi Pixel dalam suatu array, piksel ini nantinya akan disimulasikan dalam bentuk grid, dan para pengembang akan mencoba mengaksesnya menggunakan pendekatan metode Pixel Array.

PENELITIAN TERDAHULU

Meskipun metode Procedural Content Generation (PCG) telah banyak diteliti, sebagian besar penelitian sebelumnya seperti [9] berfokus pada penggunaan algoritma berbasis noise (seperti Perlin Noise) yang efektif untuk kontur alami namun lemah dalam menangani aturan penyambungan logika yang ketat (misalnya struktur jalan). Di sisi lain, studi mengenai Wave Function Collapse (WFC) yang ada, seperti [10], cenderung berfokus pada metode hibrida untuk estetika 3D dan seringkali mengabaikan analisis performa komputasi dasar pada skala grid yang ekstrim. Selain itu, penggunaan metode Pixel Array sejauh ini lebih dominan dibahas dalam konteks perangkat keras sensor [11] dan belum dieksplorasi

secara mendalam sebagai metode pengindeksan struktur data dalam algoritma PCG. Oleh karena itu, penelitian ini hadir untuk mengisi kesenjangan tersebut dengan menganalisis secara kuantitatif batas performa algoritma WFC murni yang diintegrasikan dengan metode akses Pixel Array pada berbagai dimensi grid untuk membuktikan efektivitas dan batasan komputasinya dalam pengembangan permainan 2D.

Berdasarkan kesenjangan tersebut, penelitian ini bertujuan untuk menguji efektivitas penerapan metode akses *Pixel Array* dalam algoritma *Wave Function Collapse* (WFC) murni sebagai solusi pembuatan medan permainan prosedural. Penelitian ini berfokus pada pengujian performa komputasi dan validitas aturan penyambungan ubin (*tile adjacency rules*) secara kuantitatif pada berbagai dimensi *grid*, mulai dari skala kecil (8x8) hingga skala besar (100x100). Melalui serangkaian percobaan ini, penelitian berhasil membuktikan bahwa meskipun integrasi *Pixel Array* dan WFC mampu menghasilkan variasi medan permainan yang unik dan valid secara logika pada dimensi kecil hingga menengah, terdapat batasan skalabilitas yang signifikan pada dimensi besar. Secara spesifik, penelitian ini membuktikan adanya lonjakan durasi proses yang ekstrem akibat fenomena jalan buntu (*deadlock*) pada *grid* berukuran 100x100, yang memberikan wawasan empiris mengenai batas kemampuan algoritma ini sebelum memerlukan optimasi berbasis *chunk*.

METODOLOGI

Dengan melakukan penilaian kuantitatif, algoritma akan dinilai performanya serta durasi yang diperlukan untuk menyelesaikan tugasnya dengan dimensi grid yang ditentukan. Dimensi grid akan dibikin lebih besar dari sebelumnya untuk melihat kemampuannya. Observasi dilakukan untuk mengamati perilaku dari algoritma dan performanya. Alasan kenapa observasi dipilih untuk penelitian ini karena observasi adalah metode paling tua dan paling dasar yang bertujuan untuk memperhatikan secara akurat, mencatat perilaku, dan mempertimbangkan keputusan yang akan dilakukan [12], [13].

Teknik Observasi Algoritma

Untuk mengetahui posisi suatu Pixel maka akan menggunakan beberapa persamaan dibawah ini:

x : nilai baris

y : nilai kolom

dim : dimensi

Dengan menggunakan variabel yang sudah tertera diatas, pengembangan bisa dilakukan menggunakan variable tersebut:

Index

metode ini digunakan untuk mendapatkan posisi index Pixel tersebut, persamaannya adalah:

$$index = x + y * dim \tag{1}$$

UP

metode ini digunakan untuk melihat posisi Pixel yang ada diatas Pixel saat ini (*index*), metode ini akan dijalankan dengan satu ketentuan yaitu ($y > 0$), persamaannya adalah:

$$UP = x + (y - 1) * dim \tag{2}$$

RIGHT

metode ini digunakan untuk melihat posisi Pixel yang ada dikanan Pixel saat ini (*index*), metode ini akan dijalankan dengan satu ketentuan yaitu ($x < dim - 1$), persamaannya adalah :

$$RIGHT = (x + 1) + y * dim \tag{3}$$

DOWN

metode ini dinakan untuk melihat posisi Pixel yang ada dibawah Pixel saat ini (index), metode ini akan dijalankan dengan satu ketentuan ($y < dim - 1$), persamaannya adalah :

$$DOWN = x + (y + 1) * dim \tag{4}$$

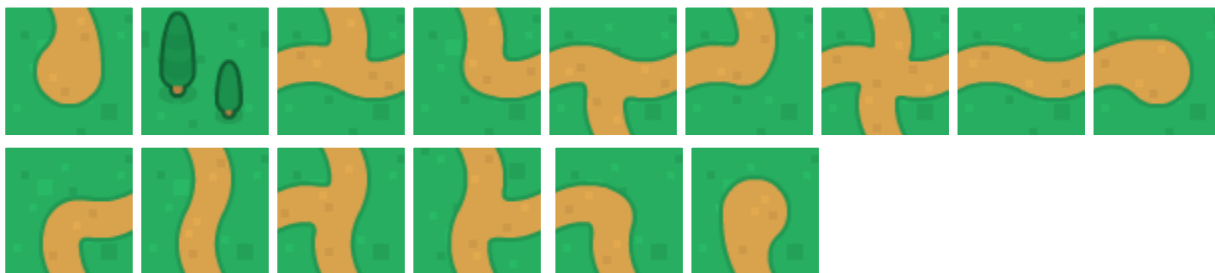
LEFT

metode ini dinakan untuk melihat posisi Pixel yang ada dikiri Pixel saat ini (index), metode ini akan dijalankan dengan satu ketentuan ($x > 0$), persamaannya adalah:

$$LEFT = (x - 1) + y * dim \tag{5}$$

Teknik Analsis Data

Setelah algoritma dijalankan, maka jalannya algoritma bisa dipantau, apakah ada kendala pada saat dijalankan, atau sudah jalan semestinya. Visualisasi dilakukan untuk membantu memudahkan analisis data, maka jumlah grid yang digunakan adalah 5 x 5. Untuk jumlah kemungkinan dari tiap sel, tergantung dari berapa banyak data yang di berikan dan kemungkinan yang bisa diambil dari sel yang sudah di *generate*, gambar yang digunakan adalah:



Gambar 1. Asset yang Digunakan

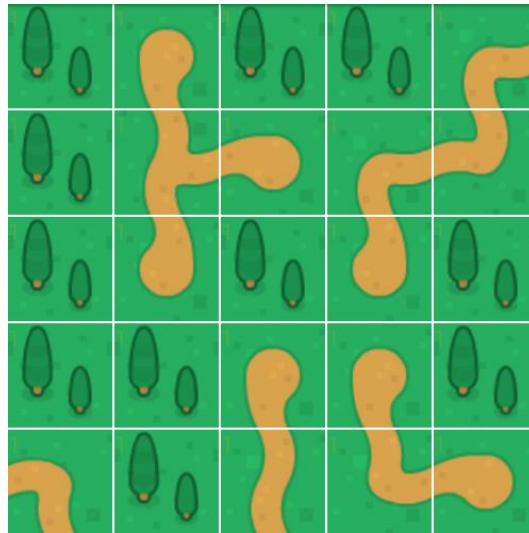
HASIL DAN PEMBAHASAN

Dengan menjalankan metode ini beberapa kali dengan perbedaan grid, didapatkan informasi kemampuan algoritma, performa dan ketepatan algoritma. Meski mendapati kesulitan dengan grid yang lebih besar, algoritma ini tetap mampu untuk menyelesaikan pembuatan medan permainan. Pixel array terbukti cocok disandingkan dengan WFC, tetapi untuk performa mungkin sedikit lebih lambat karena harus memperhatikan masing-masing dari sel secara mengulang, dan ketika sudah melakukan pemilihan sel, maka iterasi akan diulangi lagi dari awal. WFC yang disandingkan dengan Pixel Array bergantung terhadap bagaimana pengguna melakukan penanganan untuk error yang terjadi, berbeda cara untuk menangani error yang terjadi memungkinkan untuk memberikan hasil yang berbeda.

Dengan menggunakan 15 gambar tersebut, peraturan yang diterapkan adalah:

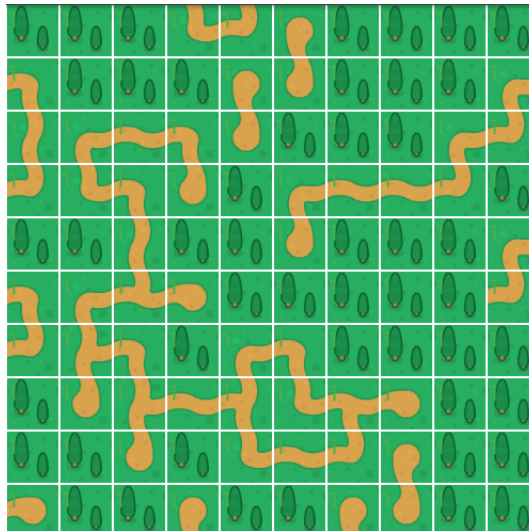
1. Setiap sisi yang merupakan jalan, hanya bisa disambung dengan sisi yang merupakan jalan juga.
2. Setiap sisi yang merupakan rumput, bisa disambung dengan sisi yang merupakan rumput juga.

Dengan menerapkan ketentuan tersebut, maka hasil dari algoritma dengan grid 5 x 5 akan menjadi seperti ini:

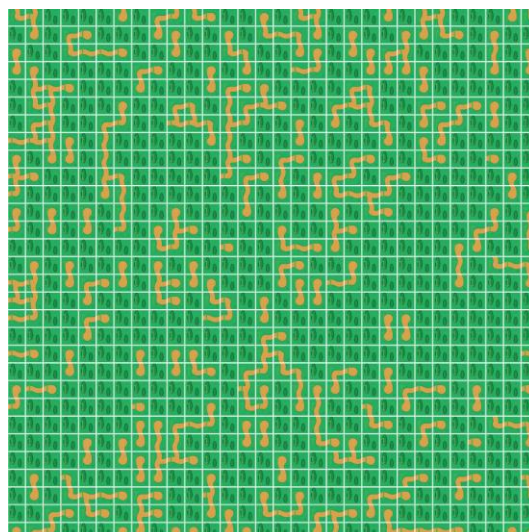


Gambar 2. Hasil Algoritma dengan Grid 5 x 5

Ukuran grid bisa diubah-ubah, sesuai dengan kebutuhan yang diperlukan para *developer*. Jika kita mengubah ukuran grid menjadi 10 x 10 dan 30 x 30, maka hasilnya akan menjadi seperti ini:



Gambar 3. Hasil Algoritma dengan Grid 10 x 10



Gambar 4. Hasil Algoritma dengan Grid 30 x 30

Wave Function Collapse memiliki kendala saat diterapkan pada dimensi grid yang lebih besar. Hal ini disebabkan oleh meningkatnya potensi jalan buntu (*dead end*), di mana algoritma mencapai kondisi kontradiksi (tidak ada opsi sel yang valid). Ketika kondisi ini terjadi, proses *generation* dianggap gagal dan harus diulang kembali dari awal. Akumulasi dari proses pengulangan inilah yang menyebabkan waktu komputasi meningkat secara drastis.

Tabel 1. Perbandingan Durasi Jalannya Algoritma

Dimensi Grid	Durasi
8x8	53 ms
20x20	818 ms
50x50	3 detik
70x70	7 detik
100x100	7 menit 26 detik

KESIMPULAN DAN SARAN

Algoritma ini sangat efektif untuk pengembangan level permainan secara prosedural karena mampu menciptakan variasi hasil yang unik dan tidak repetitif, serta memiliki potensi luas untuk diterapkan pada desain objek lain. Namun, kelemahan utamanya terletak pada waktu proses yang lambat akibat banyaknya pengulangan, sehingga performanya perlu dijaga dengan cara membatasi dimensi ruang lingkup area yang dibuat. Saran Untuk efisiensi maksimal, sebaiknya gunakan input tekstur yang sudah matang agar algoritma tidak perlu melakukan proses ulang akibat kesalahan pembacaan. Metode ini dinilai paling ideal untuk permainan jenis platformer, dungeon-based, atau permainan yang tidak memerlukan lingkungan besar untuk satu stage-nya. Untuk permainan jenis open world, sebaiknya dioptimisasi dengan sistem chunk-based generation, untuk melakukan perhitungan sendiri di chunk yang tersedia, tidak secara keseluruhan, ini bisa menghemat durasi dan kesalahan error, dengan membagi grid yang besar menjadi grid-grid kecil yang lebih efisien.

DAFTAR PUSTAKA

- [1] D. J. Chalmers and K. J. McQueen, "Consciousness and the Collapse of the Wave Function," May 2021, [Online]. Available: <http://arxiv.org/abs/2105.02314>
- [2] D. Anugrah Ramadhan and A. Dwi Indriyanti, "Procedural Content Generation pada Game World Exploration Sandbox Menggunakan Alogoritma Perlin Noise," *Journal of Informatics and Computer Science*, vol. 04, 2022.
- [3] M. Deitke *et al.*, "ProcTHOR: Large-Scale Embodied AI Using Procedural Generation," 2022.
- [4] A. Wahyudi, A. I. Pawelloi, and N. Sirimorok, "Implementasi Procedural Generation Pada Game Aksi Survival," *Jurnal Informatika dan Teknologi Pendidikan*, vol. 5, no. 2, pp. 144–151, 2025, doi: 10.59395/jitp.v5i2.142.
- [5] K. Sungkaew, P. Lungban, and S. Lamhya, "Game development software engineering: digital educational game promoting algorithmic thinking," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 5, pp. 5393–5404, Oct. 2022, doi: 10.11591/ijece.v12i5.pp5393-5404.
- [6] Y. Liu, "An Overview of Procedurally Generating Virtual Environments," *International Journal of Computer Science and Information Technology*, vol. 2, no. 1, pp. 141–147, Mar. 2024, doi: 10.62051/ijcsit.v2n1.16.
- [7] H. Juwiantho, L. Liliana, and M. Budiono, "Procedural Content Generation pada Game Tower Defense menggunakan Perlin Noise dan Algoritma Floyd Warshall," 2023.
- [8] O. Büyükkşar, D. Yıldız, and S. Demirci, "Enhancing Wave Function Collapse Algorithm for Procedural Map Generation Problem," *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, Apr. 2024, doi: 10.28948/ngumuh.1361413.
- [9] D. Anugrah Ramadhan and A. Dwi Indriyanti, "Procedural Content Generation pada Game World Exploration Sandbox Menggunakan Alogoritma Perlin Noise," *Journal of Informatics and Computer Science*, vol. 04, 2022.
- [10] O. Büyükkşar, D. Yıldız, and S. Demirci, "Enhancing Wave Function Collapse Algorithm for Procedural Map Generation Problem," *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, Apr. 2024, doi: 10.28948/ngumuh.1361413.

- [11] Y. J. Jang *et al.*, “Development of a Real-Time Pixel Array-Type Detector for Ultrahigh Dose-Rate Beams,” *Sensors*, vol. 23, no. 10, May 2023, doi: 10.3390/s23104596.
- [12] Ahmad Saádi, “Pengumpulan Data Yang Efisien pada Penelitian Tindakan Kelas: Teknik, Alat, dan Tantangan,” vol. 2, pp. 90–108, 2025.
- [13] M. Waruwu, S. N. Pu`at, P. R. Utami, E. Yanti, and M. Rusydiana, “Metode Penelitian Kuantitatif: Konsep, Jenis, Tahapan dan Kelebihan,” *Jurnal Ilmiah Profesi Pendidikan*, vol. 10, no. 1, pp. 917–932, Feb. 2025, doi: 10.29303/jipp.v10i1.3057.