

Analisis L-System dan Penerapan Teknik Graph Coloring pada Wave Function Collapse dalam Procedural Generation untuk Dunia Video Game

Rafael Marchel Darma Wijaya - 13523146¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

rafaelmarchel.dw@gmail.com, 13523146@std.stei.itb.ac.id

Abstract—Procedural Content Generation (PCG) has become an essential technique in game development, providing innovative solutions for creating dynamic and complex game environments. This paper explores the use of L-systems and Wave Function Collapse (WFC) in procedural generation, focusing specifically on their applications in generating organic and intricate game worlds. L-systems, originally designed to model the growth of organisms, are used for generating branching structures and patterns in games. Through recursive string rewriting, L-systems create visually appealing and organic shapes, making them suitable for natural environment generation in games. On the other hand, Wave Function Collapse, a technique based on quantum mechanics, generates content by collapsing superpositions of possible states, ensuring consistency in tile arrangement and relationships. This paper also explores the integration of Graph Coloring techniques, such as the Welsh-Powell algorithm, to enhance the diversity and interactivity of generated content. By employing these methods, the study demonstrates how procedural generation can create adaptive, unpredictable, and engaging game levels, offering a deeper understanding of the creative process and its potential in game design.

Keywords—Graph Coloring, L-System, Procedural Generation, Wave Function Collapse.

I. PENDAHULUAN

Game komputer semakin hadir dalam kehidupan kita. Setiap hari, ratusan juta pemain di seluruh dunia terhibur oleh *game* [1]. Proses pembuatan dunia dalam *video game* menjadi salah satu aspek yang sangat penting dalam menciptakan pengalaman permainan yang menarik. Dunia yang terstruktur dengan baik tidak hanya memberikan tantangan bagi pemain, tetapi juga menciptakan suasana bervariasi. Generasi konten prosedural (PCG) merujuk pada metode sepenuhnya otonom atau terbatas dikendalikan oleh manusia untuk menghasilkan konten *game* [2].

Salah satu teknik yang sering digunakan dalam *procedural generation* adalah *Lindenmayer System* (L-System). L-System pertama kali diperkenalkan oleh Aristid Lindenmayer pada tahun 1968 sebagai sebuah model matematis untuk menggambarkan proses pertumbuhan tanaman [3]. Teknik ini menggunakan rekursi untuk mengembangkan suatu struktur

atau pola berdasarkan aturan-aturan tertentu. Dalam dunia *video game*, L-System sering digunakan untuk menghasilkan struktur organik seperti pohon, tumbuhan, dan pola alami lainnya, yang memberi kesan dunia yang hidup dan berkembang.

Selain L-System, teknik lain yang populer dalam *procedural generation* adalah *Wave Function Collapse* (WFC). WFC adalah algoritma yang terinspirasi oleh prinsip-prinsip fisika kuantum, khususnya konsep superposisi dan pengamatan. Dalam mekanika kuantum, *wave function collapse* terjadi ketika probabilitas suatu peristiwa berkurang menjadi satu atau beberapa keadaan spesifik. Ketika pengamatan dilakukan, fungsi gelombang terkonsentrasi pada titik tertentu dan identitas partikel-partikelnya muncul [4]. Teknik ini sangat efektif untuk menciptakan peta *dungeon* atau peta dunia lainnya, di mana elemen-elemen tersebut harus saling terhubung dengan cara yang konsisten dan sesuai aturan.

Meskipun kedua teknik ini memiliki kelebihan masing-masing dalam menghasilkan dunia *game* secara prosedural, ada tantangan yang perlu dihadapi, seperti bagaimana mengelola keterhubungan antara elemen-elemen peta atau memastikan bahwa peta yang dihasilkan memiliki variasi yang cukup tanpa menjadi repetitif. Salah satu pendekatan yang dapat digunakan untuk menyelesaikan masalah tersebut adalah penerapan *Graph Coloring* dalam algoritma WFC. Dengan *Graph Coloring*, kita dapat memastikan bahwa elemen-elemen yang berdekatan dalam peta tidak memiliki kesamaan yang tidak diinginkan, seperti jenis ruangan yang tumpang tindih atau penghubung yang tidak konsisten.

Penulisan ini bertujuan untuk membahas mengenai kedua teknik ini, yakni L-System dan *Wave Function Collapse*, serta bagaimana penerapan *Graph Coloring* dapat meningkatkan hasil generasi dunia dalam *video game*. Melalui pembahasan ini, diharapkan dapat memberikan pemahaman yang lebih baik tentang bagaimana kedua algoritma ini bekerja, serta bagaimana penerapannya dapat menciptakan dunia permainan yang menarik, dinamis, dan penuh dengan variasi.

II. LANDASAN TEORI

Generasi konten prosedural (PCG) merujuk pada metode sepenuhnya otonom atau terbatas dikendalikan oleh manusia untuk menghasilkan konten *game* [2]. Teknik ini digunakan untuk menciptakan elemen-elemen dalam game seperti *level*, peta, karakter, *item*, dan elemen lainnya tanpa perlu ditentukan terlebih dahulu oleh pengembang *game*. Konsep dasar dari PCG adalah untuk menggantikan atau meminimalisir pekerjaan manual yang biasanya dilakukan oleh desainer *game* dengan cara menciptakan elemen-elemen yang tak terduga namun tetap terstruktur. Salah satu manfaat utama dari PCG adalah untuk menciptakan dunia yang lebih dinamis dan bervariasi dalam game, yang dapat menambah *replayability*.

L-System pertama kali diperkenalkan oleh Aristid Lindenmayer pada tahun 1968 sebagai sebuah model matematis untuk menggambarkan proses pertumbuhan tanaman [3]. L-systems pertama kali diperkenalkan untuk memodelkan perkembangan organisme *multicellular* sederhana (misalnya, alga) dalam hal pembelahan, pertumbuhan, dan kematian sel-sel individu [3]. Pada dasarnya, L-System menggunakan aturan penggantian (*rewrite rules*) untuk mengembangkan suatu *string* simbol menjadi bentuk yang lebih kompleks. Dalam konteks generasi konten prosedural, L-System banyak digunakan untuk menciptakan bentuk-bentuk alami seperti pohon, tanaman, dan struktur biologis lainnya. Prinsip dasar dari L-System adalah penggunaan dua komponen utama: *Axiom* dan *Rules*. *Axiom* adalah *string* awal yang digunakan sebagai titik awal, sedangkan *Rules* adalah serangkaian aturan yang mendefinisikan bagaimana simbol-simbol dalam *Axiom* akan diganti pada setiap iterasi. Contoh dasar L-System adalah sebagai berikut:

- Axiom: 'A'
- Rule: 'A' -> "AA"

Pada setiap iterasi, string 'A' akan digantikan dengan "AA", dan proses ini dapat berlanjut hingga menghasilkan struktur yang kompleks.

Wave Function Collapse (WFC) adalah algoritma yang terinspirasi oleh prinsip-prinsip fisika kuantum, khususnya konsep superposisi dan pengamatan. Dalam mekanika kuantum, *wave function collapse* terjadi ketika probabilitas suatu peristiwa berkurang menjadi satu atau beberapa keadaan spesifik. Ketika pengamatan dilakukan, fungsi gelombang terkonsentrasi pada titik tertentu dan identitas partikel-partikelnya muncul [4]. Konsep yang sama diterapkan dalam generasi prosedural. Sebuah sistem (seperti peta *dungeon* atau *level* dalam *game*) dimulai dalam keadaan "superposisi", semua kemungkinan hasil atau *tile* diperbolehkan. Seiring proses berlangsung, sistem ini secara bertahap mengurangi kemungkinan-kemungkinan ini (melakukan "*collapse*") berdasarkan aturan-aturan yang telah ditentukan, hingga mencapai konfigurasi akhir yang valid. Proses dalam WFC terdiri dari langkah-langkah berikut:

- Menentukan jenis *tile* yang akan digunakan dalam peta.
- Memilih *tile* berdasarkan prinsip entropi minimisasi, di mana *tile* dengan kemungkinan terendah dipilih terlebih dahulu.
- Menyebarkan pembatasan (*constraints*) berdasarkan

aturan penempatan *tile*, sehingga *tile* yang terpilih memengaruhi kemungkinan *tile* tetangga (*adjacent tile*).

- Melakukan proses iterasi hingga seluruh grid terisi dengan *tile* yang valid.

Graph Coloring adalah teknik yang digunakan untuk memberi warna pada simpul dalam graf dengan tujuan agar simpul-simpul yang saling berdekatan tidak memiliki warna yang sama. Dalam penerapannya pada WFC, *Graph Coloring* dapat digunakan untuk memastikan bahwa ruangan-ruangan yang berdekatan tidak memiliki jenis atau warna yang sama, menciptakan peta yang lebih dinamis dan terstruktur. Bilangan kromatik adalah nilai yang mengindikasikan jumlah warna minimum yang diperlukan untuk mewarnai graf tersebut.

Algoritma yang dipakai dalam penerapan *Graph Coloring* adalah Algoritma Welsh-Powell. Algoritma Welsh-Powell adalah salah satu algoritma yang digunakan dalam teori pewarnaan graf (*graph coloring*) untuk mencari solusi yang efisien dalam pewarnaan graf. Algoritma Welsh-Powell merupakan algoritma yang digunakan untuk menyelesaikan masalah pewarnaan graf secara efisien, meskipun tidak selalu memberikan solusi optimal. Berikut adalah penjelasan tentang langkah-langkah dasar dalam algoritma Welsh-Powell:

- Langkah pertama dalam algoritma Welsh-Powell adalah mengurutkan simpul-simpul dalam graf berdasarkan derajatnya (jumlah sisi yang terhubung ke simpul tersebut), dari yang memiliki derajat tertinggi ke yang memiliki derajat terendah.
- Setelah simpul-simpul diurutkan, algoritma akan memberikan warna pada simpul pertama dalam urutan tersebut dengan warna pertama. Kemudian, untuk setiap simpul berikutnya, algoritma akan memberikan warna yang berbeda jika simpul tersebut terhubung langsung dengan simpul yang sudah diwarnai. Proses ini akan dilanjutkan hingga semua simpul dalam graf diberi warna.

III. L-SYSTEM

L-System (*Lindenmayer System*) merupakan algoritma yang sangat terkenal dan sering digunakan dalam *procedural generation*, terutama dalam menciptakan bentuk-bentuk yang mirip dengan struktur alami seperti pohon, tumbuhan, atau bentuk-bentuk organik yang lain. Algoritma ini pertama kali diperkenalkan oleh Aristid Lindenmayer pada tahun 1968, dan digunakan dalam biologi untuk menggambarkan proses pertumbuhan organisme. Dalam konteks dunia *video game*, L-System dapat digunakan untuk menghasilkan struktur dunia yang organik atau komponen-komponen alam. Kemampuan L-System inilah yang sering dipakai untuk menciptakan lingkungan dunia *video game* yang bervariasi.

Pada dasarnya, L-System menggunakan sistem rekursi untuk mengganti dan mengembangkan suatu karakter pada *String* berdasarkan aturan-aturan yang telah ditentukan. Dua komponen utama dari L-System adalah *Axiom* dan *Rules* (aturan). *Axiom* adalah karakter atau *String* awal yang berguna menjadi titik awal pada sistem. *Axiom* bisa saja sesederhana suatu karakter seperti 'A', 'B', 'F' atau bisa serumit dan

sepanjang yang diinginkan seperti “AAA”, “FA”, “XYWB”. *Rules* adalah satu atau himpunan aturan yang memetakan dan menentukan suatu karakter pada *Axiom* dengan penggantinya pada setiap iterasi sistem. Nantinya, setiap kemunculan suatu karakter akan diganti dengan karakter atau *String* baru berdasarkan aturan yang telah ditentukan. Contohnya saja ‘A’ -> “AA”, ‘B’ -> “ABC”.

Proses dari L-System sendiri cukup sederhana. Dimulai dari *Axiom*, setiap karakter pada *Axiom* akan diganti dengan karakter atau *String* baru berdasarkan *Rules* yang telah ditentukan sebelumnya. Proses ini dapat diulang terus-menerus secara rekursif sebanyak yang diinginkan. Dengan menentukan *Rules* dan *Axiom* yang cukup baik, L-System dapat menciptakan pola yang indah dan organik.

Untuk memahami L-System lebih lanjut, kita akan melihat suatu contoh. Misal terdapat aturan yang telah didefinisikan.

- Rule 1: ‘A’ -> “AC”
- Rule 2: ‘B’ -> “ABC”
- Rule 3: ‘C’ -> “A”

Misal *Axiom*-nya adalah “AB”. Setelah dilakukan algoritma L-System sebanyak tiga iterasi, *Axiom* akan berubah menjadi berikut.

- 1: “AB” -> “ACABC”
- 2: “ACABC” -> “ACAACABCA”
- 3: “ACAACABCA” -> “ACAACACAACABCAAC”

Seperti yang terlihat, setiap iterasi, panjang *String* akan menjadi lebih panjang dan lebih kompleks daripada yang sebelumnya. Proses ini akan memanjang secara eksponensial dan dapat terus berlanjut tanpa batas.

Pada contoh ini, keluaran dari algoritma L-System hanyalah suatu *String*. Dengan menggunakan konvensi umum, *String* ini dapat dipetakan menjadi suatu pola visual. L-System dapat menciptakan suatu visual yang menyerupai bentuk organik dengan keluaran *String* ini.

Konvensi yang umum untuk memetakan suatu karakter dengan elemen visual yaitu:

- ‘F’, maju ke depan sebanyak suatu variabel panjang dan gambar suatu garis;
- ‘f’, maju ke depan sebanyak suatu variabel panjang tanpa menggambar garis;
- Karakter alfabet lain selain F dan f akan diabaikan (biasanya, berfungsi untuk memperpanjang dan menumbuhkan *String*);
- ‘+’, berputar ke kiri sejauh suatu variabel sudut;
- ‘-’, berputar ke kanan sejauh suatu variabel sudut;
- ‘[’, menyimpan posisi dan sudut saat ini ke dalam suatu *stack*;
- ‘]’, mengambil posisi dan sudut yang berada pada posisi paling atas *stack* dan kembali ke posisi dan sudut tersebut.

Dengan menggunakan konvensi ini, kita dapat memvisualisasikan *String* yang menjadi keluaran dari algoritma L-System. Misal, dengan *axiom* ‘F’ dan aturan ‘F’ -> “F+F-F”, setelah beberapa iterasi akan menghasilkan pola yang mirip pohon atau fraktal.



Fig. 1. Visualisasi L-System Sauppe pada iterasi pertama.



Fig. 2. Visualisasi L-System Sauppe pada iterasi kelima.



Fig. 3. Visualisasi L-System Sauppe pada iterasi kesepuluh.

Figur 1, 2, dan 3 adalah contoh visualisasi algoritma L-System yang menerapkan aturan dan *axiom* sebagai berikut.

- ‘V’ -> “[+++W][---W]YV”
- ‘W’ -> “+X[-W]Z”
- ‘X’ -> “-W[+X]Z”
- ‘Y’ -> “YZ”
- ‘Z’ -> “[-FFF][+FFF]F”
- *axiom* = “VZFFF”
- variabel sudut = 20 (dalam satuan derajat)

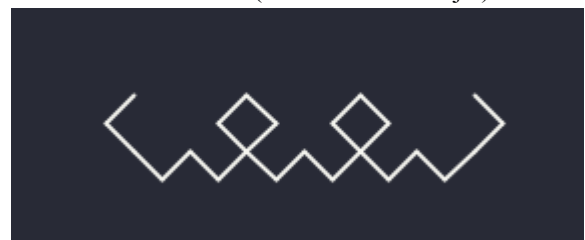


Fig. 4. Visualisasi L-System Lévy curve pada iterasi ketiga.

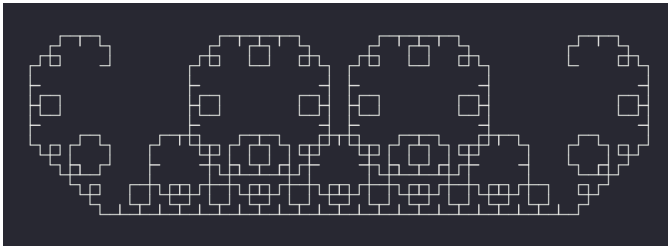


Fig. 5. Visualisasi L-System Lévy curve pada iterasi kedelapan.

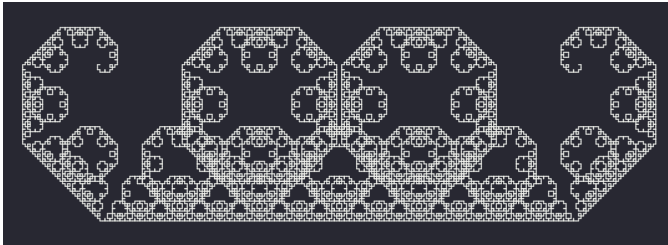


Fig. 6. Visualisasi L-System Lévy curve pada iterasi dua belas.

Figur 4, 5, dan 6 adalah contoh visualisasi algoritma L-System yang lebih sederhana dengan aturan dan *axiom* sebagai berikut.

- 'F' -> "-F++F-"
- *axiom* = "F"
- variabel sudut = 45

Dengan mencoba berbagai tipe *rules*, *axiom*, dan besar variabel sudut, kita dapat menemukan pola-pola unik lainnya. Misalkan saja aturan 'F' -> "F+F-F-" dengan *axiom* 'F' dan besar sudut putar 90.

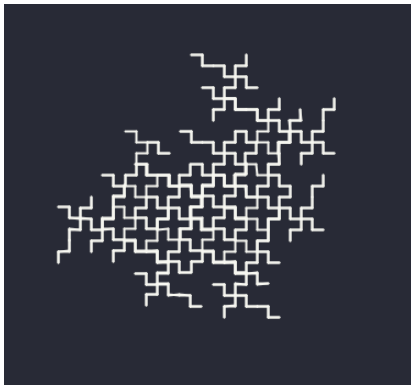


Fig. 7. Visualisasi L-System percobaan pada iterasi lima.

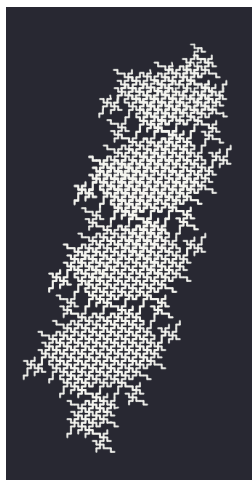


Fig.8. Visualisasi L-System percobaan pada iterasi delapan.

Kita juga dapat menambahkan faktor acak pada algoritma ini agar dapat menghasilkan pola yang tidak selalu pasti. Pertama, variabel sudut putar dapat diacak dalam suatu rentang tertentu. Kedua, suatu karakter dalam *rule* dapat memiliki lebih dari satu pengganti dan dipilih secara acak. Terakhir, karakter '[' biasanya akan memunculkan suatu cabang (*branch*). Kemunculan karakter ini, dapat diacak sehingga menciptakan suatu pola dengan cabang-cabang yang berbeda-beda.

L-System memiliki beberapa kelebihan yakni, mampu menciptakan bentuk dan pola natural atau pun fraktal, sangat fleksibel dan mudah dimodifikasi, dan mampu menciptakan pola kompleks tanpa membutuhkan tenaga komputasi yang tinggi. Kelemahan dari L-System yakni, kurang mampu menciptakan bentuk-bentuk struktur seperti bangunan atau bentuk mekanik, setiap iterasi berkembang secara eksponensial sehingga sangat cepat menjadi besar, dan perlu penyesuaian parameter yang benar-benar pas untuk menciptakan suatu bentuk yang natural dan cukup baik tanpa terlihat artifisial.

IV. GRAPH COLORING PADA WAVE FUNCTION COLLAPSE

Wave Function Collapse (WFC) adalah teknik *procedural generation* yang berdasarkan prinsip-prinsip fisika kuantum. Konsep fisika kuantum yang menjadi referensi adalah konsep superposisi yang menyebabkan suatu sistem dapat berada dalam beberapa kemungkinan keadaan sebelum "mengamati" dan meng-*collapse* menjadi satu keadaan yang pasti. Dalam konteks pembuatan konten dunia *game* secara prosedural, WFC digunakan untuk menghasilkan pola atau struktur berdasarkan input data keadaan, seperti peta *dungeon*, tekstur, atau elemen lainnya. Salah satu masalah utama dalam WFC adalah mengelola konsistensi dari pilihan tile yang saling berhubungan, memastikan bahwa tile yang berdekatan memiliki kecocokan yang sesuai agar menghasilkan struktur yang valid.

Graph Coloring adalah suatu teknik dalam teori graf yang bertujuan untuk memberikan warna yang berbeda pada setiap simpul (*node*) dalam graf, dengan syarat bahwa simpul yang berdekatan (*neighboring nodes*) tidak boleh memiliki warna yang sama. Dalam konteks WFC di dunia *video game*, simpul-simpul tersebut dapat dianggap sebagai suatu *room* dalam peta, dan warna dapat diartikan sebagai jenis *room* tersebut. Tujuannya adalah untuk memastikan bahwa *room* yang saling berhubungan memiliki warna (atau jenis) yang berbeda, menciptakan peta yang dinamis dan tidak repetitif. Pada bagian ini, akan dibahas bagaimana teknik *Graph Coloring* dapat diterapkan dalam WFC.

Dalam penerapan *Wave Function Collapse* (WFC) pada generasi peta *dungeon*, langkah pertama yang perlu dilakukan adalah mendefinisikan berbagai jenis *tile* yang akan membentuk struktur peta tersebut. *Tile* merupakan elemen dasar yang membentuk peta, dan setiap *tile* memiliki fungsi dan karakteristik tertentu yang memungkinkan peta tersebut terbentuk secara prosedural. Berikut adalah jenis-jenis *tile* yang akan digunakan, yakni mencakup ruang (*room*), void, dan jalan (*roads*). *Tile room* menggambarkan ruangan yang dapat diakses oleh pemain. Ruang adalah bagian dari *dungeon*

yang bisa berisi berbagai elemen seperti musuh, item, atau tantangan lainnya. Dalam konteks *procedural generation*, ruangan ini dihasilkan secara acak dan biasanya dapat berdampingan dengan tipe *tile* manapun. Terdapat satu jenis *room tile*. *Tile void* menggambarkan area yang tidak dapat diakses oleh pemain. *Void* adalah ruang kosong yang tidak memiliki objek atau elemen yang berarti dalam permainan. Terdapat satu jenis *void tile*. *Tile* jalan (*roads*) berfungsi sebagai penghubung antara berbagai ruangan di *dungeon*. *Tile* jalan memastikan bahwa pemain dapat berpindah antara ruang yang ada dalam peta. Terdapat satu jenis *tile* jalan dengan empat cabang (*Crossroad*). *Tile* jalan ini memiliki empat cabang yang menghubungkan empat arah: atas, kanan, bawah, dan kiri. Terdapat empat jenis *tile* jalan dengan tiga cabang. *Tile* jalan ini memiliki tiga cabang, menghubungkan tiga arah yang berbeda. Terdapat enam jenis *tile* jalan dengan dua cabang. *Tile* jalan dengan dua cabang ini hanya menghubungkan dua arah yang berbeda.



Fig.9. Bentuk *tile* (*room*, *crossroad*, dan 4 jenis *3-way roads*).



Fig.10. Bentuk *tile* (6 jenis *2-way roads* dan *void*).

Salah satu aspek penting dari algoritma ini adalah penentuan aturan penempatan *tile* yang berbasis pada hubungan antar *tile* yang berdekatan. Pada bagian ini, kita akan membahas bagaimana aturan penempatan *tile* diintegrasikan ke dalam algoritma WFC dan bagaimana proses *collapse* terjadi secara bertahap dalam peta.

Dalam penerapan WFC ini, kita perlu mendefinisikan beberapa aturan-aturan penempatan yang spesifik berdasarkan hubungan antar *tile* yang berdekatan (*adjacent*). *Tile* dengan tipe *room* dan *void* dapat bersebelahan dengan *tile* manapun. *Tile* dengan tipe *road* memiliki aturan yang lebih spesifik dalam penempatan dan interaksinya dengan *tile* lain. Pada dasarnya, *tile road* harus memiliki setidaknya satu cabang yang terhubung dengan *tile room*, *tile void*, atau dengan *tile road* lain yang memiliki cabang berlawanan. Hal ini memastikan bahwa *tile* jalan dapat terhubung dengan benar dengan *tile* lain, menciptakan keterhubungan yang valid antara ruang dan area yang dapat diakses pemain.

Langkah-langkah algoritma *Wave Function Collapse* yang digunakan untuk menghasilkan peta *dungeon* secara prosedural berdasarkan aturan penempatan *tile* yang telah dijelaskan sebelumnya adalah sebagai berikut. Pada awal algoritma, seluruh *tile* dalam *grid* peta diinisialisasi dengan status *not-collapsed*, yang berarti bahwa semua *tile* memiliki semua kemungkinan bentuk. *Grid* *dungeon* awalnya penuh dengan *not-collapsed tile*. Algoritma kemudian mencari dan memilih *tile* yang memiliki jumlah kemungkinan bentuk paling sedikit. Ini adalah langkah penting dalam WFC yang disebut entropi minimisasi. *Tile* yang paling tidak pasti (dengan kemungkinan

bentuk paling sedikit) dipilih terlebih dahulu untuk dikerjakan. Pada awalnya, semua *tile* memiliki jumlah kemungkinan yang sama, sehingga dipilih satu *tile* secara acak. Setelah memilih *tile* yang memiliki kemungkinan bentuk paling sedikit, algoritma memilih satu bentuk *tile* secara acak dari semua kemungkinan yang tersedia untuk *tile* tersebut. Setelah *tile* terpilih, *tile* tersebut akan di-*collapse*, yang berarti bahwa bentuk *tile* telah ditentukan dan tidak dapat diubah lagi. Setelah satu *tile* di-*collapse*, *tile* yang berada di sekitarnya (empat arah: atas, kanan, bawah, kiri) akan berkurang kemungkinannya. Hal ini terjadi karena aturan penempatan *tile* yang sudah didefinisikan sebelumnya. Setelah mengurangi kemungkinan *tile* yang bersebelahan, algoritma kemudian mencari *tile* yang memiliki kemungkinan paling sedikit di antara *tile* yang masih belum di-*collapse*. Proses ini mengulangi langkah-langkah sebelumnya hingga semua *tile* di *grid* terisi dan tidak ada lagi *tile* yang memiliki kemungkinan bentuk yang tersisa.

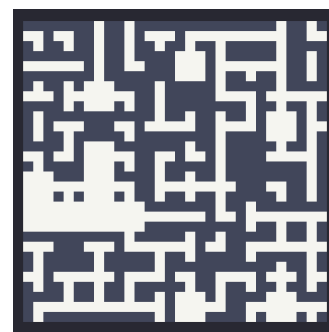


Fig.11. Hasil awal penerapan WFC

Setelah algoritma dijalankan, hasil menunjukkan *dungeon generation* yang tidak begitu baik. Perbaikan pertama yang dapat dilakukan adalah dengan meningkatkan kemungkinan terpilihnya *tile room* dan *tile void* dibandingkan *tile* jalan. Dengan menambahkan faktor kemungkinan yang memiliki bobot, algoritma akan lebih sering memilih *tile room*. Bobot yang digunakan adalah 50 untuk *tile room*, 20 untuk *tile void*, dan 5 untuk *tile* jalan.



Fig.12. Hasil penerapan WFC dengan kemungkinan yang berbobot

Perbaikan selanjutnya yang dapat dilakukan adalah dengan menghapus *tile* yang tidak terhubung sama sekali (*isolated tile*). Hal ini dapat dilakukan dengan mengecek apakah bagian sisi dari *tile* terhubung. Setiap bentuk *tile* dapat direpresentasikan dengan karakter 'W' (warna putih) atau 'B' (warna hitam) dan membentuk *grid* 3 x 3. misal *tile room* direpresentasikan dengan semua karakter 'W' pada *grid* 3 x 3 atau pada *tile crossroad* direpresentasikan sebagai berikut.

“B W B”
 “WWW”
 “B W B”

Program dapat mengecek apakah jika suatu sisi pada *tile* memiliki karakter ‘W’ bersebelahan dengan *tile* yang memiliki karakter ‘W’ pada sisi yang berdekatan. Hal ini dapat mengidentifikasi *isolated tile*. Selain itu, dapat juga diidentifikasi apakah terdapat *tile roads* yang tidak terhubung sama sekali dengan *tile room*. Hal ini dapat dicapai dengan mengecek apakah suatu *tile* adalah *tile* jalan. Lalu, cek keterhubungan *tile* tersebut dengan *tile* disekitarnya. Jika terhubung dengan *room*, maka *tile* jalan tersebut tidak perlu dihapus. Jika tidak, namun terhubung dengan *tile* jalan yang lain, dapat dilakukan pengecekan BFS (*Breadth First Search*) untuk semua jalan yang terhubung apakah terhubung dengan suatu *room*.

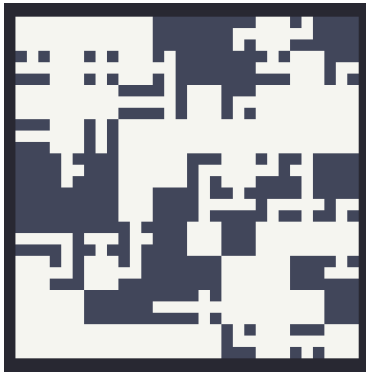


Fig.13. Hasil penerapan WFC setelah menghilangkan *isolated tile* dan *isolated roads*.

Selanjutnya, untuk menerapkan *graph coloring* pada setiap *room* di *dungeon*, perlu diidentifikasi seluruh *room* secara keseluruhan. *Tile room* yang berdekatan dan terhubung dihitung sebagai satu *room*. Dengan menggunakan BFS, dapat dicek apakah suatu *tile* adalah *tile room* dan apakah terhubung dengan *tile room* yang lain. Lalu identifikasi sebagai satu *room*.

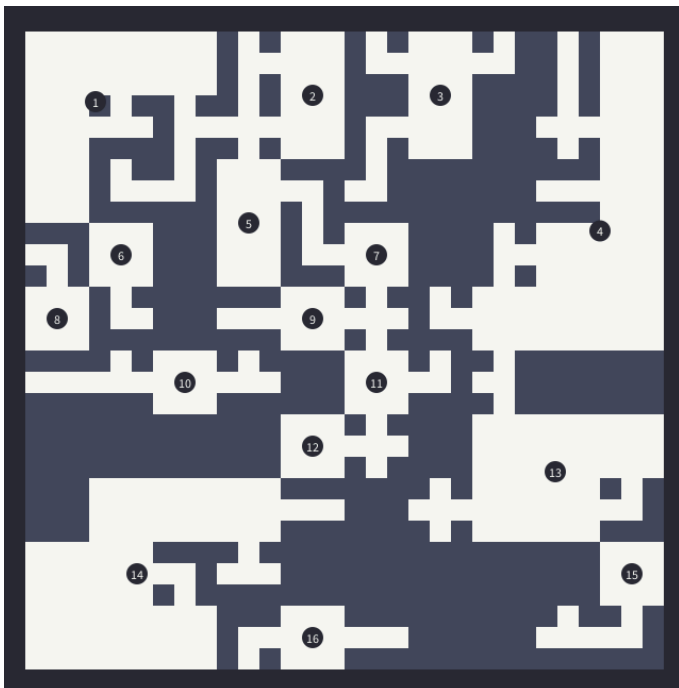


Fig.14. Hasil penerapan WFC setelah identifikasi *room*.

Dalam program identifikasi *room*, dapat juga dibuat *nodes* yang menandakan ruangan-ruangan tersebut. Kemudian, agar tercipta suatu graf dengan simpul yang saling terhubung, perlu diidentifikasi apakah ada *tile* jalan yang menghubungkan satu *room* dengan *room* yang lainnya. Jika ada, representasi keterhubungan ini dapat digambarkan sebagai sisi graf. Metode untuk menentukan keterhubungan antar ruangan juga menggunakan algoritma bersifat BFS.

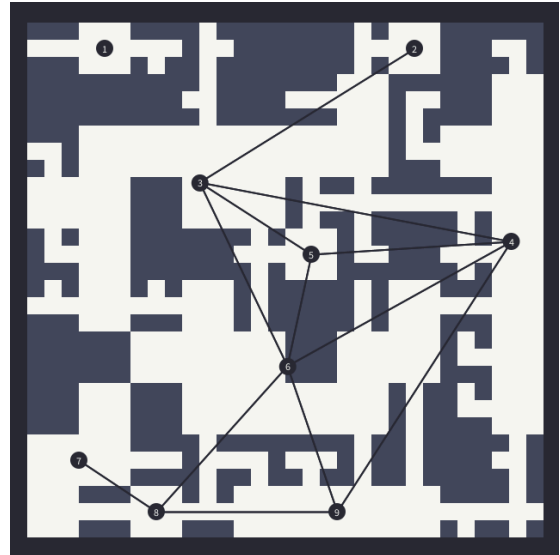


Fig.15. Hasil penerapan WFC dengan representasi graf.

Di sini dapat dilakukan perbaikan lebih lanjut untuk algoritma WFC yaitu dengan menghilangkan *room* yang tidak terhubung sama sekali. Kita hanya perlu menghilangkan simpul yang tidak memiliki sisi.

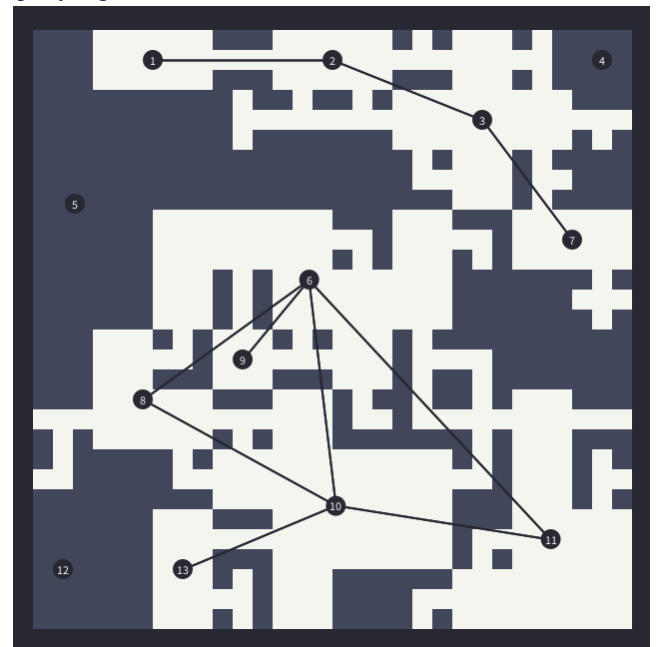


Fig.16. Hasil penerapan WFC setelah *isolated room* dihapus.

Pada figur 16, simpul yang ruangnya dihapus tidak ikut dihilangkan sebagai bentuk visualisasi. Akhirnya setelah semua perbaikan dan optimisasi, kita dapat menerapkan algoritma Welsh-Powell untuk melakukan *graph coloring* pada representasi graf dari *dungeon* WFC.

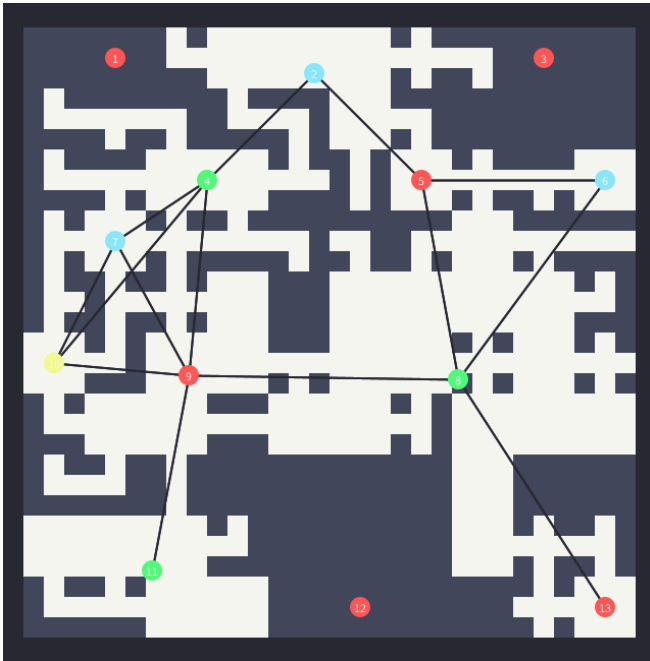


Fig.17. Hasil penerapan WFC setelah dilakukan *graph coloring*.

Dengan demikian, kita dapat memastikan bahwa setiap ruangan yang terhubung dan berdekatan, tidak akan memiliki warna yang sama. Dalam konteks pengembangan *video game*, warna dapat diganti menjadi jenis ruangan. Konteks *graph coloring* di atas merupakan bentuk menentukan bilangan kromatik. Bagaimana jika *dungeon* memiliki bilangan kromatik yang pasti, menandakan banyaknya jenis ruangan dalam satu peta?

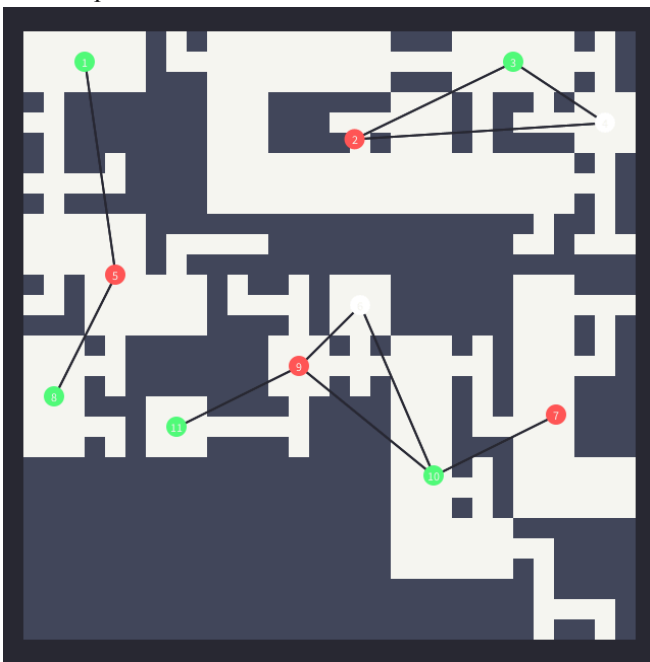


Fig.18. Hasil penerapan WFC setelah dilakukan *graph coloring* dengan bilangan kromatik 2.

Figur 18 menunjukkan *graph coloring* yang menggunakan bilangan kromatik 2. Simpul berwarna putih menandakan tidak ada warna yang tersedia untuk simpul tersebut.

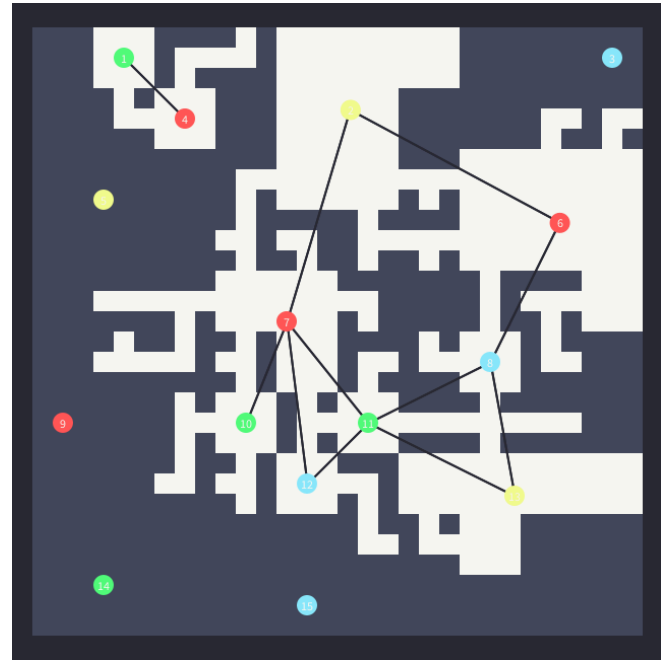


Fig.19. Hasil penerapan WFC setelah dilakukan *graph coloring* dengan bilangan kromatik 4.

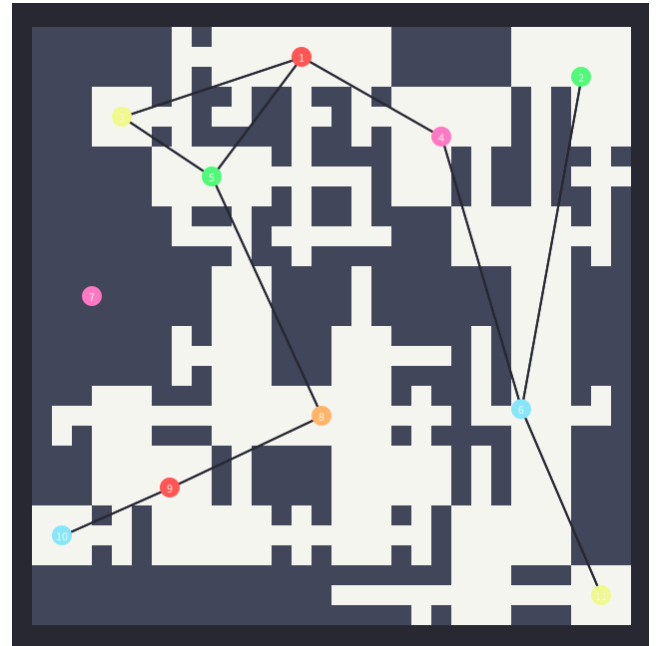


Fig.20. Hasil penerapan WFC setelah dilakukan *graph coloring* dengan bilangan kromatik 6.

V. KESIMPULAN

Dari pembahasan yang telah disampaikan, dapat disimpulkan bahwa kedua teknik, yaitu L-System dan Wave Function Collapse (WFC), memiliki potensi besar dalam dunia pengembangan *game* khususnya dalam pembuatan peta dunia yang dinamis dan menarik.

L-System, yang pada awalnya digunakan dalam biologi untuk menggambarkan pertumbuhan organisme, memiliki kemampuan dalam menghasilkan bentuk-bentuk organik seperti pohon, tanaman, dan struktur alam lainnya. Keunggulan utama dari L-System adalah kemampuannya untuk menghasilkan pola fraktal yang sangat kompleks dengan hanya

menggunakan iterasi dan aturan-aturan sederhana. Walaupun L-System memiliki keterbatasan dalam menghasilkan struktur mekanik atau bangunan yang lebih *rigid*, teknik ini sangat efektif dalam menciptakan lingkungan alami dalam permainan.

Wave Function Collapse (WFC), yang mengadaptasi prinsip-prinsip dari fisika kuantum, memungkinkan pembuatan dunia game secara prosedural dengan menjaga konsistensi antar elemen yang terhubung, seperti *room*, *void*, dan *roads* pada peta *dungeon*. WFC bekerja dengan memilih *tile* dengan kemungkinan terkecil dan "meng-collapse" *tile* tersebut sehingga menghasilkan struktur yang valid. Dalam penerapannya pada generasi *dungeon*, algoritma ini terbukti mampu menciptakan peta yang variatif, meskipun beberapa perbaikan perlu dilakukan, seperti penyesuaian bobot pemilihan *tile* dan penghapusan *tile* yang terisolasi. Penerapan *Graph Coloring* pada WFC juga menunjukkan cara efektif untuk memastikan bahwa room-room yang berdekatan tidak memiliki jenis yang sama, menghasilkan peta yang lebih terstruktur dan menyenangkan untuk dieksplorasi.

VI. APPENDIX

Sebagai tambahan material, di bawah ini merupakan *link* GitHub *repository* yang berisi kode yang digunakan untuk visualisasi menggunakan *java processing library*:

<https://github.com/V-Kleio/Procedural-Generation>

REFERENCES

- [1] Hendrixx, Mark & Meijer, Sebastiaan & Velden, Joeri & Iosup, Alexandru. (2013). Procedural Content Generation for Games: A Survey. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP). 9. 10.1145/2422956.2422957.
- [2] T. Gao, J. Zhang and Q. Mi, "Procedural Generation of Game Levels and Maps: A Review," 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Jeju Island, Korea, Republic of, 2022, pp. 050-055, doi: 10.1109/ICAIIIC54071.2022.9722624. keywords: {Costs;Scalability;Sociology;Games;Production;Cognition;Artificial intelligence;procedural content generation;game levels and maps;game artificial intelligence},
- [3] H. B. Prusinkiewicz and A. Lindenmayer, "*The Algorithmic Beauty of Plants*," *Algorithmic Botany*, 2003, pp. 1.
- [4] Javadi, Hossein. (2021). Wave function collapse and CPH Theory.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 27 Desember 2024



Rafael Marchel Darma Wijaya – 13523146