

Penerapan Teori Graf pada Pembuatan Tata Letak Labirin Bawah Tanah pada Permainan Enter the Gungeon

Chokyi Ozer 13518107
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518107@std.stei.itb.ac.id

Abstrak—Enter the Gungeon merupakan sebuah permainan video yang memiliki tingkat-tingkat yang dibuat secara prosedural. Setiap tingkat memiliki sejumlah ruangan yang saling terhubung. Setiap ruangan dapat dilihat sebagai sebuah simpul pada graf, sehingga satu tingkat pada permainan dapat dimodelkan dengan sebuah graf. Makalah ini berisi pembahasan mengenai cara Enter the Gungeon membuat sebuah tingkat terstruktur secara prosedural dengan menggunakan bentukan graf.

Kata Kunci—Pembuatan labirin, Graf, Pembuatan secara prosedural, Enter the Gungeon

I. PENDAHULUAN

Enter the Gungeon adalah sebuah permainan video tembak-menembak seperti Rogue yang dibuat oleh Dodge Roll dan di publikasi oleh Devolver Digital. Enter the Gungeon dapat dimainkan sendiri atau berdua diluar jaringan (*offline*). Permainan ini memiliki tujuan yang sederhana, yaitu menelusuri sebuah labirin bawah tanah dan melawan bos pada tiap tingkat. Setiap tingkat diisi dengan berbagai ruangan yang saling terhubung, dan setiap ruangan memiliki elemen-elemen tertentu, seperti musuh, toko, dan harta karun.



Gambar 1 Tangkapan layar dari Enter The Gungeon (<https://dodgeroll.com/gungeon/> diakses pada tanggal 2 Desember 2019)

Setiap tingkat dibuat secara acak, sehingga setiap permainan dimulai, tata letak ruangan serta isinya akan terlihat berbeda.

Akan tetapi tata letak dari ruangan-ruangan pada suatu tingkat terlihat terstruktur. Contohnya, bos-bos pada tiap tingkat akan memiliki jarak tertentu dari ruangan awal tiap tingkat. Ruangan-ruangan tanpa musuh ditempatkan secara rata agar pemain dapat memiliki waktu sejenak untuk beristirahat. Penataan ruangan juga terlihat padat dan jarak antar ruangan tidak jauh.

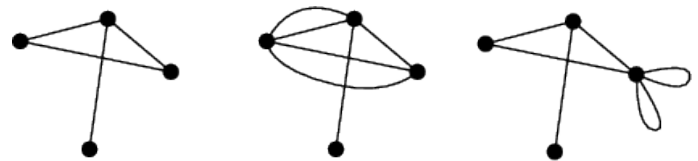
Pada makalah ini, penulis akan memaparkan cara kerja dari penataan tiap ruangan pada permainan ini.

II. LANDASAN TEORI

A. Graf

Graf merupakan kumpulan dari sejumlah titik dan busur. Titik-titik pada sebuah graf biasa dikenal dengan sebutan simpul, sementara busur dapat disebut sebagai sisi. Sisi-sisi pada graf menghubungkan dua titik bagian dari graf tersebut. Berdasarkan adanya gelang atau sisi ganda, graf dapat dibagi menjadi tiga jenis [1], yaitu

1. Graf sederhana
Graf sederhana adalah graf yang tidak mengandung sisi ganda ataupun gelang. Sisi ganda merupakan dua atau lebih sisi yang menghubungkan dua simpul tertentu. Gelang adalah sisi yang menghubungkan simpul yang sama.
2. Graf ganda
Graf yang memiliki sisi ganda, tapi tidak dapat memiliki gelang disebut sebagai sebuah graf ganda.
3. Graf semu
Graf semu merupakan sebuah graf yang dapat mengandung baik sisi ganda maupun gelang.



simple graph

multigraph

pseudograph

Gambar 2 Contoh graf sederhana (kiri), graf ganda (tengah), dan graf semu (kanan)

(<http://mathworld.wolfram.com/Graph.html> diakses pada tanggal 2 Desember 2019)

Selain itu, berdasarkan adanya orientasi arah pada sisi, graf dapat dibagi menjadi dua jenis [2], yaitu

1. Graf tak-berarah
Graf tak-berarah adalah graf biasa yang sisinya tidak berarah
2. Graf berarah
Graf yang sisinya memiliki arah, yang mengindikasikan bahwa sebuah titik menunjuk pada titik lain. Variasi dari graf bertipe ini adalah graf berorientasi, yaitu graf yang tidak memiliki arah bolak-balik [1].



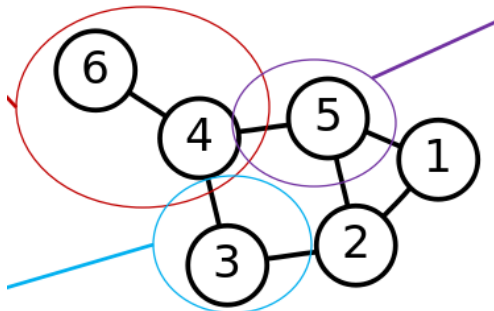
undirected graph oriented graph directed graph

Gambar 3 Contoh graf tak-berarah (kiri), graf berorientasi (tengah), dan graf berarah (kanan)

(<http://mathworld.wolfram.com/Graph.html> diakses pada tanggal 2 Desember 2019)

Pada teori graf, terdapat beberapa kosakata yang harus dikenali, yakni

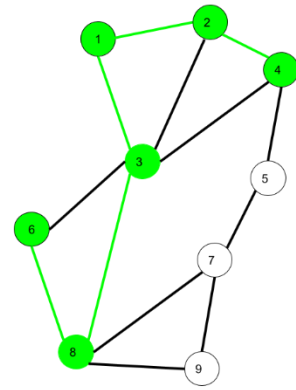
1. Ketetangaan
Dua simpul yang terhubung dengan sebuah sisi secara langsung.
2. Bersisian
Sebuah simpul dan sebuah sisi dikatakan bersisian jika sisi tersebut menghubungkan simpul tersebut dengan simpul lain.
3. Derajat
Derajat adalah banyaknya hubungan dengan tetangga / sisi yang terhubung dari sebuah simpul.



Gambar 4 Contoh ketetangaan (lingkaran merah), bersisian (lingkaran biru), dan derajat (3) (lingkaran ungu)

(https://stei.kuliah.itb.ac.id/pluginfile.php/4601/mod_resource/content/2/W14_B1_Graph.pdf diakses pada tanggal 4 Desember 2019)

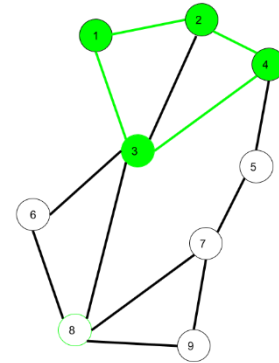
4. Upagraf
Sebuah graf G' dikatakan upagraf dari sebuah graf G jika sisi-sisi pada G' merupakan himpunan bagian dari sisi-sisi pada G .
5. Lintasan
Sebuah lintasan merupakan sisi-sisi yang dilewati untuk mencapai suatu simpul pada graf dari sebuah simpul pada graf lainnya.



Gambar 5 Contoh lintasan (hijau)

(<https://www.geeksforgeeks.org/mathematics-walks-trails-paths-cycles-and-circuits-in-graph/> diakses pada tanggal 4 Desember 2019)

6. Sirkuit
Sebuah sirkuit pada graf adalah sebuah lintasan yang berawal dan berakhir pada simpul yang sama.



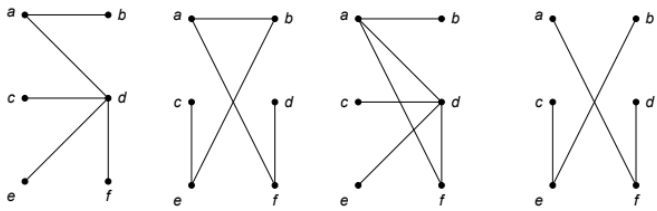
Gambar 6 Contoh sirkuit (hijau)

(<https://www.geeksforgeeks.org/mathematics-walks-trails-paths-cycles-and-circuits-in-graph/> diakses pada tanggal 4 Desember 2019)

7. Simpul Terhubung
Dua simpul dikatakan terhubung jika terdapat sebuah lintasan yang dapat menghubungkan kedua simpul.
8. Graf Terhubung
Sebuah Graf dinyatakan terhubung jika semua pasang simpul pada graf tersebut terhubung.
9. Komplemen
Komplemen dari sebuah upagraf G' terhadap graf G adalah sebuah upagraf yang himpunan sisinya merupakan sisi-sisi pada graf G yang bukan bagian dari upagraf G' .
10. Upagraf Rentang
Upagraf rentang dari graf G memiliki semua simpul yang terdapat pada G .

B. Pohon

Sebuah pohon merupakan sebuah variasi dari graf. Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit [3].



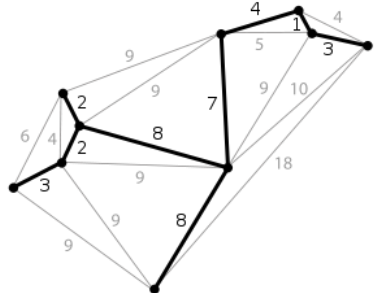
pohon pohon bukan pohon bukan pohon

Gambar 7 Contoh-contoh pohon

([http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) diakses pada tanggal 4 Desember 2019)

C. Pohon Merentang Minimum

Pohon merentang dari suatu graf didefinisikan sebagai sebuah upagraf dari graf tersebut yang merupakan sebuah pohon. Pohon merentang dari suatu graf juga harus menghubungkan semua simpul pada graf tersebut, baik langsung atau tidak langsung.



Gambar 8 Contoh pohon merentang minimum dari sebuah graf (https://en.wikipedia.org/wiki/Minimum_spanning_tree diakses pada tanggal 4 Desember 2019)

Pohon merentang minimum adalah sebuah pohon merentang dari sebuah graf terhubung-berbobot yang memiliki bobot minimum. Terdapat beberapa cara untuk mencari pohon merentang minimum, yaitu dengan algoritma Prim dan algoritma Kruskal [3].

1. Algoritma Prim

Algoritma Prim mengikuti langkah-langkah berikut:

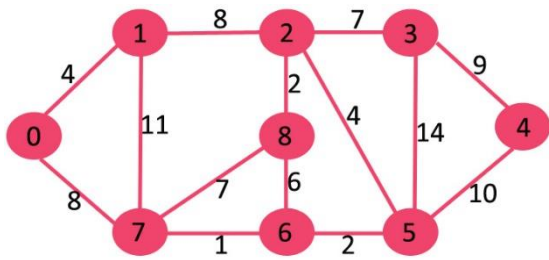
 - a. Masukkan sisi dengan bobot terendah pada graf G ke dalam pohon T.
 - b. Masukkan sisi dari graf G yang:
 - i. Bersisian dengan salah satu simpul pada pohon T
 - ii. Tidak membentuk sirkuit pada T dengan bobot paling kecil.
 - c. Ulangi langkah (b) hingga seluruh simpul pada graf G terdapat pada pohon T.
2. Algoritma Kruskal

Algoritma Prim mengikuti langkah-langkah berikut:

 - a. Urutkan semua sisi pada G dari yang memiliki bobot terkecil hingga yang terbesar.
 - b. Masukkan sisi dengan bobot minimum dari graf G jika tidak membuat sirkuit pada T.
 - c. Ulangi langkah (b) hingga seluruh simpul pada graf G terdapat pada pohon T.

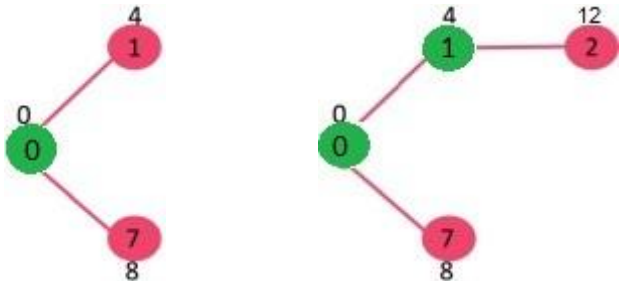
Selain algoritma diatas, terdapat juga sebuah algoritma yang dapat mencari jarak terpendek antara suatu simpul ke simpul lainnya, yaitu algoritma Dijkstra. Algoritma Dijkstra serupa dengan algoritma Prim, tetapi dioptimalkan untuk menentukan

bobot terendah dihitung berdasarkan jarak dari suatu simpul, bukan berdasarkan bobot suatu graf secara keseluruhan. Jika ingin dicari jarak dari sebuah simpul v terhadap simpul-simpul lain, maka algoritma Dijkstra bekerja melalui langkah-langkah berikut:



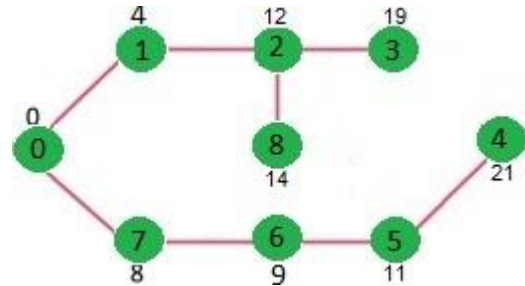
Gambar 9 Contoh graf berbobot yang akan diproses (<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/> diakses pada tanggal 3 Desember 2019)

- a. Masukkan simpul v kedalam pohon T, dengan bobot pada simpul v bernilai 0.
- b. Untuk setiap simpul u yang terhubung dengan simpul w pada pohon T, bobot pada simpul yang dimaksud memiliki nilai yang sama dengan bobot simpul yang terhubung dengan u pada pohon T dijumlahkan dengan bobot sisi yang menghubungkan u dan v pada graf G.



Gambar 10 Proses algoritma Dijkstra (<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/> diakses pada tanggal 3 Desember 2019)

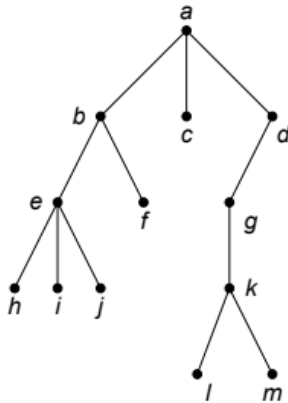
- c. Ulangi langkah (b) hingga seluruh simpul pada graf G terdapat pada pohon T.



Gambar 11 Hasil pencarian jarak terpendek menggunakan algoritma Dijkstra dari simpul 0 (<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/> diakses pada tanggal 3 Desember 2019)

D. Pohon Berakar

Sebuah pohon berakar adalah sebuah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah. Terdapat beberapa terminologi yang berhubungan dengan pohon berakar, beberapa kosakata penting adalah sebagai berikut:



Gambar 12 Sebuah pohon berakar yang memiliki nama pada tiap simpulnya

([http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) diakses pada tanggal 4 Desember 2019)

- Akar**
Akar adalah titik awal dari sebuah pohon, semua cabang akan konvergen menuju akar. Pada gambar (12), a adalah akar dari pohon tersebut.
- Anak dan Orangtua**
Sebuah simpul v dikatakan anak dari simpul lainnya u jika v bertetangga dengan u dan v lebih jauh dari akar. Sebaliknya, simpul v dikatakan orangtua dari u jika v bertetangga dengan u tapi v lebih dekat dari akar. Pada gambar (12), a adalah orangtua dari b, c, dan d. Sebaliknya, b, c, dan d adalah anak dari a.
- Saudara kandung**
Saudara kandung adalah dua atau lebih simpul yang memiliki orangtua yang sama. Pada gambar (12), b merupakan saudara kandung c dan d.
- Upapohon**
Upapohon adalah sebutan dari upagraf dari sebuah pohon. Pada gambar (12), pohon d-g-k-l-m adalah upapohon dari pohon tersebut.
- Daun**
Daun adalah sebuah simpul dari sebuah pohon yang tidak memiliki anak. Pada gambar (12), h, i, j, f, c, l, dan m adalah daun dari pohon tersebut.
- Simpul dalam**
Simpul yang memiliki setidaknya satu anak disebut sebagai sebuah simpul dalam. Pada gambar (12), a, b, d, e, f, g, dan k adalah simpul dalam dari pohon tersebut.
- Aras atau Tingkat**
Aras merupakan jarak sebuah simpul dari akar pohon tersebut. Akar dari sebuah pohon dianggap memiliki aras nol. Pada gambar (12), aras dari simpul g bernilai dua.
- Tinggi atau Kedalaman**
Tinggi adalah nilai aras maksimum pada sebuah pohon. Gambar (12) memiliki tinggi atau kedalaman empat.

9. Lebar

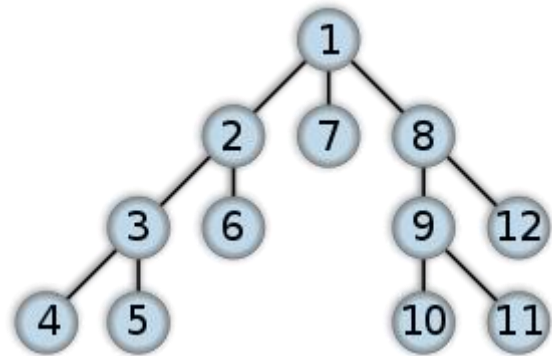
Lebar dari suatu pohon adalah jumlahnya daun pada pohon tersebut. Lebar dari pohon pada gambar (12) adalah tujuh.

Pohon merentang yang dihasilkan oleh algoritma Dijkstra adalah salah satu contoh pohon berakar.

E. Pencarian Mendalam

Pencarian mendalam merupakan suatu algoritma traversal graf yang melewati seluruh cabang dari pohon yang dimaksud sebelum melewati cabang lainnya. Algoritma pencarian mendalam mengikuti langkah langkah berikut [5]:

- Mulai dari akar pohon v.
- Kunjungi simpul w yang merupakan anak dari v.
- Ulangi pencarian mendalam dengan akar w.
- Jika pencarian mencapai daun, runut balik ke simpul yang memiliki anak w yang belum dikunjungi.
- Pencarian berakhir ketika pencarian runut balik ke simpul v dan semua anak v sudah pernah dikunjungi.



Gambar 13 Contoh alur algoritma pencarian mendalam pada pohon berakar
(https://en.wikipedia.org/wiki/Depth-first_search diakses pada tanggal 4 Desember 2019)

III. ALUR PEMBUATAN LABIRIN BAWAH TANAH PADA ENTER THE GUNGEON

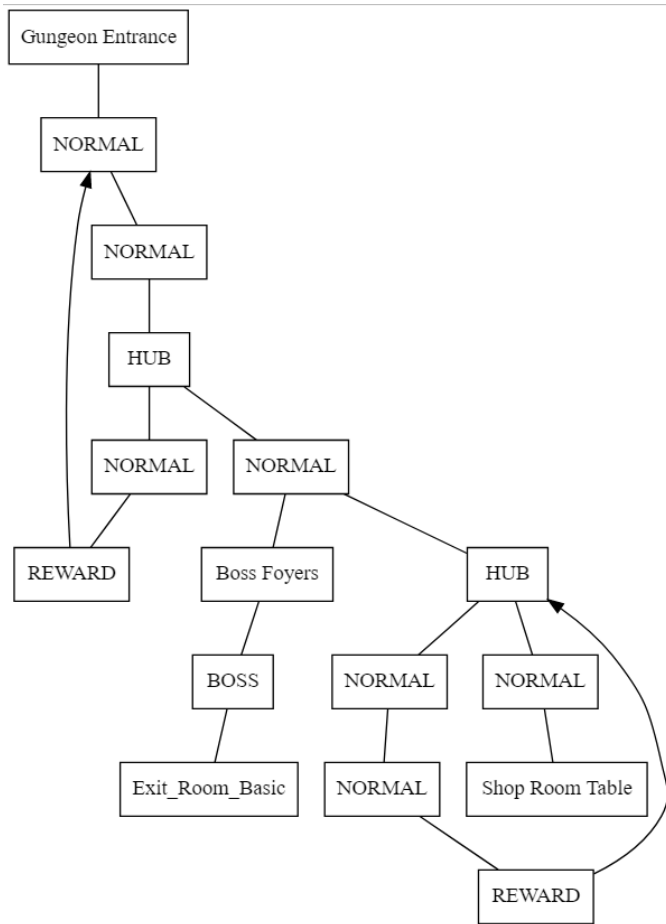
Pembuatan labirin pada tiap tingkat pada Enter the Gungeon dapat didekomposisi menjadi sejumlah langkah seperti penjelasan berikut:

A. Penentuan Alur Tata Ruang Tingkat

Setiap tingkat pada permainan Enter the Gungeon memiliki ciri khasnya sendiri. Sebagai contoh, tingkat satu *Keep of the Lead Lord* memiliki belasan ruangan untuk dijelajahi, sementara tingkat-tingkat yang dalam seperti tingkat lima *The Forge* memiliki lebih dari 20 ruangan.

Ciri khas dari setiap tingkat ditentukan dengan beberapa cara, salah satunya adalah dengan tata letak ruangan pada tiap tingkat. Setiap tingkat pada Enter the Gungeon memiliki beberapa tata letak atau alur yang dibuat sebelumnya oleh pembuat permainan. Salah satu dari alur tata letak dipilih secara acak untuk dijadikan dasar dari penempatan ruangan. Alur tata letak ini direpresentasikan dalam sebuah graf, sehingga hanya menjelaskan keterhubungan antar ruang, bukan posisinya. Oleh karena itu, posisi dari tiap ruangan dapat divariasikan sehingga menghasilkan labirin yang tampak lebih variatif.

Berdasarkan data dari Spleen, pengguna imgur, salah satu alur pada tingkat 1 adalah sebagai berikut [6]:



Gambar 14 Salah satu alur tata letak yang dapat dipilih saat pembuatan labirin bawah tanah (<https://www.boristhebrave.com/2019/07/28/dungeon-generation-in-enter-the-gungeon/> diakses pada tanggal 3 Desember 2019)

Graf yang digunakan untuk merepresentasikan alur tata letak dari suatu tingkat berbentuk graf berarah dengan varian graf berorientasi. Pada gambar (14), hanya sisi yang menutup siklus yang digambarkan berarah, tetapi pada kenyataannya semua sisi memiliki arah tertentu.

B. Penambahan Ruang pada Alur Tata Ruang

Untuk memperbanyak variasi pada alur, setiap tingkat dapat mengandung ruang-ruang tambahan yang tidak terdapat pada alur dasar. Penambahan ruangan dapat dilakukan dengan tiga cara.

1. Pemanjangan Simpul

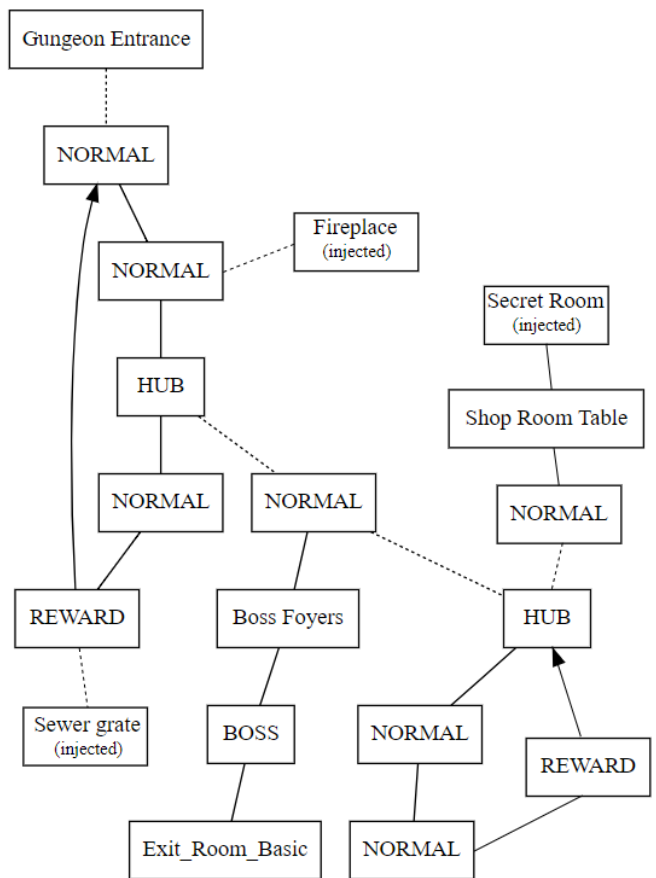
Pada tingkat-tingkat yang lebih dalam, suatu ruangan pada alur dapat diperpanjang menjadi beberapa deretan ruangan.
2. Injeksi Ruang

Ruangan dapat diinjeksi atau ditempel di sebelah ruangan lain karena beberapa kondisi yang dimiliki pemain. Kondisi-kondisi yang dimaksud meliputi memiliki kutukan yang berat, belum menyelamatkan seorang karakter, dan sebagainya. Selain itu sebuah ruangan tersembunyi juga dapat ditambahkan kedalam alur menggunakan cara ini.
3. Penambahan Lorong

Penambahan lorong dilakukan untuk mempermudah penghubungan dua ruangan. Penambahan ruangan dengan cara ini dilakukan tidak pada langkah ini, tetapi pada langkah selanjutnya.

C. Dekomposisi menjadi Bagian-Bagian

Alur yang sudah dibuat kemudian dibagi menjadi beberapa bagian. Setiap bagian pada sebuah alur merupakan antara sebuah sirkuit atau sebuah pohon. Hal tersebut dilakukan dengan mencari sirkuit terpendek pada alur tersebut, dan memisahkannya dari graf. Pencarian sirkuit terpendek dapat dilakukan dengan cara menggunakan algoritma pencarian jarak terpendek seperti algoritma Dijkstra. Hal tersebut dilakukan hingga tidak ada sirkuit yang tersisa pada alur [7]. Setelah selesai, alur akan berbentuk sebagai berikut:



Gambar 15 Hasil penambahan ruangan melalui injeksi dan dekomposisi menjadi beberapa bagian (garis putus-putus menandakan hubungan antarbagian) (<https://www.boristhebrave.com/2019/07/28/dungeon-generation-in-enter-the-gungeon/> diakses pada tanggal 3 Desember 2019)

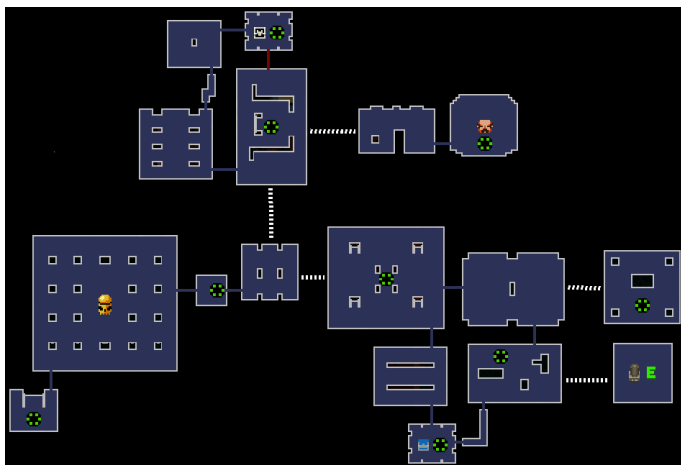
D. Penempatan Ruang

Setelah alur dibagi menjadi sejumlah bagian, penempatan ruangan dilakukan pada tiap bagian.

Untuk bagian bertipe siklus, ruangan dibuat dari kedua ujung siklus secara bergantian. Ruangan ditentukan secara acak dari katalog ratusan ruangan, dengan memprioritaskan penempatan ruangan dengan jalan keluar yang saling berseberangan. Setelah siklus sudah setengah jadi, ruangan yang dipilih untuk dibuat

adalah ruangan yang dapat menyatukan kedua belah dari siklus. Ketika semua ruangan sudah jadi, sebuah lorong sempit akan dibuat untuk menghubungkan kedua ruangan yang pertama dibuat agar dapat menutupi siklus. Lorong ini dibuat dengan menggunakan algoritma pencarian jarak terdekat antara jalan keluar kedua ruangan.

Untuk bagian yang bertipe pohon, ruangan akan dibuat dengan teknik DFS atau pencarian mendalam. Penentuan ruangan juga dilakukan secara acak dan diambil dari daftar ruangan yang sudah dibuat oleh pembuat permainan. Jika ruangan yang dipilih tidak muat, yakni ruangan yang dipilih terlalu besar atau ruang kosong yang tersisa kurang besar, maka akan dipilih ruangan baru maksimal hingga tiga kali pemilihan. Jika setelah tiga kali pemilihan masih tidak muat, maka akan dilakukan runut-balik untuk memilih ulang ruangan sebelumnya.



Gambar 16 Peta yang menunjukkan hasil penempatan ruang-ruang

(<https://www.boristhebrave.com/2019/07/28/dungeon-generation-in-enter-the-gungeon/> diakses pada tanggal 4 Desember 2019)

E. Penyatuan Bagian-Bagian

Setelah ruangan sudah terbuat semua, bagian-bagian yang terpisah akan digabung kembali sesuai dengan bentuknya pada alur. Setiap bagian akan disatukan dengan jarak terpendek. Jika diperlukan, koneksi antar bagian akan menggunakan algoritma pencarian jarak terpendek.

Setelah menempatkan ruangan, ruangan akan diisi dengan berbagai hal. Hal-hal yang diisi kedalam ruangan dipengaruhi oleh tingkat ruangan pada labirin bawah tanah, jenis ruangan itu sendiri, ataupun status dan barang yang dimiliki pemain.



Gambar 17 Peta hasil penyatuan bagian-bagian. Peta ini adalah contoh peta yang ditampilkan ke pemain pada permainan (<https://www.boristhebrave.com/2019/07/28/dungeon-generation-in-enter-the-gungeon/> diakses pada tanggal 4 Desember 2019)

IV. KESIMPULAN

Teori graf, khususnya bentukan graf dan pohon, memiliki sangat banyak kegunaan dalam dunia permainan. Salah satu diantara kegunaannya adalah pembuatan labirin secara prosedural. Hal ini dapat dilihat pada pembuatan tingkat-tingkat pada permainan Enter the Gungeon.

Salah satu algoritma yang berguna pada pembuatan labirin adalah algoritma pencarian jarak terpendek. Pembuatan tingkat dapat dipisah-pisah menjadi sejumlah bagian sebelum dihubungkan kembali menggunakan algoritma pencarian jarak terpendek.

VII. UCAPAN TERIMA KASIH

Puji syukur pada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga makalah ini dapat diselesaikan. Penulis ingin berterima kasih kepada Dra. Harlili M.Sc selaku dosen mata kuliah IF 2120 Matematika Diskrit atas bimbingannya selama satu semester. Penulis juga ingin berterima kasih kepada semua sumber referensi sehingga makalah ini bisa memaparkan informasi yang sebenar-benarnya.

DAFTAR REFERENSI

- [1] Wolfram Research, <http://mathworld.wolfram.com/Graph.html>, diakses tanggal 4 November 2019

- [2] Munir, Rinaldi, [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf), diakses tanggal 4 November 2019
- [3] Munir, Rinaldi, [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf), diakses tanggal 4 November 2019
- [4] Tim Geeks for Geeks, <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>, diakses pada tanggal 4 November 2019
- [5] Munir, Rinaldi, [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/BFS-dan-DFS-\(2019\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/BFS-dan-DFS-(2019).pdf), diakses tanggal 4 November 2019
- [6] OJSN, https://www.speedrun.com/Enter_the_Gungeon/guide/1ijr7, diakses tanggal 4 November 2019
- [7] Boris, <https://www.boristhebrave.com/2019/07/28/dungeon-generation-in-enter-the-gungeon/>, diakses tanggal 4 November 2019

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, 4 Desember 2019



Choky Ozer
13518107