

Penerapan Algoritma A* untuk Menentukan Jalan Terbaik pada Berbagai Medan di Permainan Simulasi Koloni RimWorld

Fata Nugraha 13517109
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
fata@students.itb.ac.id

Abstraksi—Masalah penentuan rute terpendek sudah menjadi permasalahan umum di dunia informatika. Secara umum, masalah ini dapat dibagi menjadi pencarian rute terpendek tanpa bobot dan dengan bobot. Algoritma A* adalah salah satu algoritma yang digunakan untuk menyelesaikan masalah ini, dan makalah ini akan membahas penerapan algoritma A* untuk menentukan rute terpendek pergerakan kolonis pada permainan simulasi koloni RimWorld.

Kata kunci—Algoritma A*, Graf, Kolonis, Medan Tempuh, RimWorld

I. PENDAHULUAN

Graf adalah salah satu cabang ilmu yang dipelajari di matematika diskrit, dan termasuk cabang ilmu dengan penerapan yang sangat banyak di dunia nyata. Salah satu penerapannya adalah untuk mendesain algoritma pencari jalan terbaik, yang sekarang ini banyak digunakan di aplikasi-aplikasi berbasis GPS. Contoh dari algoritma ini adalah algoritma A*, yang merupakan pengembangan dari algoritma Dijkstra yang bisa digunakan untuk mencari jalur tercepat pada daerah yang memiliki bobot tertentu untuk setiap tempat yang dikunjunginya. Algoritma ini juga bisa diterapkan pada permainan-permainan tertentu yang mengharuskan karakternya untuk bergerak dari suatu tempat ke tempat lain melalui berbagai medan, karena tokoh pemain dari permainan tersebut harus bisa menemukan jalan yang paling efektif untuk menghemat waktu perjalanannya.

Salah satu permainan yang memanfaatkan teori ini adalah sebuah permainan simulasi koloni yang bernama RimWorld. Pada permainan ini, setiap kolonis akan secara otomatis bergerak untuk menjalankan tugasnya masing-masing, dan sebisa mungkin akan mencari jalan terbaik yang bisa dilalui untuk menghemat waktu perjalanannya, misalnya dengan memilih rute melalui jalan beraspal daripada rute yang melalui rawa atau tanah berlumpur. Karena semua pergerakan yang dilakukan kolonis adalah pergerakan dari suatu posisi ke posisi lainnya, kita bisa melihat penerapan algoritma A* pada permainan ini.



Gambar 1.1 RimWorld

Sumber : <https://steamcdn-a.akamaihd.net/steam/apps/294100/header.jpg?t=1541837948>

II. TEORI GRAF

A. Definisi Graf

Graf adalah sebuah struktur diskrit yang terdiri dari simpul dan sisi yang menghubungkan simpul-simpul tersebut [1]. Graf juga dapat didefinisikan sebagai pasangan himpunan simpul-simpul dan himpunan sisi yang menghubungkan sepasang simpul, dengan himpunan simpulnya adalah himpunan tak kosong. Artinya, syarat agar sebuah graf bisa terdefinisi adalah setidaknya graf tersebut memiliki satu buah simpul [3]. Graf bisa dikategorikan berdasarkan ada atau tidaknya gelang atau sisi ganda pada graf tersebut. Selain itu, bisa juga dikategorikan dengan melihat apakah graf tersebut memiliki sisi berarah atau tidak.

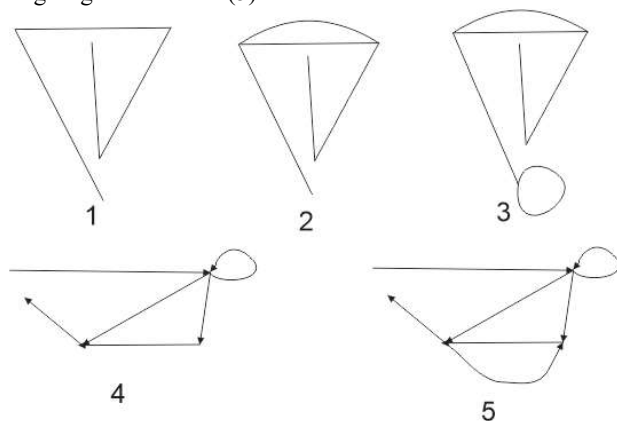
Berdasarkan ada atau tidaknya sisi gelang atau sisi ganda pada suatu graf, graf dibedakan menjadi dua kategori yaitu graf sederhana dan graf tidak sederhana. Graf sederhana adalah graf yang tidak mengandung sisi gelang atau sisi ganda. Kebalikannya, graf tak sederhana adalah graf yang mengandung sisi gelang atau sisi ganda.

Berdasarkan keberadaan sisi berarah pada graf, graf juga dibagi menjadi dua kategori. Kategori yang pertama adalah graf tidak berarah, yaitu graf yang sisi-sisinya tidak memiliki arah. Kategori yang kedua adalah graf berarah, yaitu graf yang sisi-sisinya memiliki arah. Berdasarkan dua jenis pengategorian ini, kita bisa membuat sebuah tabel yang berisikan jenis-jenis graf seperti berikut :

Tabel 2.1 Jenis-jenis graf

| Jenis | Sisi | Sisi ganda | Sisi gelang |
|--------------------|-------------|------------|-------------|
| Graf sederhana | Tak berarah | Tidak | Tidak |
| Graf ganda | Tak berarah | Ya | Tidak |
| Graf semu | Tak berarah | Ya | Ya |
| Graf berarah | Berarah | Tidak | Ya |
| Graf ganda berarah | Berarah | Ya | Ya |

Gambar di bawah ini secara berurutan menggambarkan graf sederhana (1), graf ganda (2), graf semu (3), graf berarah (4), dan graf ganda berarah (5).



Gambar 2.1 Jenis-jenis graf
Sumber : Penulis

B. Terminologi pada Graf

Terdapat beberapa istilah yang berkaitan dengan graf, yaitu:

1. Bertetangga (Adjacent)

Dua buah simpul pada graf tak berarah G dikatakan bertetangga bila keduanya terhubung langsung melalui sebuah sisi.

2. Bersisian (Incident)

Sebuah sisi dikatakan bersisian dengan sebuah simpul jika sisi tersebut menghubungkan simpul tersebut ke suatu simpul lain.

3. Simpul Terpencil (Isolated Vertex)

Simpul terpencil merupakan simpul yang sama sekali tidak memiliki sisi yang bersisian dengannya, atau tidak memiliki tetangga.

4. Graf Kosong (Null Graph atau Empty Graph)

Graf kosong adalah graf yang himpunan sisinya merupakan himpunan kosong, atau sebuah graf yang tidak memiliki sisi sama sekali, hanya memiliki simpul saja.

5. Derajat (Degree)

Derajat suatu simpul pada suatu graf sama dengan banyaknya sisi yang bersisian dengan simpul tersebut.

6. Lintasan (Path)

Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2,$

\dots, e_n, v_n sedemikian sehingga $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G .

7. Siklus (Cyclic) atau Sirkuit (Circuit)

Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus.

8. Terhubung (Connected)

Dua buah simpul dikatakan terhubung jika ada lintasan dari simpul satu ke simpul lainnya.

Graf tak berarah G disebut graf terhubung jika untuk setiap pasang simpul v_i dan v_j di dalam himpunan V terdapat lintasan dari v_i ke v_j . Jika tidak, maka G disebut graf tak terhubung.

Graf berarah G dikatakan terhubung jika graf tak berarahnya terhubung (graf tak berarah dari G diperoleh dengan menghilangkan arahnya).

Graf berarah G disebut graf terhubung kuat bila untuk setiap pasang simpul sembarang v_i dan v_j di G terhubung kuat, yaitu terdapat lintasan dari v_i ke v_j dan lintasan dari v_j ke v_i . Kalau tidak terpenuhi, G disebut graf terhubung lemah.

9. Upagraf (Subgraph) dan Komplemen Upagraf

Misalkan $G = (V, E)$ adalah sebuah graf, $G_1 = (V_1, E_1)$ adalah upagraf (subgraph) dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$. Komplemen dari upagraf G_1 terhadap graf G adalah graf $G_2 = (V_2, E_2)$ sedemikian sehingga $E_2 = E - E_1$ dan V_2 adalah himpunan simpul yang anggota-anggota E_2 bersisian dengannya.

Komponen sebuah graf adalah jumlah maksimum upagraf terhubung dalam graf G . Komponen terhubung adalah upagraf terhubung dari graf G yang tidak termuat di dalam upagraf terhubung dari G yang lebih besar. Pada graf berarah, komponen terhubung kuat adalah upagraf yang terhubung kuat dari graf G yang tidak termuat di dalam upagraf terhubung kuat dari G yang lebih besar.

10. Upagraf Merentang (Spanning Subgraph)

Upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$ dikatakan upagraf merentang jika $V_1 = V$ (G_1 mengandung semua simpul dari G).

11. Cut-Set

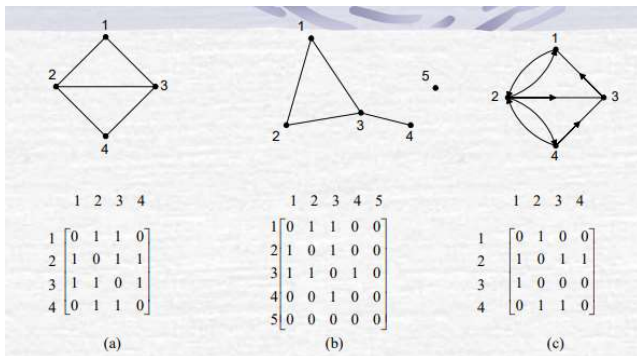
Cut-set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung. Jadi, cut-set selalu menghasilkan dua buah komponen terhubung.

12. Graf Berbobot (Weighted Graph)

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Variasi yang digunakan di permainan RimWorld adalah memberikan bobot pada setiap koordinat di peta yang berpengaruh pada kecepatan gerak kolonis.

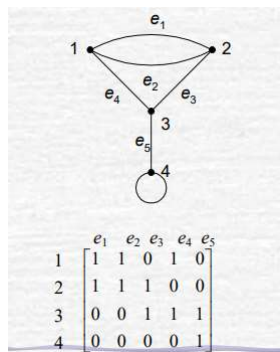
C. Representasi Graf

Sebuah graf bisa direpresentasikan dengan matriks ketetanggaan, matriks bersisian, dan senarai ketetanggaan. Berikut ini adalah contoh dari masing-masing representasi tersebut :



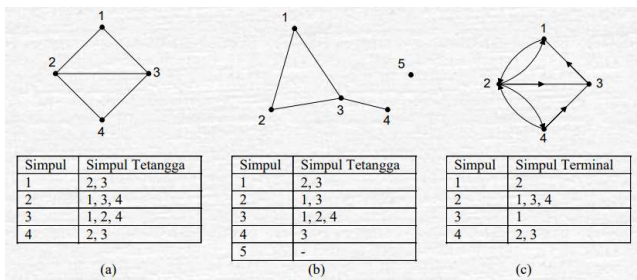
Gambar 2.2 Matriks ketetanggaan
Sumber :

[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)



Gambar 2.3 Matriks bersisian
Sumber :

[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)



Gambar 2.4 Senarai ketetanggaan
Sumber :

[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)

D. Algoritma A*

Algoritma A* adalah modifikasi dari algoritma Dijkstra yang dioptimalkan untuk satu tujuan. Algoritma Dijkstra digunakan untuk mencari jalur ke segala lokasi di peta, sedangkan algoritma A* digunakan untuk mencari jalur ke suatu lokasi di dalam peta [6]. Karena algoritma ini merupakan pengembangan dari algoritma Dijkstra, maka algoritma ini juga bisa digunakan untuk mencari jalur yang dipengaruhi oleh bobot tertentu.

Gambar berikut ini adalah contoh kode untuk algoritma A*

```

frontier = PriorityQueue()
frontier.put(start, 0)
came_from = {}
cost_so_far = {}
came_from[start] = None
cost_so_far[start] = 0

while not frontier.empty():
    current = frontier.get()

    if current == goal:
        break

    for next in graph.neighbors(current):
        new_cost = cost_so_far[current] + graph.cost(current, next)
        if next not in cost_so_far or new_cost < cost_so_far[next]:
            cost_so_far[next] = new_cost
            priority = new_cost + heuristic(goal, next)
            frontier.put(next, priority)
            came_from[next] = current

```

Gambar 2.5 Algoritma A*

Sumber : <https://www.redblobgames.com/pathfinding/a-star/introduction.html>

Gambar di bawah ini menggambarkan cara algoritma ini mencari jalan paling efektif ketika diberi suatu hambatan berupa tembok. Seperti yang bisa dilihat pada gambar tersebut, algoritma Dijkstra mencari di peta secara keseluruhan. Apabila kita hanya memerlukan satu tujuan saja, maka algoritma A* ini akan jauh lebih efektif.



Gambar 2.6 Perbandingan algoritma Dijkstra, Greedy Best-First, dan A*

Sumber : <https://www.redblobgames.com/pathfinding/a-star/introduction.html>

III. RIMWORLD

A. Perkenalan

RimWorld adalah sebuah permainan simulasi koloni yang dikembangkan oleh Tyan Sylvester dan Ludeon Studios. Permainan ini bukanlah permainan kompetitif, tetapi permainan yang menggunakan kecerdasan buatan untuk menganalisis situasi permainan saat ini dan menentukan peristiwa yang akan terjadi berikutnya. Tujuan akhir dari permainan ini adalah membuat pesawat luar angkasa dan membawa koloni kita ke luar angkasa, tetapi pada kenyataannya kebanyakan pemain tidak melakukan hal ini karena butuh waktu yang sangat lama dan kesabaran yang tinggi untuk mengembangkan suatu koloni dari awal, dan pada akhirnya memilih untuk tetap mengembangkan lebih lanjut koloni yang sudah dibuatnya, atau membuat koloni baru yang terpisah dari koloni utamanya.

B. Memulai Permainan

Pada awal permainan, kita diberikan peta dunia untuk memilih tempat awal dari koloni kita. Terdapat dua belas bioma berbeda pada permainan ini, yang dibagi menjadi tiga kategori, yaitu bioma hangat, bioma panas, dan bioma dingin. Sebuah

bioma adalah sekumpulan tempat berbentuk segi-enam yang masing-masing tempatnya memiliki karakteristiknya sendiri, seperti pegunungan, sungai, laut, atau jalan raya yang menghubungkan antar wilayah. Peta awal dunia ini juga merupakan salah satu penerapan graf, tapi hal itu tidak akan dibahas di sini.

Setelahnya, kita bisa memilih tiga kolonis awal untuk koloni kita. Setiap kolonis memiliki kelebihan dan kelemahannya masing-masing, sehingga sangat disarankan untuk memilih kombinasi yang paling optimal dari pilihan yang ada. Tergantung pada cerita awal yang kita pilih, ada beberapa variasi cerita yang memberikan lima kolonis dengan sumber daya yang lebih sedikit, atau hanya satu kolonis dengan sumber daya yang lebih banyak. Setelah selesai memilih kolonis, simulasi koloni akan dimulai dari membuat tempat tinggal dan bertahan hidup dari alam sekitar.

C. Medan Tempuh

Di permainan ini, ada berbagai macam medan yang tersebar di setiap wilayahnya, mulai dari tanah biasa, tanah berkerikil, tanah berlumpur, rawa, air dangkal, air dalam yang tidak bisa dilewati, sampai lantai buatan dari kayu atau semen yang bisa diletakkan di atas tanah yang tidak basah. Setiap medan tempuh memiliki pengali kecepatannya masing-masing, misalnya untuk setiap lantai buatan pengali kecepatannya adalah 100%. Sementara itu, tiga medan tempuh yang saya pilih untuk makalah ini adalah tanah biasa dengan pengali kecepatan sebesar 87%, tanah berlumpur dengan pengali kecepatan 52%, dan rawa dengan pengali kecepatan 36%. Masih ada banyak lagi variasi medan tempuh di permainan ini, tetapi saya memilih tiga medan tempuh ini karena jarak antar pengalinya yang cukup jauh.

D. Pergerakan Kolonis

Pada permainan ini, setiap kolonis akan bergerak secara otomatis sesuai dengan tugas yang diberikan pada mereka. Tetapi, jika kita mau, kita bisa menggerakkan setiap kolonis kita secara manual, misalnya, ketika sedang menghadapi serangan dari pihak musuh atau ketika kita akan menyerang musuh.

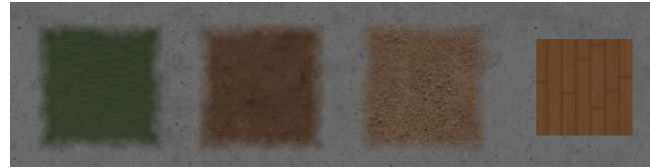
Permainan ini akan secara otomatis mencari jalur yang bisa ditempuh kolonis tersebut untuk sampai ke tempat yang hendak dituju. Selain itu, permainan ini juga secara otomatis mencari jalan dengan waktu tempuh yang paling singkat, dalam hal ini, kolonis akan berusaha menghindari medan tempuh dengan pengali yang lebih kecil dan memilih jalur memutar yang memiliki pengali kecepatan lebih besar. Tetapi ada kasus yang menyebabkan jalur dengan pengali yang lebih besar menghasilkan waktu tempuh yang lebih lama daripada melewati jalur dengan pengali yang lebih kecil, sehingga terkadang kolonis akan memilih jalur yang tidak memutar dan melalui tanah berlumpur atau rawa.

IV. UJI KASUS

A. Medan tempuh yang digunakan

Terdapat tiga jenis medan tempuh yang digunakan pada percobaan kali ini. Medan tempuh yang pertama adalah rawa, dengan pengali kecepatan sebesar 36%, dilambangkan dengan warna hijau. Medan tempuh yang kedua adalah tanah

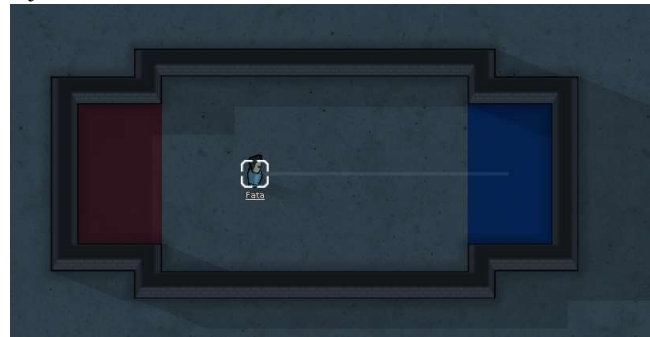
berlumpur, dengan pengali kecepatan sebesar 52%, dilambangkan dengan warna coklat tua. Medan tempuh yang ketiga adalah tanah biasa, dengan pengali kecepatan sebesar 87%, dilambangkan dengan warna coklat muda. Pada gambar di bawah ini, kotak urutan keempat adalah lantai kayu, dengan pengali kecepatan 100%, sama dengan kondisi lingkungan di sekitarnya yang berlapis beton, dengan pengali kecepatan sebesar 100%.



Gambar 4.1 Contoh Medan Tempuh yang digunakan
Sumber : Penulis

B. Kasus-kasus yang diujicobakan

Pada kasus pertama, kolonis diperintahkan untuk bergerak lurus dari karpet merah ke karpet biru. Karena seluruh medan tempuh yang ada adalah beton, maka kolonis langsung mengambil jalur yang terpendek, yaitu garis lurus menuju tujuan.



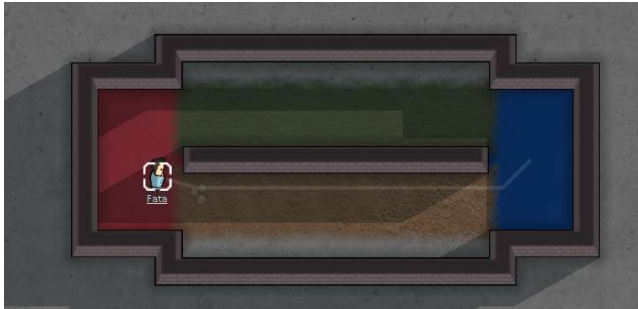
Gambar 4.2 Kasus pertama, kolonis langsung bergerak lurus menuju tujuan
Sumber : Penulis

Pada kasus kedua, terdapat medan tempuh rawa yang menghalangi jalur lurus kolonis untuk mencapai tujuan. Karena kolonis merasa bahwa waktu tempuh garis lurus akan lebih lama daripada waktu tempuh memutar, maka kolonis mengambil jalan memutar untuk menuju ke titik tujuan.



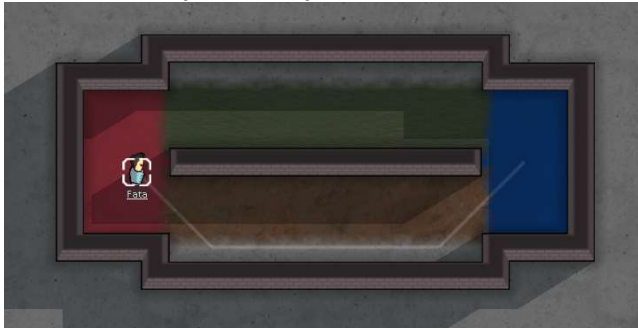
Gambar 4.3 Kolonis memilih jalan memutar untuk menghindari rawa
Sumber : Penulis

Pada kasus ketiga, terdapat dua buah medan tempuh yang dapat dilalui oleh kolonis, yaitu rawa dan tanah biasa. Karena kolonis merasa bahwa waktu tempuh melalui rawa (kecepatan 36%) akan lebih lama daripada waktu tempuh melalui tanah biasa (kecepatan 87%), maka kolonis mengambil jalur tanah biasa untuk menuju ke titik tujuan.



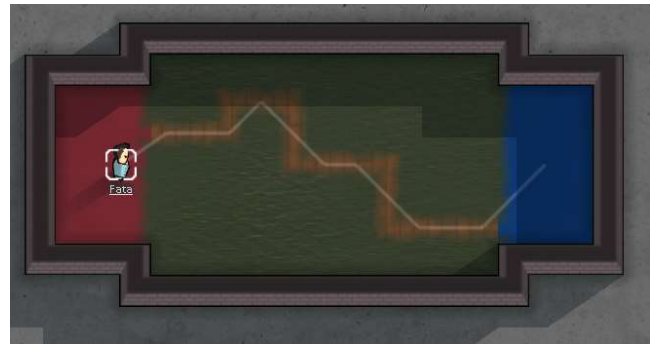
Gambar 4.4 Kasus ketiga, kolonis memilih jalur tanah biasa untuk mencapai tujuan
Sumber : Penulis

Pada kasus keempat, terdapat dua buah medan tempuh yang dapat dilalui oleh kolonis, yaitu rawa dan tanah berlumpur. Karena kolonis merasa bahwa waktu tempuh melalui rawa (kecepatan 36%) dan waktu tempuh melalui tanah berlumpur (kecepatan 52%) lebih lama daripada melalui jalan memutar di pinggir yang masih menggunakan beton (kecepatan 100%, dengan jarak yang lebih jauh), maka kolonis mengambil jalur beton untuk menuju ke titik tujuan.



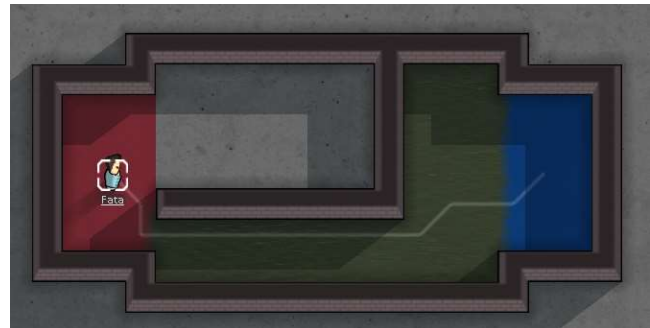
Gambar 4.5 Kasus keempat, kolonis memilih jalur beton dan memutar untuk mencapai tujuan
Sumber : Penulis

Pada kasus kelima, terdapat sebuah jalur yang dibuat dari lantai kayu (kecepatan 100%) dan medan tempuh di sekitarnya adalah rawa (kecepatan 36%). Berdasarkan gambar tersebut, terlihat jelas kolonis berusaha mengikuti jalur lantai kayu yang dibuat, karena jalur itu adalah jalur yang memiliki waktu tempuh paling singkat.



Gambar 4.6 Kolonis mengikuti jalur lantai kayu untuk menghindari rawa
Sumber : Penulis

Pada kasus keenam, terdapat sebuah wilayah yang dilapisi beton sedangkan wilayah lainnya adalah rawa. Karena kolonis mendeteksi bahwa tidak mungkin bisa mencapai tujuan jika melalui daerah berlapis beton, maka kolonis memutuskan untuk melalui jalur rawa, meskipun kecepatannya menjadi lebih lambat.



Gambar 4.7 Kolonis mengikuti jalur rawa karena hanya jalur itu yang bisa digunakan untuk mencapai tujuan
Sumber : Penulis

V. KESIMPULAN

Permainan simulasi koloni RimWorld adalah salah satu contoh permainan yang memanfaatkan teori graf, khususnya pada bagian algoritma untuk mencari jalur terpendek seperti algoritma A*. Permainan ini sebenarnya jauh lebih kompleks, dan masih memiliki banyak penerapan ilmu-ilmu matematika diskrit lainnya yang belum dibahas di makalah ini, seperti penggunaan pohon untuk menggambarkan urutan penelitian, penggunaan relasi-fungsi untuk memetakan hubungan antar kolonis, dan lain-lain. Makalah ini pun masih bisa dikembangkan lebih lanjut, karena pada makalah ini hanya membahas interaksi kolonis dengan medan tempuh yang ada di peta, belum membahas interaksi ketika kolonis bertemu objek bergerak atau kolonis lainnya sehingga rute awal yang ia pilih terhalangi.

VI. UCAPAN TERIMA KASIH

Pertama-tama, saya ucapkan puji syukur ke hadirat Tuhan Yang Maha Esa karena atas rahmat-Nya saya dapat menyelesaikan makalah ini. Lalu saya juga berterima kasih

kepada Dr. Ir. Rinaldi Munir, MT. selaku dosen mata kuliah IF2120 Matematika Diskrit yang telah memberikan bimbingan dan arahan sehingga saya dapat menyelesaikan mata kuliah ini sebaik mungkin. Tidak lupa saya juga mengucapkan terima kasih kepada orang tua, teman-teman, dan seluruh pihak yang tidak dapat saya sebutkan satu per satu. Terakhir, secara khusus saya ingin menyampaikan terima kasih kepada teman saya Yeremia Kristianto karena telah mengenalkan permainan ini kepada saya tahun lalu.

REFERENSI

- [1] K.H. Rosen, *Discrete Mathematics and its Application*, 7 th ed. New York : McGraw-Hill, 2012, pp. 641-802.
- [2] Munir, Rinaldi. 2006. Diktat Kuliah IF2120 Matematika Diskrit, Institut Teknologi Bandung : Bandung.
- [3] Slide Kuliah Graf IF2120, informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf diakses pada tanggal 9 Desember 2018
- [4] rimworldgame.com, diakses pada tanggal 9 Desember 2018
- [5] rimworldwiki.com, diakses pada tanggal 9 Desember 2018
- [6] <https://www.redblobgames.com/pathfinding/a-star/introduction.html>, diakses pada tanggal 9 Desember 2018

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, 10 Desember 2018



Fata Nugraha 13517109