

Pemanfaatan Algoritma A* untuk Penentuan Rute Perjalanan pada Event “While it’s warm” dalam Permainan Genshin Impact

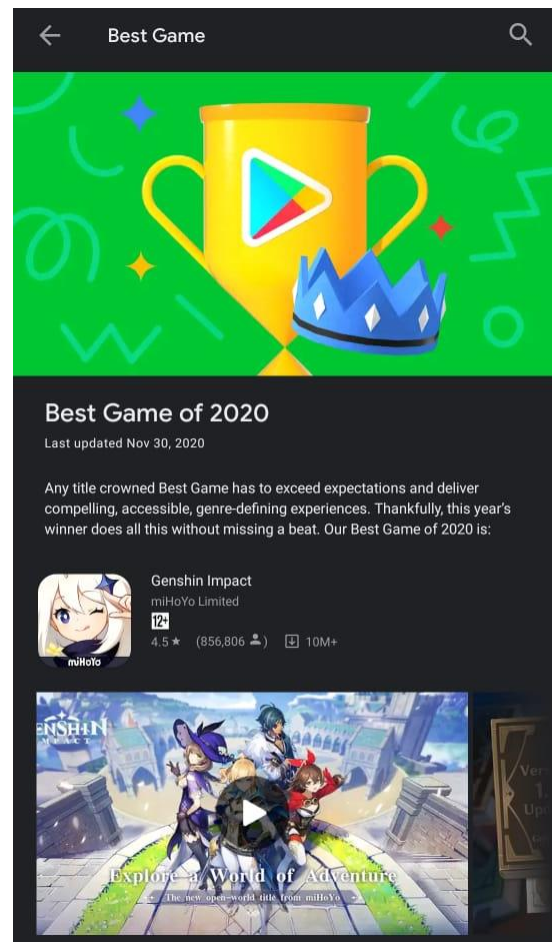
Clarisa Natalia Edelin / 13519213
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519213@std.stei.itb.ac.id

Abstract—*Genshin Impact* merupakan sebuah permainan *action RPG* yang dikembangkan oleh miHoYo. Permainan ini memiliki sistem *open world*, dimana pemain dapat dengan bebas menjelajah berbagai area yang ada di dunia. Dalam pertarungan, pemain dapat mengendalikan tim yang terdiri atas empat karakter dan setiap karakter dapat dilengkapi dengan senjata dan *artifact* untuk memperkuat karakter pemain dalam melawan musuh saat menjelajah dunia. Salah satu fitur menarik dalam permainan ini adalah *event*. *Event* merupakan kegiatan yang ada dalam permainan *Genshin Impact* yang bisa dimainkan oleh para pemain dan mendapatkan berbagai hadiah menarik. Salah satu *event* yang ada adalah “While It’s Warm”, yaitu sebuah *event* dimana pemain harus mengantarkan makanan dari satu tempat ke tempat yang lain dengan adanya ketentuan tertentu seperti batasan waktu dan tidak boleh terpengaruh oleh rekasi elemental. Oleh karena itu, dibutuhkan sebuah rute yang optimal agar pemain mampu mengantarkan makanan tepat waktu. Pada makalah ini, penulis akan membahas mengenai penentuan rute perjalanan pada *event* “While It’s Warm” di dalam permainan *Genshin Impact* dengan menggunakan algoritma A*.

Keywords—algoritma A*, rute, pengantaran makanan, Genshin Impact

I. PENDAHULUAN

Dalam menentukan rute, kita dapat menggunakan algoritma penentuan rute agar mendapatkan rute yang optimal. Algoritma penentuan rute dapat dibagi menjadi dua, yaitu *uninformed search* dan *informed search*. Namun dibandingkan semua algoritma dalam algoritma penentuan rute, algoritma A* dalam algoritma *informed search* akan menghasilkan hasil rute yang optimal dibandingkan algoritma yang lain. Algoritma A* mampu menghasilkan hasil yang optimal karena algoritma tersebut memiliki fungsi heuristik yang memperhitungkan jarak dari titik asal ke posisi kita berada saat ini dan jarak dari posisi kita saat ini ke tujuan yang ingin dituju, lalu membandingkan nilai tersebut untuk semua titik yang dapat dijelajahi dan memilih nilai heuristik minimum diantara yang lain. Dengan fungsi tersebut, algoritma A* mampu menghasilkan rute dengan ongkos dan jarak optimal.



Gambar 1.1 Penghargaan Best Game of 2020 di PlayStore yang diraih oleh Genshin Impact

(Sumber :

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fdunigames.co.id%2Fdiscover%2Farticle%2Fgenshin-impact-raih-2-penghargaan-di-google-play-awards-2020-region-asia-tenggara%2Fen&psig=AOvVaw27bwPhPcRWEgxXUZUSzIBZ&ust=1620804820354000&source=images&cd=vfe&ved=0CA0QjhxqFwoTCLC2htCOWfACFQAAAAAdAAAAABAD>)

Genshin Impact merupakan sebuah permainan dengan genre *action RPG* yang dikembangkan oleh miHoYo. Game ini dirilis pada tanggal 28 September 2020 dan telah mendapatkan beberapa penghargaan, seperti *Best Game* dan *User Choice Game* di Google Play Award 2020. Game ini merupakan game gratis yang dapat diunduh oleh semua orang di berbagai platform. *Genshin Impact* memiliki sistem dunia terbuka, dimana pemain dapat menjelajah dunia secara bebas tanpa ada batasan apapun. Dalam pertarungan, pemain dapat mengendalikan sebuah tim yang terdiri atas empat karakter. Setiap karakter dapat dilengkapi dengan senjata dan *artifact* yang dapat memperkuat karakter pemain sehingga karakter mampu melawan musuh dengan lebih mudah. Untuk memperkuat *artifact*, pemain dapat menggunakan *artifact* lainnya untuk menaikkan level artifact yang digunakan karakter. Pengumpulan *artifact* dapat dikumpulkan dua cara, yaitu dengan melakukan *domain* dan melakukan penyelidikan benda – benda di sekitar.

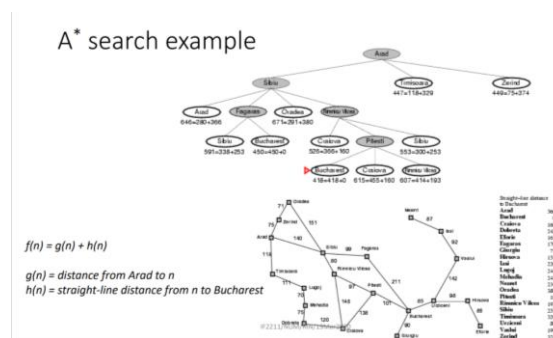
Pada permainan *Genshin Impact* juga terdapat berbagai *event* yang dapat dikerjakan oleh pemain untuk mendapat berbagai hadiah menarik. Setiap beberapa periode waktu tertentu, miHoYo akan merilis *event – event* menarik yang berbeda dari pada sebelumnya. Salah satu *event* yang pernah dirilis adalah *While It's Warm*, yaitu *event* dimana pemain harus mengantarkan makanan dari satu tempat ke tempat lain dengan batasan waktu dan ketentuan tertentu. Oleh karena itu dibutuhkan sebuah rute yang optimal agar perjalanan tidak memakan waktu lebih dari batas waktu yang diberikan. Untuk menentukan rute tersebut, penulis akan menggunakan algoritma A* untuk menentukan rute pengantaran yang optimal.

II. TEORI DASAR

A. Penentuan rute (Route/Path planning)

Algoritma yang digunakan pada penentuan rute merupakan algoritma yang digunakan untuk mencari rute dengan *cost* paling sedikit dan risiko paling rendah. Algoritma tersebut dapat dibagi menjadi dua bagian, yaitu *uninformed search* dan *informed search*. *Uninformed search* adalah sebuah metode pencarian rute yang tidak memiliki informasi pendukung mengenai tujuan yang sedang dicari, dimana pencarian dilakukan hanya dengan melihat informasi yang dimiliki sekarang. Informasi tersebut dapat berupa informasi yang beragam, oleh karena itu semua informasi akan diunifikasi menjadi satu besaran yang disebut sebagai *cost*. Algoritma yang termasuk ke dalam *uninformed search* adalah Breadth First Search, Depth First Search, Uniform Cost Search. Sedangkan untuk *informed search* merupakan sebuah metode pencarian rute yang memiliki informasi pendukung mengenai tujuan yang ingin dituju. Informasi yang telah diberikan tersebut biasa disebut sebagai nilai heuristik. Pada umumnya, nilai tersebut memberikan keterangan mengenai penilaian terhadap sebuah lokasi, dimana penilaian tersebut harus relevan dengan hal yang ingin dioptimisasikan dalam proses pencarian. Nilai heuristik juga digunakan untuk menentukan titik yang akan dieksplorasi selanjutnya, dimana hasil yang didapatkan akan menjadi lebih baik untuk Sebagian besar permasalahan. Algoritma yang termasuk *informed search* adalah Greedy Best First Search dan A*.

B. Algoritma A*



Gambar 2.2.1 Contoh pencarian rute menggunakan algoritma A*

Algoritma A* merupakan salah satu algoritma yang digunakan dalam pencarian rute dan termasuk dalam pencarian rute *informed search*, yaitu pencarian rute yang mengetahui lokasi yang ingin dituju dan dapat digunakan untuk memperhitungkan rute yang ingin dituju. Berbeda dengan algoritma lain yang termasuk *informed search*, Greedy Best First Search, dimana algoritma tersebut tidak memperhitungkan *cost* untuk mencapai tujuan. Oleh karena itu, penulis memilih algoritma A* sebagai algoritme pencarian rute paling mangkus karena sesuai dengan tujuan yang ingin dicapai, yaitu rute dengan *cost* paling sedikit dan risiko paling rendah.

Algoritma A* memiliki konsep algoritma berupa penghindaran mengekspansi jalur yang mahal. Algoritma ini menggunakan fungsi evaluasi antara lain,

$$f(n) = g(n) + h(n)$$

dimana $g(n)$ adalah *cost* yang dibutuhkan untuk dari tempat asal sampai ke n , $h(n)$ adalah estimasi *cost* yang dibutuhkan untuk sampai dari n ke tujuan, dan $f(n)$ adalah estimasi total *cost* yang dibutuhkan dari tempat asal ke tujuan melalui n . Saat mencari rute dengan menggunakan algoritma A*, algoritma akan selalu mencari nilai dengan besar $f(n)$ paling kecil dari yang lain. Sehingga algoritma A* mampu mendapatkan hasil rute dengan *cost* yang paling kecil.

C. Genshin Impact



Gambar 2.3.1 Poster Genshin Impact

(Sumber :

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fgimindo.com%2Fgenshin-impact-akan-hadir-secara-gratis-di-playstation->

4%2F&sig=AOvVaw2WRtPJahc9gw0_8q5amMdm&ust=1620804850874000&source=images&cd=vfe&ved=0CA0QjhxqFwoTCLDfyO2OwfACFQAAAAAdAAAAABAD)

Genshin Impact merupakan sebuah permainan dengan genre *action RPG* yang dibuat dan dipublikasi oleh miHoYo. Permainan ini memiliki fitur sebuah lingkungan dunia terbuka, dimana pemain dapat mengeksplorasi dunia secara bebas, dan sistem bertarung yang memiliki dasar aksi yang menggunakan *elemental magic* dan penggantian karakter. Permainan ini juga memiliki fitur gacha, dimana pemain bisa mendapatkan karakter atau senjata baru. *Genshin Impact* merupakan game *online* dan memiliki fitur *multiplayer*, dimana pemain dapat bermain bersama dengan pemain lainnya.

Genshin Impact dirilis pada bulan September 2020 di berbagai perangkat. Permainan ini mendapatkan respons positif dari para pemain, terutama karena fitur pertarungan dan kebebasan menjelajah di sistem dunia terbuka. Permainan ini telah mendapatkan berbagai penghargaan dan meraih keuntungan yang sangat besar.

Salah satu fitur yang paling terkenal dari *Genshin Impact* adalah sistem bertarung yang unik. Pemain memiliki sebuah tim yang terdiri atas 4 karakter, dimana setiap karakter memiliki jenis senjata, elemen, dan stat masing – masing. Pemain dapat memilih 4 karakter dari semua karakter yang ia miliki untuk masuk ke dalam tim dan digunakan dalam pertarungan, dimana pemain dapat melakukan kombinasi 4 karakter tersebut untuk dapat memenangkan pertarungan. Selain senjata, karakter dapat dilengkapi dengan *artifact*.

Salah satu fitur yang menarik dalam permainan *Genshin Impact* adalah *event*, dimana pemain bisa mengerjakan tugas tertentu dan mendapatkan hadiah yang menarik. *Event* diadakan dalam periode waktu tertentu dan memiliki tema yang beragam. Salah satu *event* yang dirilis adalah *While It's Warm*.



Gambar 2.3.2 Halaman event “While It’s Warm”
(Sumber : <https://genshin.mihoyo.com/en/news/detail/7244>)

Event “While It’s Warm” merupakan sebuah event dimana pemain harus mengantarkan makanan dengan kondisi tertentu. Seperti yang dapat dilihat pada gambar 2.3.3, ketentuan pengantaran barang dapat berupa waktu pengantaran yang terbatas dan tidak boleh terpengaruh oleh reaksi elemental. Karena adanya batasan waktu dalam mengantar makanan, pemain membutuhkan sebuah rute pengantaran yang optimal agar dapat mengantarkan makanan tepat waktu.



Gambar 2.3.3 Contoh ketentuan pengantaran makanan pada event “While Its Warm”
(sumber : <https://genshin.mihoyo.com/en/news/detail/7244>)

III. IMPLEMENTASI ALGORITMA A*

A. Asumsi yang digunakan dan input file

Untuk persoalan ini penulis menggunakan beberapa asumsi sebagai berikut :

- *Source code* dari game tidak dibuka dari pihak pengembang sehingga peta diasumsikan sebagai graf tidak berarah
- *Node* yang dipakai merupakan angka dari tiap daerah pada peta
- Peta dalam permainan yang digunakan dalam implementasi merupakan peta Tevyat.
- *Starting point* dapat dimulai dari nomor satu (1) pada peta dan titik tujuan adalah nomor sebelas (11)
- Peta dianggap datar dan kecepatan dari satu titik ke titik yang lain selalu konstan
- Titik koordinat yang digunakan merupakan asumsi dari penulis, bukan koordinat asli dalam permainan
- Kasus yang digunakan adalah pengantaran makanan dari *good hunter express*, dengan ketentuan waktu pengantaran maksimal 130 detik, tidak boleh terpengaruhi oleh reaksi elemental, dan tidak boleh berlari



Gambar 3.1 Peta Genshin Impact beserta dengan titik yang dapat dilalui untuk melakukan event
(sumber : dokumentasi pribadi)

Dengan asumsi yang telah dibuat, algoritma A* hanya akan diimplementasi dengan sebuah graf yang dibuat oleh penulis. Graf akan dibuat dengan menggunakan aturan sebagai berikut:

- Dua buah area dianggap terhubung satu dengan yang lain jika tidak ada area lain yang menghubungkan kedua area tersebut secara langsung
- Bobot dari graf ditentukan dengan menghitung jarak kedua titik dengan menggunakan rumus Euclidean.
- Pengukuran jarak tempuh dilakukan dengan menggunakan rumus Euclidean yang menggunakan dua titik koordinat dua tempat.
- Bobot sisi graf dianggap sama untuk dua arah (bolak-balik) karena penjelajahan peta dianggap menggunakan kecepatan konstan.

Setelah didapatkan hasil jalur antar titik yang mungkin dilalui, hasil dimuat dalam sebuah matriks keterhubungan dalam gambar sebagai berikut :

0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Gambar 3.2 Matriks keterhubungan antar titik jalur yang dapat dilalui (sumber : dokumentasi pribadi)

Setelah mendapatkan data keterhubungan area sesuai dengan Gambar 3.2, Algoritma A* dapat diterapkan dengan prosedur sebagai berikut :

1. Algoritma akan dimulai dari titik awal, yaitu area 1, di mana semua node yang bertetangga akan diekspan, kemudian nilai f(n) dari masing-masing node akan dihitung.
2. Dari node yang telah diekspan, node dengan nilai f(n) terkecil akan dipilih, dan seluruh node yang bertetangga dengan node yang terpilih akan diekspan kembali, mengecualikan node sebelumnya.
3. Ulangi langkah 2 sampai mendapatkan solusi sementara dan tidak ada nilai cost yang lebih kecil dari solusi sementara
4. Apabila sudah tidak terdapat nilai cost yang lebih kecil dari nilai cost solusi sementara, maka solusi sementara akan ditetapkan sebagai solusi akhir pencarian

B. Kode Program Algoritma A*

Untuk menghitung rute yang optimal dengan menggunakan algoritma A*, penulis menggunakan sebuah program untuk membantu perhitungan. Program dapat diakses pada halaman <https://github.com/ClarisaNatalia/Tucil3-Stima>. Pada program yang digunakan untuk menghitung rute menggunakan algoritma A*, program akan menerima dua jenis input file, yaitu file yang berisi matriks keterhubungan, seperti yang dapat dilihat pada gambar 3.2, dan file yang berisi koordinat setiap area dan area yang memiliki hubungan.

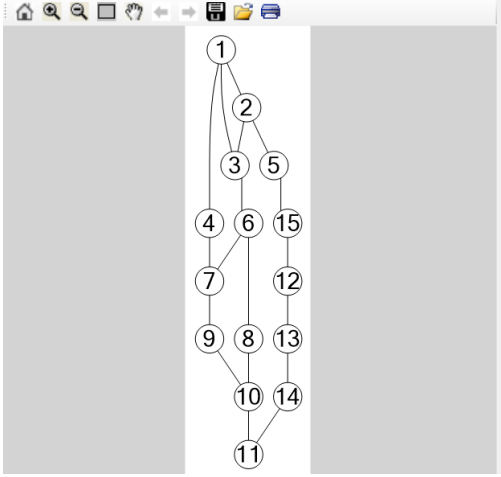
```

15
1, 3.25, 0.6
2, 2.262, 1.075
3, 2.615, 1.196
4, 2.813, 1.4
5, 3.187, 1.456
6, 3.6, 1.928
7, 3.87, 2.153
8, 3.9, 2.4
9, 3.6, 2.8125
10, 4.285, 3.15
11, 4.574, 3.563
12, 5.757, 3.6
13, 5.925, 3.712
14, 6.5, 3.8
15, 6.54, 4.2
1,2
1,3
1,4
2,3
2,5
3,6
4,7
5,15
6,7
6,8
7,9
8,10
9,10
10,11
12,13
13,14
13,11
14,11
15,12

```

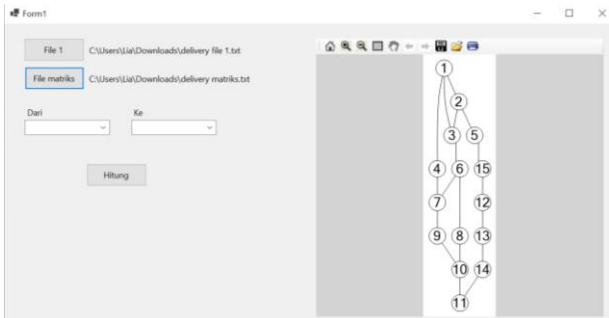
Gambar 3.3 Input file yang terdiri dari koordinat area dan keterhubungan area tertentu. (sumber : dokumentasi pribadi)

Dengan menggunakan dua file diatas, program akan menghasilkan sebuah graf representasi dari jalur yang dapat digunakan untuk menyelesaikan event. Penulis menggunakan MSAGL untuk menggambarkan grafnya. Graf tersebut bukan merupakan graf asli dari penggambaran peta seperti di 3.1, namun merupakan gambaran jalur antar area sesuai dengan keterangan input file yang berupa jalur pada penggambaran peta.



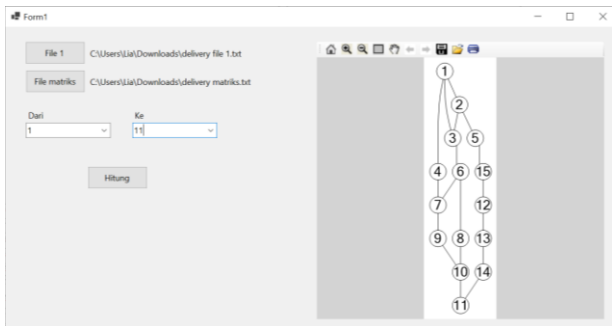
Gambar 3.4 Graf yang dihasilkan oleh program dengan menggunakan MSAGL (sumber : dokumentasi pribadi)

Setelah menerima input file, program akan menerima input titik awal dan titik tujuan yang ingin dicari rutenya.



Gambar 3.5 Tampilan program saat dijalankan (sumber : dokumentasi pribadi)

Dengan menggunakan asumsi yang telah ditentukan, titik awal penentuan rute adalah dari area dengan nomor satu (1). Pada saat kita sedang mengerjakan *event*, kita ingin bergerak dari titik 1 ke titik 11, oleh karena itu titik tujuan kita isi dengan titik 11.



Gambar 3.6 Tampilan program saat dijalankan dan telah mengisi titik awal dan titik tujuan (sumber : dokumentasi pribadi)

Jika menekan tombol hitung, maka program akan menghasilkan rute optimal dari titik 1 ke titik 15.

Pengujian dengan menggunakan algoritma A* yang digunakan oleh program adalah sebagai berikut :

1. Program akan menentukan titik 1 sebagai simpul ekspansi saat ini.
2. Program akan mengecek titik yang bertetangga dengan titik 1 dan menghitung jarak dari titik 1 ke titik yang bertetangga. Untuk titik 1, maka titik yang bertetangga hanya titik 2, lalu dihitung jarak dari titik 1 ke titik 2 dengan menggunakan rumus Euclidean, pada program menggunakan fungsi `getStraightDistance(Simpul s)`.

```

// Mendapatkan jarak lurus dari titik ke O?
// referensi
public double getStraightDistance(Simpul s)
{
    return Math.Sqrt(Math.Pow(this.getX() - s.getX(), 2) + Math.Pow(this.getY() - s.getY(), 2));
}
    
```

Gambar 3.6 Fungsi yang digunakan untuk menghitung jarak antara dua titik

3. Jarak – jarak yang telah dihitung akan dimasukkan ke dalam sebuah array yang berisi jarak dari titik 1 sampai titik bertetangga.
4. Jarak yang ada dalam array akan diurutkan dari yang terkecil sampai yang terbesar.
5. Program akan memilih jarak dengan urutan pertama pada array, yaitu jarak yang paling kecil dari titik 1 ke titik berikutnya.
6. Program akan menentukan simpul ekspansi baru, yaitu titik dengan jarak terpendek dari simpul ekspansi yang lama.
7. Program akan mengulangi langkah 1 sampai 6, sampai akhirnya sampai pada simpul tujuan.

Program yang digunakan untuk menentukan rute menggunakan algoritma A* memiliki fungsi sebagai berikut:

```

public List<Simpul> getAStar(Simpul asal, Simpul tujuan, List<Simpul> list)
{
    if (list.Count == 0) // Sudah tidak ada simpul yang lagi dikunjungi
    {
        return null;
    }
    else if (list[0].getSimpul().Equals(tujuan)) // Sudah ketemu jarak terpendek
    {
        return list[0];
    }
    else
    {
        // Mendapatkan semua simpul yang bertetangga
        List<Simpul> listBertetangga = getAllBertetangga(list[0].getSimpul());
        for (int i = 0; i < listBertetangga.Count; i++)
        {
            List<Simpul> newPath = list[i].getPath();
            Simpul E = new Simpul(newPath, listBertetangga[i]);
            // Insert simpul yang bertetangga ke list, kemudian diurut
            list.Add(E);
            sortList(list, tujuan);
        }
        list.Remove(list[0]);
        // Jika sudah lagi ke simpul asal
        if (list[0].getSimpul().Equals(asal))
        {
            list.Remove(list[0]);
        }
        return getAStar(asal, tujuan, list);
    }
}
    
```

Gambar 3.7 Kode Algoritma A* pada program (sumber : dokumentasi pribadi)

Pada tabel 2, penulis akan memperlihatkan alur perhitungan yang dilakukan oleh program dalam menentukan rute optimal dari titik nomor satu ke titik nomor sebelas.

Tabel 1 straight distance dari simpul tertentu ke simpul nomor sebelas

Simpul	Straight distance
1	3,245357
2	3,396393
3	3,072519
4	2,867454
5	2,522542
6	1,903129
7	1,575981
8	1,344189
9	1,229604
10	0,504073

11	0
12	1,1,83578
13	1,359192
14	1,940527
15	2,066622

Tabel 2 simpul ekspan dan simpul hidup

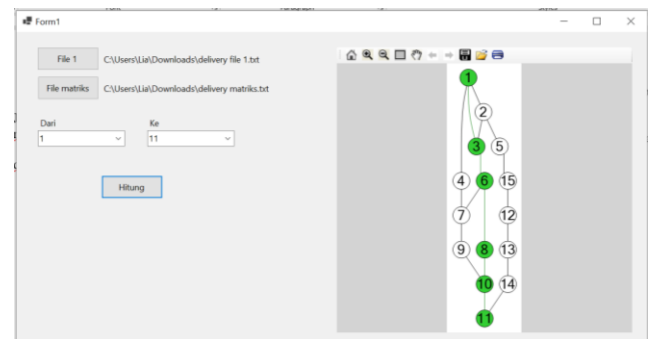
Simpul ekspan	Simpul hidup
1	2 – 1 $f(n) = 1.096252 + 3.396393 = 4.492645$ 3 – 1 $f(n) = 0.870885 + 3.072519 = 3,943404$ 4 – 1 $f(n) = 0.911575 + 2.78921 = 3,700785$ (solusi sementara)
4 – 1	7 – 4 – 1 $f(n) = 0.911575 + 1.29779 + 1.575981 = 3,785346$ (solusi sementara)
7 – 4 – 1	6 – 7 – 4 – 1 $f(n) = 0.911575 + 1.29779 + 0.351461 + 1.903129 = 4,463955$ 9 – 7 – 4 – 1 $f(n) = 0.911575 + 1.29779 + 0.712629 + 1.229604 = 4,151598$
3 – 1	6 – 3 – 1 $f(n) = 0.870885 + 1.22721 + 1.903129 = 4,001224$
6 – 3 – 1	7 – 6 – 3 – 1 $f(n) = 0.870885 + 1.22721 + 0,351461 + 1,575981 = 4,025537$ 8 – 6 – 3 – 1 $f(n) = 0.870885 + 1.22721 + 0,559271 + 1,344189 = 4,001555$ (solusi sementara)
8 – 6 – 3 – 1	10 – 8 – 6 – 3 – 1 $f(n) = 0.870885 + 1.22721 + 0,559271 + 0,843645 + 0,504073 = 4,005084$

10 – 8 – 6 – 3 – 1	11 – 10 – 8 – 6 – 3 – 1 $f(n) = 0.870885 + 1.22721 + 0,559271 + 0,843645 + 0,504073 + 0 = 4,005084$
11 – 10 – 8 – 6 – 3 – 1	Solusi telah ditemukan

Dapat dilihat pada saat simpul ekspan adalah simpul titik 1, simpul hidup yang dapat dipilih ada 3 buah, yaitu simpul titik 2, simpul titik 3, dan simpul titik 4. Program akan memilih simpul dengan nilai $f(n)$ yang paling kecil, yaitu simpul 4. Namun pada saat sampai simpul ekspannya adalah simpul 7 dengan jalur dari 4 – 1, nilai $f(n)$ yang dihasilkan lebih besar dibandingkan nilai $f(n)$ simpul 3 pada simpul hidup milik simpul ekspan 1. Oleh karena itu, program akan menetapkan simpul 3 sebagai simpul ekspannya sekarang. Pada akhirnya, dengan mengekspansi simpul 3, program mampu menemukan jalur paling optimal, dengan nilai $f(n)$ paling minimum.

C. Hasil

Berikut merupakan jalur yang dihasilkan oleh program A*



Gambar 3.8 Tampilan rute yang dihasilkan oleh program dengan menggunakan algoritma A* (sumber : dokumentasi pribadi)

Dari hasil perhitungan yang telah diperlihatkan pada tabel 2 dan hasil graf yang ditunjukkan pada gambar 3.7, dapat disimpulkan bahwa rute paling optimal untuk mencapai tujuan adalah

$$1 - 3 - 6 - 8 - 10 - 11$$

Hasil yang didapat merupakan hasil perhitungan dengan menggunakan asumsi – asumsi seperti yang telah dijabarkan penulis sebelumnya.

IV. KESIMPULAN

Pada event “While It’s Warm” pada permainan *Genshin Impact*, pemain perlu mengantarkan makanan dari satu tempat ke tempat lain dengan ketentuan tertentu, seperti ada batas waktu pengantaran dan tidak boleh dipengaruhi oleh reaksi elemental. Oleh karena itu, pemain membutuhkan suatu rute optimal agar dapat mengantarkan makanan tepat waktu. Salah satu cara untuk

menentukan rute optimal adalah dengan menggunakan algoritma A*.

Algoritma A* menggambarkan rute pemain sebagai graf tidak berarah. Dalam perhitungan rute, penulis menggunakan beberapa asumsi untuk dapat menghitung rute pengantaran. Setiap sisi graf diberi bobot sesuai dengan jarak antara kedua titik pada sisi tersebut. Setelah graf terbentuk, penentuan rute akan dilakukan dengan menggunakan Algoritma A*, dimana algoritma akan menggunakan fungsi heuristik berupa penentuan jarak Euclidean yang minimum untuk mendapatkan hasil yang optimal.

Setelah melakukan perhitungan, baik secara manual maupun menggunakan program, didapatkan hasil bahwa rute yang optimal dilalui oleh pemain agar dapat mengantarkan makanan tepat waktu adalah 1 – 3 – 6 – 8 – 10 – 11, dimana rute tersebut memiliki nilai fungsi heuristik minimum dibandingkan rute yang lainnya.

TAUTAN PROGRAM DI GITHUB

<https://github.com/ClarisaNatalia/Tucil3-Stima>

TAUTAN VIDEO DI YOUTUBE

<https://youtu.be/d8Kdzlq0GVU>

UCAPAN TERIMA KASIH

Puji syukur saya ucapkan kepada Tuhan yang Maha Esa atas selesainya makalah ini. Saya juga mengucapkan terima

kasih kepada para dosen mata kuliah strategi algoritma terutama kepada Pak Rinaldi Munir selaku dosen kelas 04 yang telah mengajarkan ilmunya. Selain itu, saya juga berterima kasih kepada semua orang yang telah membantu saya baik secara langsung maupun tidak langsung hingga sekarang.

REFERENSI

- [1] Munir, Rinaldi. 2021. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian2-2021.pdf>
- [2] <https://genshin.mihoyo.com/en/news/detail/7244>
- [3] <https://game8.co/games/Genshin-Impact/archives/312437>

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.

Jakarta, 10 Mei 2021



Clarisa Natalia Edelin 13519213