

Penerapan Algoritma A* untuk Menentukan Rute *Farming Artifact* Tercepat di *Genshin Impact*

Wilbert Fangderson - 13519025

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail (gmail): ¹13519025@std.stei.itb.ac.id ²wilbertf38@gmail.com

Abstraksi—*Genshin Impact* adalah sebuah game *action* dan *role-playing* yang lumayan terkenal di kalangan remaja. Game ini memiliki berbagai sistem yang membuatnya menjadi game favorit kebanyakan orang. Salah satu sistemnya adalah *artifact* yang memberikan kekuatan tambahan bagi pemain. Akan tetapi, mengumpulkan *artifact* akan sangat memakan banyak waktu. Oleh karena itu, Algoritma A* akan digunakan sebagai penerapan untuk menentukan rute tercepat dalam mengumpulkan *artifact* di game tersebut.

Keywords—Algoritma A*; *Genshin Impact*; rute; *artifact*

I. PENDAHULUAN

Genshin Impact merupakan sebuah permainan *free-to-play action RPG* yang dirilis pada tanggal 28 September 2020 oleh perusahaan yang bernama miHoYo. Game ini memiliki beberapa platform, diantaranya adalah Android, iOS, Windows, dan PlayStation 4, serta PlayStation 5 pada tanggal 28 April 2021.



Gambar 1. Game *Genshin Impact*

Sumber : <https://gizmologi.id/games/genshin-impact-konsep-open-world/>

Genshin Impact hadir dengan sistem bermain *Action Role Playing* dengan konsep *Open World*. *Genshin Impact* mengambil konsep dunia yang bernama Teyvat, yaitu tempat dimana pemain bisa menjelajahi dunia yang indah dan luas baik secara individu maupun bersama dengan teman-teman lintas platform. Cara bermain game ini mudah dilakukan, yaitu hanya membutuhkan *timing* dan posisi dan tepat untuk mengalahkan musuh dan level musuh juga berkembang dengan seiring naiknya level dari *world* pemain.

Sistem permainan di *Genshin Impact* bukan hanya aksi para pemain untuk menaklukkan musuh, namun juga terdapat sistem berenang, meluncur di udara, menikmati cerita dari para karakter di *Genshin Impact* sampai membangun rumah sendiri. Pemain juga dapat mengoleksi karakter-karakter yang dirilis oleh perusahaan miHoYo untuk dimasukkan ke dalam *party* dari pemain dan dapat dimainkan kapanpun dimanapun. Masing-masing dari karakter memiliki elemen yang berbeda-beda yang menjadi kunci bagi pemain untuk mengembangkan reaksi antar elemen dalam pertarungan. Elemen-elemen tersebut terbagi menjadi es (*cryo*), api (*pyro*), angin (*anemo*), listrik (*electro*), tanaman (*dendro*), air (*hydro*), serta tanah (*geo*).

Terdapat beberapa faktor yang mempengaruhi kekuatan dari sebuah karakter, diantaranya adalah senjata yang dipakai oleh karakter (*weapon*), artefak (*artifact*), talenta (*talent*), serta konstelasi yang dimiliki oleh masing-masing karakter (*constellation*). Salah satu jenis barang yang sangat dicari oleh kebanyakan pemain adalah artefak yang juga dapat ditemukan pada *open world*. Artefak dapat dijual untuk mendapatkan *mora* (mata uang pada *Genshin Impact*), maupun untuk memperkuat artefak yang lain.

Dalam makalah ini, penulis akan mengaplikasikan algoritma A* untuk menentukan rute *farming artifact* dalam game *Genshin Impact* ini.

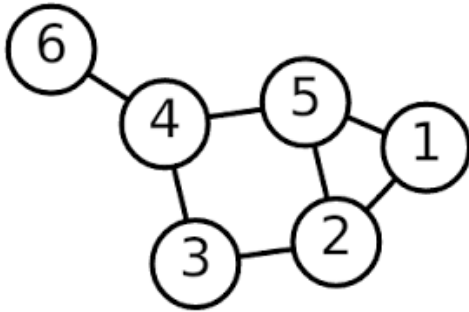


Gambar 2. Sistem artefak pada *Genshin Impact*

II. DASAR TEORI

A. Graf

Graf merupakan suatu representasi objek-objek diskrit dan hubungan antara objek-objek tersebut. Teori graf pertama kali digunakan untuk memecahkan persoalan jembatan Königsberg.



Gambar 3. Graf

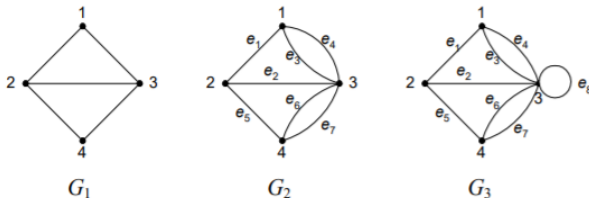
Sumber :

https://en.wikipedia.org/wiki/Vertex_%28graph_theory%29

Graf G didefinisikan sebagai (V, E) , dimana V adalah himpunan tidak kosong dari simpul-simpul, dan E adalah himpunan sisi yang menghubungkan sepasang.

Graf memiliki beberapa jenis, diantaranya adalah :

- Berdasarkan ada atau tidaknya gelang atau sisi ganda
 - Graf sederhana
Graf yang tidak mengandung gelang atau sisi ganda.
 - Graf tidak sederhana
Graf yang mengandung sisi ganda atau gelang.

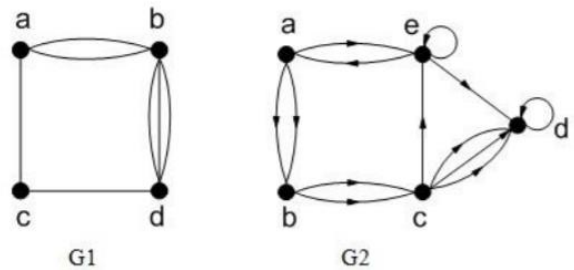


Gambar 4. Graf sederhana (G_1), Graf dengan sisi ganda (G_2), dan Graf dengan sisi ganda dan gelang (G_3)

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

- Berdasarkan orientasi arah pada sisi
 - Graf tak-berarah
Graf yang sisinya tidak mempunyai orientasi arah.
 - Graf berarah
Graf yang setiap sisinya diberikan orientasi arah



Gambar 5. Graf tak-berarah (G_1), Graf berarah (G_2)

Sumber :

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

Graf memiliki beberapa terminologi, diantaranya adalah :

- Ketetanggaan (*Adjacent*)
Dua simpul bertetangga apabila terhubung langsung.
- Bersisian (*Incidency*)
Untuk sebuah sisi $e = (v_j, v_k)$, e bersisian dengan simpul v_j dan v_k .
- Simpul Terpencil (*Isolated Vertex*)
Simpul yang tidak mempunyai sisi yang bersisian dengannya.
- Graf Kosong (*Null graph or empty graph*)
Graf yang himpunan sisinya merupakan himpunan kosong.
- Derajat (*Degree*)
Jumlah sisi yang bersisian dengan suatu simpul.
- Lintasan (*Path*)
Sisi yang ditempuh dari simpul awal (v_0) sampai simpul akhir (v_n).
- Sirkuit (*Circuit*)
Lintasan yang berawal dan berakhir pada simpul yang sama.
- Keterhubungan (*Connected*)
Untuk setiap pasang simpul v_i dan v_j , terdapat lintasan dari v_i ke v_j .
- Upagraf (*Subgraph*)
Bagian dari graf yang simpul dan sisinya merupakan himpunan bagian dari sebuah graf.
- Cut-set
Himpunan sisi yang dibuang dari sebuah graf yang menyebabkan graf tersebut tidak terhubung.

- k. Graf Berbobot (*Weighted Graph*)
Graf yang setiap sisinya diberi sebuah harga.

B. Algoritma A*

Algoritma A* merupakan sebuah algoritma *informed-search* atau algoritma traversal graf yang digunakan untuk melakukan perencanaan rute. Algoritma A* akan mencari rute dari satu tempat ke tempat lain dengan *cost* terkecil atau dengan kata lain rute terpendek, tanpa meng-*expend* rute yang lebih panjang.

Dalam Algoritma A*, simpul-simpul yang terdapat dalam sebuah graf diproses menurut berdasarkan evaluasi $f(n)$. Nilai $f(n)$ dihitung dengan persamaan :

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

dengan:

$g(n)$ = *cost* sejauh ini untuk mencapai simpul n

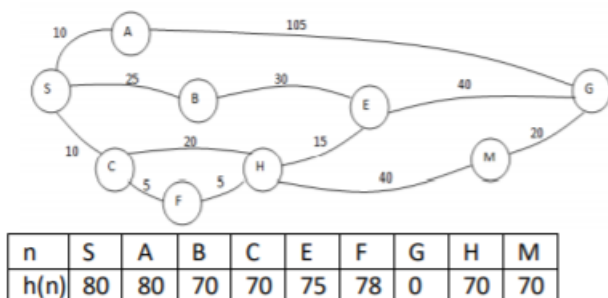
$h(n)$ = perkiraan *cost* dari simpul n ke simpul tujuan

$f(n)$ = perkiraan total *cost* untuk melewati n dan mencapai simpul tujuan

Berikut merupakan langkah-langkah algoritma A* :

1. Menentukan simpul asal
2. Dari tetangga simpul sekarang, tentukan *cost* jarak antara simpul sekarang dengan masing-masing tetangganya, kemudian ditambahkan perkiraan *cost* dari simpul sekarang ke simpul tujuan ($f(n) = g(n) + h(n)$).
3. Pilih total *cost* $f(n)$ terkecil dan ulangi langkah kedua

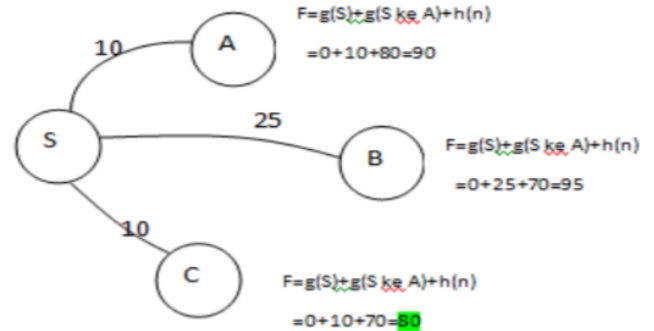
Simulasi A* :



Gambar 6.1. Contoh simulasi A*

Sumber :

<https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf>

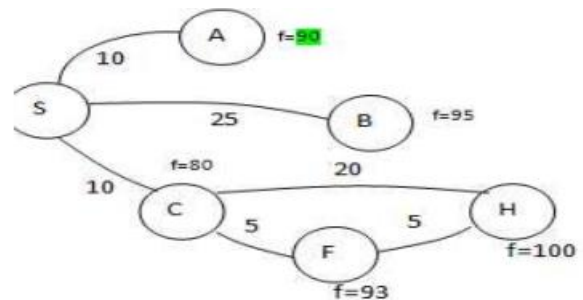


Gambar 6.2. Langkah pertama

Sumber :

<https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf>

Simpul asal (*starting point*) dimulai dari simpul S, akan dihitung $f(n)$ untuk masing-masing dari simpul S ke tetangganya, dan didapatkan *cost* terkecil adalah dari simpul S ke simpul C, yaitu 80 ($g(n) = 10$ dan $h(n) = 70$).

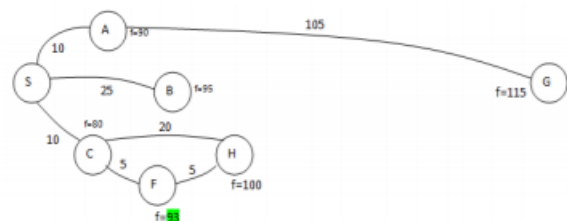


Gambar 6.3. Langkah kedua

Sumber :

<https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf>

Simpul sekarang (simpul C) akan dihitung $f(n)$ masing-masing dari simpul C ke tetangganya, dan didapatkan ternyata *cost* total antara simpul C dengan tetangganya lebih besar dibandingkan total *cost* simpul S dan simpul A, maka selanjutnya akan diambil simpul A sebagai acuan.

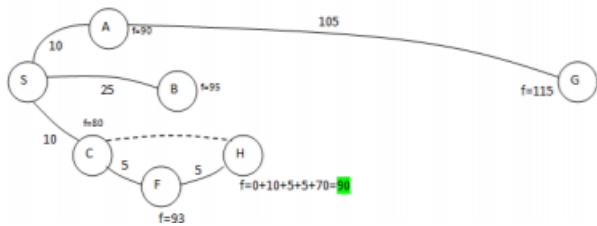


Gambar 6.4. Langkah ketiga

Sumber :

<https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf>

Simpul sekarang (simpul A) akan dihitung $f(n)$ masing-masing dari simpul A ke tetangganya, dan didapatkan ternyata cost total antara simpul A dengan tetangganya (simpul G) lebih besar dibandingkan total cost simpul S ke F (melalui simpul C), maka selanjutnya akan diambil simpul F sebagai acuan.

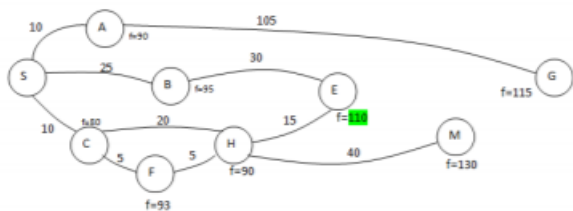


Gambar 6.5. Langkah keempat

Sumber :

<https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf>

Simpul sekarang (simpul F) akan dihitung $f(n)$ masing-masing dari simpul F ke tetangganya, dan didapatkan ternyata cost total antara simpul S hingga simpul H memiliki cost yang terkecil, sehingga selanjutnya akan diambil simpul H sebagai acuan.

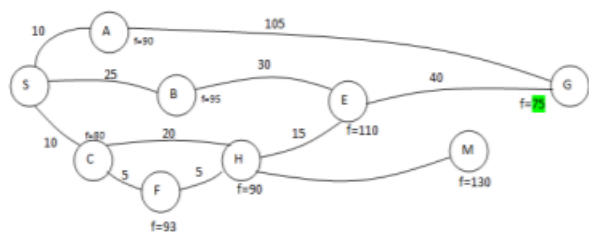


Gambar 6.6. Langkah kelima

Sumber :

<https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf>

Simpul sekarang (simpul H) akan dihitung $f(n)$ masing-masing dari simpul H ke tetangganya, dan didapatkan ternyata cost total antara simpul S hingga simpul E memiliki cost yang terkecil, sehingga selanjutnya akan diambil simpul E sebagai acuan.



Gambar 6.7. Langkah keenam

Sumber :

<https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf>

Simpul sekarang (simpul E) akan dihitung $f(n)$ masing-masing dari simpul E ke tetangganya, dan didapatkan ternyata cost total antara simpul S hingga simpul G (simpul final) memiliki cost yang terkecil. Karena simpul G sudah merupakan simpul tujuan (simpul final), maka algoritma A* berhenti dan didapatkan solusinya.

C. Artifact dalam Genshin Impact

Genshin Impact memiliki sistem artefak yang dapat menambah kekuatan dari karakter dari pemain. Sebenarnya, terdapat banyak cara untuk mendapatkan artefak-artefak ini, salah satunya adalah dengan bermain *domain* yang menghabiskan resin (energi yang dibutuhkan untuk melakukan suatu tantangan). Akan tetapi, karena resin (energi pada Genshin Impact) bersifat terbatas dan memerlukan waktu yang lama untuk mengisinya, tentunya diperlukan metode lain untuk mendapatkan artefak, seperti menjelajah dunia *Teyvat*, ataupun mengalahkan monster untuk mendapatkan artefak dari para monster.

Terdapat beberapa simbol yang bisa ditandai oleh pemain yang perlu diperhatikan dalam permainan ini. Pertama adalah simbol teleport (*starting point*), pemain dapat melakukan teleportasi ke tempat tersebut. Kedua adalah simbol clover, pada tempat ini terdapat banyak sekali artefak yang bisa didapatkan (*destination point*). Serta simbol ketiga merupakan simbol bintang, tempat ini mengandung artefak yang dapat diambil oleh pemain, namun karena jumlahnya sangat sedikit, maka pemain dapat menghindari/melewatinya apabila tidak sejalur dengan destinasi untuk menghemat waktu.



Gambar 7. Simbol di Genshin Impact

III. PENERAPAN ALGORITMA A* DALAM MENENTUKAN RUTE FARMING ARTIFACT TERCEPAT DI GENSHIN IMPACT

Pada game Genshin Impact, terdapat banyak lokasi dimana player dapat mendapatkan artifact dalam jumlah yang banyak (simbol clover). Salah satu tempatnya berada pada bagian timur Jueyun Karst.

Pada daerah timur Jueyun Karst, terdapat sebuah tempat tersembunyi yang menampung sejumlah artefak yang dapat diambil oleh pemain.

Maka, yang harus dilakukan pertama kali adalah buat sebuah sambungan simpul-simpul hingga membentuk sebuah graf, seperti pada Gambar 9.

Setelah itu, perkiraan cost dari sebuah simpul ke simpul tujuan dapat dihitung dengan satuan panjang dari peta tersebut, sehingga didapatkan seperti Tabel 2.



Gambar 8. Bagian Timur Jueyun Karst



Gambar 9. Graf Bagian Timur Jueyun Karst

Simpul	Koordinat x	Koordinat y
Teleport	360	80
A	210	60
B	540	370
C	680	420
D	780	440
E	130	560
Clover	510	590

Tabel 1. Tabel koordinat setiap simpul sesuai dengan skala satuan panjang dari peta

n	Teleport	A	B	C	D	E	Clover
$h(n)$	530	610	220	240	310	380	0

Tabel 2. Tabel perkiraan cost dengan skala satuan panjang dari peta

Untuk menentukan jalur tercepat *farming* artefak pada peta ini, dapat diterapkan algoritma A*, langkah pertama yang akan dilakukan adalah menentukan simpul asal. Teleport adalah tempat dimana pemain biasa memulai untuk melakukan *farming*, dan bisa dijadikan acuan sebagai simpul awal (*starting point*).

Langkah kedua adalah mencari tetangga dari simpul Teleport, yaitu terdapat simpul A, B, dan E, dan dihitung masing-masing cost dengan rumus $f(n) = g(n) + h(n)$.



Gambar 10.1. Langkah kedua Algoritma A* (menghitung cost tetangga dari simpul Teleport)

Dari langkah kedua, akan didapatkan cost dari teleport ke simpul A bernilai 760 (150 + 610), dari teleport ke simpul B bernilai 560 (340 + 220), serta dari teleport ke simpul E bernilai 910 (530 + 380). Karena *cost* dari teleport ke simpul B memiliki value terkecil, maka selanjutnya akan diambil simpul B sebagai acuan.



Gambar 10.2. Langkah kedua Algoritma A* (mendapatkan cost terkecil yaitu dari simpul Teleport ke simpul B)

Langkah ketiga adalah dengan menjadikan simpul B sebagai acuan, akan dicari *cost* ke tetangga simpul B dari simpul B itu sendiri. Maka akan didapatkan *cost* dari simpul B ke simpul A bernilai 1400 (340 + 450 + 610), dari simpul B ke simpul C bernilai 730 (340 + 150 + 240), serta dari simpul B ke simpul Clover bernilai 560 (340 + 220).



Gambar 10.3. Langkah ketiga Algoritma A* (menghitung cost tetangga dari simpul B)

Setelah menghitung dengan algoritma A* dengan simpul B sebagai acuan pada langkah ketiga, didapatkan cost terkecil yang mencapai simpul Clover (simpul tujuan), maka algoritma A* berakhir disini, dan didapatkan solusi rute terpendek dari teleport ke simpul tujuan seperti pada Gambar 10.4.



Gambar 10.4. Langkah ketiga Algoritma A* (mendapatkan solusi rute terpendek dari simpul awal ke simpul tujuan)

IV. KESIMPULAN

Algoritma A* merupakan algoritma yang dapat memberikan sebuah solusi optimal dalam penentuan rute terpendek dari sebuah graf berbobot. Penerapan algoritma A* dalam melakukan *farming artifact* di Genshin Impact memerlukan algoritma yang mampu mengoptimasi jarak tempuh dari sebuah titik (titik awal) hingga ke titik tujuan dan memerlukan waktu yang singkat.

Akan tetapi, apabila pemain memiliki waktu yang lebih untuk mengumpulkan semua artefak di seluruh lokasi dalam sebuah tempat, maka dapat digunakan algoritma *Branch and Bound (Travelling Salesman Problem)* untuk mendapatkan solusi dari masalah tersebut.

Penerapan Algoritma A* tidak hanya dapat diterapkan dalam penentuan rute terpendek, namun masih terdapat banyak penerapan lainnya yang bisa menggunakan algoritma A* dalam menyelesaikan sebuah kasus masalah dan mendapatkan solusi teroptimal.

UCAPAN TERIMA KASIH

Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasihnya kepada Tuhan Yang Maha Esa sehingga penulis mampu menyelesaikan masalah ini. Penulis juga ingin mengucapkan terima kasih kepada orang tua penulis atas dukungan yang diberikan kepada penulis. Penulis juga ingin menyampaikan rasa terima kasih kepada Tim Pengajar Strategi Algoritma IF2211 Tahun 2020/2021, terutama Pak Rila Mandala selaku dosen K1 yang telah membimbing penulis selama perkuliahan.

REFERENCES

- [1] <https://gizmologi.id/games/genshin-impact-konsep-open-world/> diakses terakhir pada 9 Mei 2021
- [2] Munir, Rinaldi. *Matematika Diskrit*, Informatika, Bandung. Dari : <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm> diakses pada 10 Mei 2021
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Route-Planning-Bagian2-2021.pdf> diakses pada 10 Mei 2021
- [4] <http://matdisstuff.blogspot.com/> diakses pada 10 Mei 2021
- [5] Gede Wahyu Antara Dalem, Ida Bagus. *Penerapan Algoritma A* (STAR) Menggunakan Graph untuk Menghitung Jarak Terpendek*. Dari : <https://media.neliti.com/media/publications/235453-penerapan-algoritma-a-star-menggunakan-g-fa0b1902.pdf> diakses pada 10 Mei 2021

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.

Medan, 11 Mei 2021

Wilbert Fangderson 13519025