

Penggunaan Algoritma Branch & Bound Untuk Menentukan Rute Pengambilan Anemoculus Daerah Windrise Tercepat di Genshin Impact

Feralezer L. G. Tampubolon - 13519062
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13519062@std.stei.itb.ac.id

Abstrak—Traveling Salesman Problem (TSP) adalah persoalan optimasi klasik dalam dunia komputasi. Contoh kasus TSP ini bisa kita temukan pada *video game* Genshin Impact, jika kita ingin mencari rute pengambilan Anemoculus tercepat, misalnya di daerah Windrise. Algoritma Branch & Bound digunakan untuk menyelesaikan persoalan ini.

Kata Kunci—Traveling Salesman Problem, Genshin Impact, Branch & Bound

I. PENDAHULUAN

Traveling Salesman Problem (TSP) adalah suatu persoalan optimasi klasik dalam dunia komputasi. Secara formal, persoalan ini dapat dirumuskan sebagai berikut: Jika diketahui sekumpulan kota, jarak dari setiap kota ke kota yang lain, dan kota asal, tentukan jalur perjalanan terpendek dari kota asal yang memastikan bahwa selama perjalanan, semua kota dilalui sekali saja dan perjalanan berakhir kembali di kota asal (jalur ini disebut “tur”). Persoalan ini Meskipun ide tentang persoalan TSP ini telah ada sejak era hidupnya matematikawan seperti Hamilton di tahun 1800an, persoalan ini pertama kali dibahas secara formal dan matematis pada tahun 1930an oleh Merrill M. Flood. Masalah ini semakin terkenal, terutama pada bidang IT. Komputer dengan cepat berkembang sebagai alat hitung terbaik manusia, dan komputer saat itu (hingga sekarang) hanya bisa menyelesaikan persoalan ini secara NP.

Video game adalah permainan yang dijalankan pada alat elektronik. *Video game* pertama dibuat pada tahun 1947 dengan sebutan “alat hiburan berbasis tabung sinar katoda”. Seiring berkembangnya komputer, *video game* juga menjadi semakin kompleks setiap iterasinya. Semakin banyak elemen pada *video game* yang dibuat menyerupai keadaan pada dunia nyata. Yang penting, elemen tersebut tidak menghilangkan keasyikan bermain *video game*. Salah satu elemen *video game* ini adalah peta, dan tentu saja persoalan TSP akan hadir bersama dengannya.

Genshin Impact adalah sebuah *video game* berbasis PC/PlayStation/mobile yang dikembangkan oleh miHoYo, dirilis pada tanggal 28 September 2020. *Video game* ber-genre RPG (*Role Playing Game*) ini mengisahkan perjuangan seorang pengembara, bertahan hidup di suatu dunia yang asing sembari

mencari saudara kandungnya yang secara misterius terpisah darinya. Genshin Impact dengan cepat menjadi populer, terutama di kalangan pengguna *smartphone*. Saat ini, Genshin Impact masih menjadi salah satu aplikasi populer di platform distribusi aplikasi Android dan iOS.



Gambar 1. Logo Genshin Impact (sumber: upload.wikimedia.org)

Sebagai *video game* ber-genre RPG, navigasi menjadi sesuatu yang penting dalam Genshin Impact. Genshin Impact membekali setiap pemain dengan peta agar pemain tidak tersesat. Di sisi lain, sebagai RPG, sudah pasti ada objek-objek langka yang tersebar di dunia permainan. Objek-objek ini jumlahnya terbatas, dan pemain akan mendapat hadiah jika berhasil mengoleksi semuanya. Salah satu objek yang bisa dikoleksi dengan cara seperti ini di Genshin Impact memiliki nama “Anemoculus”.



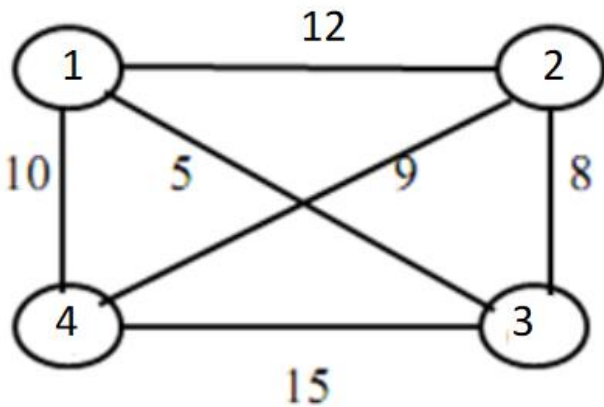
Gambar 2. Anemoculus (sumber: genshin-impact.fandom.com)

Terdapat 65 Anemoculus di dalam dunia Genshin Impact, yang terdapat pada sejumlah daerah berbeda, salah satu daerah tersebut bernama Windrise. Proses koleksi Anemoculus pada semua daerah ini secara total akan memakan waktu beberapa jam. Ini bisa dianggap memberatkan, apalagi jika pemain seharusnya sibuk/tidak memiliki banyak waktu untuk bermain. Untuk pemain-pemain seperti ini, rute pengambilan Anemoculus yang cepat menjadi sangat diperlukan. Makalah ini akan membahas rute pengambilan Anemoculus tercepat untuk daerah Windrise. Untuk melakukannya, dibutuhkan pemahaman mengenai cara menyelesaikan TSP menggunakan algoritma branch & bound.

II. LANDASAN TEORI

A. Algoritma Branch & Bound

Algoritma Branch & Bound adalah algoritma optimasi yang pada setiap pengambilan keputusan akan memilih jalur dengan *cost* (biaya) paling rendah. *Cost* di sini adalah fungsi yang dibuat secara kasus per kasus, akan berbeda untuk setiap jenis persoalan. Di sisi lain, ada juga fungsi *bound* (pembatas). Fungsi pembatas memastikan bahwa solusi yang tidak mungkin tidak akan pernah ditelusuri. Pada umumnya fungsi ini lebih sering dipakai ketika sudah ada paling tidak satu solusi yang ditemukan, dan kita ingin mencari alternatif solusi yang lebih efisien. Pada saat itu, fungsi pembatas akan mencegah kita mengambil jalur-jalur yang sudah jelas akan menghasilkan solusi yang lebih buruk daripada solusi yang telah kita temukan.



Gambar 3. Graf 4 simpul (sumber: informatika.stei.itb.ac.id/~rinaldi.munir)

Untuk kasus TSP, fungsi *cost* pada branch & bound akan dihitung menggunakan *Reduced Cost Matrix*. Fungsi *cost* di sini juga akan sekaligus berperan sebagai fungsi pembatas. Misalnya pada graf Gambar 3, kita ingin mencari tur dengan bobot minimum (lintasan terpendek), katakanlah dengan simpul awal (dan akhir) = 1. Ini berarti rute mencakup 1, lalu semua kombinasi angka 2, 3, dan 4, sebelum akhirnya kembali lagi ke 1. Pertama-tama, kita cari terlebih dahulu *Reduced Cost Matrix* untuk simpul akar (1). *Reduced Cost Matrix* menggambarkan biaya yang diperlukan untuk berpindah dari satu simpul ke simpul yang lain. Berikut ini langkah-langkah membuat *Reduced Cost Matrix* untuk simpul akar:

1. Catat biaya yang diperlukan untuk berpindah dari 1 simpul ke simpul lain (contoh, indeks (1,2) menunjukkan biaya yang diperlukan untuk berpindah dari simpul 1 ke simpul 2).

$$\begin{bmatrix} \infty & 12 & 5 & 10 \\ 12 & \infty & 8 & 9 \\ 5 & 8 & \infty & 15 \\ 10 & 9 & 15 & \infty \end{bmatrix}$$

2. Kurangi biaya pada setiap baris agar pada masing-masing baris, terdapat paling tidak satu sel bernilai 0 dan tidak ada sel bernilai negatif.

$$\begin{bmatrix} \infty & 12 & 5 & 10 \\ 12 & \infty & 8 & 9 \\ 5 & 8 & \infty & 15 \\ 10 & 9 & 15 & \infty \end{bmatrix} \begin{matrix} R1-5 \\ R2-8 \\ R3-5 \\ R4-9 \end{matrix} \begin{bmatrix} \infty & 7 & 0 & 5 \\ 4 & \infty & 0 & 1 \\ 0 & 3 & \infty & 10 \\ 1 & 0 & 6 & \infty \end{bmatrix}$$

3. Kurangi biaya pada setiap kolom agar pada masing-masing kolom, terdapat paling tidak satu sel bernilai 0 dan tidak ada sel bernilai negatif.

$$\begin{bmatrix} \infty & 7 & 0 & 5 \\ 4 & \infty & 0 & 1 \\ 0 & 3 & \infty & 10 \\ 1 & 0 & 6 & \infty \end{bmatrix} C4-1 \begin{bmatrix} \infty & 7 & 0 & 4 \\ 4 & \infty & 0 & 0 \\ 0 & 3 & \infty & 9 \\ 1 & 0 & 6 & \infty \end{bmatrix} = A$$

A adalah *Reduced Cost Matrix* untuk simpul akar. Khusus untuk simpul akar, fungsi *cost* dihitung dengan cara menjumlahkan pengurang semua baris dan kolom, sehingga:

$$\hat{c}(1) = 5 + 8 + 5 + 9 + 1 = 28$$

Diperoleh $\hat{c}(1)$, fungsi *cost* simpul akar sekaligus batas bawah bagi semua perhitungan anaknya. Jika digambarkan dalam format pohon ruang status, maka pohon ruang statusnya adalah:



Gambar 4. Pohon ruang status, iterasi ke-1

Dari simpul akar (1), kita bisa bepergian ke simpul 2, 3, dan 4. 28 sebagai batas bawah berarti $\hat{c}(2)$, $\hat{c}(3)$, dan $\hat{c}(4)$ ketiganya pasti bernilai (simpul 2, 3, dan 4 berbobot) ≥ 28 . Cari *Reduced Cost Matrix* dan fungsi *cost* untuk masing-masing simpul anak ini. Contoh berikut adalah untuk simpul 2. Pertama-tama, untuk *Reduced Cost Matrix*:

1. Dari *Reduced Cost Matrix* simpul akar, ubah semua nilai pada baris 1, kolom 2, dan $A(2,1)$ menjadi ∞ .

$$\begin{bmatrix} \infty & 7 & 0 & 4 \\ 4 & \infty & 0 & 0 \\ 0 & 3 & \infty & 9 \\ 1 & 0 & 6 & \infty \end{bmatrix} \begin{matrix} R1 + \infty \\ C2 + \infty \\ A(2,1) + \infty \end{matrix} \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 0 \\ 0 & \infty & \infty & 9 \\ 1 & \infty & 6 & \infty \end{bmatrix}$$

2. Ikuti langkah 2 dan langkah 3 pada pembuatan *Reduced Cost Matrix* simpul akar pada setiap baris/kolom yang tidak hanya terdiri dari ∞ (kalau ada baris/kolom non- ∞ yang belum memiliki sel bernilai 0, reduksi 1 baris/kolom tersebut dengan urutan baris dulu, baru kolom). Pada kasus ini, yang direduksi hanyalah baris 4.

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 0 \\ 0 & \infty & \infty & 9 \\ 1 & \infty & 6 & \infty \end{bmatrix} R_{4-1} \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 0 \\ 0 & \infty & \infty & 9 \\ 0 & \infty & 5 & \infty \end{bmatrix} = B$$

Adapun berikut ini fungsi *cost* menggunakan *Reduced Cost Matrix* untuk simpul yang bukan simpul akar:

$$\hat{c}(S) = \hat{c}(R) + A(i, j) + r$$

Dengan:

$\hat{c}(S)$ = bobot perjalanan minimum yang melalui simpul S (simpul di pohon ruang status)

$\hat{c}(R)$ = bobot perjalanan minimum yang melalui simpul R (R adalah orangtua dari S)

$A(i, j)$ = bobot sisi (i, j) pada graf yang berkoresponden dengan sisi (R, S) pada pohon ruang status

r = jumlah semua pengurang pada proses memperoleh matriks tereduksi untuk simpul S

Sehingga, diperoleh:

$$\hat{c}(2) = \hat{c}(1) + A(1,2) + r = 28 + 7 + 1 = 36$$

Lakukan langkah di atas untuk simpul 3:

$$\begin{bmatrix} \infty & 7 & 0 & 4 \\ 4 & \infty & 0 & 0 \\ 0 & 3 & \infty & 9 \\ 1 & 0 & 6 & \infty \end{bmatrix} R_1 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & \infty & 0 \\ \infty & 3 & \infty & 9 \\ 1 & 0 & \infty & \infty \end{bmatrix} C_3 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & \infty & 0 \\ \infty & 3 & \infty & 9 \\ 1 & 0 & \infty & \infty \end{bmatrix} A(3,1) + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & \infty & 0 \\ \infty & 3 & \infty & 9 \\ 1 & 0 & \infty & \infty \end{bmatrix} R_{3-3}$$

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & \infty & 0 \\ \infty & 0 & \infty & 6 \\ 1 & 0 & \infty & \infty \end{bmatrix} C_{1-1} \begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & 0 \\ \infty & 0 & \infty & 6 \\ 0 & 0 & \infty & \infty \end{bmatrix} = C$$

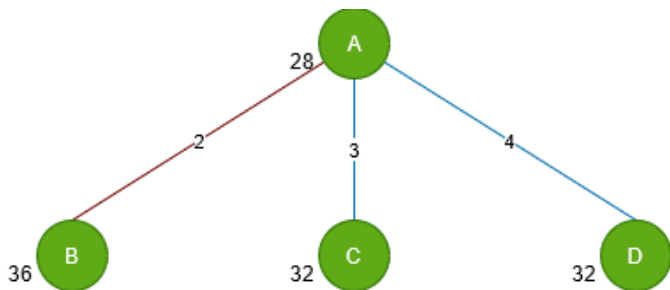
$$\hat{c}(3) = \hat{c}(1) + A(1,3) + r = 28 + 0 + 4 = 32$$

dan simpul 4:

$$\begin{bmatrix} \infty & 7 & 0 & 4 \\ 4 & \infty & 0 & 0 \\ 0 & 3 & \infty & 9 \\ 1 & 0 & 6 & \infty \end{bmatrix} R_1 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & 0 & \infty \\ 0 & 3 & \infty & \infty \\ \infty & 0 & 6 & \infty \end{bmatrix} C_4 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & 0 & \infty \\ 0 & 3 & \infty & \infty \\ \infty & 0 & 6 & \infty \end{bmatrix} A(4,1) + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & 0 & \infty \\ 0 & 3 & \infty & \infty \\ \infty & 0 & 6 & \infty \end{bmatrix} = D$$

$$\hat{c}(4) = \hat{c}(1) + A(1,4) + r = 28 + 4 + 0 = 32$$

Diperoleh $\hat{c}(2)$, $\hat{c}(3)$, dan $\hat{c}(4)$ berturut-turut memiliki bobot 36, 32, dan 32.



Gambar 5. Pohon ruang status, iterasi ke-2, A-C dan A-D bobotnya minimum

Karena $\hat{c}(3)$ dan $\hat{c}(4)$ sama-sama bernilai 32, maka pilih salah satu simpul: 3 atau 4 sebagai langkah selanjutnya.

Misalnya kita pilih simpul 3, maka simpul selanjutnya yang mungkin adalah simpul 2 dan 4, dengan fungsi *cost* simpul 2:

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & 0 \\ \infty & 0 & \infty & 6 \\ 0 & 0 & \infty & \infty \end{bmatrix} R_3 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \end{bmatrix} C_2 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \end{bmatrix} C(2,3) + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \end{bmatrix} = E$$

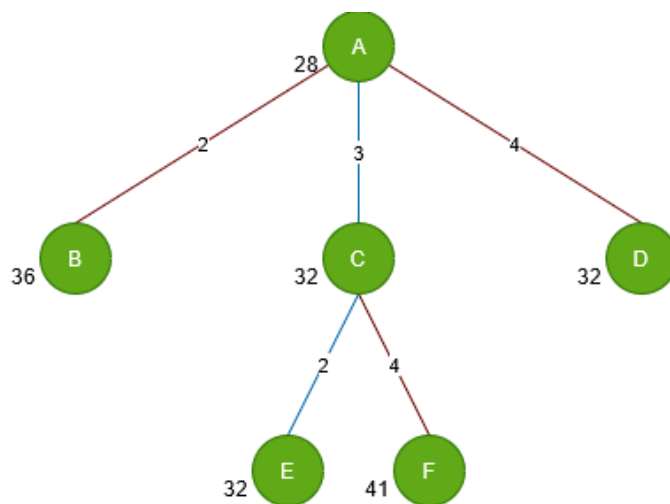
$$\hat{c}(2) = \hat{c}(3) + C(3,2) + r = 32 + 0 + 0 = 32$$

dan simpul 4:

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & 0 \\ \infty & 0 & \infty & 6 \\ 0 & 0 & \infty & \infty \end{bmatrix} R_3 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty \end{bmatrix} C_4 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty \end{bmatrix} R_{2-3}$$

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty \end{bmatrix} = F$$

$$\hat{c}(4) = \hat{c}(3) + C(3,4) + r = 32 + 6 + 3 = 41$$



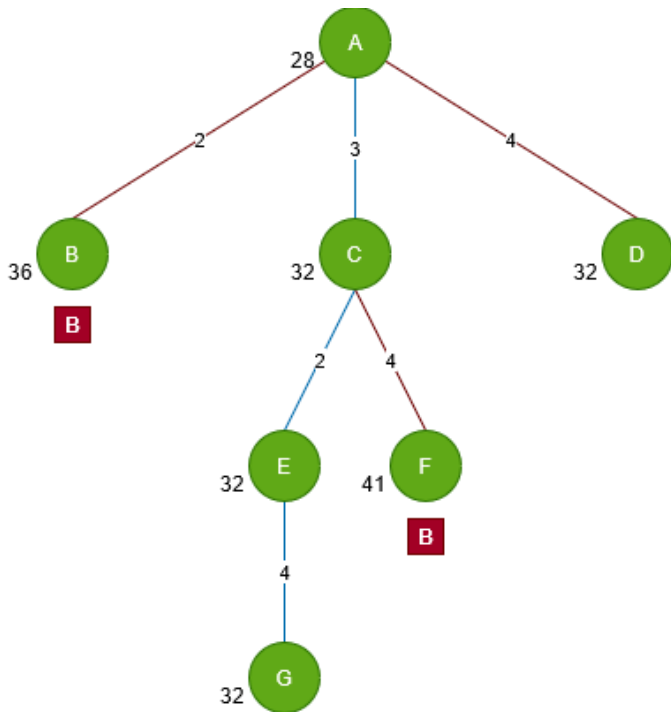
Gambar 6. Pohon ruang status, iterasi ke-3

Karena $\hat{c}(2) = 32$, maka simpul selanjutnya adalah simpul 2. Simpul yang tersisa adalah simpul 4 dengan:

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 3 & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \end{bmatrix} R_2 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \end{bmatrix} C_4 + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \end{bmatrix} E(4,2) + \infty \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \end{bmatrix} = G$$

$$\hat{c}(4) = \hat{c}(2) + E(2,4) + r = 32 + 0 + 0 = 32$$

G(4,1) tentu saja jadi satu-satunya sel yang tersisa karena setelah dari simpul 4, kita masih harus kembali ke simpul 1. Maka ditemukan tur (jalur) pertama: **1-3-2-4-1** dengan bobot (panjang lintasan) **32**. Solusi ini adalah solusi terbaik sejauh ini. Fungsi pembatas akan memberi batas = 32: Semua daun pada pohon ruang status yang bobotnya > 32 akan dimatikan.



Gambar 7. Pohon ruang status, iterasi ke-4

Perhatikan bahwa perhitungan belum selesai. Daun D yang berbobot 32 masih hidup. Singkat cerita, karena graf ini adalah graf yang bobotnya tidak mengenal arah (bobot simpul 1-2 sama dengan simpul 2-1 dst.), maka tentu saja, kalau ada tur 1-3-2-4-1, akan ada pula tur 1-4-2-3-1 dengan bobot (panjang lintasan) 32. Jika ingin dihitung, maka setelah simpul 4, akan dicari nilai fungsi *cost* simpul 2:

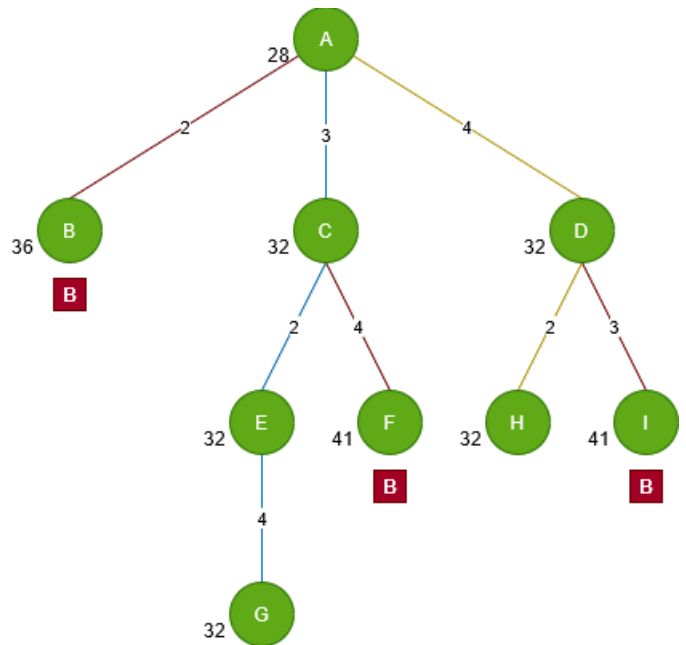
$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & 0 & \infty \\ 0 & 3 & \infty & \infty \\ \infty & 0 & 6 & \infty \end{bmatrix} \begin{matrix} R4 + \infty \\ C2 + \infty \\ D(2,4) + \infty \end{matrix} \begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & 0 & \infty \\ 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix} = H$$

$$\hat{c}(2) = \hat{c}(4) + D(4,2) + r = 32 + 0 + 0 = 32$$

dan simpul 3:

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & 0 & \infty \\ 0 & 3 & \infty & \infty \\ \infty & 0 & 6 & \infty \end{bmatrix} \begin{matrix} R4 + \infty \\ C3 + \infty \\ D(3,4) + \infty \end{matrix} \begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & \infty & \infty \\ 0 & 3 & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix} = I$$

$$\hat{c}(3) = \hat{c}(4) + D(4,3) + r = 32 + 6 + 0 = 38$$



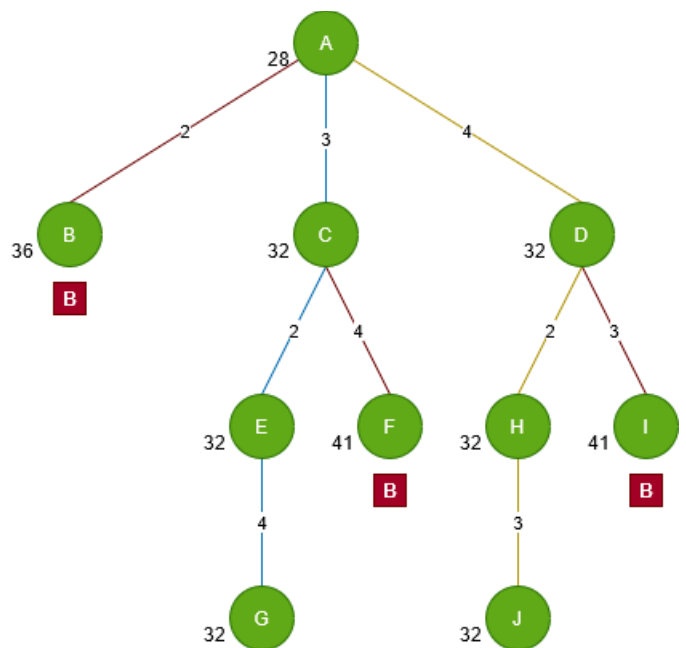
Gambar 8. Pohon ruang status, iterasi ke-5

Daun I (simpul 3) otomatis terkena fungsi pembatas karena $\hat{c}(3) = 38 > 32$. Maka yang tersisa adalah simpul 2 yang dilanjutkan ke simpul 3:

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 4 & \infty & 0 & \infty \\ 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix} \begin{matrix} R2 + \infty \\ C3 + \infty \\ H(3,2) + \infty \end{matrix} \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix} = J$$

$$\hat{c}(3) = \hat{c}(2) + H(2,3) + r = 32 + 0 + 0 = 32$$

Diperoleh tur **1-4-2-3-1** dengan bobot **32**.



Gambar 9. Pohon ruang status, iterasi ke-6, final

B. Anemoculus Daerah Windrise di Genshin Impact

Windrise adalah salah satu daerah pertama yang dikunjungi pemain pada video game Genshin Impact. Daerah ini adalah daerah *noob-friendly*, cocok untuk pemain pemula. Dalam hal ini, bioma Windrise bisa dikatakan cukup sederhana jika dibandingkan dengan daerah-daerah yang akan pemain hadapi selanjutnya: Windrise adalah suatu padang rumput luas dengan pohon raksasa di tengah-tengahnya, perbedaan ketinggian antara satu titik dengan titik yang lain nyaris tidak ada, sehingga pemain bisa mengeksplorasi daerah ini tanpa perlu memanjat (suatu fitur dari Genshin Impact yang akan sering dijumpai pada daerah-daerah selanjutnya). Dari total 65 Anemoculus di dalam dunia Genshin Impact, 7 terletak pada Windrise.



Gambar 10. Peta Windrise beserta 7 Anemoculusnya (sumber: webstatic-sea.mihoyo.com /app/ys-map-sea)

Karena masing-masing Anemoculus tersebut memiliki posisi dalam bentuk titik koordinat kartesius (x, y) , jarak antar Anemoculus dapat dihitung menggunakan *Euclidean Distance*.

C. Euclidean Distance

Euclidean Distance adalah teorema yang dapat digunakan untuk menentukan jarak antar dua titik dalam suatu bidang koordinat kartesius. Jika ada dua titik $A(x_1, y_1)$ dan $B(x_2, y_2)$, jarak d antara kedua titik tersebut dapat dicari dengan rumus:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Dengan:

- x_1 = posisi horizontal titik A
- y_1 = posisi vertikal titik A
- x_2 = posisi horizontal titik B
- y_2 = posisi vertikal titik B

III. DATA DAN ANALISIS

Pertama-tama, penulis melakukan penomoran terhadap masing-masing Anemoculus. Penulis juga mencatat koordinat kartesius masing-masing Anemoculus tersebut. Nomor 1

diberikan kepada Anemoculus yang pasti akan pemain temukan pertama kali, karena Anemoculus tersebut merupakan bagian dari *tutorial* Genshin Impact.



Gambar 11. Peta Windrise beserta 7 Anemoculus yang sudah diberi nomor (sumber: webstatic-sea.mihoyo.com /app/ys-map-sea)

TABEL I. KOORDINAT KARTESIUS ANEMOCULUS

No. Anemoculus	Koordinat	
	x	y
1	774	-500
2	767	-686
3	1040	-565
4	457	-397
5	757	-323
6	941	-272
7	692	-210

sumber: webstatic-sea.mihoyo.com /app/ys-map-sea

Dari data di atas, kita bisa tentukan jarak antara tiap-tiap Anemoculus. Jarak ini adalah matriks awal yang akan kita ubah menjadi *Reduced Cost Matrix*.

TABEL II. JARAK ANTAR ANEMOCULUS

	1	2	3	4	5	6	7
1	∞	186	274	333	178	283	301
2	186	∞	299	424	363	449	482
3	274	299	∞	607	372	309	497
4	333	424	607	∞	309	500	300
5	178	363	372	309	∞	191	130
6	283	449	309	500	191	∞	257
7	301	482	497	300	130	257	∞

catatan: dilakukan pembulatan ke bilangan bulat terdekat

Pilih 1 sebagai simpul awal, maka *Reduced Cost Matrix*nya adalah:

∞	186	274	333	178	283	301	R1-178
186	∞	299	424	363	449	482	R2-186
274	299	∞	607	372	309	497	R3-274
333	424	607	∞	309	500	300	R4-300
178	363	372	309	∞	191	130	R5-130
283	449	309	500	191	∞	257	R6-191
301	482	497	300	130	257	∞	R7-130

$$\begin{bmatrix} \infty & 8 & 96 & 155 & 0 & 105 & 123 \\ 0 & \infty & 113 & 238 & 177 & 263 & 296 \\ 0 & 25 & \infty & 333 & 98 & 35 & 223 \\ 33 & 124 & 307 & \infty & 9 & 200 & 0 \\ 48 & 233 & 242 & 179 & \infty & 61 & 0 \\ 92 & 258 & 118 & 309 & 0 & \infty & 66 \\ 171 & 352 & 367 & 170 & 0 & 127 & \infty \end{bmatrix} \begin{matrix} \text{C2-8} \\ \text{C3-96} \\ \text{C4-155} \\ \text{C6-35} \end{matrix}$$

$$\begin{bmatrix} \infty & 0 & 0 & 0 & 0 & 70 & 123 \\ 0 & \infty & 17 & 83 & 177 & 228 & 296 \\ 0 & 17 & \infty & 178 & 98 & 0 & 223 \\ 33 & 116 & 211 & \infty & 9 & 165 & 0 \\ 48 & 225 & 146 & 24 & \infty & 26 & 0 \\ 92 & 250 & 22 & 154 & 0 & \infty & 66 \\ 171 & 344 & 271 & 15 & 0 & 92 & \infty \end{bmatrix} = A$$

$$\hat{c}(1) = 178 + 186 + 274 + 300 + 130 + 191 + 130 + 8 + 96 + 155 + 35 = 1683$$

Tentu saja, perhitungan di atas cukup panjang jika ingin dilakukan secara manual. Menggunakan program dengan kode yang menerapkan algoritma Branch & Bound dengan Reduced Cost Matrix di atas, diperoleh tur **1-2-3-6-5-7-4-1** serta inversnya **1-4-7-5-6-3-2-1** dengan bobot (panjang lintasan) **1748**.

IV. KESIMPULAN

Rute pengambilan Anemoculus daerah Windrise tercepat di Genshin Impact adalah 1-2-3-6-5-7-4-1.

UCAPAN TERIMA KASIH

Penulis ingin mengucapkan syukur kepada Tuhan karena tanpa bimbingan dan penyertaan-Nya makalah ini tidak dapat selesai. Penulis juga ingin berterima kasih kepada seluruh tim pengajar IF2211 Strategi Algoritma Institut Teknologi Bandung, Semester II Tahun 2020/2021 yang telah mengajarkan ilmu-ilmu yang diperlukan penulis untuk dapat mengerjakan makalah ini dengan baik.

DAFTAR PUSTAKA

- [1] T. T. Goldsmith, Jr., et al., *Cathode-ray Tube Amusement Device*. United States Patent Office, 1948.
- [2] miHoYo 2020, *Genshin Impact – Step Into a Vast Magical World of Adventure*. miHoYo, diakses pada 11 Mei 2021 <<https://genshin.mihoyo.com/en/game>>
- [3] Fandom 2021, *Anemoculus*. Genshin Impact Wiki, diakses pada 11 Mei 2021 <<https://genshin-impact.fandom.com/wiki/Anemoculus>>
- [4] miHoYo 2021, *Teyvat Interactive Map*. miHoYo, diakses pada 11 Mei 2021 <<https://webstatic-sea.mihoyo.com/app/ys-map-sea/index.html>>
- [5] R. Munir 2021, *Pengantar Strategi Algoritma*. Homepage Rinaldi Munir, diakses pada 11 Mei 2021 <[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pengantar-Strategi-Algoritma-\(2021\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pengantar-Strategi-Algoritma-(2021).pdf)>
- [6] R. Munir 2021, *Algoritma Branch & Bound (Bagian 1)*. Homepage Rinaldi Munir, diakses pada 11 Mei 2021 <<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian1.pdf>>
- [7] R. Munir 2021, *Algoritma Branch & Bound (Bagian 2)*. Homepage Rinaldi Munir, diakses pada 11 Mei 2021 <<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian2.pdf>>
- [8] R. Munir 2021, *Algoritma Branch & Bound (Bagian 3)*. Homepage Rinaldi Munir, diakses pada 11 Mei 2021 <<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian3.pdf>>

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.

Tangerang Selatan, 11 Mei 2021



Feralezer L. G. Tampubolon 13519062