

# Penerapan Pohon Merentang Minimum untuk Menentukan Rute Terpendek untuk Membuka Teleport Waypoint di Negara Mondstadt pada Permainan Genshin Impact

Aria Bachrul Ulum Berlian 13519115<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>13519115@itb.ac.id

**Abstract**—*Genshin Impact* merupakan permainan video berbasis *role-playing game* yang sangat populer di tahun 2020. *Genshin Impact* memiliki peta dunia yang cukup luas. Di dalam permainan ini, terdapat fitur yang memungkinkan pemain berpindah dari satu tempat ke tempat yang lain dengan cara teleport untuk mengelilingi dunia Teyvat tersebut. Hanya beberapa tempat yang dapat dituju dengan teleport, salah satunya adalah teleport waypoint. Teleport waypoint ini tidak bisa dituju dengan teleport apabila pemain belum membukanya dengan cara berinteraksi langsung dengan Teleport waypoint. Makalah ini membahas cara menentukan rute terdekat untuk membuka seluruh teleport waypoint di negara Mondstadt dengan menerapkan konsep pohon merentang minimum.

**Keywords**—*peta, teleport waypoint, pohon merentang minimum, algoritma prim.*

## I. PENDAHULUAN

Pada tanggal 28 September 2020, MiHoYo, pengembang permainan video, merilis sebuah permainan video yang berjudul *Genshin Impact*. *Genshin Impact* merupakan permainan video gratis yang ber-*genre* aksi dan *role-playing game* atau biasa disebut *RPG*. Permainan ini dapat dimainkan di PC, Playstation 4, bahkan di selular.

Seperti pada permainan *RPG* pada umumnya, *Genshin Impact* memiliki dunia yang dapat dijelajahi pemain. Dunia *Genshin Impact* yakni Teyvat, memiliki wilayah yang sangat luas. Untuk memudahkan pemain mengelilingi dunia, MiHoYo membuat fitur teleport. Pemain dapat berpindah dari manapun menuju ke tempat-tempat tertentu dengan menggunakan teleport. Tempat-tempat yang dapat dituju salah satunya adalah teleport waypoint. Teleport waypoint tersebar di banyak tempat di Teyvat. Sayangnya, pemain tidak bisa begitu saja berpindah ke teleport waypoint jika pemain belum membukanya. Pemain dapat membuka teleport waypoint dengan cara berinteraksi langsung dengan mesin tersebut. Namun untuk mencapai teleport waypoint tersebut, pemain harus menempuh jarak dengan cara jalan, berlari, memanjat maupun terbang menggunakan *glider*. Perjalanan ini cukup melelahkan untuk kebanyakan pemain. Oleh karena itu, diperlukan rute terpendek

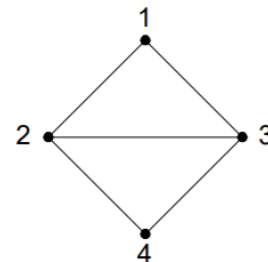
untuk mencapai seluruh teleport waypoint.

Rute terpendek untuk mencapai seluruh teleport waypoint dapat ditentukan dengan menerapkan algoritma prim untuk mendapatkan pohon merentang minimum. Makalah ini bertujuan untuk mencari rute terpendek untuk mencapai seluruh teleport waypoint di wilayah Negara Mondstadt.

## II. LANDASAN TEORI

### A. Graf

Graf adalah struktur yang terdiri dari simpul (*vertex* atau *node*) dan sisi (*edge*). Graf  $G$  terdiri dari pasangan  $V$  dan  $E$ .  $V$  adalah himpunan tidak kosong dari simpul-simpul.  $E$  adalah himpunan sisi yang menghubungkan dua simpul.

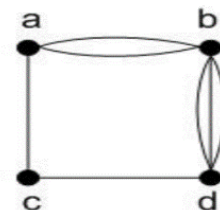


Gambar 2.1 Graf sederhana

Sumber: [1]

### B. Graf Tak Berarah

Graf tak berarah adalah graf yang sisinya tidak memiliki arah.



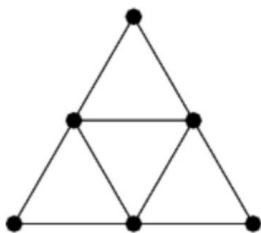
Gambar 2.2 Graf Tak Berarah

Sumber: [1]

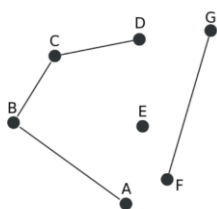
### C. Graf Terhubung

Graf terhubung adalah graf yang setiap simpulnya memiliki lintasan yang menghubungkan antar simpulnya. Jika ada dua buah simpul dalam graf  $G$  yang tidak memiliki lintasan yang menghubungkan antara keduanya maka graf tersebut dikatakan tidak terhubung.

Berikut ini contoh graf yang terhubung.



**Gambar 2.3** Graf Terhubung  
Sumber: [1]



**Gambar 2.4** Graf Tidak Terhubung  
Sumber: [1]

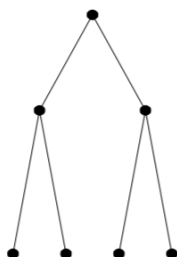
### D. Graf Berbobot

Graf berbobot adalah graf yang di setiap sisinya memiliki nilai atau harga. Harga tersebut bisa berupa jarak antara dua tempat, waktu tempuh, biaya, dll.

### E. Pohon

Pohon adalah salah satu jenis dari graf. Pohon adalah sebuah graf terhubung dan tak berarah dan tidak mengandung sirkuit<sup>[2]</sup>. Pohon  $G(V, E)$  memiliki beberapa sifat. Sifat yang pertama adalah untuk setiap pasangan simpul dalam pohon  $G$  dihubungkan dengan satu lintasan. Sifat yang kedua yaitu  $G$  adalah graf terhubung dan memiliki  $n-1$  buah sisi. Sifat yang keempat adalah  $G$  tidak memiliki sirkuit. Sifat berikutnya adalah penambahan satu sisi akan membuat satu sirkuit baru. Sifat yang terakhir adalah semua sisi dalam pohon  $G$  adalah sebuah jembatan.

Contoh pohon adalah pada silsilah keluarga, direktori berkas pada computer, dan pohon biner.



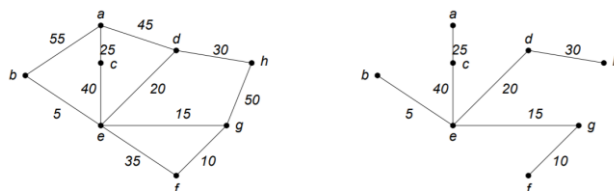
**Gambar 2.5** Pohon  
Sumber: [2]

### F. Pohon Merentang

Pohon merentang adalah pohon yang diperoleh dari sebuah graf. Dengan kata lain, pohon merentang adalah upagraf dari sebuah graf. Cara memperoleh pohon merentang adalah dengan memutus sirkuit-sirkuit pada graf.

### G. Pohon Merentang Minimum

Pohon merentang minimum adalah pohon merentang yang diperoleh dari graf berbobot dan memiliki total bobot paling kecil. Graf berbobot memungkinkan memiliki lebih dari satu pohon merentang minimum. Untuk mendapatkan pohon merentang minimum, graf berbobot harus diputuskan sirkuit-sirkuitnya. Terdapat beberapa algoritma untuk membantu membuat pohon merentang minimum, salah satunya adalah algoritma prim.

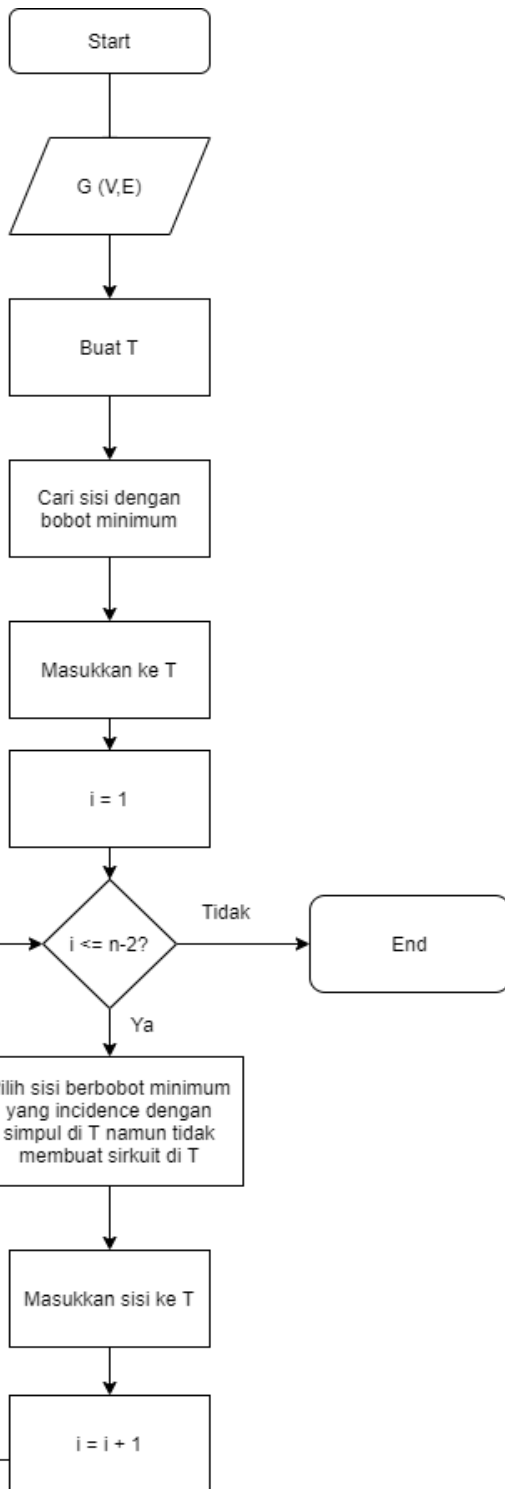


**Gambar 2.6** Pohon Merentang Minimum dari Graf  
Sumber: [2]

### H. Algoritma Prim

Algoritma prim adalah algoritma yang digunakan untuk mencari pohon minimum merentang dari sebuah graf. Ide utama dalam algoritma prim adalah untuk memilih sisi (*edge*) yang memiliki bobot paling kecil dari graf yang terhubung dengan pohon merentang yang telah terbentuk. Namun, sisi yang memiliki bobot paling kecil tersebut tidak boleh membentuk sirkuit saat dihubungkan dengan pohon merentang.

Algoritma prim memiliki tiga langkah utama. Langkah pertama yaitu mengambil sisi (*edge*) yang memiliki bobot paling kecil lalu dimasukan ke dalam pohon merentang ( $T$ ). Langkah kedua adalah memilih sisi yang memiliki bobot minimum dan bersisian dengan simpul di  $T$  namun tidak membuat sirkuit di  $T$ . Langkah terakhir adalah mengulang langkah kedua hingga  $n-2$  kali.



**Gambar 2.7** Diagram Alir Algoritma Prim

### III. PENCARIAN RUTE TELEPORT WAYPOINT DI NEGARA MONDSTANDT

#### A. Teleport Waypoint

Menurut penjelasan dari permainan Genshin Impact, teleport waypoint adalah mesin aneh yang tersebar di seluruh Teyvat. Mesin ini hanya bisa digunakan saat petualang sudah berinteraksi langsung. Tidak ada cara lain untuk berinteraksi selain dengan cara mengunjungi mesin-mesin ini. Saat mesin sudah terbuka, pemain dapat langsung berpindah ke salah satu

teleport waypoint dari manapun.

Pada *update* 1.1, terdapat 61 buah teleport waypoint yang tersebar di Teyvat. Dua puluh satu diantaranya terdapat di Negara Mondstadt. Sedangkan sisanya berada di Negara Liyue.



**Gambar 3.1** Peta Negara Mondstadt dengan Teleport Waypoint Dilingkari Merah

#### B. Penentuan Jarak Antar Teleport Waypoint

Untuk mengetahui rute terdekat untuk mengunjungi semua teleport waypoint, perlu dicari tahu terlebih dahulu jarak antar teleport waypoint. Tidak ada cara untuk mengetahui jarak virtual antar dua titik di dalam permainan Genshin Impact. Permainan ini hanya memberi informasi jarak pemain menuju misi yang akan diselesaikan. Oleh karena itu, jarak antar teleport waypoint diperoleh dengan menggunakan aplikasi pemroses gambar seperti Adobe Photoshop atau Corel Draw. Dengan aplikasi ini, kita dapat mengetahui jarak antar dua titik dengan mudah. Jarak antar kedua titik tersebut diukur dalam satuan pixel. Jarak antar dua titik yang terdekat adalah dengan menarik garis lurus antara kedua titik tersebut.

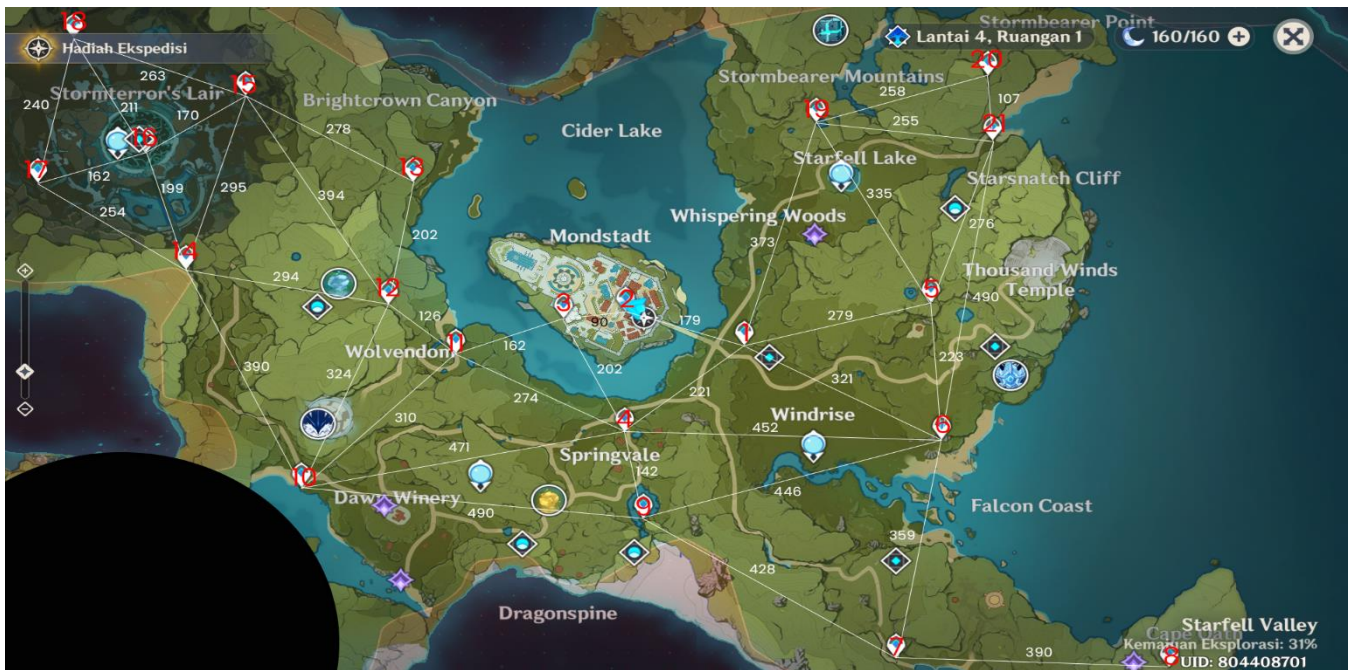


**Gambar 3.2** Penentuan Jarak dengan Adobe Photoshop

#### C. Penentuan Rute Terdekat

Langkah pertama dalam menentukan rute terdekat untuk mencapai seluruh teleport waypoint adalah dengan menganggap teleport waypoint sebagai simpul dalam graf dan jarak terdekat antar dua teleport waypoint sebagai sisi. Graf ini ditunjukkan pada gambar berikut. Setiap simpul diberi penamaan berupa angka untuk memudahkan.





**Gambar 3.3** Graf Teleport Waypoint Mondstadt

Selanjutnya, pohon merentang minimum dicari dengan menerapkan algoritma prim.

Langkah pertama dalam algoritma prim adalah menentukan sisi dengan bobot minimum. Seperti yang terlihat pada gambar 3.3, sisi dengan bobot terkecil adalah sisi yang menghubungkan simpul 2 dan 3 dengan bobot 90.



**Gambar 3.4** Langkah 1

Langkah kedua adalah mencari sisi yang berbobot minimum yang bersisian dengan simpul 2 atau 3. Dalam hal ini, yang terpilih adalah sisi antara sisi 3 dan 11 dengan bobot 162.



**Gambar 3.5** Langkah 2

Langkah ketiga adalah melakukan hal yang sama persis dengan langkah kedua. Sisi dengan bobot minimum yang bersisian dengan simpul 2, 3, atau 11 adalah sisi yang menghubungkan sisi 11 dengan 12 yang berbobot 126.



**Gambar 3.6** Langkah 3

Langkah keempat masih sama seperti langkah sebelumnya, yakni mencari sisi dengan bobot minimum yang bersisian dengan simpul 2, 3, 11, atau 12, namun tidak membuat sirkuit pada pohon merentang yang sedang dibuat. Dari gambar 3.3, dapat dilihat bahwa sisi yang memenuhi syarat adalah sisi

berbobot 179 yang menghubungkan simpul 2 dan 1.



**Gambar 3.7** Langkah 4

Langkah kelima masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul 2, 3, 11, 12, atau 1 namun tidak membuat sirkuit di T adalah sisi antara simpul 2 dan 4 yang berbobot 193.



**Gambar 3.8** Langkah 5

Langkah keenam masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul 2, 3, 11, 12, 1 atau 4 namun tidak membuat sirkuit di T adalah sisi antara simpul 12 dan 14 yang berbobot 202. Sisi antara simpul 3 dan 4 juga berbobot 202. Namun, sisi antara simpul 3 dan 4 akan menyebabkan sirkuit pada pohon merentang.



**Gambar 3.9** Langkah 6

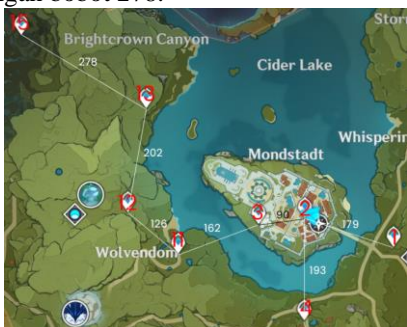


Langkah ketujuh masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul 2, 3, 11, 12, 1, 4 atau 13 namun tidak membuat sirkuit di T adalah sisi antara simpul 4 dan 9 dengan bobot 142.



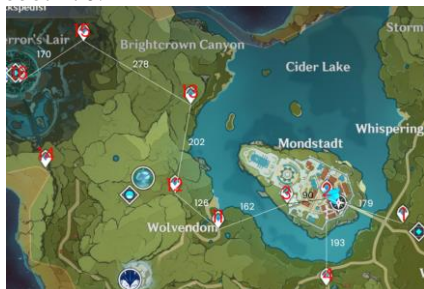
**Gambar 3.10** Langkah 7

Langkah kedelapan masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul 2, 3, 11, 12, 1, 4, 13, atau 9 namun tidak membuat sirkuit di T adalah sisi antara simpul 13 dan 15 dengan bobot 278.



**Gambar 3.11** Langkah 8

Oleh karena pengulangan belum mencapai  $n-2$  dua kali, yakni  $21-2 = 19$ , maka langkah kesembilan masih sama seperti langkah-langkah sebelumnya. Sisi yang dicari adalah sisi yang memiliki bobot minimum dan bersisian dengan simpul 2, 3, 11, 12, 1, 4, 13, 9, atau 15 namun tidak membentuk sirkuit di pohon merentang. Sisi yang dimaksud adalah sisi antara simpul 15 dan 16 yang berbobot 170.



**Gambar 3.12** Langkah 9

Langkah selanjutnya, yakni langkah kesepuluh masih mengulang langkah sebelumnya. Pada langkah ini, dicari sisi berbobot minimum yang bersisian dengan simpul-simpul di pohon merentang namun tidak membentuk sirkuit. Sisi yang ditemukan adalah sisi yang menghubungkan simpul 16 dan simpul 17 dengan bobot 162.



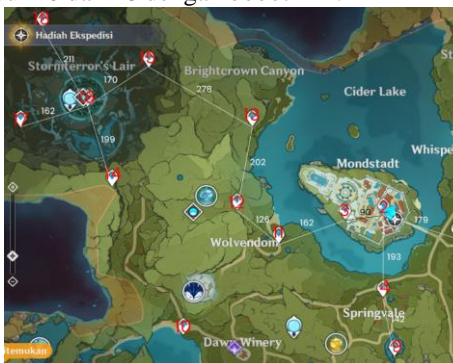
**Gambar 3.13** Langkah 10

Langkah kesebelas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 16 dan 14 dengan bobot 199.



**Gambar 3.14** Langkah 11

Langkah kedua belas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 16 dan 18 dengan bobot 211.



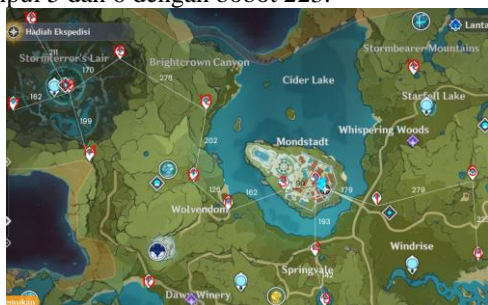
**Gambar 3.15** Langkah 12

Langkah ketiga belas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 1 dan 5 dengan bobot 279.



**Gambar 3.16** Langkah 13

Langkah keempat belas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 5 dan 6 dengan bobot 223.



**Gambar 3.17** Langkah 14

Langkah kelima belas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 5 dan 21 dengan bobot 276.



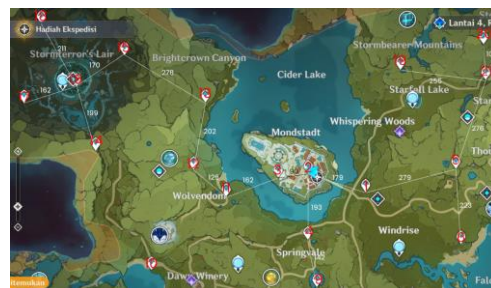
**Gambar 3.18** Langkah 15

Langkah keenam belas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 21 dan 20 dengan bobot 107.



**Gambar 3.19** Langkah 16

Langkah ketujuh belas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 21 dan 19 dengan bobot 255.



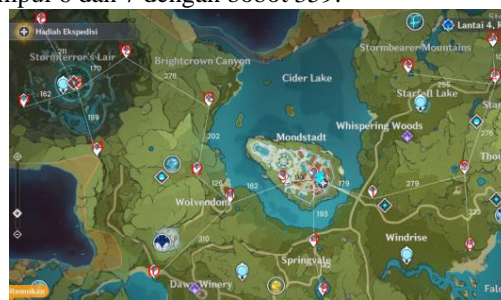
**Gambar 3.20** Langkah 17

Langkah kedelapan belas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 11 dan 10 dengan bobot 310.



**Gambar 3.21** Langkah 18

Langkah kesembilan belas masih sama seperti langkah sebelumnya. Dari gambar 3.3 dapat dilihat bahwa sisi yang berbobot minimum dan bersisian dengan simpul-simpul di pohon merentang namun tidak membuat sirkuit di T adalah sisi antara simpul 6 dan 7 dengan bobot 359.



**Gambar 3.22** Langkah 19

Langkah selanjutnya adalah langkah terakhir karena pengulangan telah dilakukan sebanyak 19 kali. Langkah ini adalah memasukkan sisi yang menghubungkan simpul 7 dan 8. Sisi ini adalah sisi yang memiliki bobot paling kecil dan tidak membentuk sirkuit pada pohon merentang. Berikut ini adalah hasil pohon merentang minimum yang didapat dari graf dengan menggunakan algoritma prim. Pohon merentang ini adalah rute terpendek untuk mencapai seluruh teleport waypoint di wilayah Mondstadt.





Gambar 3.23 Rute Terpendek

#### IV. KESIMPULAN

Konsep pohon merentang minimum dapat diterapkan pada peta permainan Genshin Impact. Dengan menerapkan konsep ini kita dapat menentukan rute terpendek untuk mencapai seluruh teleport waypoint. Rute terpendek untuk membuka seluruh teleport waypoint di Negara Mondstadt dapat dilihat pada gambar 3.23.

#### V. PENUTUP

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas berkat, rahmat dan karunia-Nya lah makalah ini dapat diselesaikan dengan baik dan tepat waktu. Penulis menyampaikan terima kasih kepada Fariska Zakhrativa Ruskanda, S.T., M.T. selaku dosen pengampu mata kuliah IF2120 yang telah memberikan ilmunya kepada penulis sehingga penulis dapat menulis makalah ini. Penulis juga mengucapkan terima kasih kepada Dr. Ir. Rinaldi Munir, MT. atas website yang berisi sumber materi. Tidak lupa juga penulis mengucapkan terima kasih kepada keluarga penulis yang senantiasa mendukung penulis dalam menjalani perkuliahan.

#### REFERENCES

- [1] Rinaldi Munir. 2020. Graf (Bag.1). Diakses pada 10 Desember 2020 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf> pukul 06.00
- [2] Rinaldi Munir. 2020. Pohon (Bag.1). Diakses pada 10 Desember 2020 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Pohon-2020-Bag1.pdf> pukul 09.16

#### 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, 3 Desember 2020

Aria Bachrul Ulum Berlian 13519115