

Penerapan Algoritma Shortest Path pada Perhitungan Rute Optimal di Map Zelda : Breath of The Wild

Fahd Muhammad Zahid - 13524078

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: blacklythning12@gmail.com , 13524078@std.stei.itb.ac.id

Abstract—*The Legend of Zelda: Breath of the Wild* merupakan game dunia terbuka dengan peta sangat luas dan banyak lokasi penting seperti shrine, tower, dan village. Pemain kerap kebingungan menjelajahi peta secara efisien. Dalam makalah ini, peta game dimodelkan sebagai graf berbobot, di mana simpul mewakili lokasi dan sisi mewakili jarak antar lokasi. Dengan menerapkan algoritma Minimum Spanning Tree (MST), khususnya algoritma Kruskal, diperoleh rute optimal untuk mengunjungi seluruh lokasi dengan total jarak minimum. Hasil pemodelan menunjukkan bahwa struktur MST memiliki total bobot 71,96 km dan membentuk koneksi efisien tanpa siklus. Meskipun terdapat asumsi seperti jarak Euclidean dan medan datar, pendekatan ini memberikan dasar konseptual dalam menyusun strategi eksplorasi yang sistematis di dalam permainan.

Keywords—Minimum Spanning Tree, algoritma Kruskal, teori graf, *The Legend of Zelda: Breath of the Wild*, rute optimal, jarak Euclidean

I. PENDAHULUAN

Game *The Legend of Zelda: Breath Of The Wild* merupakan game open-world yang sangat luas dengan banyak struktur bangunan yang dapat dikunjungi, seperti shrine, stable, tower, dan kota-kota yang tersebar di seluruh peta Hyrule. Saat memulai game ini untuk pertama kali, pemain sering kali merasa kebingungan dan kelelahan karena untuk berjelajah dari satu titik ke titik lain membutuhkan waktu yang relatif lama. Oleh karena itu, penulis merumuskan suatu permasalahan, yaitu bagaimana cara merencanakan urutan perjalanan dengan total jarak tempuh minimum untuk menjelajahi seluruh lokasi tersebut secara efisien



Gambar 1. Tangkapan layar game *The Legend of Zelda : Breath Of the Wild*.

(<https://wccftech.com/zelda-breath-wild-full-hyrule-map-revealed-new-screenshots-released/> diakses pada tanggal 19 Juni 2025)

Permasalahan ini dapat dimodelkan sebagai graf berbobot, dimana setiap lokasi penting di wakili oleh node dan setiap jalur antar lokasi diwakili oleh edge dengan bobot berupa jarak antar lokasi. Dalam konteks ini, pendekatan yang relevan adalah *Minimum Spanning Tree* (MST), yaitu pohon terbentang minimum yang menghubungkan semua node dengan total bobot minimum.

Makalah ini menerapkan algoritma kruskal. Melalui representasi graf peta *Zelda BOTW*, penelitian ini bertujuan mengevaluasi penerapan algoritma MST dalam menghasilkan rute kunjungan efisien yang dapat membantu pemain menjelajahi seluruh area dengan jarak minimum.

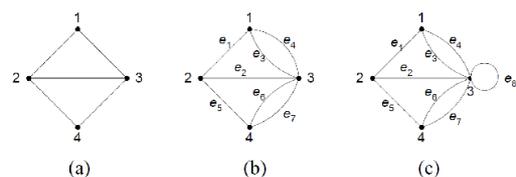
II. TEORI PENDUKUNG

A. Graf

Secara sederhana, graf dapat dijelaskan dideskripsikan sebagai titik-titik yang disambungkan oleh beberapa garis. Titik dalam sebuah graf dapat disebut juga sebagai nodes (*vertices*) dan garis dalam sebuah graf dapat disebut juga sebagai edges.

Graf biasanya dipakai untuk merepresentasikan suatu hubungan antar objek. Di dalam makalah ini, graf digunakan untuk merepresentasikan hubungan jarak antar lokasi dimana node merepresentasikan lokasi struktur dan edge merepresentasikan jarak antar node.

Graf dapat dibagi menjadi tiga [5], yaitu :



Gambar 2. (a) Contoh Graf Sederhana, (b) Contoh Graf Ganda, (c) Contoh Graf Semu.

(https://www.researchgate.net/figure/a-simple-graph-b-multigraph-and-c-pseudograph-1_fig1_369174013 diakses pada tanggal 19 Juni 2025).

- Graf Sederhana (*simple graph*)

Graf Sederhana adalah graf yang tidak memiliki sisi ganda maupun gelang antara pasangan simpul yang sama. Graf sederhana G terdiri dari V yang tidak kosong, serta himpunan sisi E yang merupakan pasangan dari elemen-elemen V . Graf sederhana dapat dinotasikan sebagai berikut :

$$G = (V, E)$$

- Graf Ganda (*multiple graph*)

Graf ganda adalah graf yang mengandung sisi ganda, tapi tidak memiliki gelang. Sisi ganda (*multiple edge*) adalah kondisi dimana dua *edge* atau lebih terhubung kepada dua *node* tertentu.

- Graf Semu (*Pseudograph*)

Graf semu adalah graf yang dapat mengandung sisi ganda maupun sisi gelang. Gelang adalah *edge* yang terhubung kepada *node* yang sama.

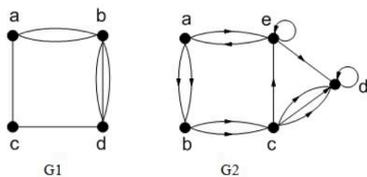
Di samping itu, dapat dibagi dua lagi berdasarkan orientasi arah pada *edge*, graf dapat dibagi menjadi dua jenis [5], yaitu :

- Graf tak Berarah

Graf tak berarah adalah graf yang tidak memiliki orientasi arah.

- Graf Berarah

Graf berarah adalah graf yang memiliki orientasi arah.



Gambar 3. (G1) Contoh Graf tak Berarah, (G2) Contoh Graf Berarah

(Sumber : [5], diakses pada tanggal 19 Juni 2025)

Selain itu, terdapat beberapa terminologi di dalam graf yang harus diketahui [5], yaitu :

- Ketetanggaan (*Adjacent*)

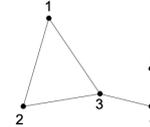
Jika terdapat dua simpul, dua simpul tersebut dikatakan bertetanggaan apabila keduanya terhubung langsung. Jika dilihat di **Gambar 3**, di graf $G1$, a dan c bertetanggaan karena terhubung langsung.

- Bersisian (*Incidency*)

Untuk sembarang sisi, dimana $e = (v_j, v_k)$ dikatakan bersisian apabila e bersisian dengan simpul v_j , atau e bersisian dengan simpul v_k [5]. Sebagai contoh, dapat dilihat di **Gambar 3**, di graf $G1$, misalkan sisi yang menghubungkan a dengan c adalah $e1$. Maka, $e1$ bersisian dengan *node* a dan $e1$ bersisian dengan *node* b.

- Simpul Terpencil (*Isolated Vertex*)

Simpul terpencil adalah simpul tidak bersisian dengan sisi manapun. Contohnya sebagai berikut :



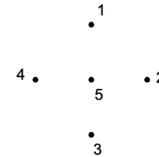
Gambar 4. Contoh graf dengan simpul terpencil.

(Sumber : [5], diakses pada tanggal 19 Juni 2025).

Dapat dilihat dari contoh di **Gambar 4** bahwa terdapat *edge* yang tidak bersisian dengan sisi manapun (*edge* ke-5).

- Graf Kosong (*null graph*)

Graf kosong adalah graf dengan simpul yang terpencil semua.



Gambar 5. Contoh Graf kosong.

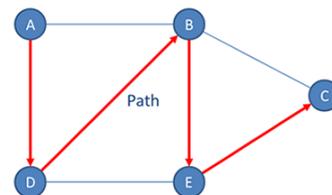
(Sumber : [5], diakses pada tanggal 19 Juni 2025).

- Derajat (*Degree*)

Derajat adalah jumlah *edge* yang bersisian dengan *node* tersebut. Jika dilihat dari **Gambar 3**, di graf $G1$, *node* a memiliki 3 sisi yang bersisian dengannya. Maka, dapat disimpulkan bahwa derajat yang dimiliki oleh *node* a adalah 3.

- Lintasan

Lintasan adalah rangkaian *edge* yang dimulai dari suatu *node* dan dilanjutkan ke *node*-*node* yang lainnya melalui *edge*-nya hingga sampai ke *node* yang ingin dituju.

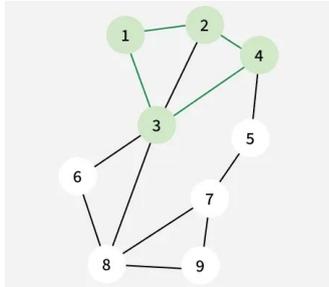


Gambar 6. Contoh Lintasan

(<http://www.computersciencebytes.com/array-variables/graphs/> diakses pada tanggal 19 Juni 2025)

- Sirkuit

Sirkuit adalah suatu lintasan yang berawal dan berakhir pada *node* yang sama.



Gambar 7. Contoh Sirkuit

(<https://www.geeksforgeeks.org/walks-trails-paths-cycles-and-circuits-in-graph/> diakses pada tanggal 19 Juni 2025)

- Keterhubungan

Suatu keterhubungan dari dua simpul atau lebih dapat dibuktikan jika terdapat sebuah lintasan yang menghubungkan kedua simpul tersebut.

- Upagraf

Suatu graf $G' = (V', E')$ dapat dikatakan sebagai upagraf dari graf $G = (V, E)$, jika memenuhi dua syarat berikut :

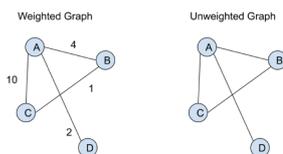
- (1) $V' \subseteq V$, yaitu himpunan simpul G' merupakan bagian dari himpunan simpul pada G ; dan
- (2) $E' \subseteq E$, yaitu himpunan edge G' merupakan bagian dari himpunan edge G , di mana setiap sisi dalam E' hanya menghubungkan *node-node* yang juga berada dalam V' .

B. Graf Berbobot

Menurut [1], graf berbobot termasuk graf sederhana, bedanya tiap edge diasosiasikan oleh suatu bilangan riil (yang merupakan bobotnya) di dalam graf. Secara matematis, fungsi dari bobot dapat dinotasikan sebagai berikut :

$$w : E \rightarrow \mathbb{R}$$

Di dalam makalah ini, graf berbobot dipakai untuk merepresentasikan jarak antar lokasi di dalam peta game *The Legend of Zelda: Breath Of The Wild*.



Gambar 8. Contoh Graf berbobot (kiri), Contoh Graf tak berbobot (kanan)

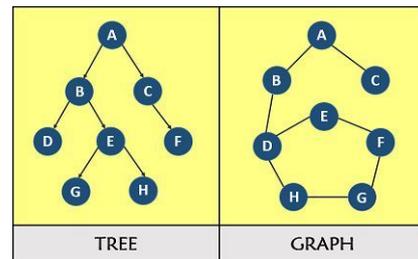
(Sumber : [5])

C. Graf Padat dan Graf Jarang

Graf padat dan graf jarang adalah klasifikasi graf berdasarkan kepadatan edge-nya. Pada graf tak berarah dengan n simpul, jumlah maksimum edge adalah $\frac{n(n-1)}{2}$. Graf disebut pada jika edge-nya mendekati jumlah maksimum ini, sementara graf jarang memiliki edge jauh lebih sedikit. Kepadatan memengaruhi efisiensi representasi graf dan pemilihan algoritma.

D. Pohon (Tree)

Pohon adalah graf terhubung yang tidak mengandung sirkuit [2].



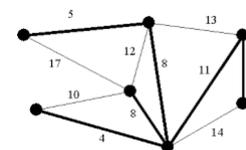
Gambar 9. Contoh Pohon (kiri), Contoh Graf (kanan).

(<https://techdifferences.com/difference-between-tree-and-graph.html> diakses pada tanggal 20 Juni 2025).

E. Pohon Berentang Minimum (Minimum Spanning Tree)

Secara singkat, *Minimum Spanning Tree* (MST) pada suatu graf berbobot merupakan spanning tree yang memiliki total penjumlahan *edge* terkecil di antara semua kemungkinan spanning tree yang dapat dibentuk dari graf tersebut. MST harus mencakup seluruh node dalam graf, namun tidak boleh membentuk siklus.

Dalam makalah ini, MST digunakan untuk menentukan jalur dengan total jarak minimum yang dapat digunakan untuk menjelajahi seluruh lokasi pada peta *The Legend of Zelda: Breath Of The Wild*. Dengan menggunakan MST, diperoleh rute optimal yang mencakup semua lokasi penting.



Gambar 10. Contoh Minimum Spanning Tree

(<https://people.eecs.berkeley.edu/~jfc/cs174lecs/lec19/lec19.html> diakses pada tanggal 20 Juni 2025)

F. Algoritma Prim

Algoritma Prim adalah salah satu algoritma implementasi dari MST. Algoritma ini dimulai dengan memilih node bebas yang ada di graf, kemudian memilih edge dengan bobot terkecil. Proses ini diulangi secara terus-menerus hingga seluruh simpul dalam graf telah terhubung tanpa membentuk siklus.

Berdasarkan [2], pseudocode algoritma prim dapat

```
Prim's Algorithm
procedure Prim( $G$ : weighted connected undirected graph with  $n$  vertices)
   $T :=$  a minimum-weight edge
  for  $i := 1$  to  $n - 2$  do
     $e :=$  an edge of minimum weight incident to a vertex in  $T$  and not forming a
    simple circuit in  $T$  if added to  $T$ 
     $T := T$  with  $e$  added
  return  $T$  {  $T$  is a minimum spanning tree of  $G$  }
```

direpresentasikan sebagai berikut :

G. Algoritma Kruskal

Algoritma *kruskal* juga merupakan salah satu algoritma implementasi dari MST. Algoritma ini dimulai dengan mengurutkan seluruh *edge* dalam graf secara *ascending* (menaik). Kemudian, secara bertahap, algoritma memilih sisi dengan bobot terkecil dan menambahkannya ke MST selama penambahan sisi tersebut tidak membentuk siklus. Proses ini dilanjutkan hingga jumlah sisi dalam MST mencapai $n - 1$, dimana n merupakan jumlah node dalam graf.

Berdasarkan [2], pseudocode algoritma *kruskal* dapat direpresentasikan sebagai berikut :

```
Kruskal's Algorithm
 $T :=$  empty graph
for  $i := 1$  to  $n - 1$  do
   $e :=$  any edge in  $G$  with smallest weight that does not form a simple circuit
  when added to  $T$ 
   $T := T$  with  $e$  added
return  $T$  {  $T$  is a minimum spanning tree of  $G$  }
```

H. Struktur Bangunan Di Legend of Zelda : Breath Of the Wild

• Shrine

Berdasarkan [6], Shrine adalah struktur bangunan yang terdapat dalam game *The Legend of Zelda Breath Of the Wild*. Shrine dihuni oleh *Sheikah Monk* (biksu sheikah) dan dikenal sebagai kuil ujian. Shrine ini tersebar di seluruh wilayah Hyrule dan memerlukan *Sheikah Slate* milik Link untuk dapat memasukinya. Secara keseluruhan, terdapat 120 *shrine* di seluruh peta *The Legend of Zelda : Breath Of the Wild*.



Gambar 11. Tangkapan layar dari struktur bangunan shrine. (<https://www.zeldadungeon.net/breath-of-the-wild-walkthrough/shrine-locations/> diakses pada tanggal 20 Juni 2025)

• Stable

Berdasarkan [7], Stable merupakan tempat dimana pemain dapat mendaftarkan kuda, menyimpan kuda yang sudah terdaftar, atau mengambil kembali kuda yang telah disimpan untuk digunakan. Secara keseluruhan, terdapat lima belas stable di *The Legend of Zelda : Breath Of the Wild*.



Gambar 12. Tangkapan layar dari struktur bangunan stable.

(<https://zelda.fandom.com/wiki/Stable> diakses pada tanggal 20 Juni 2025).

• Village

Village merupakan desa yang menyediakan tempat tinggal NPC, side-quests, dan toko. Selain itu, pemain dapat berinteraksi dan membeli barang-barang yang tersedia di dalam toko-toko yang terdapat di village.



Gambar 13. Tangkapan layar dari salah satu village di game *The Legend of Zelda : Breath Of the Wild*, Hateno Village.

(https://zelda-archive.fandom.com/wiki/Hateno_Village diakses pada tanggal 20 Juni 2025)

• Sheikah Tower

Berdasarkan [8], *Sheikah Tower* adalah struktur kuno yang tinggi dan tersebar di seluruh game *The Legend of Zelda : Breath of the Wild*. Menara-menara ini dibangun untuk mendeteksi tanda-tanda kebangkitan *Calamity Ganon* dan akan muncul dari tanah ketika dialiri *Ancient Energy*. Saat diaktifkan, menara ini akan bersinar terang dengan cahaya biru, dan ujung-ujungnya yang bercabang akan terbuka seperti antena satelit.



Gambar 14. Tangkapan layar dari struktur bangunan Sheikah Tower.

(https://zelda.fandom.com/wiki/Sheikah_Tower diakses pada tanggal 20 Juni 2025)

Gambar 15. Peta *The Legend of Zelda : Breath Of the Wild* (Sumber : [3]).

B. Asumsi Pemodelan

Untuk menyederhanakan kompleksitas lingkungan dalam game ke dalam bentuk graf matematis, beberapa asumsi berikut digunakan :

- **Permukaan diasumsikan datar.** Perbedaan elevasi seperti pegunungan, jurang, atau bangunan bertingkat diabaikan.
- **Jarak antar titik dihitung sebagai jarak lurus (Euclidean distance)**
- **Struktur yang berada pada lokasi yang sama dianggap satu simpul**
- **Graf diasumsikan tidak berarah dan terhubung sempurna**
- **Edge antar simpul yang mewakili jarak lebih dari tiga kotak (> 1575 meter) dihilangkan**

C. Pengumpulan dan Pemrosesan Data

Dengan menggunakan [4], struktur bangunan dalam game *The Legend of Zelda : Breath Of the Wild* dapat terlihat dengan memilih struktur apa saja yang ingin dilihat. Dalam makalah ini, struktur bangunan yang akan direpresentasikan sebagai node adalah *shrine*, *sheikah tower*, *village*, dan *stables*. *Edge* dalam graf akan merepresentasikan jarak (bobot) antar lokasi struktur yang ada di dalam model peta. Jarak didapatkan dengan mengukur panjang menggunakan fitur penggaris di aplikasi *GoodNotes*. Dalam pengukuran tersebut, satu kotak di grid memiliki ukuran 0.6 x 0.6 satuan di *GoodNotes* yang dikonversikan ke ukuran sebenarnya yaitu 525 meter x 525 meter. Oleh karena itu, rumus konversi dari satuan penggaris ke satuan meter adalah sebagai berikut :

$$\text{Jarak (meter)} = \left(\frac{\text{Jarak pada GoodNotes}}{0.6} \right) \times 525$$

D. Pembentukan Graf Berbobot

Dalam makalah ini, setiap struktur lokasi direpresentasikan dengan warna yang berbeda. Untuk *shrine* direpresentasikan dengan warna hijau, *sheikah tower* direpresentasikan dengan warna biru, *village* direpresentasikan dengan warna merah, *stables* direpresentasikan dengan warna coklat, dan untuk beberapa struktur yang ada di titik yang sama berwarna hitam. Ini dilakukan agar pemodelan peta dapat dibuat lebih mudah. Selain itu, hubungan antar *node* berupa *edge* dengan bobot yang dihitung menggunakan rumus konversi sebelumnya. Graf yang dibentuk tidak berarah dan berbobot, serta diasumsikan terhubung. Untuk menjaga realisme, hanya *edge* dengan bobot ≤ 1575 meter yang dimasukkan.

E. Penerapan Algoritma MST

Algoritma *Minimum Spanning Tree* (MST) diterapkan untuk mencari rute optimal dengan total jarak minimum yang menghubungkan seluruh lokasi penting. Algoritma yang digunakan adalah :

III.

METODOLOGI

A. Representasi Peta

Berdasarkan [3], peta *The Legend of Zelda : Breath Of the Wild* memiliki ukuran sebesar 10 km x 8 km. Akan tetapi, itu hanya area yang dapat kita akses sebagai pemain, ukuran aslinya itu sekitar 16 km x 16 km.

Di dalam makalah ini, peta direpresentasikan dalam bentuk grid berukuran 28 x 25 kotak. Setiap kotak diasumsikan memiliki ukuran 525 meter x 525 meter, berdasarkan konversi dari ukuran kotak pada peta digital yang diukur menggunakan fitur penggaris di aplikasi *GoodNotes*. Dalam pengukuran tersebut, satu kotak pada *grid* itu setara dengan 0.6 x 0.6 satuan penggaris dan dikonversikan ke ukuran dunia nyata game sebesar 525 meter. Oleh karena itu, total ukuran peta menjadi 14.7 km x 13.125 km dengan total luas sekitar 192.56 km².

Pemilihan skala ini didasarkan pada fakta bahwa beberapa komunitas memperkirakan ukuran peta penuh game, termasuk terrain yang tidak langsung dapat dijelajahi berkisar antara 170 hingga 200 km². Oleh karena itu, pemodelan ini dianggap cukup representatif dan realistis untuk mendekati skala dunia permainan dalam simulasi graf.



- Algoritma Kruskal, yang mengurutkan sisi dari bobot terkecil hingga terbesar dan menambahkannya ke MST selama tidak membentuk siklus.

- Algoritma prim, yang dimulai dengan memilih node bebas, kemudian memilih edge dengan bobot terkecil hingga semua node tersambung semua dan tidak membentuk siklus.

Pertama, semua *edge* akan diberi bobot sesuai dengan hasil konversi berdasarkan rumus konversi sebelumnya. Setelah itu, *edge* akan diurutkan dari yang paling kecil hingga paling besar. Selanjutnya, dengan bantuan program di *python*, MST akan dibuat. Terakhir, MST akan digambar ulang melalui aplikasi *GoodNotes*.

F. Evaluasi dan Visualisasi

Hasil dari algoritma MST dianalisis berdasarkan :

- Total Bobot
- Struktur jaringan yang dihasilkan
- Perbandingan hasil Algoritma Kruskal

IV. HASIL DAN ANALISIS

A. Visualisasi Graf

Gambar di bawah ini menunjukkan hasil representasi graf berbobot dari peta *The Legend of Zelda : Breath Of the Wild* yang telah dibangun berdasarkan lokasi-lokasi struktur penting dalam permainan. Setiap *node* pada graf merepresentasikan suatu titik lokasi seperti *shrine*, *sheikah tower*, atau *village*, sedangkan *edge* menghubungkan dua lokasi dengan bobot berupa jarak antar lokasi dengan bobot berupa jarak lokasi dalam satuan meter.

Dalam graf tersebut :

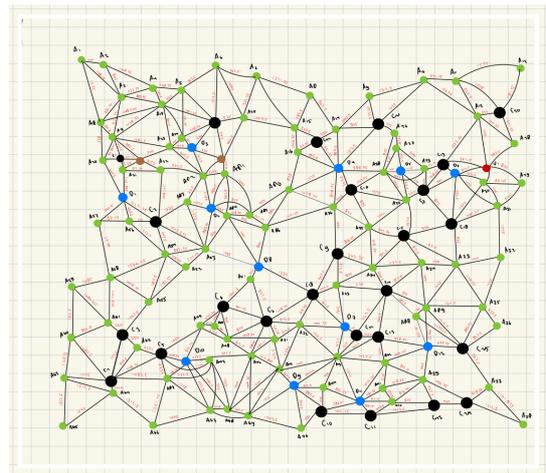
- Hijau (A) merepresentasikan *shrine*
- Coklat (E) merepresentasikan *stable*
- Merah (V) merepresentasikan *village*
- Biru (D) merepresentasikan *sheikah tower*
- Hitam (C) merepresentasikan node yang mewakili beberapa struktur sekaligus (campuran)



Gambar 10. Visualisasi graf pada peta *The Legend of Zelda: Breath of the Wild*. Setiap simpul merepresentasikan lokasi penting seperti *shrine* (hijau), *stable* (coklat), *Sheikah Tower* (biru), dan *village* (merah). Simpul hitam digunakan untuk mewakili gabungan beberapa struktur. Sisi pada graf menunjukkan hubungan antar lokasi yang memiliki jarak maksimum 1575 meter. Graf ini menjadi dasar dalam penerapan algoritma *Minimum Spanning Tree (MST)* untuk perhitungan rute optimal.

(<https://zeldamaps.com/?game=BotW> diakses pada tanggal 20 Juni 2025).

Dengan menggunakan rumus konversi sebelumnya, bobot tiap *edge* dapat direpresentasikan sebagai berikut :



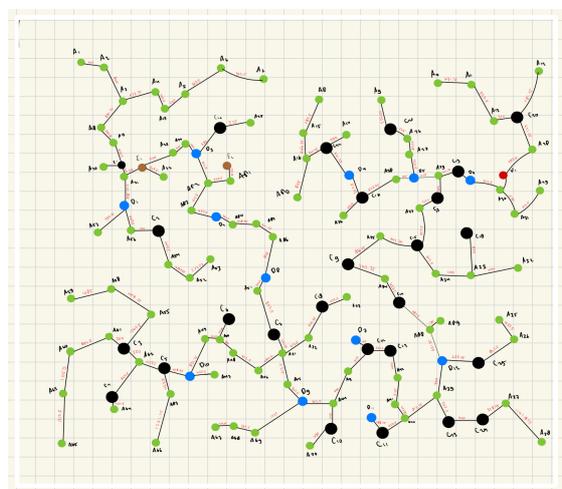
Gambar 11. Graf hasil pemberian bobot menggunakan rumus konversi yang ada di bab 3C.

Graf terdiri dari 129 node dan 318 edge.

B. Hasil MST

- Hasil Algoritma Prim dan Kruskal

Setelah semua *edge* berbobot sudah diurutkan didapatkan hasil MST melalui program *python* sebagai berikut :



Gambar 12. Graf hasil penerapan algoritma Kruskal untuk membentuk Minimum Spanning Tree (MST). Sisi-sisi dipilih berdasarkan bobot terkecil secara global dan membentuk struktur koneksi optimal antar lokasi tanpa siklus. Bobot tiap sisi ditampilkan dalam satuan meter.

Setelah data sudah diproses dan dimasukkan ke dalam kode *python*, terlihat bahwa hasil dari algoritma prim maupun kruskal sama saja dan total bobot yang dihasilkan adalah 71964.81 meter, atau kalau dikonversikan menjadi 71.96 km.

C. Interpretasi Hasil

Dengan total bobot sepanjang 71.96 km, hasil MST ini memberikan estimasi jarak minimum yang diperlukan untuk menjangkau seluruh lokasi penting dalam *The Legend of Zelda : Breath Of the Wild* secara efisien, tanpa perlu mengunjungi titik yang sama lebih dari sekali. Struktur graf yang terbentuk melalui algoritma kruskal maupun prim menunjukkan pola koneksi yang menyebar secara merata dan efisien dari titik pusat ke seluruh penjuru wilayah. Beberapa simpul seperti C5, C15, C3, dan lain lain menunjukkan memiliki derajat node yang lebih tinggi dibandingkan node yang lain. Hal ini menunjukkan juga bahwa desain dunia game yang mengandalkan beberapa lokasi kunci sebagai titik transisi antarwilayah. Pola jalur yang dipilih dalam MST menghindari siklus dan menunjukkan bagaimana algoritma kruskal secara sistematis memilih sisi dengan bobot terkecil untuk membentuk struktur koneksi yang optimal. Meskipun hasil ini memberikan gambaran rute eksplorasi paling efisien secara matematis, perlu dicatat bahwa pendekatan ini hanya mempertimbangkan jarak lurus antar titik. Dengan demikian, hasil MST ini bersifat konseptual, namun tetap dapat digunakan sebagai dasar dalam menyusun strategi eksplorasi yang hemat waktu dan sumber daya di dalam permainan.

D. Analisis Kelebihan dan Kekurangan

Dalam pencarian Minimum Spanning Tree (MST), baik algoritma Kruskal maupun Prim merupakan metode yang populer dan banyak digunakan. Algoritma Prim bekerja dengan memulai dari satu simpul dan secara bertahap menambahkan simpul terdekat yang belum termasuk ke dalam MST sehingga cocok digunakan pada graf padat di mana banyak sisi menghubungkan simpul-simpul. Sementara itu, algoritma Kruskal bekerja dengan memilih sisi-sisi berbobot terkecil dari seluruh graf, kemudian menyusunnya secara bertahap selama tidak membentuk siklus, hingga seluruh simpul terhubung. Pendekatan ini menjadikan Kruskal lebih efisien untuk graf jarang karena tidak perlu mengakses semua tetangga suatu simpul.

Kelebihan algoritma Kruskal terletak pada fleksibilitasnya yang tidak memerlukan simpul awal, serta kemampuannya menangani graf yang tidak sepenuhnya terhubung dalam satu komponen. Selain itu, jika data graf disimpan sebagai daftar sisi (*edge list*), maka implementasi Kruskal menjadi sangat sederhana dan efisien. Di sisi lain, algoritma Prim unggul dalam graf padat karena dapat mengakses simpul-simpul

terdekat dengan cepat menggunakan struktur *heap* atau *adjacency matrix*. Namun, kekurangannya adalah ketergantungannya pada simpul awal dan kecenderungan menjadi kurang efisien pada graf yang sangat besar namun *sparse*. Namun, keduanya memiliki keterbatasan dalam konteks penerapan pada game *The Legend of Zelda : Breath Of the Wild*. Salah satu kelemahan utama adalah asumsi bahwa bobot sisi merepresentasikan jarak *Euclidean* (garis lurus) antar titik, padahal dalam kondisi nyata di game, jalur sebenarnya bisa jauh berbeda karena adanya gunung, sungai, tebing, atau area berbahaya yang tidak bisa dilalui secara langsung. Asumsi ini menyebabkan hasil MST menjadi pendekatan matematis yang belum tentu sesuai dengan rute aktual yang ditempuh oleh pemain.

Meskipun begitu, eksperimen yang dilakukan menunjukkan bahwa algoritma Kruskal dan Prim menghasilkan struktur MST yang identik dengan total bobot sebesar 71.96 km, membuktikan bahwa kedua algoritma akan menghasilkan solusi yang sama selama graf terhubung dan bobot sisi tidak ambigu. Namun dalam *gameplay* asli, terdapat fitur teleportasi yang membuat pergerakan pemain jauh lebih efisien dibandingkan sekadar mengikuti jalur terpendek. Setelah pemain membuka shrine atau *sheikah tower*, lokasi tersebut dapat dijadikan titik teleportasi, memungkinkan pemain berpindah antar titik secara instan tanpa harus menempuh jalur darat. Fitur ini secara signifikan mengurangi kebutuhan untuk mengikuti jalur terhubung secara fisik dan menyebabkan struktur MST menjadi tidak relevan dalam situasi tertentu. Selain itu, ada fitur *gliding* yang membuat pemain dapat melayang di udara menggunakan alat dari game tersebut, yang membuat pemain dapat melakukan perjalanan cepat. Oleh karena itu, meskipun MST memberi gambaran rute optimal secara teoritis, ia tetap memiliki keterbatasan signifikan jika dibandingkan dengan mekanisme mobilitas yang sebenarnya tersedia di dalam permainan.

V.

KESIMPULAN

Berdasarkan hasil implementasi dan analisis yang telah dilakukan, dapat disimpulkan bahwa algoritma Kruskal merupakan metode yang efektif untuk membentuk Minimum Spanning Tree (MST) dari graf berbobot yang merepresentasikan hubungan antar lokasi dalam peta *The Legend of Zelda: Breath of the Wild*. Hasil MST menunjukkan struktur koneksi optimal yang menghubungkan seluruh lokasi penting dengan total bobot minimum sebesar 71.96 km, tanpa membentuk siklus.

Dibandingkan dengan algoritma Prim, algoritma Kruskal menunjukkan hasil akhir yang identik dalam kasus ini, namun dipilih karena lebih sesuai untuk graf jarang seperti yang digunakan dalam pemodelan peta game. Meski begitu,

terdapat beberapa keterbatasan dalam pendekatan ini, seperti asumsi jarak antar lokasi yang bersifat linier (Euclidean) dan tidak mempertimbangkan kondisi medan, jalur aktual, maupun mekanisme teleportasi dalam game.

Meskipun demikian, penerapan MST memberikan pendekatan konseptual yang berguna dalam merancang rute eksplorasi yang efisien dan sistematis. Untuk pengembangan selanjutnya, pendekatan ini dapat diperluas dengan mempertimbangkan aspek-aspek tambahan dari dunia game agar hasil analisis lebih mendekati kondisi nyata.

VI. UCAPAN TERIMAKASIH

Dengan penuh rasa syukur kepada Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, penulis dapat menyelesaikan makalah ini dengan baik. Ucapan terima kasih disampaikan kepada Bapak Arrival Dwi Sentosa, S.Kom., M.T., M.Sc selaku dosen pengampu mata kuliah IF2120 Matematika Diskrit yang telah memberikan arahan dan bimbingan sepanjang perkuliahan. Penulis juga menyampaikan apresiasi kepada seluruh pihak dan sumber referensi yang telah membantu dalam penyusunan informasi pada makalah ini.

REFERENSI

- [1] E. Lehman, F. T. Leighton, and A. R. Meyer, *Mathematics for Computer Science*. Cambridge, MA: Massachusetts Institute of Technology, 2010. [Diakses 18 Juni 2025]
- [2] K. H. Rosen, *Discrete Mathematics and Its Applications*, 7th ed. New York: McGraw-Hill, 2012. [Diakses 18 Juni 2025]
- [3] T. Color, "The size of the Breath of the Wild Map," [tacolor.xyz](https://www.tacolor.xyz/The_size_of_the_Breath_of_the_Wild_Map.html). [Online]. Available: https://www.tacolor.xyz/The_size_of_the_Breath_of_the_Wild_Map.html. [Diakses 18 Juni 2025]
- [4] Zeldamaps.com, "Breath of the Wild Interactive Map," [Online]. Available: <https://zeldamaps.com/?game=BotW>. [Diakses: 18-Jun-2025].

- [5] Rinaldi Munir, *Graf Bagian 1, Mata Kuliah Matematika Diskrit, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2024–2025*. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf> [diakses 18 Juni 2025]
- [6] "Shrine (Breath of the Wild)," *Zelda Wiki, Fandom*, [Online]. Tersedia: [https://zelda-archive.fandom.com/wiki/Shrine_\(Breath_of_the_Wild\)](https://zelda-archive.fandom.com/wiki/Shrine_(Breath_of_the_Wild)). [Diakses: 20 Juni 2025].
- [7] "Stable," *Zelda Dungeon Wiki*, [Online]. Tersedia: <https://www.zeldadungeon.net/wiki/Stable>. [Diakses: 20 Juni 2025].
- [8] "Sheikah Tower," *Zelda Wiki, Fandom*, [Online]. Tersedia: https://zelda.fandom.com/wiki/Sheikah_Tower. [Diakses: 20 Juni 2025].

LAMPIRAN

Lampiran A. Repositori Kode Program
Kode lengkap yang digunakan dalam pembuatan makalah ini dapat diakses melalui repositori berikut :

<https://github.com/fahdmz/MakalahMatematikaDiskrit.git>

Berikut link video :

<https://youtu.be/PmaJREX40bQ?feature=shared>

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, 1 Juni 2025



Ttd

Fahd Muhammad Zahid / 13524078