Implementasi graf berbobot dan Algoritma A* untuk menentukan rute terpendek *transport* dari *Royal City* ke Caerleon dalam permainan Albion Online

Muhammad Aulia Azka - 13523137
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: auliaazka2506@gmail.com , 13523137@std.stei.itb.ac.id

***Abstract*—*Albion Online* adalah permainan daring multipemain masif yang cukup populer. Pada permainan ini terdapat kegiatan bernama *Transport*. *Transport* adalah kegiatan di mana pemain membawa barang dari kota utama (Lymhurst, Bridgewatch, Martlock, dan Fortsterling) ke kota lain di pusat peta bernama Caerleon. Makalah ini membahas bagaimana implementasi dari Algoritma A* dan Graf berbobot dalam menentukan rute tercepat dari kota utama ke Caerleon agar kegiatan *Transport* ini menjadi efektif dan cepat. Algoritma A* dipilih karena kemampuannya yang efisien dalam menemukan jalur terpendek dengan memanfaatkan fungsi heuristik. Dalam eksperimen yang dilakukan, struktur peta dalam permainan dimodelkan sebagai graf berbobot, dengan simpul sebagai pintu masuk atau zona koneksi, dan sisi sebagai jalur antar zona dengan bobot direpresentasikan sebagai jarak tempuh antar pintu.**

***Kata kunci*—A* graf *Transport* rute**

I. PENDAHULUAN

Albion Online adalah permainan daring multipemain masif yang cukup populer. Pada permainan ini terdapat kegiatan bernama *Transport*. *Transport* adalah kegiatan di mana pemain membawa barang dari *Royal City* atau kota utama (*Lymhurst*, *Bridgewatch*, *Martlock*, dan *Fortsterling*) ke kota lain di pusat peta bernama *Caerleon*. Kenapa kegiatan *Transport* ini sangat penting untuk dilakukan, karena kegiatan *Transport* ini bertujuan untuk menjual barang di *Black Market* yang hanya terdapat di kota Caerleon. Menjual barang di *Black Market* memiliki keuntungan yang lebih besar dibandingkan menjual barang di *Market* biasa di kota

Albion Online memiliki sistem peta dunia yang cukup kompleks, terdiri atas berbagai zona dan koneksi antar zona. Setiap zona memiliki jalur masuk atau keluar yang spesifik, yang menghubungkan antar zona. Dalam hal ini, penentuan rute dari *Royal City* menuju Caerleon menjadi permasalahan yang cukup menarik karena pemain harus menavigasi zona-zona ini dengan cepat agar perjalanan dari *Royal City* ke kota Caerleon menjadi perjalanan yang cukup efektif dan cepat. Masalah ini dapat dimodelkan sebagai persoalan pencarian jalur terpendek pada graf berbobot.

Salah satu algoritma yang sangat relevan untuk menyelesaikan masalah ini adalah **A***, yang menggabungkan pendekatan pencarian dari *Uniform Cost Search* dan *Greedy Best First Search* dengan memanfaatkan fungsi heuristik untuk memperkirakan jarak ke tujuan. Algoritma ini dikenal cukup efisien dan optimal dalam pencarian jalur terpendek pada graf, terutama jika fungsi heuristik yang digunakan cukup baik.

Dalam makalah ini, penulis mengimplementasikan algoritma A* dan konsep Graf untuk memodelkan dan menyelesaikan permasalahan pencarian rute tercepat dari *Royal City* ke Caerleon dalam permainan *Albion Online*.

II. DASAR TEORI

A. *Graf*

Graf adalah struktur data yang biasanya merepresentasikan objek – objek diskrit dan hubungan antar objek – objek tersebut. Graf terdiri atas dua komponen yaitu simpul dan sisi. Simpul adalah komponen yang merepresentasikan Objek. Sisi adalah komponen yang merepresentasikan hubungan antar objek tersebut. Secara formal, Graf didefinisikan sebagai G = (V, E) dalam hal ini:

V = himpunan tidak-kosong dari simpul – simpul

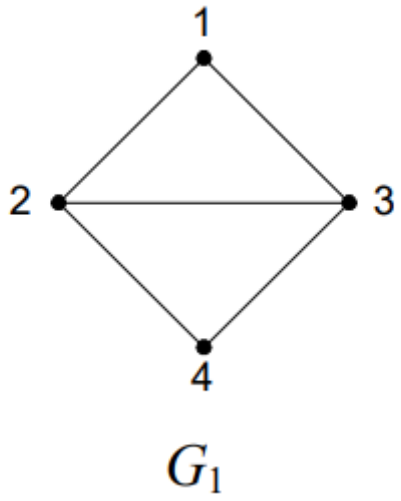
V = {v1, v2, v3, ..., vn}

Karena V tidak boleh himpunan kosong, maka dapat diartikan bahwa graf tidak boleh tidak mengandung simpul.

E = himpunan sisi yang menghubungkan antar simpul

E = {e1, e2, e3, ..., en}

Variabel E boleh kosong, artinya sebuah graf boleh tidak mengandung sisi sama sekali



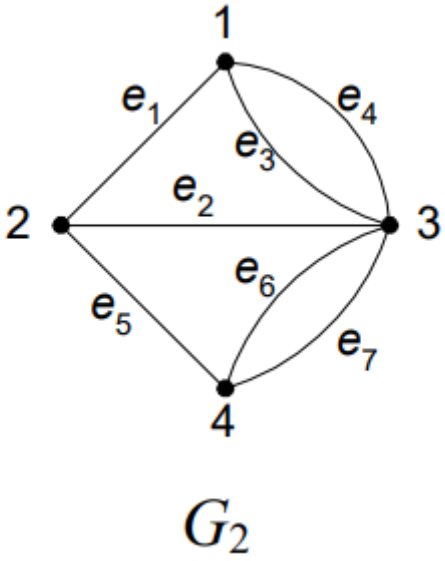
Gambar 1. Contoh Graf, diambil dari

(https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf)

Pada Gambar 1, G1 adalah graf dengan jumlah simpul 4 dan jumlah sisi 5.

V = {1, 2, 3, 4}

E = {(1, 3), (1, 2), (2, 3), (2, 4), (3, 4)}

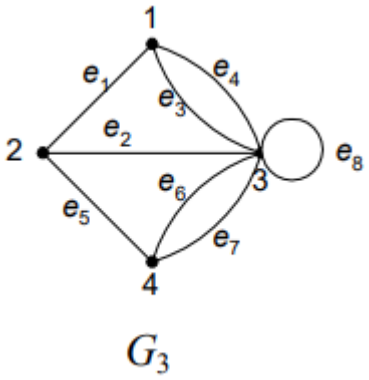


Gambar 2. Contoh Graf yang mempunyai sisi ganda, diambil dari

(https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf)

Pada Gambar 3, Graf G2 mempunyai sisi e3 = (1, 3) dan sisi e4 = (1, 3). Sisi – sisi ini dinamakan dengan sisi ganda karena kedua sisi ini menghubungi dua buah simpul yang sama, yaitu simpul 1 dan simpul 3.

Berdasarkan ada tidaknya sisi gelang atau sisi ganda di dalam graf, maka graf dibedakan menjadi dua jenis yaitu graf sederhana dan graf tak-sederhana. Graf sederhana adalah graf yang tidak mengandung sisi gelang maupun sisi ganda. Graf G1 pada gambar 1 adalah contoh graf sederhana. Graf tak-sederhana adalah graf yang mengandung sisi ganda atau sisi gelang. Contoh graf yang memiliki sisi ganda adalah Graf G2 pada gambar 2.

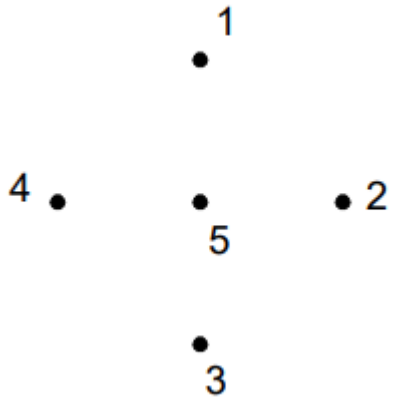


Gambar 3. Contoh Graf dengan sisi gelang, diambil dari (https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf)

Pada gambbar 3, Graf G3 3 adalah salah satu contoh graf yang memiliki sisi gelang. Pada graf G3 ini sisi gelangnya adalah sisi e8 = (3, 3). Dinyatakan sisi gelang karena ia berawal dan berakhir di simpul yang sama.

Berdasarkan orierantasi arah pada sisi, graf dibedakan atas dua jenis yaitu graf tak-berarah dan berarah. Graf tak-berarah adalah graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah. Graf berarah adalah graf yang setiap sisinya mempunyai orientasi arah.

Pada graf terdapat beberapa terminologi yaitu ketetanggaan (*Adjacent*), Bersisian (*Incidency*), Simpul Terpencil (*Isolated Vertex*), Graf kosong (*null graph*), dan Derajat (*Degree*). Ketetanggan adalah istilah di mana dua buah simpul dikatakan bertetangga jika kedua simpul tersebut terhubung langsung. Pada gambar 1, simpul 1 betetangga dengan simpul 2 dan 3, simpul 2 bertetangga dengan simpul 1 dan simpul tetapi simpul 4 tidak bertetangga dengan simpul 1.Bersisian adalah istilah dimana untuk sembarang sisi e = (vj, vk) dapat dikatakan dengan e bersisian dengan simpul vj dan e bersisian dengan simpul vk. Pada gambar 1, sisi (1, 3) bersisian dengan simpul 1 dan simpul 3, sisi (1, 2) bersisian dengan simpul 1 dan simpul 3, dan sisi (3, 4) bersisian dengan simpul 3 dan simpul 4. Simpul Terpencil adalah istilah dimana terdapat simpul yang tidak mempunyai sisi sama sekali. Graf kosong adalah istilah dimana suatu graf mempunyai himpunan sisinya merupakan himpunan kosong. Artinya graf tersebut tidak memiliki sisi. Derajat adalah istilah jumlah sisi yang berisisan dengan suatu simpul. Pada gambar 1, terdapat dua sisi yang bersisian dengan simpul 1, yaitu sisi (1, 2) dan sisi (1, 3). Sisi dilambangkan sebagai d(x), x adalah simpul yang ditinjau.



Gambar 4. Contoh Graf kosong, diambil dari (https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf)

Pada ilmu graf, terdapat satu konsep yang bernama Lemma Jabat Tangan. Konsep ini menyatakan bahwa jumlah derajat semua simpul pada suatu graf adalah genap, yang dapat diartikan juga adalah dua kali jumlah sisi pada graf tersebut. Jika suatu graf dinyatakan sebagai G = (V, E), V adalah himpunan simpul dari graf G dan E adalah himpunan sisi dari graf G, maka berlaku

$$\sum_{v \in V} d(v) = 2|E|$$

Gambar 5. Persamaan yang menyatakan jumlah derajat semua simpul sama dengan 2 kali jumlah sisi graf tersebut, diambil dari (https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf)

d(v) adalah derajat simpul v

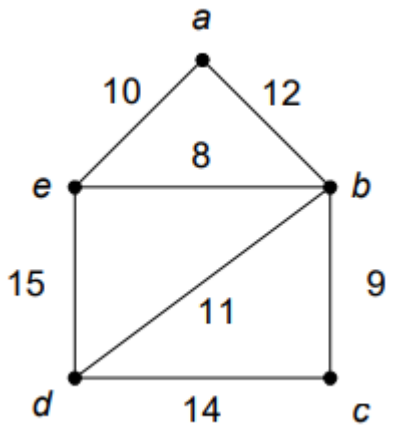
|E| adalah banyaknya sisi dalam graf

Dari konsep ini, didapatkan suatu teorema yang menyatakan bahwa untuk sembarang Graf, banyaknya simpul berderajat ganjil dari graf tersebut selalu genap. Jadi dapat diartikan, tidak mungkin sebuah graf memiliki simpul berderajat ganjil yang berjumlah ganjil. Contoh jika ingin membuat graf dengan lima buah simpul dengan masing – masing derajatnya adalah 2, 3, 1, 1, 2. Graf tersebut tidak mungkin dibuat karena jika dijumlahkan semua derajatnya maka hasilnya, jumlah semua derajat simpulnya ganjil.

Sirkuit atau siklus adalah istilah dimana suatu lintasan yang berawal dan berakhir pada simpul yang sama. Keterhubungan (*Connected*) adalah kondisi dimana dua buah simpul v1 dan

simpul v2 terhubung jika terdapat lintasan dari v1 ke v2. Misal ada sebuah graf G, graf G disebut graf terhubung jika untuk setiap pasangan simpulnya dalam himpunan V terdapat lintasan dari vi ke vj. Jika tidak maka G disebut graf tak-terhubung. Untuk graf berarah G, dikatakan terhubung jika graf tidak berarahnya terhubung. Dua simpul pada sebuah graf disebut terhubung kuat jika terdapat lintasan berarah dari simpul satu dan simpul dua dan juga lintasan berarah dari simpul dua ke simpul satu. Jika simpul tersebut tidak terhubung kuat tetapi terhubung pada graf tidak berarahnya, maka simpul tersebut dikatan simpul lemah. Misal terdapat graf berarah yang disebut G, graf G disebut graf terhubung kuat apabila untuk setiap pasang simpul sembarang di G terhubung kuat. Jika tidak, maka disebut graf terhubung lemah.

Graf berbobot adalah graf yang tiap sisinya memiliki bobot atau harga. Representasi bobot atau harga ini bisa berbagai macam, tergantung dengan apa yang direpresentasikan graf tersebut. Misal graf merepresentasikan suatu daerah antar kota, kota direpresentasikan sebagai simpul dan jalan yang menghubungkan antar kota direpresentasikan sebagai sisi berbobot yang bisa jadi bobot tersebut adalah panjang jalan.



Gambar 6. Ilustrasi graf berbobot, diambil dari (https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf)

Terdapat beberapa graf khusus yang memiliki ciri khas yang membedakannya dengan graf umum, yaitu graf lengkap, graf lingkaran, graf teratur, dan graf *bipartite*. Graf lengkap adalah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul. Ini artinya graf tersebut memiliki jumlah sisi n(n – 1)/2, n adalah jumlah simpul pada graf, ini juga dapat diartikan graf tersebut memiliki derajat (n – 1) untuk masing – masing simpul. Graf lingkaran adalah graf sederhana yang setiap simpulnya berderajat 2. Graf teratur adalah graf yang setiap simpulnya mempunyai derajat yang sama disebut graf teratur. Graf bipartite adalah yang himpunan simpulnya dapat dibagi menjadi dua.

Graf memiliki 3 cara dalam merepresentasikan, yaitu Matriks ketetanggaan, Matriks bersisian, dan Senarai ketetanggan. Matriks ketetanggaan adalah matriks yang kolom dan barisnya merepresentasikan simpul. Misal terdapat matriks ketetanggaan A dengan baris dan kolomnya adalah i dan j, nilai dari A[i][j] adalah 1 atau 0 untuk graf tidak berbobot. Nilai 1 berarti simpul i dan j bertetangaan sedangkan nilai 0 berarti simpul i dan j tidak bertetanggaan. Jika graf adalah graf berbobot, maka nilai A[i][j] adalah bobot dari sisi (i, j). Matriks bersisian adalah matriks dengan barisnya merepresentasikan simpul dan kolomnya merepresentasikan sisi. Misal terdapat matriks bersisian A dengan baris dan kolomnya i dan j, maka nilai A[i][j] adalah 1 atau 0. Nilai 1 berarti simpul i bersisian dengan sisi j, sedangkan jika nilainya 0 berarti simpul i tidak bersisian dengan sisi j.

	A	B	C	D
A	0	1	3	0
B	1	0	5	2
C	3	5	0	1
D	0	2	1	0

Gambar 7. Ilustrasi matriks ketetanggaan yang merepresentasikan graf berbobot, diambil dari (https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf)

	e1	e2	e3	e4	e5	e6
1	1	1	0	1	0	0
2	1	1	1	0	0	0
3	0	0	1	1	1	0
4	0	0	0	0	1	1

Gambar 8. Ilustrasi matriks bersisian yang merepresentasikan graf tidak berbobot, diambil dari (https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf)

B. *Algoritma A* (A star)*

Algoritma A* adalah salah satu algoritma Penentuan rute (*Path Planning*) yang cukup populer dan banyak digunakan. Algoritma ini banyak digunakan untuk memecahkan permasalahan pencarian lintasan/rute terpendek. Algoritma A* termasuk ke dalam kategori *Informed Search*, yaitu pencarian yang menggunakan heuristik untuk memperkirakan jarak atau biaya (*cost*) dari titik sekarang menuju tujuan. Dengan bantuan heuristik, A* dapat memprioritaskan pencarian jalur yang "menjanjikan" sehingga dapat mempercepat proses pencarian solusi.

Algoritma A* menggunakan sebuah fungsi yang disebut fungsi evaluasi yang bertujuan untuk menghitung total biaya (*cost*)

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

Gambar 9. Fungsi evaluasi yang digunakan Algoritma A* untuk menghitung total biaya, diambil dari

(https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/22-Route-Planning-(2025)-Bagian2.pdf)

g(n) = biaya dari simpul awal ke simpul n

h(n) = *heuristic estimate*, yaitu estimasi biaya dari simpul n ke simpul tujuan

f(n) = estimasi total biaya dari simpul awal menuju tujuan melalui simpul n

Cara kerja dari Algoritma A* adalah, ia akan mencoba menyeimbangkan antara eksplorasi dan eksploitas. Eksplorasi jalur yang terlihat menjanjikan menuju tujuan. Algoritmaa A* tetap akan mempertimbangkan jalur yang sejauh ini telah terbukti murah.

Heuristik dikatakan admissible jika nilai heuristik pada simpul n lebih rendah atau sama dengan biaya sebenarnya dari n ke tujuan

$$h(n) \leq h^*(n)$$

Gambar 10. h(n) <= h*(n), diambil dari

(https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/22-Route-Planning-(2025)-Bagian2.pdf)

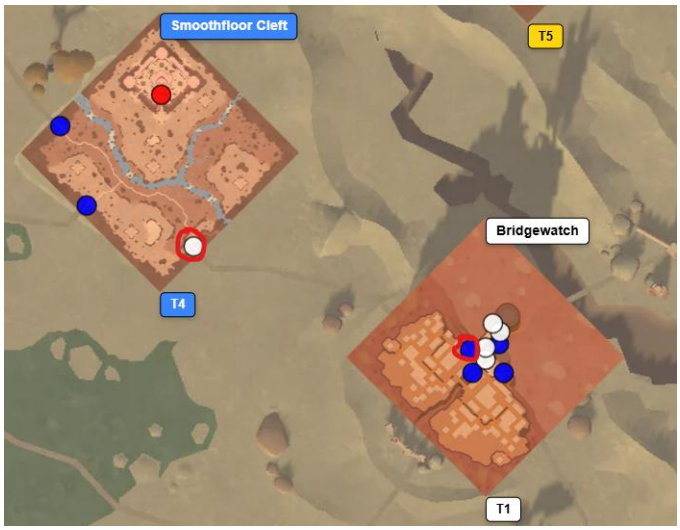
Artinya, heuristik tidak boleh melebih – lebihkan biaya ke tujuan. Dengan heuristik yang admissible, Algoritma A* dijamin akan menemukan solusi yang optimal. Contoh heuristik yang *admissible* adalah Manhattan Distance.

III. Analisis

Sebelum dilakukannya pengujian atau eksperimen, diperlukannya desain dari graf yang akan merepresentasikan rute dari *Royal City* ke Caerleon dan juga menentukan heuristik yang *admissible*.

A. *Desain Graf*

Untuk menentukan desain graf, penulis harus memahami desain dari map permainan Albion Online dan mekanisme perpindahan antar zona di permainan Albion Online.

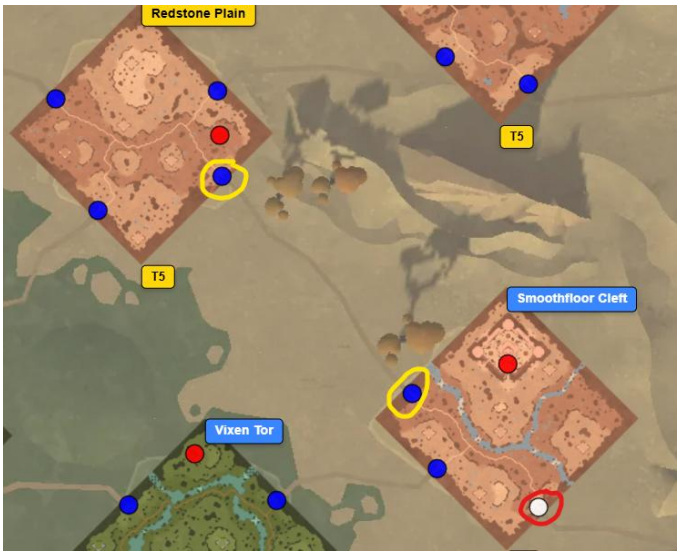


Gambar 11. Gambar kota BridgeWatch dan zona smoothfloor cleft, diambil dari

(https://albionfreemarket.com/albion-map)

Perhatikan gambar 11, pintu yang menghubungkan antar kota Bridgewatch dengan zona Smoothfloor Cleft adalah titik biru yang penulis tandai dengan lingkaran merah. Jika pemain memasuki pintu tersebut, maka akan terdapat *loading screen* lalu pemain akan masuk ke zona *Smoothfloor Cleft*. Proses *loading screen* ini berlangsung cukup cepat dan tiap zona kira – kira memiliki waktu yang sama. Hal ini menyebabkan jika graf yang diimplementasi adalah graf yang simpulnya adalah kota dan sisinya adalah yang menghubungkan kota tersebut, maka implementasi ini tidak mungkin dan tidak relevan, alasannya karena proses *loading screen* ini memiliki waktu yang relatif sama untuk setiap zona, jadi untuk implementasi graf berbobot tidak dapat dilakukan. Oleh karena itu penulis melakukan pendekatan lain dalam imlementasi grafnya. Penulis anggap titik putih yang berada diantara titik – titik biru (posisinya ditengah) pada kota Bridgewatch adalah simpul yang bernama BW (Bridgewatch). Simpul ini akan menjadi

simpul awal pencarian rute. Lalu titik – titik biru pada kota Bridgewatch adalah pintu yang menghubungkan ke zona lain. Pemain perlu waktu untuk berjalan dari titik putih ke titik biru. Titik biru pada kota Bridgewatch yang ditandai dengan lingkaran merah dan titik putih pada zona *Smoothfloor Cleft* yang ditandai dengan lingkaran merah, akan merepresentasikan sebuah simpul yang bernama simpul BW/SC (Bridgewatch/Smoothfloor Cleft) simpul inilah yang akan bersisian dengan simpul BW dengan bobot sesuai jarak yang ditempuh dari simpul BW ke simpul BW/SC.

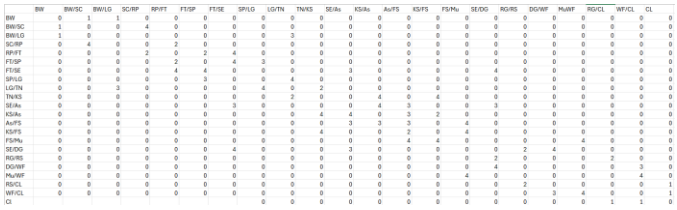


Gambar 12. Gambar zona Smoothfloor Cleft dan Redstone Plain, diambil dari

(https://albionfreemarket.com/albion-map)

Perhatikan Gambar 12, titik warna putih adalah simpul BW/SC yang tadi sudah didefinisikan. Titik biru yang ditandai lingkaran berwarna kuning pada zona *Smoothfloor Cleft* dan titik biru yang ditandai lingkaran berwarna kuning pada zona *Redstone Plain* akan menjadi simpul SC/RP (*Smoothfloor Cleft*/*Redstone Plain*) artinya simpul tersebut yang menghubungkan zona *Smoothfloor Cleft* dan *Redstone Plain*. Simpul tersebut juga bersisian dengan simpul BW/SC.

Bobot dari sebuah sisi pada graf nantinya akan menyatakan jarak antara simpul yang berisian dengan graf tersebut. Sayangnya penulis belum dapat menemukan data resmi terkait jaraknya. Untuk itu penulis akan menentukan jarak lewat perhitungan kasar yang menghitung jarak lurus antara pintu atau pada gambar 11 dan gambar 12 adalah titik menggunakan alat hitung jarak konvensional.



Gambar 13. Matriks ketetanggaan dari graf berbobot rute kota BridgeWatch ke Caerleon.

B. Menentukan fungsi heuristik h(n)

Fungsi heuristik yang ditentukan berdasarkan dari jumlah Langkah yang dilewati (jumlah simpul) dari simpul ke n hingga simpul tujuan yaitu simpul Caerleon (CL). Jadi, nilai h(n) menyatakan berapa banyak simpul yang harus dilewati untuk mencapai simpul CL, tanpa mempertimbangkan bobot/jarak antar simpul. Fungsi heuristik ini digunakan karena cukup sederhana dan *admissible*.

	Simpul	h(n) ke CL
23	CL	0
22	RS/CL	1
20	WF/CL	1
24	Cl	2
21	Mu/WF	2
18	DG/WF	2
17	RG/RS	2
16	MuWF	2
19	RG/CL	3
10	SE/DG	3
15	FS/Mu	3
7	FT/SE	4
9	SE/As	4
13	KS/FS	4
14	As/FS	4
12	KS/As	5
6	FT/SP	5
5	RP/FT	5
11	TN/KS	5
8	SP/LG	6
4	LG/TN	6
3	SC/RP	6
2	BW/LG	7
1	BW/SC	7
0	BW	8

Gambar 14. Tabel fungsi heuristik h(n) untuk setiap simpul

C. *Eksperimen*

Untuk melakukan eksperimen, penulis membuat sebuah program sederhana untuk menghitung rute terdekat dari kota BridgeWatch ke Caerleon. Di program tersebut ada fungsi untuk membuat graf berdasarkan matriks ketetanggaan yang sudah penulis buat beserta implementasi algoritma A*.

```
ASUS@LAPTOP-GQMSEO6T MINGW64 /d/CompSCI_ITB/Materi/Semester_4/STIMA/makalah
$ "C:\Users\ASUS\AppData\Local\Programs\Python\Python312\python.exe" Algoritma_Astar.py
Nodes in graph: ['BW', 'BW/SC', 'BW/LG', 'SC/RP', 'RP/FT', 'FT/SP', 'FT/SE', 'SP/LG',
G/RS', 'DG/WF', 'MU/WF', 'RS/CL', 'WF/CL', 'CL']
Nodes in heuristics: ['CL', 'RS/CL', 'WF/CL', 'MU/WF', 'DG/WF', 'DG/RS', 'MUWF', 'SE/D
, 'TN/KS', 'SP/LG', 'LG/TN', 'SC/RP', 'BW/LG', 'BW/SC', 'BW']

Start: BW, Goal: CL

Rute terpendek: BW → BW/SC → SC/RP → RP/FT → FT/SE → SE/DG → DG/RS → RS/CL → CL
Total biaya: 20
```

Gambar 15. Keluaran program.

Berdasarkan keluaran program, maka rutenya adalah, dimulai dari BridgeWatch lalu ke zona Smoothfloor Cleft lalu ke zona Redstone Plain lalu ke zona Flatriver Trough lalu ke zona Shiftshadow Expanse lalu ke Deadvein Gully lalu ke zona Roastcorpse Steepe lalu sampai ke Caerleon. Berarti terdapat total 9 simpul yang dilalui (termasuk simpul mulai). Berdasarkan Algoritma A* hasil eksperimen mendapatkan hasil yang optimal dengan total biaya 20. Jadi terbukti dengan algoritma A* dan fungsi heuristik berupa jumlah simpul yang akan dilalui mampu menghasilkan rute terdekat yang optimal

IV. KESIMPULAN

Algoritma A* berhasil menemukan rute terpendek dari Royal City (BW) ke Carleon (CL) dalam model graf tidak berarah berbobot, dengan total biaya 20. Fungsi Heuristik h(n) yang sederhana sudah cukup membuat A* optimal dan lebih efisien daripada pencarian tanpa heuristik.

LAMPIRAN

Source code:

https://github.com/Azzkaaaa/makalah

UCAPAN TERIMA KASIH

Terima kasih kepada Tuhan yang Maha Esa karena atas karunia dan berkatnya penulis dapat menyelesaikan makalah ini dengan cukup baik. Tak lupa penulis ucapkan terima kasih kepada keluarga dan teman – teman karena telah memberikan dukungan dan semangat kepada penulis sehingga penulis dapat menyelesaikan masalah ini. Terima kasih juga kepada Dr. Ir. Rinaldi, M.T. selaku dosen pengampu mata kuliah matematika diskrit karena telah memberikan ilmu dan pengetahuan sehingga penulis dapat menerapkannya untuk menulis makalah ini.

REFRENSI

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf
Diakses pada 16 Juni 2025

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/21-Graf-Bagian2-2024.pdf
Diakses pada 16 juni 2025

[3] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/22-Route-Planning-(2025)-Bagian2.pdf
Diakses pada 18 juni 2025

[4] https://albionfreemarket.com/albion-map
Diakses pada 18 juni 2025

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

Muhammad Aulia Azka 13523137