

Penentuan Rute Terpendek Perjalanan Antarbintang sebagai Aplikasi dari *Traveling Salesman Problem*

Vito Ghifari - 13520153¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹13520153@std.stei.itb.ac.id

Abstrak—Perjalanan antarbintang merupakan ide teoretis tentang perjalanan wahana berawak untuk mengunjungi bintang-bintang dalam galaksi. Perjalanan ini akan memakan waktu yang sangat lama karena jauhnya jarak antarbintang. Namun, perjalanan antarbintang dapat dimodelkan sebagai *traveling salesman problem* (TSP) agar rute terpendek dapat ditemukan. Penyelesaian dari TSP sendiri dilakukan dengan banyak strategi dengan dua di antaranya adalah *brute force* dan *nearest neighbour*.

Keywords—Graf, Bintang, Perjalanan Antarbintang, *Traveling Salesman Problem*.

I. PENDAHULUAN

Bintang merupakan objek alam semesta yang memancarkan energi dalam jumlah besar akibat reaksi fusi pada intinya. Energi yang dipancarkan oleh bintang berupa cahaya atau gelombang elektromagnetik. Kecuali Matahari, cahaya bintang yang mencapai Bumi akan teramati sebagai titik bercahaya di malam hari. Bintang yang teramati hanya sebagai titik ini disebabkan oleh jauhnya jarak bintang tersebut dari Bumi. Karena masalah jarak tersebut, satuan meter tidak lagi digunakan untuk mengukur jarak ke bintang. Satuan yang lain, yaitu tahun cahaya, digunakan sebagai standar jarak untuk objek astronomi di luar tata surya.

Karena bintang-bintang terpisah dengan jarak yang sangat jauh, muncul ide untuk melakukan perjalanan antarbintang. Perjalanan antarbintang mengacu kepada teori tentang perjalanan menggunakan wahana antariksa berawak untuk berpindah dari suatu bintang ke bintang yang lain. Perjalanan antarbintang ini digunakan pada film-film fiksi ilmiah seperti *Star Wars* dan *Star Trek*. Kedua film tersebut secara tersirat menjelaskan bahwa perjalanan antarbintang dilakukan dengan menggunakan wahana yang bergerak pada kecepatan cahaya atau lebih cepat daripada cahaya. Pada kenyataannya, secara fisika, perjalanan antarbintang seperti ini belum dapat diaplikasikan karena tidak ada hal yang lebih cepat daripada cahaya. Selain itu, belum ada teknologi yang diciptakan khusus untuk perjalanan antarbintang.

Jika suatu saat perjalanan antarbintang sudah dapat dilakukan, akan muncul pertanyaan baru. Bintang mana yang akan kita tuju terlebih dahulu? Pertanyaan ini mudah dijawab, yaitu dengan menuju bintang yang paling dekat dengan tata surya. Namun, jika tujuan perjalanan antarbintang ada lebih dari satu, bagaimana urutan bintang yang akan dikunjungi? *Traveling Salesman Problem* (TSP) muncul sebagai model untuk pemecahan masalah ini. Hanya saja, kita memerlukan data

jarak antarbintang sebagai acuan. Jarak antarbintang itu sendiri tidak tercatat secara eksplisit untuk bintang-bintang di langit. Masalah ini dapat diselesaikan dengan mengetahui data lain seperti koordinat bintang dan jarak bintang tersebut dari tata surya. Setelah itu, jarak antarbintang dapat diketahui dengan memanfaatkan persamaan-persamaan yang ada.

II. LANDASAN TEORI

A. Definisi Graf

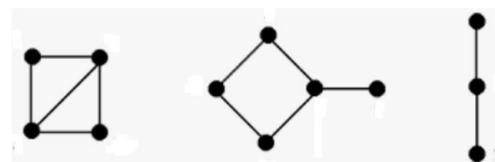
Pada matematika diskrit, graf merupakan himpunan simpul (*vertices*) yang dihubungkan oleh himpunan sisi (*edges*). Graf dinyatakan melalui persamaan $G = (V, E)$ dengan G adalah graf, V merupakan himpunan tidak kosong dari simpul-simpul, dan E merupakan himpunan sisi yang menghubungkan sepasang simpul pada graf tersebut.

B. Jenis-jenis Graf

Graf terbagi menjadi dua menurut adanya gelang dan sisi ganda pada suatu graf:

1. Graf sederhana

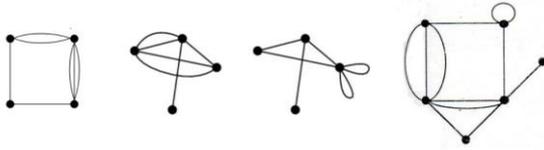
Definisi dari graf sederhana adalah graf yang tidak mempunyai gelang maupun sisi ganda. Gambar 1 merupakan contoh bentuk graf sederhana.



Gambar 1. Contoh graf sederhana (Sumber: [1])

2. Graf tak sederhana

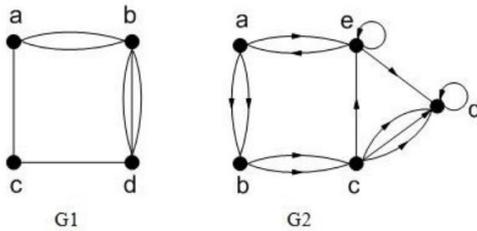
Definisi dari graf tak sederhana adalah graf yang mempunyai gelang ataupun sisi ganda. Graf tak sederhana terbagi lagi menjadi dua, yaitu graf ganda dan graf semu. Graf ganda adalah graf yang mengandung sisi ganda, sedangkan graf semu adalah graf yang mengandung sisi gelang. Contoh graf tak sederhana dapat dilihat pada gambar 2.



Gambar 2. Contoh graf tak sederhana (Sumber: [1])

Sementara itu, berdasarkan orientasi arah pada sisi, graf dibedakan menjadi dua jenis:

1. Graf tak berarah
Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak berarah.
2. Graf berarah
Graf yang setiap sisinya diberikan orientasi arah disebut graf berarah.

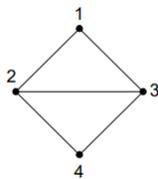


Gambar 3. G1 merupakan graf tak berarah, sedangkan G2 merupakan graf berarah. (Sumber: [1])

C. Terminologi Graf

Pada teori graf, dikenal istilah-istilah berikut.

1. Ketetanggaan (*Adjacent*)
Dua buah simpul pada suatu graf dinyatakan bertetangga apabila kedua simpul tersebut dihubungkan oleh sebuah sisi atau lebih.
2. Bersisian (*Incidency*)
Sisi $e = (v_j, v_k)$ pada suatu graf dapat dikatakan bersisian dengan simpul v_j dan v_k .

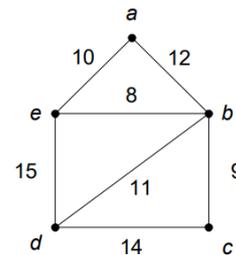


Gambar 4. Sisi (1, 2) bersisian dengan sisi 1 dan sisi 2. (Sumber: [1])

3. Simpul terpencil (*Isolated vertex*)
Simpul terpencil merupakan simpul yang tidak mempunyai sisi yang bersisian dengan simpul tersebut.
4. Graf kosong (*Empty graph*)
Sebuah graf didefinisikan sebagai graf kosong apabila himpunan sisinya merupakan himpunan kosong, artinya tidak ada simpul yang bersisian.
5. Derajat (*Degree*)
Derajat pada teori graf digunakan sebagai istilah untuk jumlah sisi yang bersisian dengan simpul tersebut. Contohnya, empat sisi pada suatu graf bersisian dengan

simpul A, maka simpul tersebut berderajat empat. Pada graf berarah, derajat simpul dibedakan menjadi derajat masuk (*in-degree*) dan derajat keluar (*out-degree*).

6. Lintasan (*Path*)
Lintasan pada graf merupakan sisi-sisi yang ditempuh dari simpul awal v_0 ke simpul tujuan v_n pada graf G . Tinjau kembali gambar 4. Lintasan yang terbentuk dari simpul 1 menuju simpul 4 adalah 1, 3, 4 atau 1, 2, 4.
7. Siklus atau sirkuit (*Cycle/circuit*)
Lintasan yang berawal dan berakhir pada simpul yang sama disebut siklus atau sirkuit.
8. Keterhubungan (*Connected*)
Simpul v_1 dan v_2 pada graf G dinyatakan terhubung apabila terdapat lintasan dari v_1 ke v_2 . Graf G disebut sebagai graf terhubung apabila sembarang v_1 dan v_2 pada G terdapat lintasan. Jika tidak memenuhi syarat tersebut, graf G disebut graf tak terhubung. Untuk graf berarah, simpul v_1 dan v_2 pada graf G dinyatakan terhubung kuat apabila terdapat lintasan dari v_1 ke v_2 dan lintasan dari v_2 dan v_1 . Jika salah satu lintasan tidak ada, kedua simpul tersebut terhubung lemah.
9. Upagraf (*Subgraph*)
Suatu graf $G_1 = (V_1, E_1)$ merupakan upagraf dari $G = (V, E)$ apabila $V_1 \subseteq V$ dan $E_1 \subseteq E$.
10. Komplemen upagraf
Misalkan graf $G_1 = (V_1, E_1)$ adalah upagraf dari $G = (V, E)$. $G_2 = (V_2, E_2)$ merupakan komplemen dari G_1 apabila $E_2 = E - E_1$ dan V_2 adalah simpul yang bersisian dengan E_2 .
11. Upagraf merentang
Upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$ disebut sebagai upagraf merentang apabila $V_1 = V$.
12. *Cut-set*
Cut-set dari graf terhubung G adalah himpunan sisi yang jika dibuang dari G menyebabkan graf G tidak terhubung. *Cut-set* selalu menghasilkan dua buah komponen.
13. Graf berbobot
Graf berbobot merupakan graf yang setiap sisinya diberi sebuah harga (bobot).



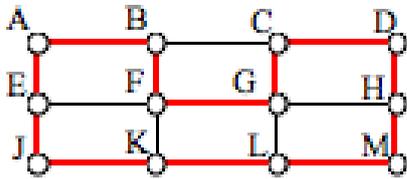
Gambar 5. Graf berbobot (Sumber: [1])

14. Graf lengkap
Graf lengkap adalah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Graf lengkap dilambangkan dengan K_n dengan n sebagai jumlah simpul. Sementara itu, jumlah sisi pada K_n adalah

$$e = n(n - 1)/2.$$

D. Lintasan dan Sirkuit Hamilton

Lintasan Hamilton merupakan lintasan yang melewati setiap simpul pada graf tepat satu kali. Sirkuit Hamilton adalah sirkuit yang melewati setiap simpul pada graf tepat satu kali, kecuali simpul asal yang dilalui dua kali. Graf yang mempunyai lintasan Hamilton disebut sebagai graf semi-Hamilton, sedangkan graf yang mempunyai sirkuit Hamilton disebut sebagai graf Hamilton. Tinjau gambar 6 yang merupakan penggambaran sebuah graf. Graf ini mempunyai beberapa sirkuit Hamilton, salah satunya ditandai dengan warna merah.

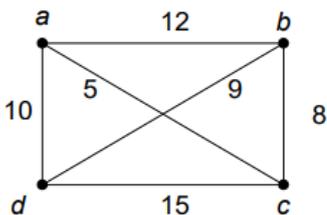


Gambar 6. Graf dengan sirkuit Hamilton (Sumber: <https://courses.lumenlearning.com/mathforliberalartscorequisite/chapter/hamiltonian-circuits/>)

Syarat cukup supaya graf sederhana G dengan n buah simpul ($n \geq 3$) dikatakan graf Hamilton adalah bila derajat setiap simpul paling sedikit berjumlah $n/2$. Suatu graf G yang merupakan graf Hamilton akan mempunyai $\frac{(n-1)!}{2}$ buah sirkuit Hamilton. Sementara itu, pada graf yang sama, terdapat $\frac{n-1}{2}$ buah sirkuit Hamilton yang saling lepas.

E. Traveling Salesman Problem

Traveling salesman problem (TSP) atau persoalan pedagang keliling merupakan aplikasi dari graf yang menggunakan pemodelan lintasan Hamilton. Sebagai contoh, ada sejumlah kota dan jarak satu kota ke kota lainnya. Pedagang keliling yang berada di kota asal ingin menggunakan rute terpendek untuk menyinggahi semua kota tepat satu kali dan kembali ke kota awal. Persoalan ini terlihat mudah jika kota yang disinggahi hanya sedikit, misalnya kurang dari 5 kota. Namun, seiring bertambahnya jumlah kota, rute alternatif yang dapat digunakan pedagang keliling tersebut menjadi jauh lebih banyak. Jika dihubungkan dengan sirkuit Hamilton, jumlah rute yang tersedia bila terdapat kota sebanyak n adalah $\frac{(n-1)!}{2}$. Hanya dalam 10 kota saja, jumlah rute yang tersedia menjadi $\frac{9!}{2} = 181440$ buah.



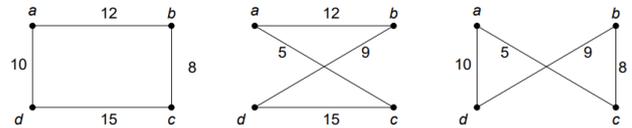
Gambar 7. Graf berbobot sebagai pemodelan TSP (Sumber: [2])

Gambar 7 mencontohkan masalah pedagang keliling. Graf lengkap ini mempunyai 4 simpul $a, b, c,$ dan d sebagai permasalahan kota. Karena terdapat 4 kota, rute yang terbentuk adalah sebanyak $\frac{(4-1)!}{2} = 3$ buah rute. Perhatikan gambar 8 sebagai kemungkinan sirkuit Hamilton yang terbentuk dari graf pada gambar 7. Bobot dari masing-masing sirkuit Hamilton yang terbentuk jika dipilih simpul a sebagai titik asal adalah

$$I_1 = (a, b, c, d, a) \rightarrow 12 + 8 + 15 + 10 = 45,$$

$$I_2 = (a, b, d, c, a) \rightarrow 12 + 9 + 15 + 5 = 41,$$

$$I_3 = (a, c, b, d, a) \rightarrow 5 + 8 + 9 + 10 = 32.$$



Gambar 8. Sirkuit Hamilton yang terbentuk. Dari kiri ke kanan: I_1, I_2, I_3 (Sumber: [2])

Setelah masing-masing panjang lintasan dari sirkuit Hamilton diketahui, kita dapat membandingkan panjang lintasan tersebut. Urutan lintasan yang terbentuk jika ketiga lintasan tersebut diurutkan membesar adalah I_3, I_2, I_1 . Oleh karena itu, solusi dari persoalan pedagang keliling yang dimodelkan dengan graf pada gambar 7 jika simpul a sebagai titik asal adalah lintasan $I_3 = (a, b, c, d, a)$ dengan bobot total I_3 sebesar 32.

Untuk mendapatkan solusi dari *traveling salesman problem*, ada beberapa metode yang digunakan, di antaranya

1. Brute force

Brute force merupakan algoritma penyelesaian persoalan pedagang keliling dengan menghitung semua kemungkinan bobot sirkuit Hamilton yang ada pada suatu graf. Algoritma ini merupakan algoritma yang menjamin dapat menemukan jarak paling optimal. Hanya saja, karena semua kemungkinan harus diperhitungkan, *brute force* sangat tidak efisien. Kompleksitas waktu dari algoritma ini adalah $O(n!)$.

2. Tetangga terdekat (*nearest neighbour*)

Strategi *nearest neighbour* sebagai solusi dari TSP adalah dengan memilih simpul awal, lalu mencari bobot terendah pada simpul yang dapat dituju. Kemudian, pada simpul tujuan, akan dicari kembali sisi dengan bobot terendah selain sisi yang tujuannya pernah dikunjungi. Ulangi proses ini hingga tidak ada lagi simpul yang belum dituju. Algoritma *nearest neighbour* mempunyai kompleksitas waktu sebesar $O(n^2)$. Kekurangan dari algoritma ini yaitu tidak adanya jaminan untuk mendapatkan hasil optimal. Akan tetapi, penggunaan *nearest neighbour* menjamin bahwa solusi yang dihasilkan memenuhi 15-20% batas bawah Held-Karp [3].

3. Greedy heuristic algorithm

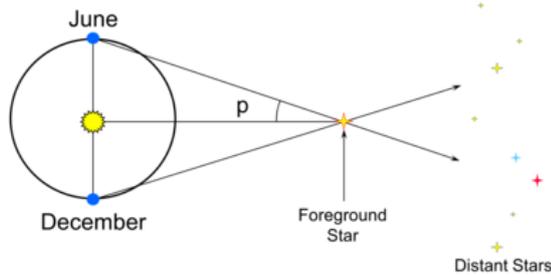
Algoritma ini memanfaatkan pencarian yang menghasilkan *local optima*. Hasil ini kemudian dioptimalkan untuk mencari *global optima*. Cara kerja algoritma *greedy heuristic* adalah dengan mengurutkan bobot sisi dari paling kecil. Setelah itu, pilih sisi dengan

bobot terkecil berikutnya dan lakukan hal ini terus menerus dengan syarat tidak ada *loop* yang terbentuk. Kompleksitas waktu dari algoritma *greedy heuristic* adalah $O(N^2 \log_2(N))$. Sama seperti algoritma tetangga terdekat, *greedy heuristic* tidak menjamin solusi yang ditemukan adalah lintasan terpendek.

III. PEMBAHASAN

A. Jarak Bintang

Metode penentuan jarak bintang pertama kali dilakukan dengan paralaks. Metode ini tetap digunakan hingga saat ini untuk menghitung jarak bintang-bintang terdekat. Prinsip paralaks untuk penentuan jarak bintang berdasar pada perubahan posisi bintang yang relatif dekat terhadap bintang latar yang relatif sangat jauh [4].



Gambar 9. Paralaks bintang (Sumber: [4])

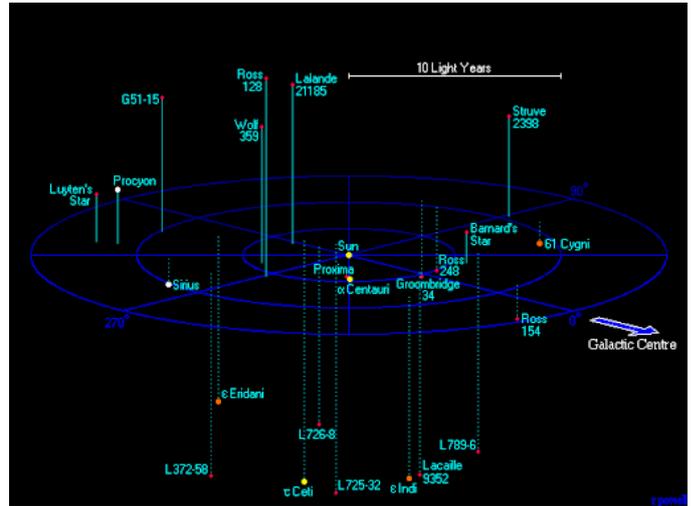
Perubahan posisi bintang diamati pada waktu terpisah sebesar 6 bulan. Dalam 6 bulan ini, bintang yang diamati terlihat berpindah sebesar $2p$ derajat terhadap bintang latar. Jika jarak Bumi ke Matahari sebesar D dan jarak Matahari ke suatu bintang sebesar d , akan terbentuk segitiga yang mempunyai persamaan

$$\tan p = \frac{D}{d}$$

dengan p adalah sudut paralaks. Dalam hal ini, D atau jarak Bumi-Matahari didefinisikan sebagai 1 satuan astronomi (SA). Karena jarak bintang yang sangat jauh, sudut p yang terbentuk sangat kecil, sehingga nilai $\tan p$ bernilai hampir sama dengan p . Maka, jarak bintang dari Matahari dapat didefinisikan sebagai

$$\begin{aligned} \tan p &= \frac{1}{d}, \\ \tan p \approx p &= \frac{1}{d}, \\ d &= \frac{1}{p}. \end{aligned}$$

Pada umumnya, p dinyatakan dalam satuan detik busur (*arcsecond*), dengan 1 detik busur bernilai $1/3600$ dari 1 derajat. Nilai p yang dinyatakan dalam detik busur akan menghasilkan jarak bintang dalam satuan parsec (*parallax of one second*). Satuan parsec sendiri dapat dikonversi ke satuan tahun cahaya (*light-year/ly*) dengan perbandingan 1 parsec $\approx 3,26$ tahun cahaya. Gambar 10 mengilustrasikan persebaran dari 33 bintang yang ada dalam jarak 12,5 tahun cahaya [5]. Sebagai perbandingan, Proxima Centauri yang merupakan bintang terdekat berjarak 4,22 tahun cahaya dari Matahari.



Gambar 10. Visualisasi posisi bintang-bintang terdekat dari tata surya (Sumber: [5])

B. Sistem Koordinat Benda Langit

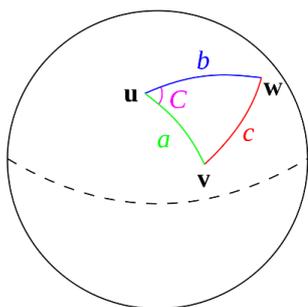
Salah satu data yang mempunyai relasi dengan jarak untuk mendapatkan jarak antar bintang adalah koordinat bintang. Dalam astronomi, dikenal empat jenis sistem koordinat untuk menyatakan posisi objek di ruang angkasa. Keempat jenis sistem tersebut adalah

1. Koordinat horizon
Koordinat horizon mengacu pada koordinat benda langit yang relatif terhadap pengamat di Bumi. Posisi suatu benda langit dalam dapat dinyatakan dalam dua arah, yaitu sudut terhadap arah utara sebagai Azimuth dan ketinggian terhadap horizon sebagai *altitude*.
2. Koordinat ekuator
Untuk objek luar angkasa yang relatif tidak berpindah seperti bintang, koordinat ekuatorial digunakan untuk mendefinisikan lokasi dari objek tersebut di luar angkasa. Objek luar angkasa yang dinyatakan dalam koordinat ekuator mempunyai dua arah dalam bidang fundamental. Arah pertama adalah asensio rekta (*right ascension/RA*) yang didefinisikan sebagai sudut suatu objek terhadap titik Aries ke arah timur sejajar dengan bidang ekuator. Arah kedua adalah deklinasi, yaitu besar sudut suatu objek terhadap bidang ekuator.
3. Koordinat ekliptika
Pada umumnya, koordinat ekliptika digunakan untuk mendefinisikan posisi benda-benda langit dalam tata surya. Ini disebabkan oleh kemiringan sumbu rotasi terhadap sumbu orbit Bumi. Planet atau benda langit lainnya di dalam tata surya akan lebih mudah dinyatakan dalam koordinat ini.
4. Koordinat galaktik
Sistem koordinat galaktik menggunakan acuan bidang galaksi untuk menyatakan koordinat suatu benda langit. Utamanya, koordinat galaktik digunakan untuk mendefinisikan posisi benda-benda langit dalam Galaksi Bimasakti seperti bintang dan nebula.

Sebagai sistem koordinat yang paling sering digunakan, koordinat ekuatorial dari bintang-bintang yang relatif dekat dari tata surya telah dicatat pada berbagai macam katalog bintang. Salah satu katalog yang terkenal adalah katalog Hipparcos. Katalog ini merupakan hasil dari pemetaan langit oleh satelit Hipparcos yang memanfaatkan efek paralaks untuk menentukan jarak bintang dengan ketelitian tinggi. Hasil yang didapatkan dari pemetaan ini adalah tercatatnya data kuantitatif dari 118.218 bintang [6], termasuk koordinat ekuatorial dan besar paralaks masing-masing bintang.

C. Jarak Antarbintang

Keempat sistem koordinat benda langit sangat dekat kaitannya dengan trigonometri bola. Pada trigonometri bola, dikenal persamaan cosinus untuk menghitung panjang suatu sisi. Panjang ini penting untuk diketahui agar jarak antarbintang dapat diperhitungkan.



Gambar 11. Segitiga bola dengan titik u, v, w, dan sisi a, b, c, dan sudut C pada titik u. (Sumber: https://en.wikipedia.org/wiki/Spherical_law_of_cosines)

Sisi a, b, dan c pada gambar 11 adalah panjang dari lingkaran besar yang menghubungkan titik u, v, dan w. C adalah sudut yang dibentuk antara sisi a dan b, atau bisa disebut sebagai sudut vuw. Persamaan cosinus yang dimaksud pada segitiga bola ini adalah

$$\cos c = \cos a \cos b + \sin a \sin b \cos C .$$

Untuk menghubungkan persamaan ini dengan jarak antarbintang, ada langkah-langkah yang dilewatkan pada makalah ini untuk mempersingkat pengerjaan. Persamaan akhir yang didapatkan dari rumus cosinus ini setelah dihubungkan dengan koordinat ekuatorial bintang adalah

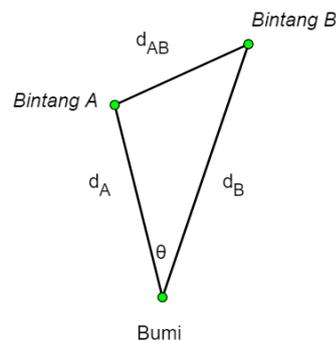
$$\cos \theta = \sin \delta_A \sin \delta_B + \cos \delta_A \cos \delta_B \cos(\Delta RA)$$

dengan θ adalah sudut pisah antara bintang A dan bintang B, δ_A dan δ_B sebagai deklinasi bintang A dan bintang B, dan $\Delta RA = RA_A - RA_B$. Gambar 12 mendeskripsikan tentang relasi sudut θ ini terhadap jarak antarbintang. Dengan rumus cosinus untuk segitiga datar, persamaan yang terbentuk adalah

$$d_{AB}^2 = d_A^2 + d_B^2 - 2 d_A d_B \cos \theta .$$

Kedua persamaan yang ini akan membentuk satu rumus untuk menghitung jarak bintang A ke bintang B dengan

$$d_{AB} = \sqrt{d_A^2 + d_B^2 - 2 d_A d_B (\sin \delta_A \sin \delta_B + \cos \delta_A \cos \delta_B \cos(\Delta RA))}$$



Gambar 12. Segitiga yang terbentuk antara bintang A, bintang B, dan Bumi (Sumber: dokumen pribadi)

IV. PENERAPAN TRAVELING SALESMAN PROBLEM

A. Penyederhanaan Masalah

Faktor-faktor lain untuk perhitungan solusi dari *traveling salesman problem* harus diperhatikan. Hanya saja, penulis menyederhanakan masalah agar perhitungan yang dilakukan tidak kompleks. Di bawah ini adalah bentuk penyederhanaan masalah yang digunakan.

1. Menganggap bahwa posisi bintang dan Matahari tidak berubah. Artinya, koordinat dan jarak bintang tidak akan berubah akibat rotasi galaksi.
2. Menggunakan wahana yang mempunyai bahan bakar yang tak terbatas, sehingga pengisian bahan bakar tidak diperlukan.
3. Menempuh lintasan berupa garis lurus sepanjang perjalanan.

B. Perhitungan Jarak Antarbintang

Tabel 1 merepresentasikan koordinat ekuatorial dan jarak sepuluh bintang terdekat dari tata surya menggunakan katalog Hipparcos, dengan sedikit pengecualian pada Alpha Centauri. Penyebutan Alpha Centauri sebenarnya sebenarnya merujuk kepada sistem tiga bintang yang terdiri atas Alpha Centauri A, Alpha Centauri B, dan Proxima Centauri. Pada penelitian ini, Alpha Centauri disederhanakan sehingga istilah Alpha Centauri mengacu pada bintang paling terang di sistem tersebut, yaitu Alpha Centauri A.

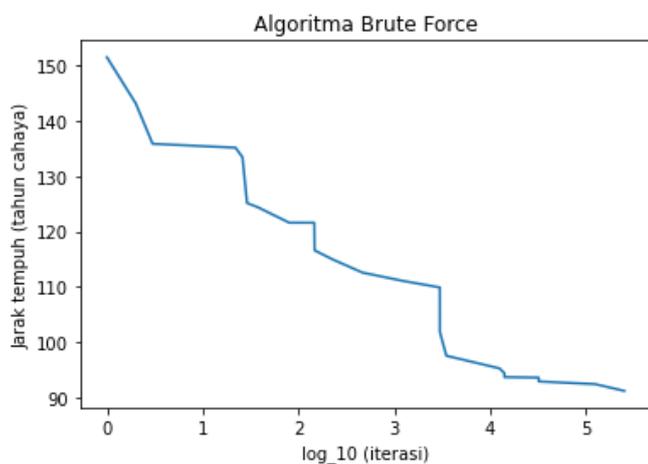
Nama Bintang	Jarak (ly)	Asensio Rekta (°)	Deklinasi (°)
Alpha Centauri	4.39	219.9204	-60.8351
Bintang Barnard	5.94	269.454	4.668288
Lalande 21185	8.31	165.8359	35.98146
Sirius	8.6	101.2885	-16.7131
Ross 154	9.69	282.454	-23.8358
Epsilon Eridani	10.49	53.23509	-9.45831
Lacaille 9352	10.73	346.4465	-35.8563
Ross 128	10.88	176.9335	0.807526
61 Cygni	11.35	316.7118	38.74149
Procyon	11.4	114.8272	5.227508

Tabel 1. Data jarak dan koordinat ekuatorial dari 10 bintang terdekat (Sumber: [7] dengan penyesuaian)

Dengan data yang tersedia pada tabel 1, jarak antar bintang untuk kesepuluh bintang tersebut dapat diketahui. Data jarak antar bintang ini hanya ditampilkan pada lampiran karena jumlah kolom yang dihasilkan cukup banyak.

C. Solusi dengan Brute Force

Pada percobaan pertama, *brute force* digunakan untuk mencari rute terpendek perjalanan antar bintang. Algoritma ini diimplementasikan pada program dalam bahasa Python yang dibantu *library* Pandas. *Library* Pandas digunakan sebagai tabulasi data bintang yang didapatkan dari katalog Hipparcos dan perhitungan jarak antar bintang. Pencarian solusi dengan *brute force* memakan waktu sebesar 119 detik. Algoritma yang digunakan penulis belum optimal karena rute yang diperhitungkan tidak bersifat dua arah. Efeknya, total iterasi yang dilakukan bukan $\frac{(11-1)!}{2}$ iterasi, melainkan $(11 - 1)! = 3.628.800$ iterasi. Sebagai contoh, rute A, B, C, D, A akan dianggap berbeda dengan rute A, D, C, B, A dan akan dihitung secara terpisah. Algoritma ini berhasil mendapatkan rute terpendek untuk perjalanan antar bintang pada data yang telah tersedia, yaitu sebesar 91,17 tahun cahaya. Rute terpendek ini berhasil ditemukan pada iterasi ke-251.463 dari total 3.628.800 iterasi. Gambar 13 menyatakan hubungan logaritmik dari nomor iterasi dengan penemuan rute yang lebih pendek. Apabila ditemukan jarak yang lebih pendek dari iterasi sebelumnya, jarak dan nomor iterasinya dicatat, serta dijadikan jarak terpendek sementara. Proses ini berulang hingga iterasi terakhir.



Gambar 13. Grafik logaritmik dari jumlah iterasi terhadap penemuan rute yang lebih pendek (Sumber: dokumen pribadi)

Untuk percobaan kedua, penulis menggunakan algoritma tetangga terdekat sebagai perbandingan terhadap solusi optimal. Iterasi yang dilakukan hanyalah untuk mencari jarak terpendek ke bintang yang belum pernah dikunjungi sehingga tidak memakan banyak waktu seperti algoritma *brute force*. Pada kasus ini, algoritma tetangga terdekat tidak berhasil mendapatkan rute terpendek. Hasil yang didapatkan dari algoritma ini adalah 103,23 tahun cahaya. Jarak tempuh ini lebih jauh sekitar 13,2% daripada jarak tempuh minimal dari algoritma *brute force*. Tabel 2 menunjukkan perbandingan hasil dari kedua algoritma.

	<i>Brute Force</i>	<i>Nearest Neighbour</i>
Tujuan ke-1	Alpha Centauri	Alpha Centauri
Tujuan ke-2	Ross 128	Bintang Barnard
Tujuan ke-3	Lalande 21185	Lalande 21185
Tujuan ke-4	Procyon	Sirius
Tujuan ke-5	Sirius	Ross 154
Tujuan ke-6	Epsilon Eridani	Epsilon Eridani
Tujuan ke-7	Lacaille 9352	Lacaille 9352
Tujuan ke-8	Ross 154	Ross 128
Tujuan ke-8	Bintang Barnard	61 Cygni
Tujuan ke-10	61 Cygni	Procyon
Tujuan ke-11	Matahari	Matahari
Panjang rute	91,17 ly	103,23 ly
Perbedaan	0%	13,2%
Waktu eksekusi	119 detik	~0.004 detik

Tabel 2. Perbandingan algoritma brute force dan tetangga terdekat.

V. PENUTUP

A. Kesimpulan

Perjalanan antar bintang adalah teori untuk melakukan perjalanan menuju bintang-bintang. Karena keterbatasan teknologi, pada saat ini perjalanan antar bintang hanyalah fiksi ilmiah. Akan tetapi, perjalanan antar bintang dapat dimodelkan dengan graf berbobot dan lintasan Hamilton. Jika permasalahannya adalah perjalanan antar bintang dari Matahari dan kembali ke Matahari, *traveling salesman problem* dapat digunakan untuk menentukan rute terpendek perjalanan antar bintang. Data yang dibutuhkan untuk menyelesaikan persoalan ini adalah koordinat ekuatorial bintang dan jarak bintang tersebut terhadap tata surya. Katalog Hipparcos digunakan sebagai acuan data koordinat dan jarak bintang. Sementara itu, bintang yang digunakan dari katalog Hipparcos adalah sepuluh bintang terdekat dari tata surya. Penentuan jarak antar bintang dilakukan dengan mendapatkan jarak bintang dan sudut pisahnya melalui rumus cosinus pada segitiga bola. Beberapa asumsi dilakukan sebagai penyederhanaan masalah, contohnya adalah posisi dan jarak bintang yang tidak berubah.

Perhitungan rute terpendek dilakukan pada dua percobaan dengan algoritma yang berbeda. Percobaan pertama adalah mencari rute terpendek dengan *brute force*. Algoritma ini berhasil mendapatkan rute terpendek sebesar 91,17 tahun cahaya. Pada percobaan kedua, algoritma tetangga terdekat digunakan sebagai perbandingan dari algoritma pada percobaan pertama. Rute yang didapatkan dari percobaan kedua adalah sebesar 103,23 tahun cahaya. Namun, secara kompleksitas waktu, algoritma tetangga terdekat lebih baik digunakan jika rute paling optimal tidak diperlukan.

VI. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan karunia, sehingga penulis dapat menyelesaikan makalah yang berjudul "Penentuan Rute Terpendek Perjalanan Antarbintang sebagai Aplikasi dari *Traveling Salesman*

Problem” selesai tepat pada waktunya. Tidak lupa penulis mengucapkan terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi, S.T, M.Sc sebagai dosen pengajar mata kuliah IF2120 Matematika Diskrit Kelas 3 atas bimbingan dan pengajaran yang telah dilakukan di kelas Matematika Diskrit ini. Penulis juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, MT. sebagai salah satu dosen pengampu Matematika Diskrit yang memberikan referensi dan sumber pembelajaran Matematika Diskrit melalui situs beliau. Terakhir, penulis mengucapkan terima kasih kepada orang tua, keluarga, dan seluruh pihak yang membantu penulis dalam menyelesaikan makalah ini.

REFERENCES

- [1] Rinaldi Munir, “Graf (Bag. 1)”, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>, diakses 10 Desember 2021 pukul 20.23.
- [2] Rinaldi Munir, “Graf (Bag. 3)”, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian3.pdf>, diakses 10 Desember 2021, pukul 21.59.
- [3] Haider Abdulkarim, Ibrahim Fadhil Alshammari, “Comparison of Algorithms for Solving Traveling Salesman Problem”, https://www.researchgate.net/publication/280597707_Comparison_of_Algorithms_for_Solving_Traveling_Salesman_Problem, diakses 12 Desember 14.19.
- [4] Richard Pogge, “Lecture 5: Distances of the Stars”, <http://www.astronomy.ohio-state.edu/~pogge/Ast162/Unit1/distances.html>, diakses 12 Desember 15.43.
- [5] Richard Powell, “The Universe within 12.5 Light Years The Nearest Stars”, <http://www.atlasoftheuniverse.com/12lys.html>, diakses 12 Desember 18.23.
- [6] R Nave, “Hipparcos”, <http://hyperphysics.phy-astr.gsu.edu/hbase/Solar/Hipparcos.html>, diakses 13 Desember 16.14.
- [7] Konivat, “Hipparcos Star Astronomy Catalog”, <https://www.kaggle.com/konivat/hipparcos-star-catalog>, diakses 8 Desember.

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, 13 Desember 2021



13520153 Vito Ghifari

LAMPIRAN

	A	B	C	D	E	F	G	H	I	J
A	0	6.46	10.43	9.55	8.1	12.62	10.41	10.24	14.4	13.04
B	6.46	0	10.88	14.39	5.54	15.68	11.54	12.62	9.28	16.91
C	10.43	10.88	0	10.93	15.96	15.79	19.04	6.51	15.25	9.63
D	9.55	14.39	10.93	0	17.16	7.84	14.77	12.19	18.89	5.22
E	8.1	5.54	15.96	17.16	0	17.62	9.6	16.28	12.19	20.71
F	12.62	15.68	15.79	7.84	17.62	0	11.51	18.82	16.86	11.48
G	10.41	11.54	19.04	14.77	9.6	11.51	0	20.53	14.13	19.52
H	10.24	12.62	6.51	12.19	16.28	18.82	20.53	0	19.81	11.51
I	14.4	9.28	15.25	18.89	12.19	16.86	14.13	19.81	0	20.76
J	13.04	16.91	9.63	5.22	20.71	11.48	19.52	11.51	20.76	0

Tabel 3. Jarak antarbintang untuk 10 bintang terdekat, dinyatakan dalam tahun cahaya. A: Alpha Centauri, B: Bintang Barnard, C: Lalande 21185, D: Sirius, E: Ross 153, F: Epsilon Eridani, G: Lacaille 9352, H: Ross 128, I: 61 Cygni, J: Procyon.
(Sumber: Dokumen pribadi)