Khazanah
Ilmu
Berazam

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

ANALISIS PERBANDINGAN PADA ALGORITMA BELLMAN FORD DAN DIJKSTRA PADA GOOGLE MAP

Bayu Irawan Ahmad Prasetyo[1], Andi Maslan[2]

[1]Fakultas Teknik Informatika, Universitas Putera Batam, Jl. R Soeprapto Muka Kuning Kota Batam
e-mail: irawanbayu30@gmail.com
[2]Dosen Fakultas Teknik Informatika, Universitas Putera Batam, Jl. R Soeprapto Muka Kuning Kota Batam
e-mail: lanmasco@gmail.com

ABSTRACT

The Bellman Ford algorithm and the Dijkstra algorithm are often used in determining the shortest path. Bellman Ford's Algorithm and Dijkstra's Algorithm have the same way of working because both algorithms have the same strategy, Greedy. In the Greedy strategy the first step taken is to choose the side with the smallest weight among all sides connected from the initial node to the neighbor node. Compared to the Dijkstra algorithm, the Bellman Ford algorithm is superior because the Bellman Ford algorithm can complete calculations on graphs that have negative values. Determination of the shortest path using the Bellman Ford algorithm and the Dijkstra algorithm as a route reference used in determining the route from the Tembesi campus to the Tiban campus. This research was conducted to compare the two algorithms on the Google Map in terms of both the complexity of the space and the complexity of the time needed to provide a solution.

Keywords*: Bellman Ford, Dijkstra, Shortest Track Path, Google Map.*

PENDAHULUAN

Dalam kehidupan sehari-hari sering sekali melakukan perjalanan dari suatu tempat ke tempat lain. Kebutuhan akan perjalanan ini menuntut adanya pemilihan lintasan terpendek dari suatu tempat ke tempat lainnya dalam waktu yang singkat sehingga dapat mengefisienkan jarak, waktu, dan biaya yang dibutuhkan untuk mencapai tempat tujuan tersebut.

Perkembangan ilmu pengetahuan dan teknologi saat ini, banyak sekali algoritma-algoritma yang sering digunakan untuk memecahkan permasalahan jalur lintas terpendek seperti algoritma Dijkstra, algorima Ant Colony, algoritma Floyd Warsall, algoritma Bellman-Ford, algoritma A-Star dan lainnya. Algoritma-algoritma ini dibuat untuk mencari cara paling efisien untuk menganalisa pemilihan jalur tercepat dengan berbagai penambahan variable lainnya.

Jalur lintas terpendek merupakan bagian dari teori graph, jika diberikan graph berbobot, masalah jarak terpendek adalah bagaimana mendapatkan jalur pada graph yang meminimalkan jumlah bobot sisi pembentuk jalur tersebut.

Algoritma yang baik adalah algoritma yang memberikan solusi dengan hasil pemilihan jalur lintas terpendek dengan nilai sebenarnya dan efisien. Efisien Algoritma ditinjau dari dua hal yaitu kompleksitas ruang dan kompleksitas waktu. Kompleksitas ruang adalah besarnya memori yang dibutuhkan dalam menjalankan Algoritma, sedangkan kompleksitas waktu ialah waktu yang dibutuhkan oleh Algoritma dalam memberikan solusi jalur lintas terpendek. Algoritma Bellman Ford dan Algoritma Dijkstra memiliki sisi efisiensi yang berbeda dalam penentuan rute pada Google Map. Oleh karena itu, kedua algoritma tersebut diangkat dalam penelitian ini untuk membandingkannya dalam pemilihan jalur lintas terpendek.

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

Rumusan Masalah

Berikut rumusan masalah yang akan digunakan sebagai pembahasan pada penelitian ini:

1. Bagaimana cara kerja pencarian lintasan terpendek menggunakan algoritma Bellman-Ford dan algoritma Dijkstra di Google Map.
2. Bagaimana perbandingan algoritma Bellman-Ford dan algoritma Dijkstra dari sisi kompleksitas ruang dan kompleksitas waktu (*running time*).

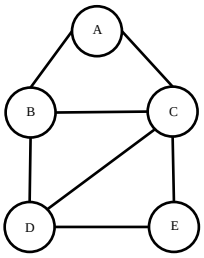
TINJAUAN PUSTAKA

Teori Graph

Graph adalah struktur diskrit yang terdiri dari adanya simpul (*vertex*) dan adanya sebuah sisi (*edge*), graph adalah pasangan himpunan (V, E) dimana V merupakan sebuah himpunan yang tidak kosong dari sebuah *vertex* dan E adalah himpunan sisi yang meghubungkan sepasang simpul dalam graph tersebut (Harahap & Khairina, 2017: 18).

Menurut (Harahap & Khairina, 2017: 18) Jika dilihat arah sisi, secara umum graph dapat dibedakan menjadi 2 jenis, yaitu

1. Graph Tak Berarah

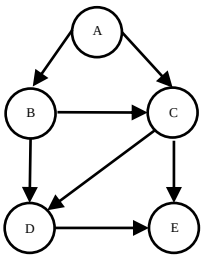


Gambar 1 Graph Tak Berarah
Sumber: Data olahan Peneliti (2020)

Graph tak berarah adalah graph yang sisinya tidak memiliki arah atau graph yang sisinya tidak memiliki anak panah. Graph tak berarah, urutan pasangan simpul yang dihubungkan oleh sisi diabaikan. Maka bisa disebutkan adalah sisi yang sama.

(vj, vk) = (vk, vj)

2. Graph Berarah



Gambar 2 Graph Berarah
Sumber: Data olahan Peneliti (2020)

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

Graph berarah adalah graph yang sisinya mempunyai orientasi arah yang jelas atau pada sisinya miliki arah anak panah. Sisi yang berarah disebut dengan busur (*orc*). Pada graph berarah (vj, vk) dan (vk, vj) dinyatakan 2 buah busur yang berbeda, atau dengan kata lain:

$$\mathbf{(vj, vk) \neq (vk, vj)}$$

Jalur Lintas Terpendek (*Shortest Path*)

Jalur lintas terpendek (*Shortest Path*) memiliki arti sebagai peroses meminimalisasikan bobot pada sebuah lintasan graph. Pada jurnal (Harahap & Khairina, 2017: 18) terdapat beberapa persoalan lintasan terpendek antara lain:

1. Lintasan terpendek antara dua buah simpul
2. Lintasan terpendek antara semua pasangan simpul
3. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain
4. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu

Algoritma Bellman Ford

Algoritma Bellman Ford yang ditemukan oleh Richard E. Bellman, seorang ahli matematika yang terlahir di New York 1920 (Hutasoit, 2019: 21)

Algortima Bellman Ford merupakan pengembangan dari algoritma Dijkstra, algoritma Bellman Ford akan benar jika dan hanya jika graph tidak terdapat *cycle* dengan bobot negatif yang dicapai dari sumber (Hamdi & Prihandoko, 2018: 28).

Algoritma Dijkstra

Algoritma Dijkstra ditemukan oleh Edger Wybe Dijkstra, seorang ilmuan dari belanda. Algoritma ini dikembangkan di tahun 1956 dan dipublikasikan secara umum untuk pertama kalinya di tahun 1959.

Dalam pencarian jalur terpendek Algoritma Dijkstra bekerja dengan mencari bobot yang paling minimal dari suatu graf berbobot, jarak terpendek akan didapatkan dari dua atau lebih titik dari suatu graf dan nilai total yang didapat adalah bernilai paling kecil(Hamdi & Prihandoko, 2018: 27)

Google Map API

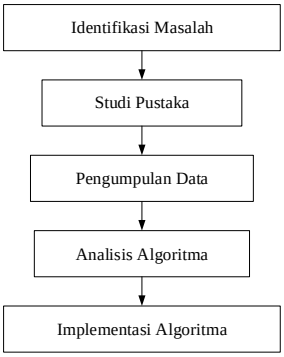
Google Map API adalah sebuah layanan dari Google yang diberikan kepada penggunanya agar dapat mengakses Google Map untuk kebutuhan pengembangan aplikasi. Terdapat beberapa fitur yang ada pada Google Map API dimana pengguna dapat memanipulasi peta, menambahkan konten dan mengizinkan pengguna untuk membangun aplikasi dengan mengambil variable yang berisi informasi dari Google Map tersebut.

METODE PENELITIAN

Desain Penelitian

Desain penelitian yang dilakukan pada penelitian ini terdiri dari beberapa langkah, yaitu:

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041



Gambar 3 Desain Penelitian
Sumber: Data Penelitian (2020)

1. Identifikasi Masalah
   Tahap awal dalam desain penelitian ini ialah mengidentifikasi masalah mengenai perbandingan dua algoritma dengan menggunakan google map sebagai media pengujian.

2. Studi Pustaka
   Studi pustaka dilakukan dengan memahami teori dasar melalui sumber terpercaya yaitu buku, jurnal penelitian terdahulu, dan beberapa sumber terpercaya lainnya yang berhubungan pembahasan penelitian.

3. Pengumpulan Data
   Adapun beberapa cara yang dilakukan dalam pengumpulan data adalah melakukan obeservasi dan dokumentasi data.

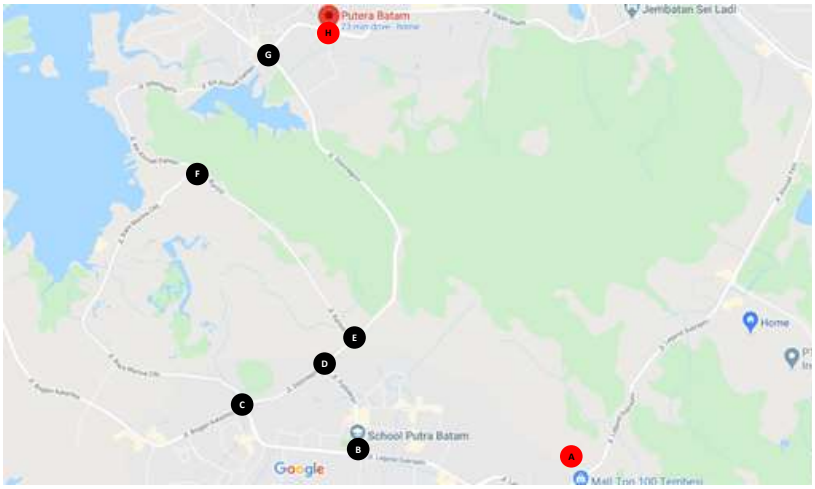
4. Analisis Algoritma
   Analisis yang dilakukan adalah perhitungan secara manual setiap algoritma dengan rute yang ditentukan. Hasil analisis dapat digunakan sebagai acuan dalam mengimplentasikan algoritma di sistem yang akan dirancang.

5. Implementasi Algoritma
   Mengimpelentasi algoritma bellman ford dan Dijkstra baik dalam perhitungan manual dan penerapan pada Google map API untuk pencarian rute terpendek.

Pengumpulan Data

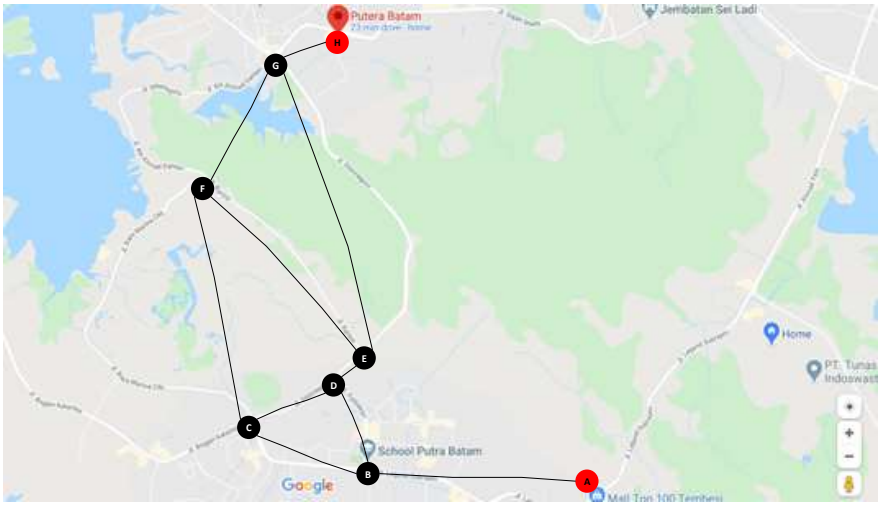
Proses mengumpulkan data pada penelitian ini dilakukan dengan cara mengumpulkan data sekunder yang didapat dari google map. Rute yang digunakan dalam penilitian ini yaitu rute dari kampus Universitas Putera Batam Tembesi yang berada di jalan Letjen R. Soeprato menuju kampus Universitas putera batam yang berada di Jalan Gajah Mada. Titik awal ditandai dengan simpul A dan simpul akhir ditandai dengan simpul H sesuai dengan Gambar 3.2 dibawah ini.

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041



Gambar 4 Map Dengan Titik Simpul
Sumber: Data Peneliti

Pembentukan graf yang akan digunakan di penelitian ini diambil dari sebuah peta yang telah memiliki vertex dan simpul. Simpul ditandai dengan setiap adanya persimpangan jalan dan vertex merupakan bentang panjang jalan antar simpul.



Gambar 5 Graf Rute Lintasan
Sumber: Data Penelitian

Gambar diatas merupakan pembentukan graf dari map yang didapat dari google map, graf ini belum memiliki bobot yang memiliki 8 simpul dan 10 garis (vertex). Table dibawah ini berisi penjelasan detail mengenai Gambar 3.2.

Tabel 1 Simpul Tetangga

No.	Simpul	Lokasi	Simpul Tetangga
1	A	Jalan Letjend Soeprapto	B
2	B	Simpang Tiga Jalan Letjend soeprapti (Simp. Makam Pahlawan) dan Jalan Pahlawan	C, D

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

3	C	Simpang Empat jalan Jalan Letjend Soeprapto (Simp. Basecamp) dan jalan Raya Marina City	F, D
4	D	Simpang Tiga Jalan. Pahlawan dan jalan Diponegoro	E
5	E	Simpang Tiga Jalan Diponegoro dan Jalan Kartini	F, G
6	F	Simpang Tiga Jalan Raya Marina City dan Jalan Kyai Haji Ahmad Dahlan	G
7	G	Simpang Empat Jalan Kyai Haji dan Jalan Gajah Mada	H
8	H	Jalan Gajah Mada	G

Sumber: Data Penelitian (2020)

Setiap simpul tersebut memiliki nilai koordinat. Nilai koordinat tersebut didapat dari google map, nilai koordinat tersebut tertera pada tabel 3.3 dibawah ini.

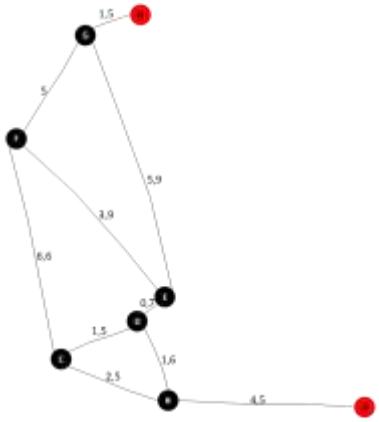
Tabel 2 Nilai Koordinat

Simpul	Latitude	Longitude
A	1.0432921	103.9976036
B	1.045420	103.968224
C	1.052830	103.952032
D	1.058792	103.963926
E	1.062568	103.968048
F	1.083457	103.939590
G	1.105216	103.955164
H	1.1081319	103.9603179

Sumber: Data Peneliti (2020)

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

Selanjutnya setiap sisi antar simpul dihitung sehingga menghasilkan jarak-jarak yang tertera di gambar dibawah ini. Graf tersebut akan ditambahkan bobot pada setiap vertex nya. Bobot yang akan ditambahkan ini didapat dari jarak antar simpul yang memiliki satuan dalam kilometer.



Gambar 6 Graf Dengan Nilai Bobot
Sumber: Data Penelitian (2020)

Gambar diatas merupakan pembentukan graf dari map yng didapat dari google map, graf ini belum memiliki bobot yang memiliki 8 simpul dan 10 garis (vertex). Table dibawah ini berisi penjelasan detail mengenai Gambar 3.

Tabel 3 Simpul Tetangga

No.	Simpul	Lokasi	Simpul Tetangga
1	A	Jalan Letjend Soeprapto	B
2	B	Simpang Tiga Jalan Letjend soeprapti (Simp. Makam Pahlawan) dan Jalan Pahlawan	C, D
3	C	Simpang Empat jalan Jalan Letjend Soeprapto (Simp. Basecamp) dan jalan Raya Marina City	F, D
4	D	Simpang Tiga Jalan. Pahlawan dan jalan Diponegoro	E
5	E	Simpang Tiga Jalan Diponegoro dan Jalan Kartini	F, G
6	F	Simpang Tiga Jalan Raya Marina City dan Jalan Kyai Haji Ahmad Dahlan	G
7	G	Simpang Empat Jalan Kyai Haji dan Jalan Gajah Mada	H
8	H	Jalan Gajah Mada	G

Sumber: Data Penelitian (2020)

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

Gambar 6 menunjukan hasil pembuatan graf dari map yang dijadikan contoh kasus, yang selanjutnya akan digunakan untuk menghitung jalur terpendek antara titik A sampai titik H dengan menggunakan algoritma Bellman Fold dan Algoritma Dijkstra.

1.1 Analisis Sistem Algoritma Bellman Ford

Untuk memulai perhitungan maka dilakukan tahap inisialisasi, simpul awal diberikan "0" dan simpul lain dengan "∞".

Tabel 4 Literasi Ke-0

		Simpul (Km)							
		A	B	C	D	E	F	G	H
L	0	0	∞	∞	∞	∞	∞	∞	∞

Sumber: Data Peneliti (2020)

Kemudian dalam perhitungan ini menggunakan rumus Bellman Ford, yaitu:
M [i,v] = min(M [i-1,v] , (M [i-1,n]+ Cvn))

Perhitungan pada literasi pertama dimana masing-masing simpul masih "∞" dan akan ditambahkan bobot jarak yang ditempuh dari simpul awal ke simpul tujuan. Berikut perhitungan literasi pertama, diawali dengan simpul A yang tehubung dengan simpul B.

M [1,B] = min(M [0,B] , (M [0,A]+ C_{AB}))
= min(∞, (0+4,5 km))
=min(∞, 4,5 km)
=4,5 km

Tabel 5 Literasi Ke-1

		Simpul (Km)							
L		A	B	C	D	E	F	G	H
i	0	0	∞	∞	∞	∞	∞	∞	∞
t -	1	0	4,5	∞	∞	∞	∞	∞	∞

Sumber: Data Peneliti (2020)

Berikutnya literasi ke-2, simpul yang terhubung dengan B yaitu C, dan D, maka berikut ini perhitungan literasi keduanya.

M [2,C] = min(M [1,C] , (M [1,B]+ C_{BC}))
= min(∞, (4,5+2,5 km))
=min(∞, 7 km)
=7 km

M [2,D] = min(M [1,D] , (M [1,B]+ C_{BD}))
= min(∞, (4,5+1,6 km))
=min(∞, 6,1 km)
=6,1 km

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

Tabel 6 Literasi Ke-2

		Simpul (Km)							
		A	B	C	D	E	F	G	H
Lit	0	0	∞	∞	∞	∞	∞	∞	∞
	1	0	4,5	∞	∞	∞	∞	∞	∞
	2	0	4,5	7	6,1	∞	∞	∞	∞

Sumber: Data Peneliti (2020)

Selanjutnya lakukan literasi berulang hingga semua simpul telah bernilai "0" dan mendapatkan rute akhir, sehingga didapat hasil akhir pada tabel 3.4 di bawah ini.

Tabel 7 Hasil Akhir Literasi

		Simpul (Km)							
		A	B	C	D	E	F	G	H
Literasi	0	0	∞	∞	∞	∞	∞	∞	∞
	1	**0**	4,5	∞	∞	∞	∞	∞	∞
	2	0	**4,5**	7	6,1	∞	∞	∞	∞
	3	0	4,5	7	**6,1**	6,8	∞	∞	∞
	4	0	4,5	7	6,1	**6,8**	10,7	12,7	∞
	5	0	4,5	**7**	6,1	6,8	10,7	12,7	∞
	6	0	4,5	7	6,1	6,8	**10,7**	12,7	∞
	7	0	4,5	7	6,1	6,8	10,7	**12,7**	14,2
	8	0	4,5	7	6,1	6,8	10,7	12,7	**14,2**

Sumber: Data Penelitian (2020)

Analisis Sistem Algoritma Dijkstra

Pada perhitungan tahap inisialisasi dengan algoritma Dijkstra, simpul awal diberi inisial dengan "1" dan simpul lain dengan inisial "0", kemudian setiap node yang sudah dipilih diinisialkan dengan "1".

Tabel 8 Perhitungan Dengan Algoritma Dijkstra

Literasi Ke-1								
Node	A	B	C	D	E	F	G	H
Status	1	0	0	0	0	0	0	0
Bobot	-	4,5	-	-	-	-	-	-
Prod-	A	A	-	-	-	-	-	-
Literasi Ke-2								
Node	A	B	C	D	E	F	G	H
Status	1	1	0	0	0	0	0	0
Bobot	-	4,5	7	6,1	-	-	-	-
Prod-	A	A	B	B	-	-	-	-

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

Literasi Ke-3								
Node	A	B	C	D	E	F	G	H
Status	1	1	0	1	0	0	0	0
Bobot	-	4,5	7	6,1	6,8	-	-	-
Prod-	A	A	B	B	D	-	-	-
Literasi Ke-4								
Node	A	B	C	D	E	F	G	H
Status	1	1	0	1	1	0	0	0
Bobot	-	4,5	7	6,1	6,8	10,7	12,7	-
Prod-	A	A	B	B	D	E	E	-
Literasi Ke-5								
Node	A	B	C	D	E	F	G	H
Status	1	1	1	1	1	0	0	0
Bobot	-	4,5	7	6,1	6,8	10,7	12,7	-
Prod-	A	A	B	B	D	E	E	-
Literasi Ke-6								
Node	A	B	C	D	E	F	G	H
Status	1	1	1	1	1	1	0	0
Bobot	-	4,5	7	6,1	6,8	10,7	12,7	-
Prod-	A	A	B	B	D	E	E	-
Literasi Ke-7								
Node	A	B	C	D	E	F	G	H
Status	1	1	1	1	1	1	1	0
Bobot	-	4,5	7	6,1	6,8	10,7	12,7	14,2
Prod-	A	A	B	B	D	E	E	G
Literasi Ke-8								
Node	A	B	C	D	E	F	G	H
Status	1	1	1	1	1	1	1	1
Bobot	-	4,5	7	6,1	6,8	10,7	12,7	14,2

Prod-	A	A	B	B	D	E	E	G

Sumber: Data Peneliti (2020)

Perhitungan berhenti karena semua node telah dihitung. Untuk melihat jalur mana yang terpilih dapat ditelusuri dari predecessor nya. Sehingga akan didapat data dibawah ini.

A KE B : A-B : 4,5
A KE C : A-B-C : 7
A KE D : A-B-D : 6,1
A KE E : A-B-D-E : 6,8
A KE F : A-B-D-E-F : 10,7
A KE G: A-B-D-E-G : 12,7
A KE H: A-B-D-E-G-H : 14,2

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

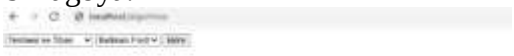
HASIL DAN PEMBAHASAN

Hasil Penelitian

Hasil perancangan sistem dalam penelitian ini berupa bentuk tampilan dari program yang telah dirancang dengan memanfaatkan Google Map API dan mengimplementasikan algoritma Bellman Ford dan Algoritma Dijkstra pada sistem.

Tampilan Awal Sistem

Tampilan awal sistem adalah pemilihan rute yang akan dicari solusi rute terpendeknya dan algoritma yang akan digunakan. Pilihan rute terdiri dari rute kampus Tembesi ke kampus Tiban, rute kampus Tembesi ke kampus Nagoya, dan rute dari kampus Tiban ke kampus Nagoya.



Gambar 7 Tampilan Awal Program
Sumber: Data Penelitian (2020)

Tampilan Pencarian Rute Terpendek Algoritma Bellman Ford

Pada tampilan ini, rute yang dipilih adalah rute dari kampus Tembesi ke kampus Tiban dengan algoritma Bellman Ford. Pada tampilan diawali dengan peta Kota Batam dengan titik-titik simpul yang akan dilalui. Dibawah map kota terdapat tabel berisi perhitungan algoritma bellman ford yang didapat dari rute tersebut.

Gambar 8 Tampilan Rute Kampus Tembesi ke Kampus Tiban Dengan Algoritma Bellman Ford
Sumber: Data Peneliti (2020)

Tampilan Pencarian Rute Terpendek Algoritma Dijkstra

Pada tampilan ini, rute yang dipilih masih sama yaitu rute dengan titik awal kampus Tembesi dan titik akhir kampus Tiban dengan algoritma Dijkstra sebagai metode pencarian rute terpendeknya.

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

Gambar 9 Tampilan Rute Kampus Tembesi Ke Kampus Tiban Dengan Algoritma Dijkstra

Sumber: Data Peneliti (2020)

Pengujian Perbandingan

Berikut kriteria yang kan digunakan untuk mengukur performa masing-masing algoritma dalam mencari rute tependek:

1. Kompleksitas Ruang (Space Complexity)
   Besar beban memori akan memengaruhi kinerja dari sebuah algoritma untuk mencari solusi jalur terpendek.
2. Kompleksitas Waktu (Time Complexity)
   Kompleksitas waktu merupakan waktu yang digunakan Algoritma dalam menemukan solusi jalur lintas terpendek.

Tabel 9 Tabel Perbandingan Kinerja Algoritma

Rute	Bellman Ford		Dijkstra	
	Memori (MB)	Waktu (s)	Memori (MB)	Waktu (s)
Tembesi - Tiban	10.6MB	3.60s	12.3MB	3.95s
Tembesi - Nagoya	10.6MB	3.74s	12.3MB	4.12s
Tiban - Nagoya	10.6MB	3.54	12.3MB	3.83s

Sumber: Data Peneliti (2020)

SIMPULAN

Simpulan

Berdasarkan pembahasan dan pengujian menentukan jalur lintas terpendek dengan algoritma Bellman Ford dan Algoritma Dijkstra pada Google Map, didapat beberapa kesimpulan dari peneliti sebagai berikut:

1. Algoritma Bellman Ford dapat digunakan pada graf yang mengandung simpul negative, sedangkan algoritma Dijkstra tidak dapat melakukan perhitungan apabila ada sisi yang memuat nilai negative.

Open Journal System
e-ISSN : 2621-9441
p-ISSN : 2623-1041

2. Berdasarkan hasil pengujian kompleksitas waktu, waktu meningkat seiring dengan bertambah jumlah simpul yang ada tiap rute. Dengan hasil algoritma Bellman Ford lebih unggul karena memiliki waktu lebih cepat dibandingkan algoritma Dijkstra.
3. Untuk hasil pengujian kompleksitas ruang, algoritma Bellman Ford lebih memiliki ukuran ruang yang lebih kecil dibandingkan dengan algoritma Dijkstra.

DAFTAR PUSTAKA

Hamdi, S., & Prihandoko. (2018). Analisis Algoritma Dijktra dan Algoritma Bellman-Ford Sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran (Studi Kasus: Kecamatan Praya Kota). *Jurnal Ilmiah Ilmu-Ilmu Teknik*, *8*(1), 26–32.

Harahap, M. K., & Khairina, N. (2017). Pencarian Jalur Terpendek dengan Algoritma Dijkstra. *SinkrOn*, *2*(2), 18. https://doi.org/10.33395/sinkron.v2i2.61

Hutasoit, E. T. H. (2019). Pencarian Rute Terpendek Menggunakan Algoritma Bellman-Ford (Studi Kasus: PT. JNE Medan). *Jurnal Sistem Komputer Dan Informatika (JSON)*, *1*(1), 20. https://doi.org/10.30865/json.v1i1.1367