

Algoritma *Dijkstra* dan *Bellman-Ford* dalam Sistem Pemetaan *Barbershop* di Kota Lhokseumawe

Rozzi Kesuma Dinata*, Bustami**, Ar Razi***, Muhammad Arasyi****

Fakultas Teknik, Universitas Malikussaleh

*rozzi@unimal.ac.id, bustami@unimal.ac.id**, ***ar.razi@unimal.ac.id, ****muhammad.170170083@mhs.unimal.ac.id

ABSTRACT

A Barbershop service provider is a company that provides hair care to the community. Many people are currently doing business in this field, and many business actors are opening Barbershops in a variety of locations, ranging from campuses to office districts to densely populated towns. In Lhokseumawe City, there are 12 Barbershops. The application's benefit is that it can identify the shortest path from the user's location to the selected Barbershop, as well as the Barbershop's location and a brief description of the Barbershops in Lhokseumawe City. Only the system's defined nodes can be used to find the fastest route to the Barbershop. Dijkstra's method was chosen because it works against all current alternative functions and provides the shortest path from all nodes, ensuring that the shortest path is produced optimally. Because the Bellman-Ford algorithm is a variant of the BFS (best-first-search) algorithm, which is also employed in the search for the closest distance when the search for the closest distance has a negative weight, it was chosen. The same results were obtained in picking the route based on the results of the route selection test. However, when the two techniques are compared in terms of program execution time, Dijkstra's algorithm is faster than the Bellman-Ford algorithm.

Keyword: *Nearest Distance, Dijkstra's Algorithm, Bellman-Ford Algorithm, GPS*

1. Pendahuluan

Sistem Informasi Geografis (SIG) merupakan sistem informasi terkomputerisasi yang memungkinkan pengumpulan, pengambilan sampel, manipulasi, pengambilan, analisis, dan penyajian data referensi geografis serta persiapan, representasi, dan interpretasi fakta tentang permukaan bumi [1].

Penyedia layanan *Barbershop* adalah perusahaan yang memberikan perawatan rambut kepada masyarakat. Banyak orang yang saat ini melakukan bisnis di bidang ini, dan banyak pelaku usaha yang membuka *Barbershop* di berbagai lokasi, mulai dari kampus hingga kawasan perkantoran hingga kota-kota padat penduduk. *Barbershop* merupakan tempat yang biasa dikunjungi oleh para pendatang yang datang dari kota atau provinsi lain. Bagi pengunjung yang berasal dari Lhokseumawe mungkin tidak masalah untuk mencari *Barbershop*, bisa melakukan pencarian secara mendadak, boleh saja, namun tidak dengan pengunjung yang berasal dari kota atau provinsi yang jauh dari kota Lhokseumawe, menjadi kendala tersendiri karena mencari *Barbershop* adalah suatu hal yang penting, mendapatkan *Barbershop* yang bagus dan sesuai dengan kondisi yang diinginkan.

Dalam kasus lain, masih banyak pengguna yang tidak mengetahui lokasi *Barbershop* di kota Lhokseumawe. Yang membuat sebagian besar pengguna mengambil rute yang lebih panjang untuk mencapai lokasi *Barbershop*. Sebagian besar pengguna perjalanan ingin menempuh jarak tercepat, alasannya karena hemat bahan bakar, hemat waktu dan tenaga. Dengan acuan untuk menentukan rute mana yang akan dilalui, Anda dapat memberikan kemudahan bagi pengguna untuk memilih rute tercepat untuk mencapai lokasi yang diinginkan dan tidak tersesat di tengah jalan.

Penerapan Sistem Informasi Geografis (SIG) berbasis *mobile web* merupakan salah satu cara untuk membantu mengakses berbagai macam tempat [2][3], karena perangkat *mobile* merupakan perangkat yang sudah lazim dimiliki oleh setiap orang saat ini [4][5].

Sistem informasi geografis, layanan Internet, dan perangkat seluler adalah contoh layanan berbasis lokasi [6]. Pemahaman tentang Layanan Berbasis Lokasi atau dikenal dengan *Location Based Service* (LBS) menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang digunakan [7][8]. LBS merupakan layanan informasi dapat diakses melalui *mobile device* menggunakan *mobile network* yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut [9][10].

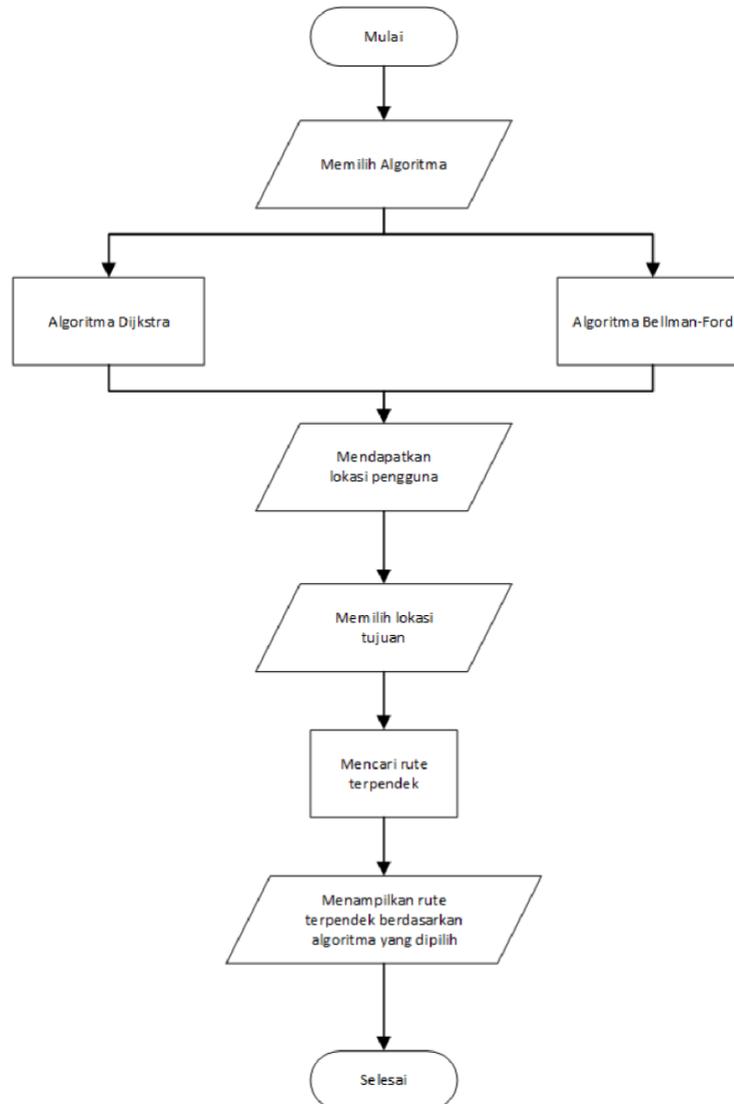
Oleh karena itu penelitian ini bertujuan untuk membuat suatu teknologi informasi berupa aplikasi pencarian *Barbershop* yang dilengkapi dengan pencarian jalur terpendek menggunakan Algoritma *Dijkstra* dan Algoritma *Bellman-Ford*. Melihat kelebihan dan kekurangan dari algoritma *Dijkstra* dan algoritma *Bellman-Ford*, perlu membandingkan keduanya untuk menunjukkan hasil dari dua algoritma berbasis deterministik yang salah satunya akan menjadi solusi optimal. Selain itu, tujuan penelitian ini juga untuk memudahkan

pelanggan untuk menemukan *Barbershop* yang dilengkapi dengan teknologi *Global Positioning System* (GPS) yang berfungsi sebagai navigasi global yang dapat mengarahkan pelanggan untuk menggunakan rute tercepat menuju *Barbershop* yang dituju.

2. Metode Penelitian

2.1 Skema Sistem

Skema Sistem Informasi Geografis Pemetaan *Barbershop* di Kota Lhokseumawe Menggunakan Algoritma Dijkstra dan Algoritma *Bellman-Ford* Berbasis Android, ditampilkan pada Gambar 1.



Gambar 1. Skema Sistem

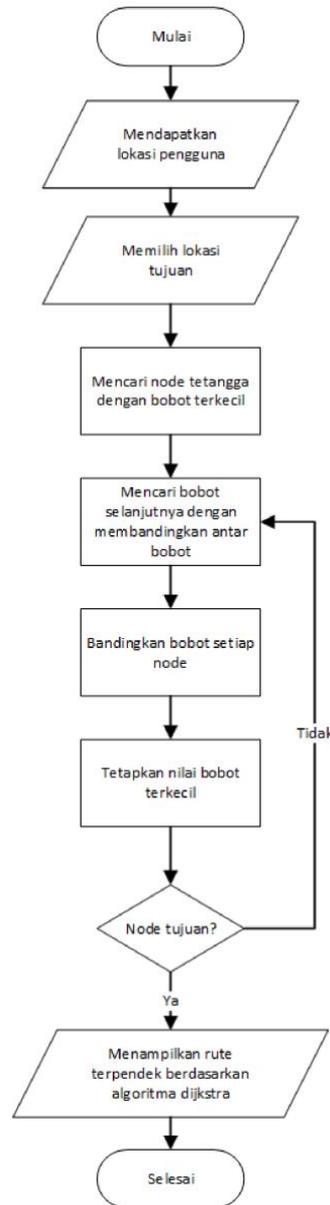
Dalam penelitian ini akan diambil sample titik keberangkatan yaitu dari Gedung Teknik Informatika Universitas Malikussaleh menuju *Barbershop* yang ada di Kota Lhokseumawe sebagai titik tujuan menggunakan Algoritma Dijkstra dan Algoritma *Bellman-Ford* untuk mendapatkan rute terpendeknya. Mengenai proses merepresentasikan lokasi-lokasi tujuan dan jalan-jalan di Kota Lhokseumawe dalam bentuk graf berlabel dan berbobot. Berikut langkah-langkah merepresentasikan lokasi-lokasi dan jalan-jalan di Kota Lhokseumawe dalam bentuk graf berlabel dan berbobot.

1. Mengambil dan menyimpan foto peta Kota Lhokseumawe melalui google earth.
2. Menentukan titik Gedung Teknik Informatika Universitas Malikussaleh (titik asal) dan *Barbershop* yang ada di Kota Lhokseumawe (titik tujuan).
3. Menarik garis lurus untuk menghubungkan titik asal dengan titik tujuan.
4. Menentukan titik-titik pemotongan ruas jalan yang akan dilalui dari titik asal menuju titik tujuan.

5. Mengumpulkan data berupa:
 - a. Koordinat dan alamat/nama lokasi-lokasi *Barbershop* sebagai alamat tujuan, dan koordinat setiap titik pemotongan jalan yang diberi symbol T_n , dimana n merupakan bilangan bulat serta bobot setiap titik (T_n) yang terhubung.
 - b. Informasi *Barbershop* berupa gambar, nama, alamat, tarif, dan no hp pemilik.

2.2 Flowchart Algoritma Dijkstra

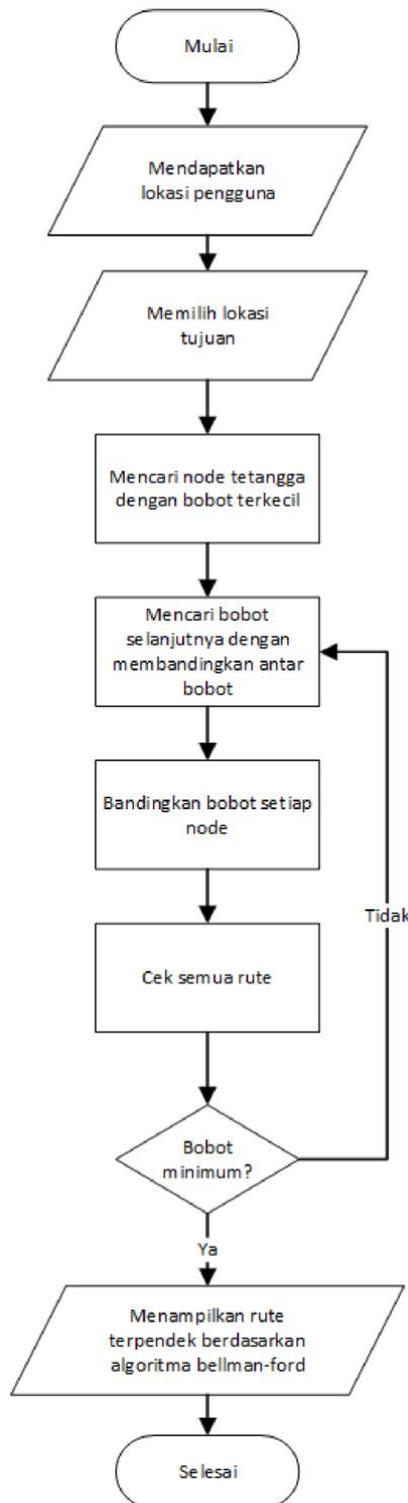
Adapun flowchart Algoritma Dijkstra pada Sistem Informasi Geografis Pemetaan *Barbershop* di Kota Lhokseumawe ditampilkan pada Gambar 2.



Gambar 2. Flowchart Algoritma Dijkstra

2.3 Flowchart Algoritma Bellman-Ford

Adapun flowchart Algoritma *Bellman-Ford* pada Sistem Informasi Geografis Pemetaan *Barbershop* di Kota Lhokseumawe ditampilkan pada Gambar 3.

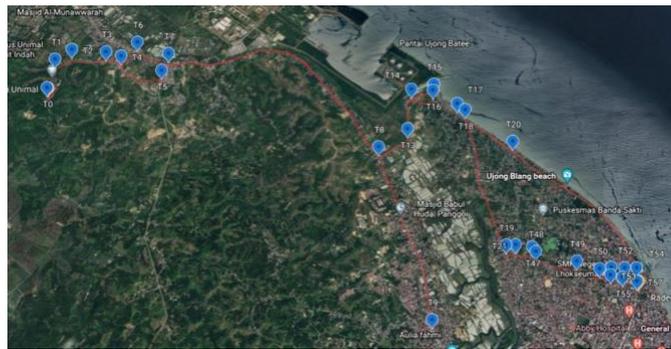


Gambar 3. Flowchart Algoritma *Bellman-Ford*

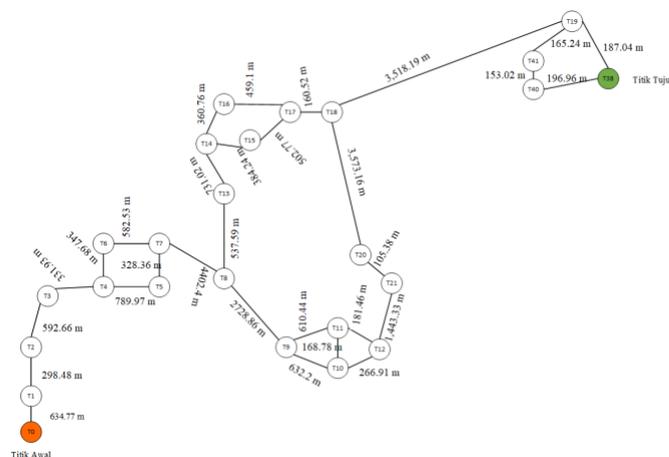
3. Hasil dan Pembahasan

3.1 Analisis Hasil Perhitungan Algoritma Dijkstra

Pada perhitungan Algoritma Dijkstra dalam penelitian ini di tetapkan titik keberangkatan berada di Gedung Teknik Informatika dan titik tujuan di tetapkan di Raden The Barber. Berikut penjelasan proses pencarian rute terpendek dari Gedung Teknik Informatika ke Raden The Barber.



Gambar 4. Peta Google Earth Rute dari Gedung Teknik Informatika Menuju Raden The Barber



Gambar 5. Graf Berlabel dan Berbobot Dari Gedung Teknik Informatika Menuju Raden The Barber-Dijkstra

a. Iterasi 0

Pada iterasi ini titik awal adalah gedung Teknik Informatika yang di beri tanda T0 diambil sebagai titik permanen pertama.

b. Iterasi 1

Nilai yang mungkin berubah adalah pada titik-titik yang berhubungan langsung dengan titik T0 iterasi sebelumnya. Titik yang berhubungan langsung dengan titik T0 adalah T1. Dengan menggunakan rumus Algoritma Dijkstra berikut.

$$dist[x] \leftarrow \min \{ dist[x], dist[u] + c[u,x] \} \dots (1)$$

Dist[x] adalah nilai minimum dari nilai dist[x] sebelumnya dibandingkan dengan dist[u] yang merupakan titik permanen sebelumnya + c[u,x] yang merupakan jarak yang menghubungkan titik permanen sebelumnya ke titik selanjutnya. Dalam hal ini dist[T1] = 8 yang artinya belum terdefenisi, dist[T0] = 0 dan c[T0,T1] = 634.77, maka :

$$Dist [T1] = \min(\omega , dist[T0]+ c[T0, T1])$$

$$= \min(\omega , 0+ 634.77) = 634.77 \text{ m, karena hanya ada titik T1 yang terhubung langsung dengan titik T0 maka titik selanjutnya adalah T1 dengan jarak 634.77 m.}$$

c. Hasil perhitungan Iterasi 1 sampai Iterasi 16

Nilai yang diperoleh dari iterasi 1 sampai dengan iterasi 16 adalah mulai dari titik T0 pada iterasi 0 merupakan titik keberangkatan dan diambil sebagai titik permanen pertama yang bernilai 0. Pada iterasi 1 titik T1 merupakan satu-satunya titik yang terhubung langsung dengan titik T0, sehingga titik selanjutnya adalah titik T1. Pada iterasi 2 titik T2 merupakan satu-satunya titik yang terhubung langsung dengan titik T1, sehingga

titik selanjutnya adalah titik T2. Pada iterasi 3 titik T3 merupakan satu-satunya yang terhubung langsung dengan titik T2, sehingga titik selanjutnya adalah titik T3.

Pada iterasi 4 titik T4 merupakan satu-satunya yang terhubung langsung dengan titik T3, sehingga titik selanjutnya adalah titik T4. Pada iterasi 5 terdapat dua titik yang terhubung langsung dengan titik T4 yaitu titik T5 dan titik T6. Setelah melakukan perbandingan titik T6 memiliki nilai lebih kecil dari pada nilai titik T5, sehingga titik selanjutnya adalah titik T6. Pada iterasi 6 titik T7 merupakan satu-satunya yang terhubung langsung dengan titik T6, sehingga titik selanjutnya adalah titik T7.

Pada iterasi 7 terdapat dua titik yang terhubung langsung dengan titik T7 yaitu titik T5 dan titik T8. Setelah melakukan perbandingan titik T5 memiliki nilai lebih kecil dari pada nilai titik T8, sehingga titik selanjutnya adalah titik T5. Pada iterasi selanjutnya titik T4 satu-satunya yang terhubung langsung dengan titik T5, sehingga titik selanjutnya adalah T4. Hanya saja pada iterasi ini titik T8 sudah di proses sebelumnya dan nilai titik T8 lebih kecil apabila menggunakan jalur yang terhubung dengan titik T7. Sehingga titik T5 tidak dapat dilewati. Dan T8 menjadi titik selanjutnya setelah titik T7. Pada iterasi 8 titik T8 merupakan satu-satunya yang terhubung langsung dengan titik T7, sehingga titik selanjutnya adalah titik T8.

Pada iterasi 9 terdapat dua titik yang terhubung langsung dengan titik T8 yaitu titik T9 dan titik T13. Setelah dilakukan perbandingan titik T13 memiliki nilai lebih kecil dari titik T9, sehingga titik selanjutnya adalah titik T13. Pada iterasi 10 titik T14 merupakan satu-satunya yang terhubung langsung dengan titik T13, sehingga titik selanjutnya adalah titik T14. Pada iterasi 11 terdapat dua titik yang terhubung langsung dengan titik T14 yaitu titik T15 dan titik T16. Setelah dilakukan perbandingan titik T16 memiliki nilai lebih kecil dari titik T15, sehingga titik selanjutnya adalah titik T16.

Pada iterasi 12 titik T17 merupakan satu-satunya yang terhubung langsung dengan titik T16, sehingga titik selanjutnya adalah titik T17. Pada iterasi 13 terdapat dua titik yang terhubung langsung dengan titik T17 yaitu titik T15 dan titik T18. Setelah dilakukan perbandingan titik T18 memiliki nilai lebih kecil dari titik T15, sehingga titik selanjutnya adalah titik T18.

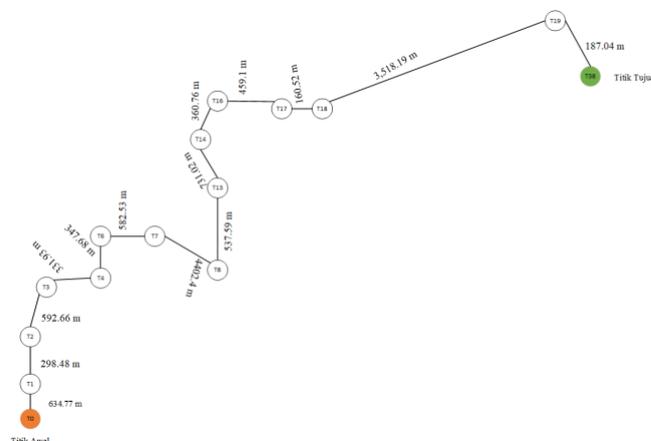
Pada iterasi 14 titik T19 merupakan satu-satunya yang terhubung langsung dengan titik T18, sehingga titik selanjutnya adalah titik T19. Pada iterasi 15 terdapat dua titik yang terhubung langsung dengan titik T19 yaitu titik T38 dan titik T41. Setelah dilakukan perbandingan titik T41 memiliki nilai lebih kecil dari titik T38, sehingga titik selanjutnya adalah titik T41.

Pada iterasi 16 titik T40 merupakan satu-satunya yang terhubung langsung dengan titik T41, sehingga titik selanjutnya adalah titik T40. Pada iterasi 17 titik T38 merupakan satu-satunya yang terhubung langsung dengan titik T38, sehingga titik selanjutnya adalah titik T38.

Hanya saja pada iterasi ini titik T38 sudah di proses sebelumnya pada iterasi 14 dan nilai titik T38 lebih kecil apabila menggunakan jalur yang terhubung dengan titik T19. Sehingga titik T41 tidak dapat dilewati. Dan T38 menjadi titik selanjutnya setelah titik T19. Pada bagian ini iterasi di berhentikan karena tujuan tercapai. Jadi, rute terpendek dari gedung Teknik Informatika yang ditandai dengan simbol T0 menuju Raden the Barber yang ditandai dengan simbol T38 dapat di tempuh dengan total jarak 12.762,8 meter dengan rincian sebagai berikut :

T0 → T1 → T2 → T3 → T4 → T6 → T7 → T8 → T13 → T14 → T16 → T17 → T18 → T19 → T38

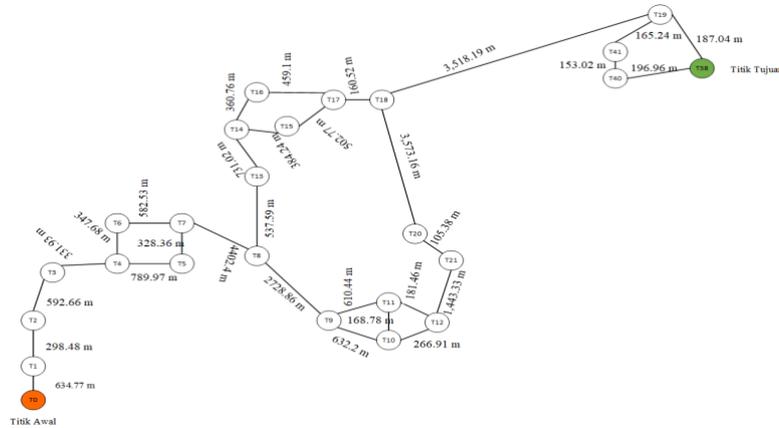
Adapun rute terpendek Dijkstra dalam bentuk graf ditampilkan pada Gambar 6.



Gambar 6. Rute Terpendek Dijkstra Dalam Bentuk Graf

3.2 Analisis Hasil Perhitungan Algoritma *Bellman-Ford*

Pada perhitungan Algoritma *Bellman-Ford* dalam penelitian ini ditetapkan titik keberangkatan berada di Gedung Teknik Informatika dan titik tujuan di tetapkan di Raden The Barber. Berikut penjelasan proses pencarian rute terpendek dari Gedung Teknik Informatika ke Raden The Barber pada Gambar 7.



Gambar 7. Graf Berlabel dan Berbobot Dari Gedung Teknik Informatika Menuju Raden The Barber-*Bellman-Ford*

- a. Langkah 1 : Buat verteks awal = 0 dan verteks lainnya dengan nilai tak terhingga.
- b. Langkah 2 : Perhitungan *Bellman-Ford*-Langkah 2 seperti pada Tabel 1.

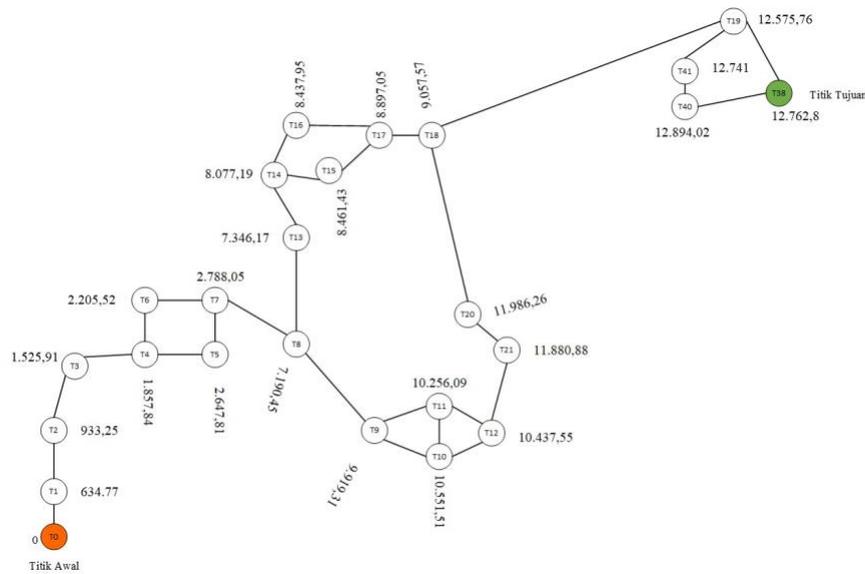
Tabel 1. Perhitungan *Bellman-Ford*-Langkah 2

No	Verteks	Melewati Verteks	Total Jarak (m)
1	T0	-	0
2	T1	T0	634,77
3	T2	T1	933,25
4	T3	T2	1.525,91
.	.	.	.
.	.	.	.
.	.	.	.
24	T41	T19	13.762,38
25	T40	T41	13.915,4
26	T38	T40	14.112,36

- c. Langkah 3 : Hitung Kembali semua *verteks* yang belum terlewati, hasilnya ditampilkan pada Tabel 2.

Tabel 2. Perhitungan *Bellman-Ford*-Langkah 3

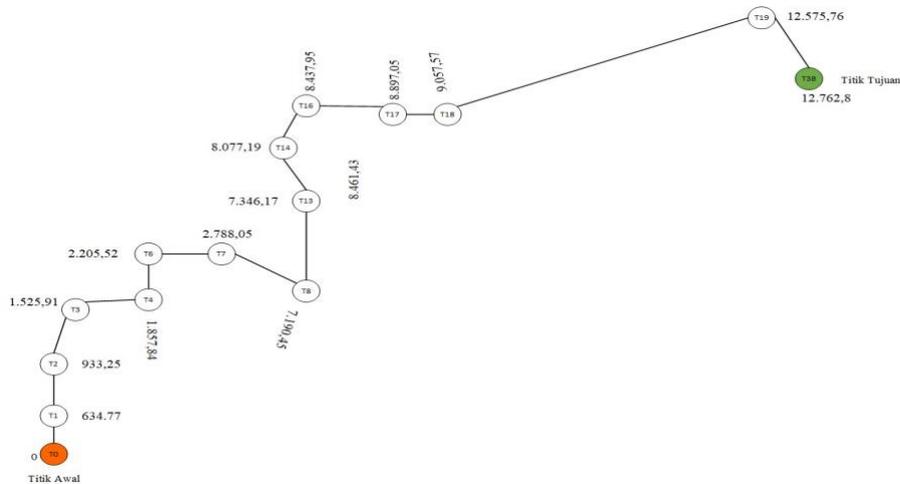
No	Verteks	Melewati Verteks	Total Jarak (m)
1	T0	-	0
2	T1	T0	634,77
3	T2	T1	933,25
4	T3	T2	1.525,91
5	T4	T3	1.857,84
6	T6	T4	2.205,52
7	T7	T5	2.976,17
.	.	.	.
.	.	.	.
.	.	.	.
27	T38	T19	12.762,8
28	T41	T19	12.741
29	T40	T41	12.894,02
30	T38	T40	14.112,36



Gambar 8. Hasil Langkah 3 Algoritma *Bellman-Ford*

Jadi, rute terpendek dari gedung Teknik Informatika yang ditandai dengan simbol T0 menuju Raden the Barber yang ditandai dengan simbol T38 dapat di tempuh dengan total jarak 12.762,8 meter dengan rincian sebagai berikut:

T0 → T1 → T2 → T3 → T4 → T6 → T7 → T8 → T13 → T14 → T16 → T17 → T18 → T19 → T38



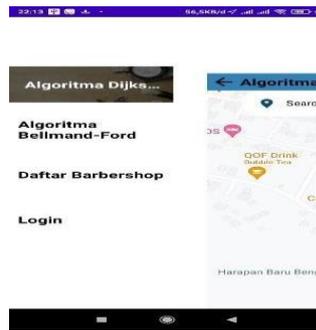
Gambar 9. Rute Terpendek *Bellmad-Ford* Dalam Bentuk Graph

Dari hasil pencarian rute terdekat pada masing-masing algoritma yang telah dilakukan, maka dapat diperoleh perbandingan dari keduanya. Perbandingan dari pengujian pemilihan rute, didapatkan hasil yang sama dalam melakukan pemilihan jalur. Namun pada perbandingan waktu eksekusi program, kedua algoritma menunjukkan bahwa Algoritma Dijkstra bisa dieksekusi lebih cepat daripada Algoritma *Bellman-Ford*. Waktu eksekusi program Dijkstra bisa di eksekusi program Dijkstra ini membutuhkan waktu sekitar 0.941 detik hingga 1.409 detik. Waktu eksekusi program *Bellman-Ford* dalam mencari rute dikisaran -0.993 hingga 2.280 detik.

3.3 Implementasi Antar Muka Sistem

Penelitian ini mengimplementasikan algoritma Dijkstra dan *Bellman-Ford* menggunakan platform berbasis Android dalam pembuatan Sistem Informasi Geografis Pemetaan *Barbershop* di Kota Lhokseumawe.

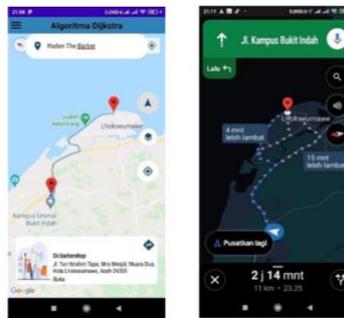
a. Halaman Pengguna



Gambar 10. Halaman menu pengguna

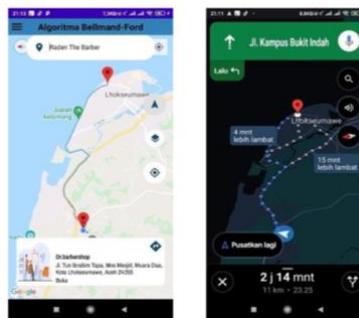
Terdapat tiga menu utama yang dapat digunakan oleh pengguna aplikasi yaitu Pilihan untuk menggunakan Algoritma Dijkstra atau Algoritma *Bellman-Ford*, *Barbershop* dan juga login.

b. Tampilan Rute Terdekat dengan Algoritma Dijkstra dan Navigasi Perjalanan



Gambar 11. Tampilan Rute Terdekat-Dijkstra dan Navigasi Perjalanan

c. Tampilan Rute Terdekat dengan *Bellman-Ford* dan Navigasi Perjalanan



Gambar 12. Tampilan Rute Terdekat- *Bellman-Ford* dan Navigasi Perjalanan

Pada halaman ini akan ditampilkan rute terdekat setelah pengguna memilih lokasi *Barbershop* yang ingin di tuju. Dalam aplikasi pengguna juga dapat meminta bantuan Google Maps untuk menampilkan navigasi perjalanan untuk mengarahkan pengguna menuju lokasi yang diinginkan.

4. Kesimpulan

Berdasarkan hasil analisis perbandingan algoritma dijkstra dan *bellman-ford* pada sistem informasi geografis pemetaan *Barbershop* di kota Lhokseumawe diketahui bahwa sistem yang di buat Berbasis Android dapat membantu pengguna dalam mencari *Barbershop* yang ada di Kota Lhokseumawe dan memberi informasi *Barbershop* dan hasil perhitungan jarak antara algoritma *Bellman-Ford* dan Dijkstra identik dalam hal menghitung rute terpendek dan memilih rute atau jalur yang harus ditempuh namun kedua algoritma memberikan jawaban yang sama. Adapun hasil pengujian terhadap rute terpendek pada algoritma Dijkstra lebih

sedikit dibandingkan pada algoritma *Bellman-Ford*. Waktu eksekusi program Dijkstra bisa di membutuhkan waktu sekitar 0.941 detik hingga 1.409 detik, sedangkan waktu eksekusi program *Bellman-Ford* dalam mencari rute dikisaran -0.993 hingga 2.280 detik.

Referensi

- [1] B. A. Herlambang, S. O. Pradana, R. R. Waliyansyah. "Sistem Informasi Geografis Pencarian Rute Optimum Lokasi Obyek Wisata Kota Semarang Menggunakan Algoritma Dijkstra." *Journal of Applied Science and Technology*, 2(01), 16-25. 2022
- [2] H. Paunsyah, H. Mubarak, R. N. Shofa. "Penentuan Jalur Terpendek menggunakan Google Maps API pada Sistem Informasi Geografis (SIG) Panti Sosial di Kota Tasikmalaya." *Innovation in Research of Informatics (INNOVATICS)*, 1(1). 2019.
- [3] N. Hasdyna, and R. K. Dinata, "Analisis Matthew Correlation Coefficient pada K-Nearest Neighbor dalam Klasifikasi Ikan Hias." *INFORMAL: Informatics Journal*, 5(2), 57-64, 2020.
- [4] A. L. Turnip. "Sistem Informasi Geografis Pendorong Darah Tetap di Bandar Lampung Menggunakan Algoritma Dijkstra". *Journal of Engineering, Computer Science and Information Technology (JECSIT)*, 1(1). 2021
- [5] R. Pramudita, N. Safitri. "Algoritma Bellman-Ford Untuk Menentukan Jalur Tercepat Dalam Sistem Informasi Geografis." *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 6(2), 105-114. 2018.
- [6] S. Retno, and N. Hasdyna, "Profile Matching in Government Scholarship Acceptance System for Student in Aceh Utara," *Journal of Informatics and Telecommunication Engineering*, 5(2), 268-275, 2022.
- [7] V. L. Pratama, D. A. Dermawan. "Sistem Informasi Geografis Pencarian Rute Terdekat Bengkel Motor di Kota Surabaya Menggunakan Algoritma Bellman-Ford." *Journal of Informatics and Computer Science (JINACS)*, 3(04), 580-599. 2022
- [8] N. Hasdyna, I. Amal, "Algoritma Brute Force dalam Sistem Informasi Lowongan Kerja Berbasis Web Di Kota Lhokseumawe," *Jurnal Elektro Dan Informatika*, 2(1), 6-11, 2021.
- [9] M. W. Ridho, A. Usman. "Aplikasi Pencarian Jadwal Dokter dan Fasilitator Rumah Sakit di Kota Medan Menggunakan Algoritma Bellman Ford Berbasis Android." In *Seminar Nasional Teknologi Informasi & Komunikasi* (Vol. 1, No. 1, pp. 534-540). 2020.
- [10] M.P. Dinanda, A. Muzakir, I. Effendy. "Perbandingan ALgoritma Bellman-ford dengan Algoritma Branch and Bounda dalam pencarian Jalur terdekat." In *Bina Darma Conference on Computer Science (BDCCS)* (Vol. 1, No. 2, pp. 344-352). 2019.