Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

journal homepage: http://journal.lembagakita.org/index.php/jtik

Implementasi Algoritma *Dijkstra* pada Aplikasi Go-Tahu dengan Pencarian Rute Terpendek ke Pabrik Tahu

Putriana Mayang Sari [1], Fauziah [2], Aris Gunaryati [3]

[1,2,3] *Program Studi Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional.*

a r t i c l e i n f o

Article history:
Received 28 October 2020
Received in revised form
30 November 2020
Accepted 3 December 2020
Available *online* April 2021

DOI:
https://doi.org/10.35870/jtik.v5i2.210

Keywords:
Dijkstra's Algorithm, Android, Flutter, Shortest Line, Tofu Factory.

Kata Kunci:
Algoritma Dijkstra, Android, Flutter, Jalur Terpendek, Pabrik Tahu.

a b s t r a c t

Currently, the food business sector is increasing, one of which is tofu producers in South Tangerang. Many people who want to buy tofu with good quality but do not know the closest distance to the factory is located. In this research, we will use Dijkstra's Algorithm which is applied to the Android software to determine the shortest distance from one point to the tofu factory which is the chosen destination. Using the Dijkstra algorithm, an application will be designed, namely a mobile-based Go-Tofu for finding the closest route to the tofu factory. The route search process is carried out with a graph that has a weight and an area that is connected to a predetermined route. In the application test, it produces the shortest route from a house to the tofu factory with the smallest total weight of 11 kilometers based on the test results in the study.

a b s t r a k

Pada saat ini sektor usaha pangan semakin banyak salah satunya yaitu produsen tahu yang berada di Tangerang Selatan. Banyak orang yang ingin membeli tahu dengan kualitas yang baik tetapi tidak mengetahui jarak terdekat menuju pabrik tersebut berada. Pada penelitian ini, akan menggunakan Algoritma Dijkstra yang diterapkan pada perangkat lunak Android untuk menentukan jarak terdekat dari satu titik menuju pabrik tahu yang menjadi tujuan pilihan. Dengan menggunakan algoritma Dijkstra akan dirancang sebuah aplikasi yaitu Go-Tahu berbasis mobile untuk pencarian rute terdekat ke pabrik tahu. Proses pencarian rute dilakukan dengan bentuk graf yang memiliki bobot dan juga penitikan area yang saling terhubung dengan rute yang telah ditentukan. Pada pengujian aplikasi menghasilkan rute terpendek dari suatu rumah menuju pabrik tahu dengan total bobot terkecil yaitu 11 kilometer berdasarkan hasil pengujian pada penelitian.

*Corresponding author. Email: putrianamay14@gmail.com [1].
© E-ISSN: 2580-1643.

1. Latar Belakang

Masyarakat Indonesia umumnya sangat menggemari tahu sebagai makanan pendamping, sehingga masyarakat kerap kali ingin membeli tahu langsung di pabriknya demi mendapatkan kualitas tahu yang baru. Tetapi pabrik tahu yang jarang diketahui lokasinya ini menjadi sebuah permasalahan.

Maka dengan adanya permasalahan tersebut penulis menawarkan sebuah solusi dengan membuat sebuah aplikasi berbasis *mobile* dengan nama Go-Tahu, yang berfungsi untuk mencari rute terdekat menuju ke sebuah pabrik tahu dari suatu titik tertentu. Teori graph merupakan teori yang memiliki bagian salah satunya yaitu jalur lintas terpendek, ketika diberikan graph berbobot, permasalahan jarak terpendek yaitu bagaimana mendapatkan jalur pada graph yang dapat meminimalkan hasil jumlah bobot sisi pembentuk jalur tersebut [1].

Aplikasi Go-Tahu akan dirancang menggunakan Algoritma *Dijkstra* yang merupakan salah satu algoritma yang dapat digunakan untuk menyelesaikan sebuah permasalahan jarak terpendek (*shortest path problem*) pada sebuah graf yang terarah (directed graph) [2].

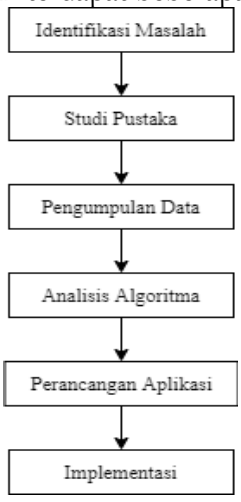
Banyak penelitian terdahulu membahas tentang algoritma untuk pencarian rute menuju titik tertentu dan menghasilkan rute terpendek salah satunya penelitian yang dilakukan oleh M.N Parapat dkk, yang membahas pencarian rute terpendek menggunakan algoritma *Dijkstra* untuk jasa pengiriman barang, dimana algoritma ini yang cukup popular dan sering digunakan karena dapat menyelesaikan pencarian jalur terpendek dari satu ke semua simpul yang ada pada suatu graf bearah dengan bobot nilai [3].

Penelitian serupa juga dilakukan oleh Fitria dalam menentukan lintasan terdekat menuju tempat pariwisata yang bertujuan untuk membantu instansi pemerintah, para pengguna jalan, perusahaan yang bergerak dibidang pariwisata, traveling, salesman, dan lain sebagainya terutama seseorang yang sangat membutuhkan informasi dalam mencari lintasan atau rute terdekat [4].

Penelitian yang dilakukan oleh R. Dwi yaitu membuat sebuah aplikasi pencarian rute terpendek pada Bus Trans Semarang dengan menerapkan metode *Dijkstra* [5]. Algoritma *Dijkstra* juga dapat di implementasikan dalam pencarian rute untuk menemukan rute terpendek menuju lokasi SPBU di Kota Padang bagi pengguna kendaraan bermotor [6]. Dari penelitian terdahulu tersebut maka penulis akan mengembangkan penelitian kali ini dengan membuat sebuah aplikasi berbasis *mobile* beranama Go-Tahu dengan menggunakan algoritma *Dijkstra* dan menggunakan Google Maps yang bertujuan untuk memudahkan dalam pencarian rute terpendek menuju pabrik tahu.

2. Metode Penelitian

Pada penelitian ini terdapat beberapa langkah, yaitu:



Gambar 1. Desain Penelitian

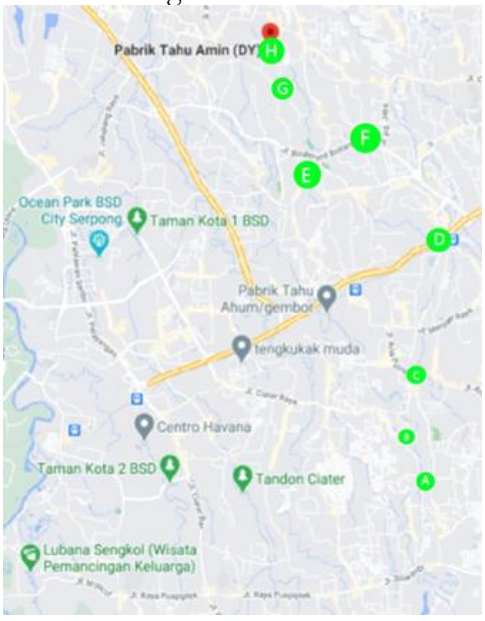
Berdasarkan gambar 1 maka dapat dijabarkan langkah-langkah dari desain penelitian yaitu sebagai berikut:

1) Identifikasi Masalah
   Pada tahapan ini merupakan tahap awal dari penelitian yaitu identifikasi masalah mengenai pencarian jalur terpendek menuju Pabrik Tahu di Tangerang Selatan.
2) Studi Pustaka
   Tahapan kedua adalah studi pustaka merupakan tahap memahami teori-teori dasar dalam penelitian melalui beberapa sumber terpercaya antara lain yaitu buku, jurnal penelitian terdahalu, dan beberapa sumber terpercaya lainnya yang sejenis dengan pembahasan penelitian.

3) Pengumpulan Data
   Tahapan ketiga adalah tahap pengumpulan data terdiri dari dua cara yaitu melakukan observasi dan dokumentasi data.
4) Analisis algoritma
   Tahapan keempat adalah analisis yaitu melalukan perhitungan secara manual pada algoritma dengan rute yang ditentukan. Sehingga, hasil analisis dapat digunakan sebagai acuan dalam mengimplentasikan algoritma di sistem yang akan dirancang.
5) Perancangan Aplikasi
   Tahapan kelima merupakan proses pembuatan tampilan aplikasi, alur aplikasi dan pembuatan fitur-fitur dari aplikasi penelitian ini.
6) Implementasi
   Tahapan terakhir yaitu impelentasi algoritma *Dijkstra* baik dalam perhitungan manual dan penerapan pada aplikasi Go-Tahu untuk pencarian rute terpendek.

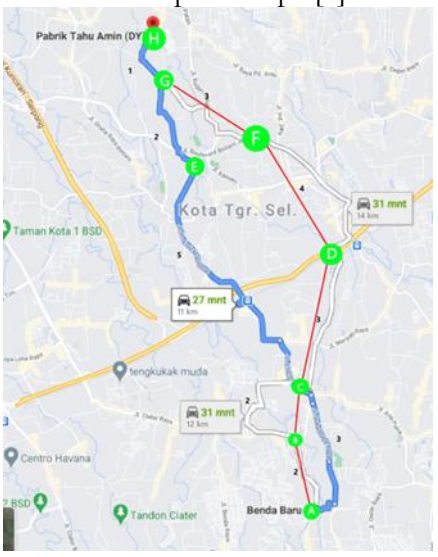
Pengumpulan Data

Pada proses pengumpulan data dalam penelitian ini dilakukan dengan cara mengumpulkan data jalan yang bersumber dari Google Maps. Data tersebut akan digunakan untuk menggambarkan lokasi dengan jalur terdekat ke pabrik tahu. Rute yang digunakan dalam penelitian ini yaitu dari Benda Baru menuju Pabrik Tahu Amin (DY). Titik awal ditandai dengan simpul A dan simpul akhir ditandai dengan simpul H sesuai dengan Gambar 2.



Gambar 2. Map dengan titik simpul

Pada gambar 2 simpul ditandai dengan titik hijau pada setiap persimpangan jalan. Setelah membuat titik simpul maka langkah selanjutnya adalah pembentukan graf yang terdapat *vertex* yaitu merupakan bentang panjang jalan antar simpul dan kemudian menghitung jarak antar simpul yang digunakan sebagai bobot untuk melakukan perhitungan manual pada algoritma sesuai Gambar 3. Terdapat dua macam proses dalam perhitungan jarak terpendek yaitu pemberian label dan pemeriksaan node pada maps [7].



Gambar 3. Graf Dengan Nilai Bobot

Pada gambar 3 merupakan pembentukan graf yang didapat dari google map, graf tersebut memiliki 8 simpul dan 9 garis (*vertex*). Selanjutnya, dilakukan perhitungan antar simpul sehingga menghasilkan jarak-jarak seperti pada gambar 3. Jarak antar simpul digunakan sebagai bobot untuk proses perhitungan.

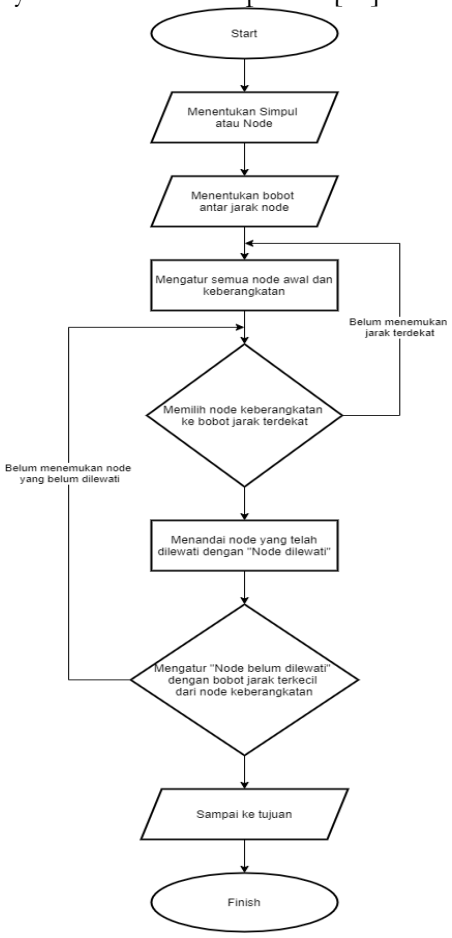
Data yang akan digunakan kedalam aplikasi adalah menggunkan Google Maps API merupakan sebuah layanan dari Google yang diberikan kepada penggunanya agar dapat mengakses Google Map untuk kebutuhan pengembangan aplikasi. Adapun fitur dari Google Map API adalah pengguna dapat menambahkan konten, mengizinkan pengguna menggunakan variabel yang berisi dari Google Map dan memanipulasinya untuk kepentingan penelitian [1].

Analisis Algoritma

Algoritma yang digunakan pada penelitian ini adalah *Dijkstra*. Algoritma djikstra bekerja dengan mencari sebuah bobot dengan nilai terkecil dari suatu graf

berbobot sehingga didapatkan jarak terpendek dari dua atau lebih titik dari suatu graf dengan hasil rute terpendek [8].

Algoritma *Dijkstra* merupakan sebuah algoritma untuk menentukan jarak terpendek dari satu *vertex* ke *vertex* (jarak antar *vertex* adalah nilai bobot dari setiap edge pada graph) yang lainnya pada suatu graph berbobot [9]. Dan algoritma *Dijkstra* merupakan salah satu algoritma dalam pencarian rute terpendek dari sebuah simpul ke sinpul lain dalam graf berbobot yang hanya memiliki bobot positif [10].



Gambar 4. Flowchart Algoritma *Dijkstra*

Adapun tahapan algoritma *Dijkstra* ini dapat dilakukan dengan tahapan berikut:

1) Langkah pertama yaitu menentukan titik awal yang akan menjadi node pertama atau node awal, kemudian menentukan bobot jarak pada node awal ke node terdekat satu per satu.
2) Menentukan bobot jarak dari setiap node ke node lainnya, kemudian menetapkan nilai 0 pada node pertama dan nilai tak terhingga pada node lainnya.
3) Mengatur semua node yang belum terlewati dan node awal dengan tanda "node keberangkatan"
4) Pada node keberangkatan, memperhitungkan node lainnya yang paling dekat dengan keberangkatan yang belum terlewati dan memperhitungkan jarak dari awal keberangkatan.
5) Dicek jika jaraknya lebih kecil dari jarak sebelumnya (yang telah dihitung sebelumnnya) menghapus data lama dan menyimpan ulang data jarak sebelumnya dengan data jarak yang lebih pendek.
6) Setelah selesai menghitung dan mempertimbangkan setiap jarak pada node terdekat, kemudian menandai node yang sudah dilalui dengan "Node dilewati". Node yang sudah dilewati tidak akan dicek kembali, dan akan menyimpan bobot jarak yang terkecil.
7) Mengatur "Node belum dilewati", dengan bobot jarak paling kecil (dari node keberangkatan) sebagai "Node Keberangkatan" untuk keberangkatan selanjutnya dan ulangi langkah.

Pada tahap perhitungan algoritma *Dijkstra*, terdapat 8 langkah hingga mendapatkan hasil rute terpendek. Proses perhitungan algoritma *Dijkstra* pada Tabel 1 adalah sebagai berikut:

1) Pada langkah pertama memilih node A dengan bobot nilai terkecilnya adalah 0, yang berarti node A bernilai 0A dikarenakan berada di node A.
2) Selanjutnya, node A terhubung ke node B dan C dengan nilai *vertex* masing-masing 2 dan 3, maka menjadi 2A dan 3A.
3) Node D sampai dengan H tidak saling terhubung dengan node A, maka ditandai dengan simbol infinity (tidak terhingga).
4) Pada langkah kedua, memilih bobot terkecil yang ada pada langkah pertama yaitu 2A berada di node B.
5) Setelah memilih bobot terkecil, langkah selanjutnya menghitung node yang terhubung dengan node B dan menambahkan bobotnya dengan bobot terkecil terpilih.
6) Node yang terhubung dengan B yaitu node C bernilai 2, bobot tersebut ditambahkan dengan bobot terkecil yaitu 2+2=4 hasil dari penjumlahan tersebut dibandingkan dengan perhitungan yang sudah ada yaitu 3A. Sesuai dengan ketentuan *Dijkstra* jika bobot setelah dijumlahkan lebih besar dari bobot sebelumnya

maka akan dipilih bobot sebelumnya yang lebih kecil, begitupun berlaku sebaliknya.

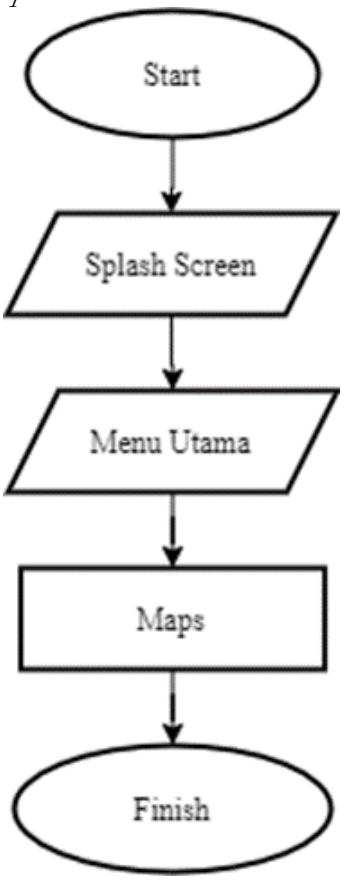
7) Untuk langkah ketiga sampai ke-delapan mengulangi langkah yang sama.
Pada langkah ke-delapan semua node telah terlewati, dan menemukan hasil jalur terpendek yaitu A-C-E-G-H dengan bobot nilai 11 atau jika pada jarak asli yaitu 11 kilometer.

Pada gambar 5 merupakan tampilan flowchart dari aplikasi GoTahu, pada tahap pertama menampilkan halaman splash screen selama tiga detik, lalu menuju halaman utama yang terdapat tombol Tentang Aplikasi menjelaskan tentang informasi mengenai aplikasi dan juga tombol Maps yang akan menuju halaman maps yang terdapat peta menuju pabrik tahu yang ada di Tangerang Selatan.

Tabel 1. Perhitungan Algoritma Djikstra

Langkah	Pilih	Terkecil	A	B	C	D	E	F	G	H
1	A	0	0A	2A	3A	∞	∞	∞	∞	∞
2	B	2A	0A	**2A**	3A	∞	∞	∞	∞	∞
3	C	3A	0A	2A	**3A**	6C	8C	∞	∞	∞
4	D	6C	0A	2A	3A	**6C**	8C	10D	∞	∞
5	E	8C	0A	2A	3A	6C	**8C**	10D	10E	∞
6	F	10D	0A	2A	3A	6C	8C	**10D**	10E	∞
7	G	10E	0A	2A	3A	6C	8C	10D	**10E**	11G
8	H	11G	0A	2A	3A	6C	8C	10D	10E	**11G**
	stop									

Perancangan Aplikasi



Gambar 5. Flowchart Aplikasi

3. Hasil dan Pembahasan

Pada tahapan ini akan membahas mengenai hasil penelitian berupa analisis kebutuhan sistem, tampilan aplikasi, pengujian, serta perbandingan dengan perhitungan algoritma *Greedy*.

Analisis Kebutuhan Sistem

Pada tahapan ini bertujuan untuk mengidentifikasikan kebutuhan yang diperlukan dalam merancang sistem aplikasi yang akan dibuat pada penelitian ini. Adapun kebutuhan sistem yang diperlukan yaitu berupa perangkat lunak (*software*) dan perangkat keras (*hardware*). Berikut adalah kebutuhan *software* dan *hardware* dalam penelitian ini:

Tabel 1. Kebutuhan Perangkat Keras

Perangkat	Spesifikasi
Processor	Intel Core i3-5000 series atau lebih tinggi
Ram	6GB atau lebih tinggi
Harddisk	250GB atau lebih tinggi
Sistem Operasi	Windows 7/8/10
Smartphone	Android

Tabel 2 adalah kebutuhan perangkat keras yang

digunakan untuk membuat penelitian dan aplikasi android penelitian.

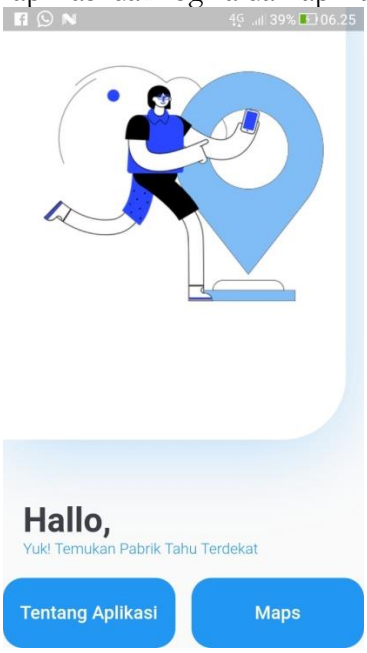
Tabel 2. Kebutuhan Perangkat Lunak

Perangkat	Penggunaan
Android Studio	IDE sistem
Flutter	Framework sistem
Dart	Bahasa sistem

Tabel 3 adalah kebutuhan perangkat lunak yang digunakan untuk mengerjakan penelitian dan membuat aplikasi android penelitian. Kebutuhan dari perangkat lunak yaitu aplikasi IDE Android Studio agar dapat menggunakan framework flutter dan bahasa dart untuk membuat aplikasi android.

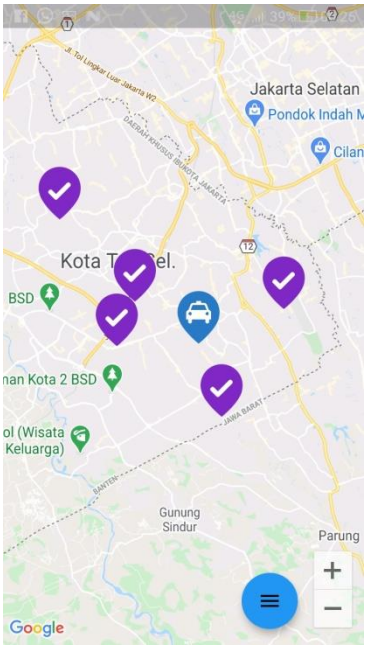
Tampilan Aplikasi

Dalam membuat rancangan sistem aplikasi, digunakan framework flutter untuk membuat desain, kode sistem aplikasi dan logika dari aplikasi.



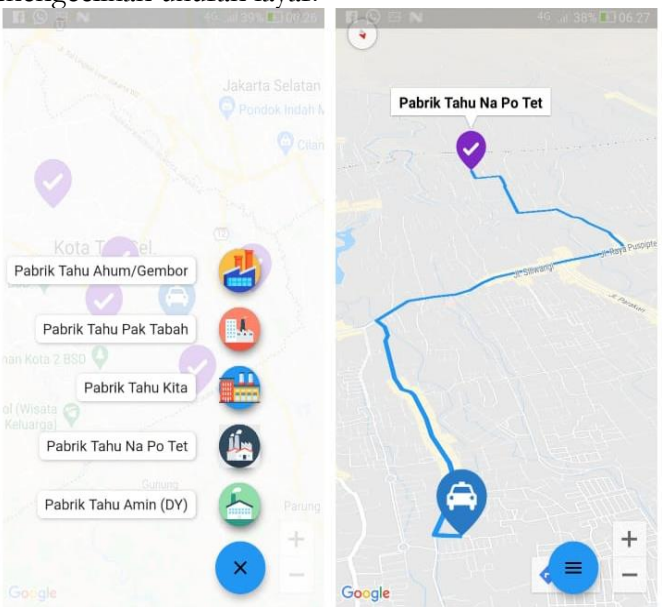
Gambar 6. Tampilan Halaman Menu Utama

Pada gambar 6 adalah tampilan halaman menu utama dari aplikasi Go-Tahu yang terdiri dari 2 buah tombol yaitu tombol Tentang Aplikasi yang akan menampilkan informasi singkat mengenai aplikasi Go-Tahu dan tombol Maps yang berfungsi untuk menampilkan halaman map untuk pencarian jarak terdekat menuju pabrik tahu di Tangerang Selatan.



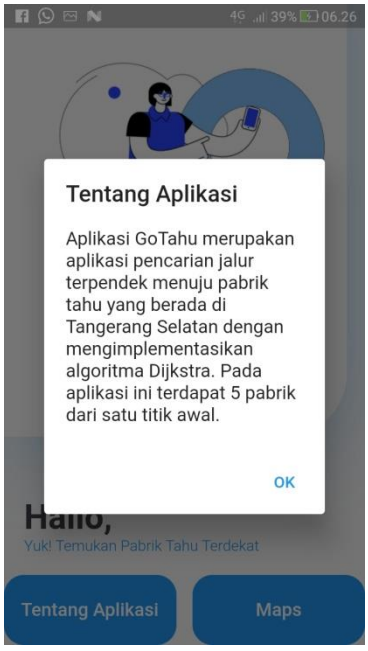
Gambar 7. Tampilan Halaman Maps

Pada gambar 7 adalah tampilan halaman Maps dimana terdapat lima titik ungu dengan icon ceklis dan satu titik biru dengan icon mobil, titik ungu menandakan tujuan atau pabrik yang berada di Tangerang Selatan dan titik biru merupakan titik awal dari pencarian rute. Kemudian terdapat 3 tombol yang memiliki fungsi berbeda yaitu tombol biru dengan tiga garis untuk menamoilkan daftar pabrik, tombol tambah untuk memperbesar ukuran layar dan tombol minus untuk mengecilkan ukuran layar.



Gambar 8. Tampilan Halaman Maps

Pada gambar 8 adalah tampilan halaman maps yang dimana pengguna dapat melihat rute terpendek dari daftar pabrik tersebut. Dan tampilan rute terdekat menuju pabrik tahu dari titik awal.



Gambar 9. Tampilan Tentang Aplikasi

Pada gambar 9 adalah tampilan menu tentang aplikasi yang dimana terdapat informasi tentang aplikasi dan juga terdapat 1 buah tombol OK untuk kembali ke halaman menu utama.

Pengujian

Pengujian pada penelitian ini dilakukan melalui beberapa tahap, diantaranya:

1) Pengujian *Device*
   Pengujian *device* dilakukan pada beberapa *device* android yang memiliki sistem operasi yang berbeda agar mengetahui keberhasilan aplikasi pada setiap *device* android yang berbeda. Maka dari itu, pengujian ini dilakukan terhadap 5 *device*.

Tabel 3. Pengujian Device

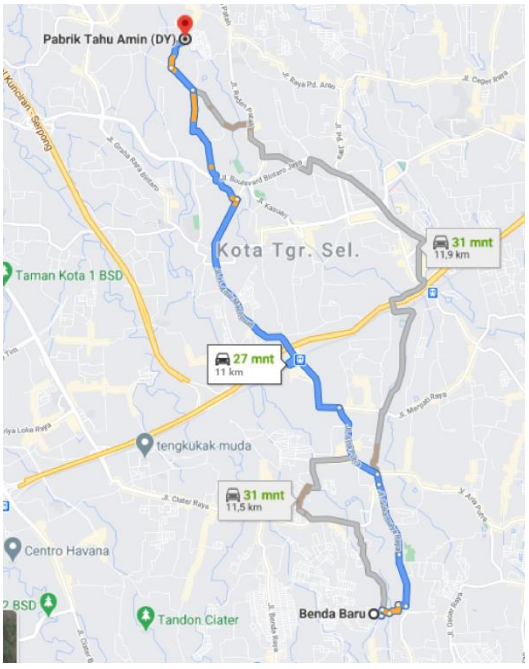
Perangkat	Sistem Operasi	Hasil
1	Android Q	Berhasil
2	Android Pie	Berhasil
3	Android Nougat	Berhasil
4	Android Marshmallow	Berhasil
5	Android Lollipop	Berhasil

2) Pengujian aplikasi
   Pengujian aplikasi ini dilakukan menggunakan metode blackbox *testing* untuk mengetahui fungsionalitas aplikasi tercapai atau tidak.

Tabel 4. Pengujian Aplikasi

Pertanyaan	Pengujian	Hasil
Gambar	Gambar tertampilkan dengan jelas	Berhasil
Tombol	Tombol responsif	Berhasil
Teks	Teks tertampilkan dengan jelas	Berhasil
Rute Terpendek	Menemukan rute terpendek	Berhasil

3) Pengujian Akurasi
   Pengujian akurasi ini didapatkan dari data sebenarnya di google maps yang merupakan rute terdekat sehingga sesuai dengan hasil penelitian yaitu 11km.



Gambar 10. Pengujian Akurasi Dengan Google Maps

Perbandingan

Untuk membuktikan hasil dari algoritma *Dijkstra* maka dilakukan perbandingan dengan algoritma Greedy dalam mencari rute terpendek menuju Pabrik Tahu di Tangerang Selatan. Pada algoritma Greedy merupakan algoritma dengan proses langkah demi

langkah yang dimana pada setiap langkah tersebut akan dipilih keputusan yang paling optimal [9]. Hasil perbandingan sesuai dengan Tabel 3.

Tabel 5. Hasil Perhitungan Algoritma Greedy

Langkah	A	B	C	D	E	F	G	H	Node Yang Dilalui
A		2							B
BC			2						C
CD				3					D
DF						4			F
FG							3		G
GH								1	H

Hasil rute yang didapatkan pada perhitungan algoritma Greedy adalah A-B-C-D-F-G-H, dengan total bobot 15 yang berarti lebih besar dari pada yang dihasilkan algoritma *Dijkstra*, dikarenakan dalam pencarian rute terpendek pada algoritma Greedy hanya menghitung local maximum, tidak menghitung keseluruhan node dan bobot yang ada sampai proses akhir, sesuai dengan prinsipnya "*take what you can get now!*". Sehingga dapat menyebabkan solusi tidak optimal atau bukan rute yang terpendek.

4. Kesimpulan dan Saran

Berdasarkan pembahasan dan pengujian dalam menentukan rute terpendek menggunakan algoritma *Dijkstra* dan perbandingan menggunakan algoritma Greedy pada aplikasi Go-Tahu, terdapat beberapa kesimpulan algoritma yang digunakan didalam aplikasi yaitu algoritma *Dijkstra* dapat menemukan rute terpendek menuju Pabrik Tahu di Tangerang Selatan. Algoritma Djikstra yang diterapkan kedalam aplikasi Go-tahu menghasilkan rute yang efektif karena merupakan jalan utama yang dapat dilalui kendaraan roda empat. Berdasarkan pengujian yang telah dilakukan Algoritma *Dijkstra* menghasilkan rute terpendek dengan total bobot 11 atau jika dikonversi kedalam jarak sebenarnya yaitu satuan kilometer adalah 11km.

5. Daftar Pustaka

[1] Prasetyo, B.I.A. and Maslan, A., 2020. Analisis Perbandingan Pada Algoritma Bellman Ford Dan Dijkstra Pada Google Map. Khazanah Ilmu Berazam, 3(2), pp.337-349.

[2] Cantona, A., Fauziah, F. and Winarsih, W., 2020. Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta. Jurnal Teknologi dan Manajemen Informatika, 6(1), pp.27-34.

[3] Parapat, M.N., Kusbianto, D. and Rahmad, C., 2017. Rancang Bangun Aplikasi Pencarian Rute Terpendek Jasa Kiriman Barang Berbasis Mobile Dengan Metode Algoritma Dijkstra. Jurnal Informatika Polinema, 3(3), pp.15-15.

[4] Triansyah, A., 2013. Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan. JSI: Jurnal Sistem Informasi (E-Journal), 5(2).

[5] Ardana, D. and Saputra, R., 2016, October. Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang. In Seminar Nasional Ilmu Komputer (SNIK 2016) (pp. 299-306).

[6] Junanda, B., Kurniadi, D. and Huda, Y., 2018. Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Pada Sistem Informasi Geografis Pemetaan Stasiun Pengisian Bahan Bakar Umum. VoteTEKNIKA: Jurnal Vocational Teknik Elektronika dan Informatika, 4(1).

[7] Serdano, A., Zarlis, M. and Hartama, D., 2019, August. Perbandingan Algoritma Dijkstra dan Bellman-Ford Dalam Pencarian Jarak Terpendek Pada SPBU. In Seminar Nasional Sains dan Teknologi Informasi (SENSASI) (Vol. 2, No. 1).

[8] Hamdi, S. and Prihandoko, P., 2018. Analisis Algoritma Dijktra dan Algoritma Bellman-Ford Sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran (Studi Kasus: Kecamatan Praya Kota). Energy, 8(1), pp.26-32.

[9] Harahap, M.K. and Khairina, N., 2017. Pencarian Jalur Terpendek dengan Algoritma Dijkstra. SinkrOn, 2(2), pp.18-23.

[10] Aulia, R., Syahputra, E.R. and Dafitri, H., 2015. Sistem Pencarian Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra Berbasis Android (Studi Kasus: rumah Sakit di Kota Medan). In Medan. Prosiding: SNASTIKOM (Seminar nasional teknologi informasi & komunikasi), jilid (Vol. 1, pp. 150-155).