MIND (Multimedia Artificial Intelligent Networking Database) Journal

MIND Journal | ISSN (p): 2528-0015 | ISSN (e): 2528-0902 | Vol. 5 | No. 2 | Halaman 108 - 120
DOI : https://doi.org/10.26760/mindjournal.v5i2.108 Desember 2020

Algoritma Dijkstra untuk Penentuan Jarak Tempuh Terpendek Pengantaran Katering Pabrik

THETA DINNARWATY PUTRI, WINARNO SUGENG, EKA SAFITRI

Program Studi Informatika Institut Teknologi Nasional Bandung
Email: theta@itenas.ac.id

Received 25 Oktober 2020 **|** *Revised* 11 November 2020 **|** *Accepted* 30 November 2020

ABSTRAK

Algoritma Dijkstra digunakan untuk menemukan jalur terpendek antara titik pada graf dan persamaan Haversine digunakan untuk mengukur jarak dari lokasi awal menuju lima lokasi tujuan yang mana lokasi tersebut merupakan pabrik yang berada di kota Cikarang dan lokasinya ada di sekitaran penyedia rumah catering. Perhitungan dilakukan setelah sistem mendapatkan koordinat latitude dan longitude pengguna dan lokasi pabrik yang dituju. Pada penelitian ini, lokasi pengguna dan lokasi pabrik dilakukan di kota Cikarang. Sistem mampu menampilkan prediksi jarak dan waktu tempuh untuk rekomendasi dari urutan lima pengantaran dengan penerapan metode algoritma Dijkstra dimana proses yang dilakukan sistem adalah memperhitungkan jarak menggunakan Haversine Formula, sehingga didapatkan waktu tempuh berdasarkan parameter kemacetan. selain itu API mampu memvisualisasikan rute setiap tujuan dari titik lokasi katering.

***Kata kunci**: Dijkstra, Formula Haversine, PHP, Jarak terpendek, Kecerdasan Buatan.*

ABSTRACT

The algorithm is used to find the shortest path between points on a graph. The Haversine formula is used to measure the distance from the initial location to the five destination locations where the factory is located in Cikarang and the location is around the location catering house. Calculations are carried out after the system gets the user's latitude and longitude coordinates and the intended factory. In this study, the location of the user and the location of the factory were carried out in the city of Cikarang. The system is suitable to display distance and travel time predictions for recommendations from the order of five deliveries by applying the Dijkstra algorithm method. The process that is carried out by the system, calculates the distance using Haversine formula. Thus, the travel time is obtained bases on congestion parameters. In Addition, besides the API is able to visualize the route of each destination from the catering location point.

***Keywords**: Dijkstra, Haversine Formula, PHP, Shortest Path, Artificial Intelligence.*

1. PENDAHULUAN

Pangan merupakan usaha yang prospektif di Indonesia, sehingga perkembangan pangan dalam bidang industri menghasilkan minat untuk menekuni bisnis bidang ini. Bisnis katering saat ini cukup banyak **(Harahap & Khairina, 2017)** Beberapa faktor seperti faktor manajemen yang masih sederhana, strategi pemasaran dan berpindahnya konsumen untuk mencari alternatif produsen lain menyebabkan katering kurang mampu bertahan lama **(Budihartono, 2016)**. Kurangnya pelayanan yang diberikan seperti konsumen tidak merasa puas atas keterlambatan dalam pengantaran makanan sehingga kualitas makanan menjadi berkurang **(Kamil, Anra, & Sastypratiwi, 2015)**. Manajemen waktu juga harus diperhatikan baik dalam satu kali pengantaran menggunakan satu kendaraan agar dapat menghemat bahan bakar serta dari segi jumlah armada yang digunakan **(Wijayanti, Prihandono, & Kusnandar, 2015)**. Setiap jarak antar pabrik tentu berbeda namun untuk mengejar waktu jam makan karyawan agar tidak adanya keterlambatan diperlukan perhitungan jarak terpendek untuk memilih jalur terbaik **(Prianto & Kusnadi, 2018)**. Pemilihan jarak terpendek dihitung dari titik awal pengiriman titk pengantaran pabrik yang mana untuk meminimalisir waktu pengantaran katering pabrik di jam makan karyawan.

Penentuan jarak terpendek dilakukan dengan menggunakan peta yang sudah ada **(Gonzalez, 2016)** dengan cara memilih jalur terbaik dan terpendek dari titik awal ke titik tujuan, sehingga didapatkan hasil yaitu waktu tempuh perjalanan dan urutan pengantaran yang mana terlebih dahulu diantar **(Junanda, 2016)**. Namun hal tersebut dirasa kurang maksimal karena memakan waktu untuk menentukan sendiri dari jalur - jalur yang ada dan memperkirakan sendiri perhitungan jarak terpendek dari tempat asal ke tempat yang dituju **(Effensi & Rosmala, 2018)**. Hal tersebut dikarenakan permasalahan yang muncul seperti pemutusan jalur terpendek dengan adanya kegiatan masyarakat, kemacetan, atau jalan rusak **(Cantona, Fauziah, & Winarsih, 2020)**. Untuk itu diperlukan sistem yang dapat menentukan jarak terpendek dari titik asal ke titik tujuan untuk dapat membantu menemukan rute terpendek dan terbaik **(Ismantohadi & Iryanto, 2018)**. Dalam hal ini digunakan algoritma *Dijkstra* dalam penentuan jarak terpendek tersebut dengan menerapkan *Haversine* formula yang diterapkan pada penentuan pengantaran katering dari titik awal yaitu rumah katering dan titik tujuan yaitu lima pabrik yang lokasinya berada di sekitar rumah katering tersebut yang mana masih di dalam satu kota yang sama yaitu kota Cikarang. Studi kasus dari penelitian ini secara langsung diambil dari satu perusahaan katering di kota Cikarang, kemampuan dari katering tersebut adalah lima pengantaran dalam satu waktu. Permasalahan utama dari perusahan katering tersebut adalah menentukan urutan pengantaran, hal mana kendala kondisi jalan dalam hala ini kemacetan jalan setiap waktu berbeda, sehingga waktu pengiriman sangat berhubungan dengan penjadwalan. Oleh karena itu diperlukan sistem optimasi penjadwalan untuk mendapatkan waktu optimum dalam pengantaran katering berdasarkan rute atau jarak tempuh terpendek serta meminimalisir waktu pengantaran katering di jam makan siang pabrik di satu kota untuk tujuan lokasi pabrik yang berada di sekitaran lokasi rumah pabrik.

Setiap jarak antar pabrik tentu berbeda namun untuk mengejar waktu jam makan karyawan agar tidak adanya keterlambatan diperlukan perhitungan jarak terpendek untuk memilih jalur terbaik **(Prianto & Kusnadi, 2018)**. Algoritma *Dijkstra* digunakan sebagai metode untuk memecahkan permasalahan pencarian rute terpendek dengan memperhatikan kondisi jalan yang ada.Dengan menerapkan *Haversine* formula untuk membantu menghitung jarak antara titik di permukaan Bumi menggunakan *latitude* dan *longitude* yang ada. Penelitian ini bertujuan untuk membangun sistem optimasi

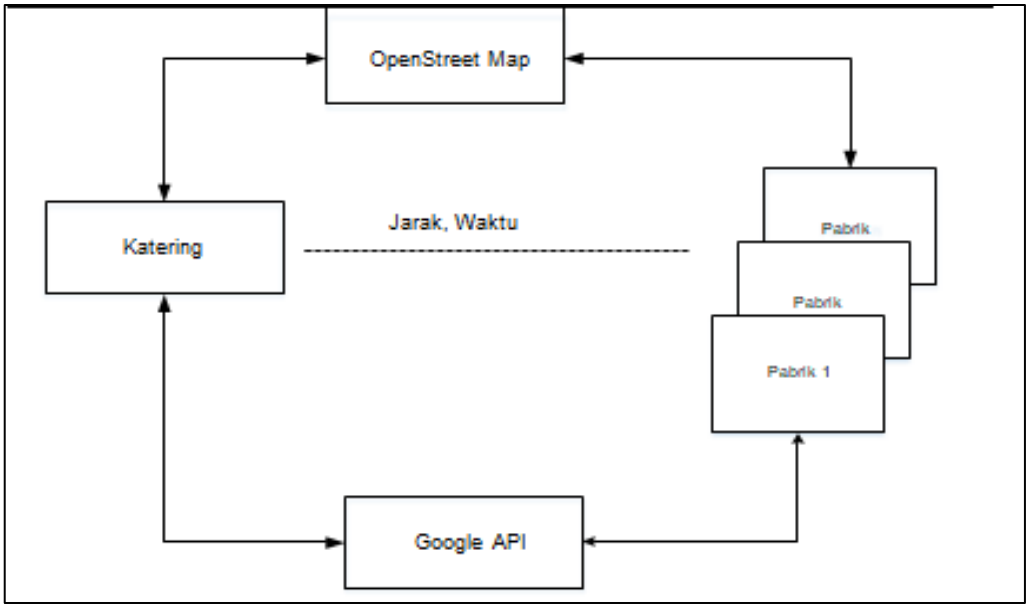
penjadwalan untuk mendapatkan waktu optimum dalam pengantaran katering berdasarkan rute atau jarak tempuh terpendek dengan menerapkan algotima *Dijkstra* dan *Haversine* formula*,* serta sistem dapat menampilkan model lintasan terpendek dan waktu yang dibutuhkan dalam pengantaran katering menuju setiap lokasi pabrik.

2. METODE PENELITIAN

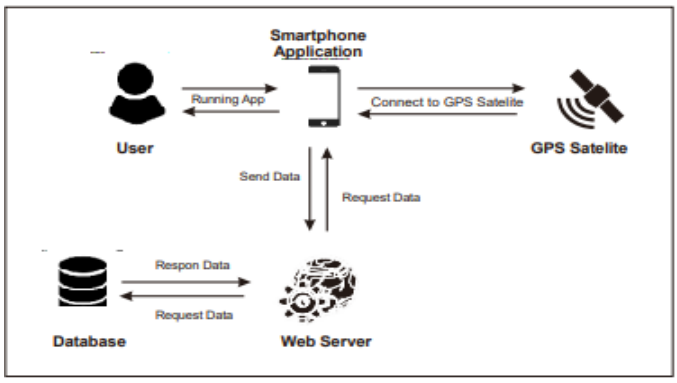
Penelitian dilakukan di kota Cikarang yang mana membutuhkan data jalanan, persimpangan yang ada, serta info jalur satu arah pada kota tersebut. Proses selanjutnya adalah mengolah data tersebut untuk dapat digambar. Data *latitude* dan *longitude* dari map sangat dibutuhkan untuk proses penggambaran lokasi pabrik untuk selanjutnya dilakukan proses *intersection* dengan batas wilayah kota Cikarang. Data yang telah masuk lalu diproses dengan menghitung jarak tempuh kendaraan. Informasi jarak di gunakan untuk mengetahui waktu yang optimum menuju lokasi pabrik. Status kemacetan diputuskan dengan menyesuaikan nilai kecepatan kendaraan dengan tabel klasifikasi jalan. Informasi ini tertuju pada pihak pengguna.

Proses selanjutnya adalah melakukan rancang bangun sistem untuk satu kali pengantaran makanan dari satu perusahaan katering menuju ke banyak pabrik. Sistem mengolah data jarak, dan waktu tempuh setiap tujuan. Terdapat parameter bobot kemacetan untuk menghasilkan urutan rekomendasi pengantaran yang didahulukan. Titik A yaitu lokasi awal katering berada, menuju beberapa pabrik dengan notasi yang berbeda yaitu (P_1, P_2, P_3, …., P_n).

Pada Gambar 1 dijelaskan bahwa katering sebagai titik awal lokasi mencari urutan lokasi tujuan yaitu beberapa pabrik yang tidak ditentukan jumlahnya (Pabrik 1, Pabrik 2, …. Pabrik n) dimana penentuan urutan diilakukan berdasarkan waktu tempuh paling optimum.



Gambar 1. *Quick Design* Proses Kerja Sistem yang Dibangun



Gambar 2. Skema Umum Perancangan Sistem Penentuan Jarak

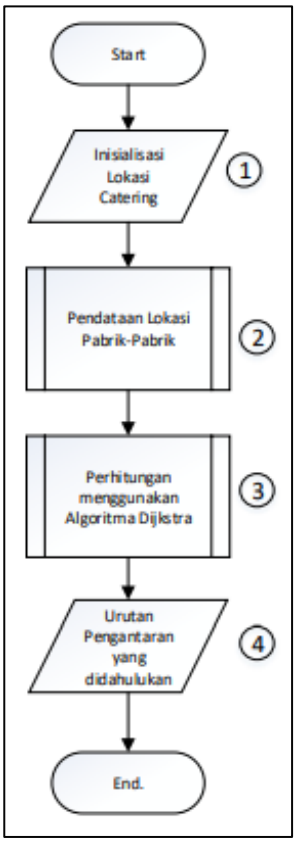
Pada Gambar 2 dijelaskan bahwa sistem diimplementasikan pada *mobile device* sebagai *client* dan *web application* sebagai *server*. Setelah GPS menerima transmisi sinyal satelit dan mendapatkan koordinat *user*, *user* kemudian diarahkan untuk memasukkan tujuan secara manual sehingga *Cloud Database* membaca data besaran koordinat *latitude* dan *longitude.* Setelah itu dilakukan proses perhitungan Algoritma *Dijkstra* untuk mendapatkan rute terpendek. Input awal dan tujuan berupa *node* diproses dengan menghitungkan nilai *Dijkstra* dan nilai *heuristic* yang nodenya termasuk pada *openset*. Setelah itu dilakukan perhitungan dan didapat *node* yang dapat dimungkinkan untuk dilalui dan ditampilkan pada peta sebagai *path* terpilih. Logika Algoritma *Dijkstra* yaitu lokasi yang berdekatan dijadikan satu simpul pada graf sedangkan jalan yang berpotongan secara langsung dijadikan sisi dalam graf. Setelah membentuk model graf kemudian menentukan jarak terpendek menggunakan perhitungan *Haversine*. *Haversine Formula* digunakan untuk mengetahui jarak antar titik di permukaan Bumi dimana *latitude* dan *longitude* digunakan sebagai variabel masukan**(Junanda, 2016)** .

Gambar 3 merupakan *flowchart* keseluruhan sistem penerapan rute terpendek dengan tahapan sebagai berikut.

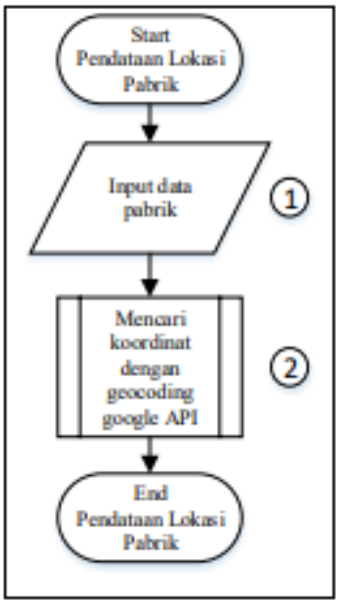
1. Pengguna melakukan pendataan lokasi pabrik-pabrik kemudian *Geometry Location* mengembalikan informasi spasial termasuk nilai koordinat dari lokasi yang dimasukkan oleh pengguna.
2. Sistem menampilkan lokasi awal dan tujuan beserta koordinat.
3. Dari koordinat lokasi awal dan tujuan yang telah didapat, sistem memproses dengan melakukan perhitungan jarak berdasarkan rumus *Haversine* formula dan menerapkan Algoritma *Dijkstra* untuk pencarian rute terpendek.
4. Sistem menampilkan data urutan pengantaran yang didahulukan, rute pada peta dan disertakan jarak, waktu tempuh pada antarmuka aplikasi *smartphone*.

Gambar 4 merupakan subproses dari pendataan lokasi pabrik. Pendataan disini dimaksudkan untuk mengetahui data spasial berupa koordinat setiap pabrik sebelum memulai aplikasi pada *smartphone*.

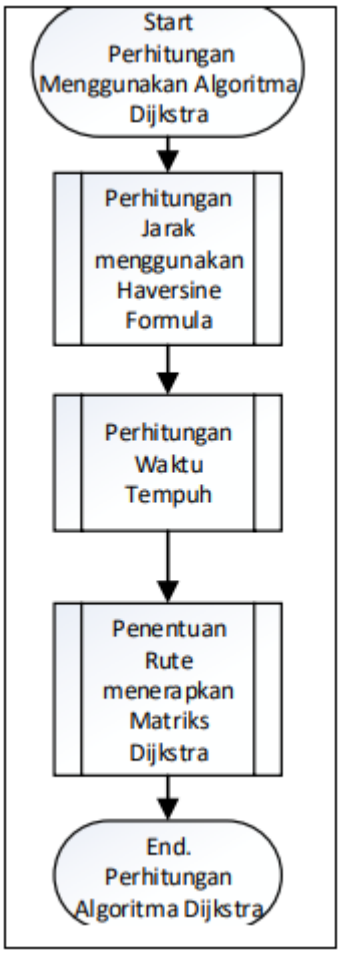
1. Data yang dibutuhkan yaitu nama pabrik, alamat lokasi pabrik beserta titik koordinat lokasi pabrik.
2. Koordinat didapat dari *Google* API dengan proses *geocoding*. Untuk jumlah data yang dimasukkan dan disimpan pada sistem adalah minimal satu dan tidak ada batasan maksimum.



Gambar 3. *Flowchart* Sistem Penerapan Rute Terpendek



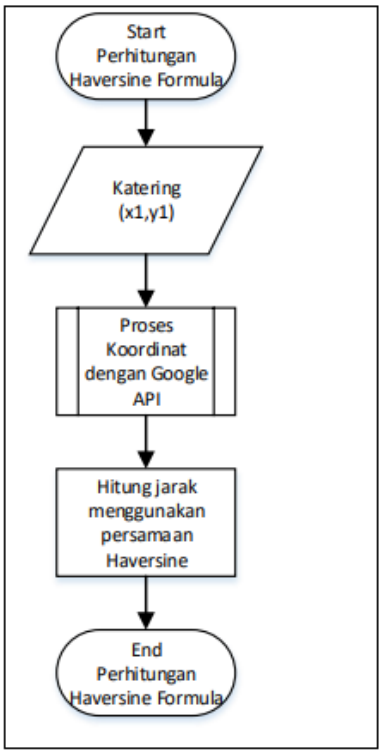
Gambar 4. *Flowchart* Pendataan Lokasi Pabrik



Gambar 5. *Flowchart* Perhitungan Menggunakan Algoritma Djikstra

Gambar 5 yang merupakan *flowchart* subproses perhitungan *algoritma dijkstra*. Sebelum sampai pada penentuan rute, ada beberapa tahapan untuk menghasilkan keluaran sistem yang diharapkan. Jarak diperhitungkan menggunakan *Haversine* formula yang merupakan salah satu persamaan akurat untuk menentukan jarak antara titik - titik bumi dengan memperhitungan lengkung bumi untuk penjelasannya dapat dilihat pada Gambar 6. Setelah didapatkan jarak tempuh, perhitungan waktu tempuh dapat dimulai dengan dipengaruhi bobot kemacetan. Algoritma *Dijkstra* dapat merekomendasikan lintasan terpendek berdasarkan bobot waktu tempuh yang didapat dengan menerapkan matriks *Dijkstra*.

Pada Gambar 6 dijelaskan mengenai proses perhitungan jarak tempuh menggunakan persamaan *Haversine formula* dengan parameter katering dan pabrik-pabrik yang dituju. Variabel yang telah diinisialisasikan dengan x dan y adalah koordinat *latitude* dan *longitude*. Pada titik awal katering divariabelkan sebagai x1,y1 dan koordinat pabrik diinsialisasikan sebagai x2n dan y2n karena jumlah pabrik tidak dibatasi. Proses perhitungan jarak dilakukan dari *node* terdekat lokasi awal dan *node* terdekat lokasi tujuan. Setelah mendapatkan koordinat, perhitungan dilakukan secara berulang dari tujuan pabrik awal hingga pabrik ke-*n*.



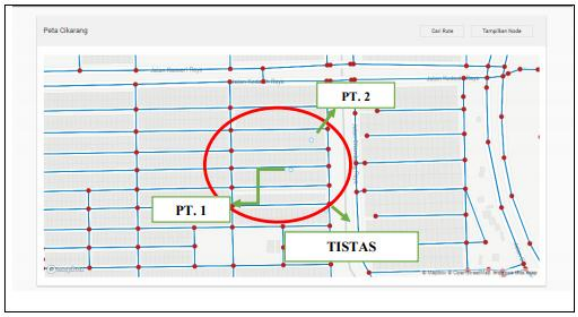
Gambar 6. *Flowchart* Perhitungan *Haversine* Formula

3. HASIL DAN PEMBAHASAN

3.1. Studi Kasus

Pada subbab ini dilakukan studi kasus untuk mencari jarak tempuh, waktu tempuh, dan perhitungan matriks *dijkstra*. Pada kasus ini dilakukan satu titik lokasi perusahaan katering dengan pengantaran menuju dua perusahaan pabrik untuk diketahui mana yang dapat didahulukan pengantarannya. Berikut ini adalah tahapan-tahapan yang dibahas dalam studi kasus ini.

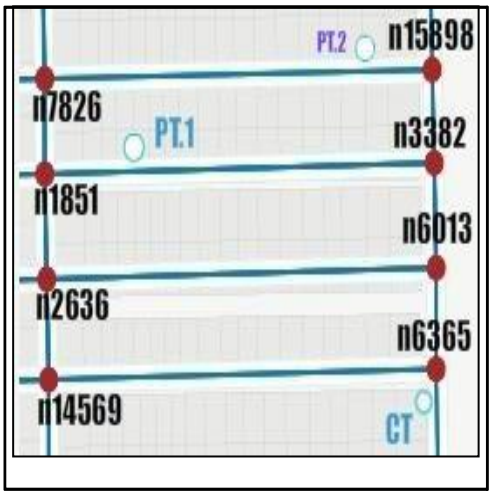
1. Mencari jarak tempuh setiap *node*.



Gambar 7. Pencarian Jarak Tempuh Setiap *Node*

Gambar 7 merupakan tampilan peta kota Cikarang yang didalamnya terdapat gambar anak panah berwarna hijau menunjukan titik lokasi Tistas sebagai lokasi awal katering. PT1 dan PT2 merupakan lokasi tujuan. Persimpangan jalan pada peta disebut *node* dimana *node-node* tersebut ditandai oleh titik merah pada tampilan peta.

2. Tampilan Peta yang Diperbesar



Gambar 8. Peta yang Diperbesar

Gambar 8 merupakan tampilan peta pada Gambar 7 yang telah diperbesar sehingga nama-nama *node* terlihat jelas. Sebelum dilakukan tahapan proses pencarian jarak antar *node* terdapat delapan *node* yang terdeteksi yaitu N6365, N14569, N2636, N1851, N6013, N3382, N15898.

Tabel 1. Inisialisasi Penamaan *Node*

Nama *node* asli	Inisialisasi
N6365	a
N6013	b
N3382	c
N14569	d
N15898	e
N2636	f
N1851	g
N7862	h

Tabel 1 merupakan inisialisasikan nama-nama *node* yang asli menjadi nama lain yaitu huruf abjad berurut, bertujuan untuk memudahkan saat penggambaran tabel proses matriks pencarian bobot terkecil setiap *node* yang masing-masing punya keterhubungan.

Tabel 2. *Latitude* dan *Longitude* Lokasi

Perusahaan	Lokasi	*Latitude*	*Longitude*	*Node* Terdekat
Tistas Catering (Start)	Jalan Kedasih XI blok G1 no.2c	-6.301847	107.163662	a
PT1	Jalan Kedasih VII no.22	-6.301295	107.163995	g
PT2	Jalan Kedasih V no.5	-6.300991	107.164768	e

Tabel 2 merupakan keterangan koordinat *latitude* dan *longitude* yang didapat dari sistem menggunakan *Google geocoding*. Setelah koordinat lokasi didapatkan, sistem mencari *node* yang paling dekat untuk dijadikan titik acuan lokasi *node*.Seperti pada kasus ini

bahwa lokasi CT berdekatan dengan *node* a dan PT2 berdekatan dengan *node* e. Berikut dijelaskan perhitungan jarak tempuh menggunakan rumus *Haversine Formula.*

Tabel 3. Bobot Kemacetan

Kondisi Jalan	Bobot
Normal	1
Ramai Lancar	0.8
Macet	0.65
Macet Parah	0.4

Tabel 3 merupakan penjelasan setiap nilai bobot kemacetan. Bobot ruas jalan didapatkan dari asumsi ruas jalan yang memiliki kapasitas maksimal kendaraan yang terdiri dari kendaraan beroda empat dengan kecepatan rata – rata 40 km/jam. sehingga dipastikan pada ruas jalan akan terjadi kemacetan dengan begitu dapat dipilih salah satu jalur alternative dengan diiasumsikan perbedaan bobot setiap persimpangan jalan bahwa kondisi normal merupakan jalan lancar sehingga diberi bobot 1 yaitu 100% kondisi dari kecepatan rata-rata yang didata. Bobot 0.8 merupakan 80% kondisi jalan ramai lancar. Bobot 0.65 merupakan kondisi 65% dari kondisi normal yang berarti jalan macet dan 0.4 adalah macet parah.

Tabel 4. Jarak dan Hubungan Setiap *Node*

Node	**Koordinat**	**Hubungan**		**Jarak antar *node* (km)**
		Node	**Bobot Kemacetan**	
a	-6.301840782165527 107.16505432128906	b	0.4	0.03
		h	1	0.158
b	-6.301568984985452 107.16505432128906	c	0.4	0.031
		g	0.65	0.159
		a	0.65	0.03
c	-6.330718040466309 107.1871109008789	d	1	0.028
		f	0.4	0.159
		b	0.8	0.031
d	-6.301036357879639 107.1650390625	e	0.4	0.158
		f	1	0.028
e	-6.309691905975342	d	0.4	0.158
	107.1411890930175781	f	0.8	0.028
f	-6.311672687530518 107.14118194580078	e	0.65	0.028
		c	0.65	0.159
		g	0.65	0.031
g	-6301599502563477 107.16361236572266	b	1	0.159
		f	0.65	0.031
		h	0.8	0.03
h	-6.296785831451416	a	0.65	1.5
	107.15249633789062	g	1	0.03

Tabel 4 yang berisikan hubungan setiap *node* dan bobot kemacetan yang telah ditentukan dari sistem. Jarak didapat dari perhitungan yang telah dicontohkan sebelumnya. Berikut tahapan-tahapan Pencarian rute terpendek dan waktu tempuh dengan Algoritma *Dijkstra* menggunakan bobot kemacetan.

1. Menentukan kecepatan rata-rata kendaraan .
2. Menghitung matriks *Dijkstra* set *node* awal N6365 (lokasi katering)
3. Perhitungan bobot waktu tempuh (jam)
4. Konversikan waktu tempuh dari jam menjadi menit

Selanjutnya nilai bobot kemacetan antara *node* a dengan *node* b adalah 0.4 dan jarak yang didapat adalah 0.03 km. Setelah waktu tempuh antar *node* didapat, selanjutnya dilakukan penentuan rute terpendek menggunakan matriks *dijkstra*. Berikut dijabarkan terlebih dahulu hasil waktu tempuh yang telah diperhitungkan yang dapat dilihat pada Tabel 5.

Tabel 5. Waktu Tempuh Seluruh *Node*

***Node* keberangkatan**	***Node* tujuan**	**Bobot Kemacetan**	**Jarak tempuh**	**Waktu tempuh**
a	b	0,4	0,03	0,1125
	c	1	0,158	0,237
b	c	0,4	0,031	0,11625
	g	0,65	0,159	0,366923
	a	0,65	0,03	0,069231
c	d	1	0,028	0,042
	f	0,4	0,159	0,59625
	b	0,8	0,031	0,058125
d	e	0,4	0,158	0,5925
	c	1	0,028	0,042
e	d	0,4	0,158	0,5925
	f	0,8	0,028	0,0525
f	e	0,65	0,028	0,064615
	c	0,65	0,159	0,366923
	g	0,65	0,031	0,071538
g	b	1	0,159	0,2385
	f	0,65	0,031	0,071538
	h	0,8	0,03	0,05625
h	a	0,65	1,5	3,461538
	g	1	0,03	0,045

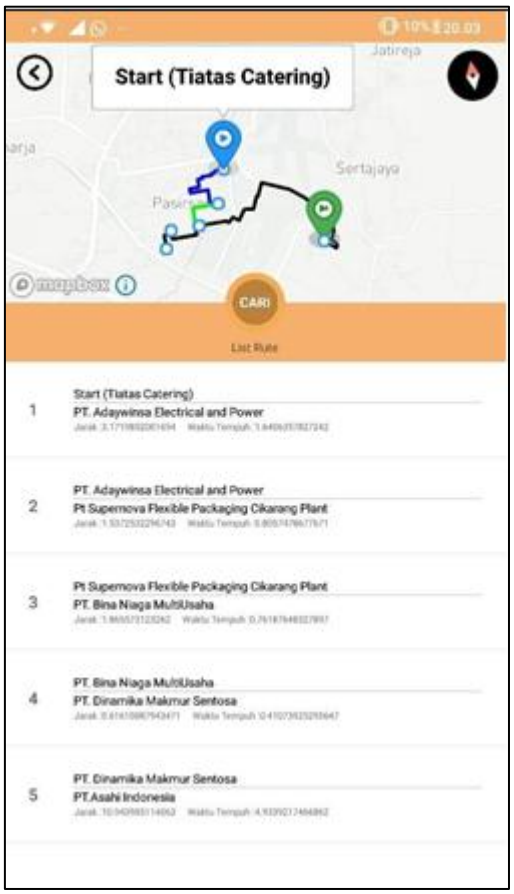
Tabel 5 merupakan hasil waktu tempuh seluruh *node* yang telah didapat dari perhitungan bobot tempuh. Dimana parameternya terdapat *node* keberangkatan menuju masing-

masing titik tujuan. Titik tujuan tersebut adalah *node* yang terhubung. Waktu tempuh yang didapat tersebut merupakan bahan yang diolah pada *matriks dijkstra*.

Tabel 6. Hasil Pengujian Pengantaran Katering Pabrik

Pengujian		Koordinat Katering		Koordinat Pabrik			Jarak Tempuh (km)	Waktu Tempuh (menit)	Hasil Urutan ke-
		Latitude	Longitude	Nama Pabrik	Latitude	Longitude			
1	Tistas Catering (start)	-6.3019297	107.165008400000003	PT Supernova Flexible Packaging Cikarang Plant	-6.3098703,107	107.16020159999994	1.58	3.25	2
		-6.3019297	107.16020159999994	PT. Adaywinsa Electrical and Power	-6.3072119	107.16299679999997	1.64	3.17	1
		-6.3019297	107.16020159999994	PT. Bina Niaga MultiUsaha	-6.3097986	107.15524069999992	1.91	3.75	4
		-6.3019297	107.16020159999994	PT. Dinamika Makmur Sentosa	-6.31251399999	107.15442610000002	1.74	3.60	3
		-6.3019297	107.16020159999994	PT.Asahi Indonesia	-6.3161217	107.17980080000007	3.17	6.70	5

Terdapat 5 lokasi pabrik yang dimasukkan dalam daftar tujuan pengantaran katering di waktu yang sama. Sehingga dicarikan urutan pabrik mana yang pertama sampai terakhir untuk diantarkan makanannya. Hasil pengujian pertama didapatkan urutan pertama yaitu PT Supernova Flexible Packaging Cikarang Plant, urutan kedua adalah PT Supernova Flexible Packaging Cikarang Plant, urutan ketiga adalah PT. Bina Niaga MultiUsaha, PT. Dinamika Makmur Sentosa dan pengantaran terakhir adalah PT.Asahi Indonesia. Urutan didapatkan berdasarkan jarak tempuh dan waktu tempuh dimana urutan pertama adalah waktu dan jarak yang bernilai kecil. Gambar 9 merupakan hasil implementasi pada *smartphone* berdasarkan pengujian yang ada pada Tabel 6. Dimana urutan pertama hingga akhir sama seperti keterangan Tabel 6.



Gambar 9. Hasil Implementasi Pengujian Pengantaran Katering Pabrik

4. KESIMPULAN

Dari hasil pengujian sistem didapatkan kesimpulan yaitu sistem dapat menerapkan Algoritma *Dijkstra* untuk optimasi penjadwalan berdasarkan jarak tempuh terpendek dengan perhitungan jarak menggunakan *Haversine Formula*. Berdasarkan perancangan studi kasus yang telah diterapkan pada sistem, sistem berhasil memberi alternatif antrian pengantaran ke beberapa pabrik dengan maksimal lima pabrik dalam satu kali pengantaran, dimana urutan selanjutnya dijadikan pengantaran selanjutnya. Setelah di masukkan dua lokasi yang berdekatan, sistem menunjukkan kegagalan yaitu sistem tidak dapat melakukan proses perhitungan sehingga hasil yang didapat adalah jarak tempuh dengan nilai 999999.0 km dan waktu tempuh 99999.0 menit. Dikarenakan pada kasus ini lokasi tujuan sangat berdekatan sehingga sistem hanya membaca satu koordinat dari node terdekat tersebut.

DAFTAR RUJUKAN

Budihartono, E. (2016). Penerapan Algoritma Dijkstra untuk Sistem Pendukung Keputusan Bagi Penentuan Jalur Terpendek Pengiriman Paket Barang pada Travel. *Teknik Komputer Politeknik Harapan Bersama*.

Cantona, A., Fauziah, F., & Winarsih, W. (2020). Implementasi Algoritma Dijkstra pada Pencarian Rute Terpendek ke Musem di Jakarta.

Effensi, I., & Rosmala, D. (2018). Pengukuran Tingkat Keakurasian Jarak dengan Menerapkan Algoritma Dijkstra pada Sistem Zonasi.

Gonzalez, A. (2016). Measurement of Areas on A Sphere using Fibonnaci and Latitude - Longitude Lattices.

Harahap, M. K., & Khairina, N. (2017). Pencarian Jalur Terpendek dengan Algoritma Dijkstra. *Jurnal & Penelitian Teknik Informatika*.

Ismantohadi, E., & Iryanto. (2018). Penerapan Algoritma Dijkstra Untuk Penentuan Jalur Terbaik Evakuasi Tsunami - Studi Kasus : Kelurahan Sanur Bali. *Jurnal Teknologi Terapan*.

Junanda, B. (2016). *Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Pada Sistem Infirmasi Geografis Pementaan Statiun Pengisian Bahan Bakar Umum .* Universitas Negeri Padang.

Kamil, M. I., Anra, H., & Sastypratiwi, H. (2015). Rancang Bangun Aplikasi Pencarian Rute Terpendek Lokasi Wisata Kuliner Kota Pontianak Berbasis Mobile.

Prianto, C., & Kusnadi, M. (2018). Penerapan Algoritma Dijktra untuk Menentukan Rute Terbaik pada Mobile E-Parking Berbasis Sistem Informasi Geografis.

Wijayanti, S., Prihandono, B., & Kusnandar, D. (2015). Mencari Lintasan Terpendek dan Optimalisasi Kendaraan Pengangkut Sampah di Pontianak.