

# Penerapan Rute Pada Physical Distancing Menggunakan Metode Dynamic Dijkstra

## *Application Of Physical Distancing-Based Routes Using Dynamic Dijkstra Method*

1<sup>st</sup> Sulthan Muzakki Adytia  
Sadarisman

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
sulthanmuzakki@student.telkomuni-  
versity.ac.id

2<sup>nd</sup> Surya Michrandi Nasution  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

michrandi@telkomuniversity.ac.id

3<sup>rd</sup> Casi Setianingsih  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

### Abstrak

*Physical Distancing* sangatlah penting di masa sekarang untuk mengurangi penyebaran Virus COVID-19, salah satu caranya adalah dengan menjaga jarak. Pada masa pandemi ini, masalah yang dialami kebanyakan orang adalah sulit menemukan restoran yang sepi pengunjung atau taat terhadap peraturan *Physical Distancing*. Pembuatan aplikasi untuk membantu mencari restoran yang sepi sangat dibutuhkan oleh masyarakat. Pembuatan aplikasi ini dapat menggunakan Algoritma Dynamic Dijkstra. Metode ini biasanya digunakan untuk menentukan rute terdekat, Dynamic Dijkstra terbukti efektif melalui hasil penelitian yang telah dilakukan oleh peneliti sebelumnya. Penentuan jalur ini dilakukan dengan menggunakan data yang telah didapat dari beberapa restoran. Dalam penelitian ini, pemilihan restoran yang sepi diharapkan dapat memberi solusi kepada masyarakat untuk berbelanja dengan jarak terdekat dan aman. Hasil yang diperoleh dari penelitian ini adalah website dapat menerapkan pemetaan dan juga dapat menentukan jalur terdekat ke lokasi tujuan dengan data yang diambil dari Google Maps. Hasil pada pengujian Alpha memiliki nilai 100% dan juga pada pengujian Beta dengan hasil validitas yang valid dan juga hasil uji reabilitas yang reable.

**Kata Kunci:** *Dynamic Dijkstra, Physical Distancing, Restoran.*

### Abstract

*Physical distancing is very important nowadays to reduce the spread of the COVID-19 Virus, one way is to keep your distance. During this pandemic, the problem that most people experience is that it is difficult to find a restaurant that is empty of visitors or obeys Physical Distancing regulations. Making applications to help find restaurants that are deserted is very much needed by the community. Making this application can use the Dynamic Dijkstra Algorithm. This method is usually used to determine the closest route, Dynamic Dijkstra has proven to be effective through the results of research that has been carried out by previous researchers. Determination of this path is done using data that has been obtained from several restaurants. In this study, the selection of a quiet restaurant is expected to provide a solution for the community to shop at the closest and safe distance. The results obtained from this research are the website can apply mapping and can also determine the closest path to the destination location with data taken from Google Maps. The results on the Alpha test have a value of 100% and also on the Beta test with valid validity results and also reliable*

*reliability test results.*

**Keyword:** *Dynamic Dijkstra, Physical Distancing, Restaurant.*

## I. PENDAHULUAN

Pada tahun 2020 terjadi peristiwa munculnya Virus COVID-19 secara global yang menyebabkan lumpuhnya sebagian kegiatan masyarakat Indonesia. Pada bulan Maret 2020, "Kasus pertama COVID-19 di Indonesia pada Senin, 2 Maret 2020 ditularkan melalui transmisi dari manusia ke manusia" (Joko Widodo, 2020). Sejak itu, Indonesia mulai terjangkit virus COVID-19 namun, karena kurangnya pengetahuan pemerintah terhadap penanggulangannya menyebabkan peningkatan kasus virus COVID-19 di Indonesia sehingga menyebabkan lumpuhnya perekonomian Indonesia.

Solusi yang diberikan saat ini adalah *Physical Distancing*. Maka dari itu, pemerintah Indonesia telah melakukan Pembatasan Sosial Berskala Besar (PSBB) terhadap tempat ramai seperti Pusat Perbelanjaan dan restoran. Wabah virus COVID-19 telah mengakibatkan penurunan pendapatan restoran sebesar 70-80%.

Permasalahan yang dihadapi para pengunjung adalah sedikitnya informasi mengenai lokasi restoran yang sepi atau telah menaati protokol kesehatan untuk berbelanja dengan aman dan nyaman. Pencarian lokasi restoran dapat menggunakan *Google Maps* [1], namun masih terdapat beberapa kekurangan untuk menentukan lokasi restoran yang sepi pengunjung. Penentuan ini dapat dilakukan menggunakan berbagai macam Algoritma atau Metode, salah satunya adalah Algoritma Dynamic Dijkstra.

Algoritma Dynamic Dijkstra adalah salah satu dari berbagai macam jenis metode yang dapat digunakan karena metode ini dapat memilih solusi terbaik dan memiliki kompleksitas yang

lebih sedikit dibandingkan dengan metode lainnya seperti algoritma Bellman Ford dan Floyd Warshall, sehingga dalam pembuatan ini dapat digunakan metode Dynamic Dijkstra untuk menentukan rute ke restoran yang sepi atau telah menaati protokol kesehatan yang ada disekitar [2].

## II. KAJIAN TEORI

### a. *Physical Distancing*

*Physical Distancing* merupakan sebuah kegiatan menjaga jarak yang bertujuan untuk meminimalisir penyebaran Virus COVID-19, WHO dan para ahli merekomendasikan setidaknya lebih dari 1-2 meter untuk saling menjaga jarak. Karena Virus *COVID-19* lebih mudah menyebar ketika berada didalam kerumunan [3]. Aturan 1-2 meter juga didasarkan oleh kerja lama pada dikotomi tetesan pernapasan, tetesan pernapasan dibagi menjadi 2, yaitu yang besar dan yang kecil. Ukuran tetesan mempengaruhi seberapa jauh jangkauan dari orang yang terinfeksi, tetesan besar dipancarkan jauh melalui udara dan memiliki jarak kisaran 1-2 meter [4]

### b. Algoritma Dijkstra

Algoritma Dijkstra biasanya digunakan untuk pencarian rute terpendek, pencarian rute terpendek ini menggunakan materi teori Graf. Algoritma Dijkstra ditemukan oleh Edger Dijkstra, yaitu seorang ilmuwan asal Belanda [5]. Algoritma yang paling banyak digunakan untuk menentukan masalah pemilihan rute, baik penentuan rute terpendek maupun rute yang paling optimal adalah algoritma Dijkstra. Algoritma Dijkstra digunakan untuk mengidentifikasi rute terpendek. Algoritma Dijkstra memungkinkan menemukan rute terpendek antara dua titik [6]. Dijkstra adalah sebuah algoritma yang digunakan untuk mencari rute terpendek pada sebuah Graf berarah (*Directed Graph*). Dijkstra menggunakan prinsip

algoritma *Greedy* untuk menentukan suatu rute terpendek. *Greedy* mempunyai prinsip tersendiri pada algoritma Dijkstra, bahwa setiap langkah dipilih sisi dengan berbobot terkecil yang terhubung kedalam suatu simpul yang sudah menjadi solusi dengan simpul lainnya yang belum menjadi solusi [7]. Algoritma Dijkstra membandingkan kemungkinan rute yang akan dilalui dan dihitung setiap kemungkinan jarak. Rute dengan jarak terpendek akan menjadi pilihan terbaik [8].

### C. Teori Graf

*Input* dari algoritma Dijkstra adalah sebuah graf  $G$  yang diwakilkan dengan titik, sudut dan simpul yang saling terhubung oleh sebuah garis. Suatu graf dapat dinyatakan dengan  $V$  (*Vortex*) yang merupakan sebuah simpul dan titik (*Node*),  $E$  (*Edges*) yang merupakan sebuah sisi dan  $G$  yang merupakan Graf. Sebagai contoh dari graf yaitu sebuah keterhubungan antar gedung yang ditandai dengan titik atau simpul, sedangkan setiap sisinya dapat diberi nilai sisi (*Edge*) [9]. Nilai yang didapat bisa dinyatakan dalam bentuk jarak antar titik tersebut, daripada itu simpul (*Vertex*) dapat digunakan dalam pemberian nilai *heuristic* pada graf [10].

### D. Django

Django adalah sebuah *framework* yang

dapat merangkap *front-end* dan *back-end* atau biasa disebut *full-stack*. Django menggunakan Bahasa pemrograman Python. *Framework* ini dapat mempermudah *programmer* karena pembuatan web yang lebih cepat [11].

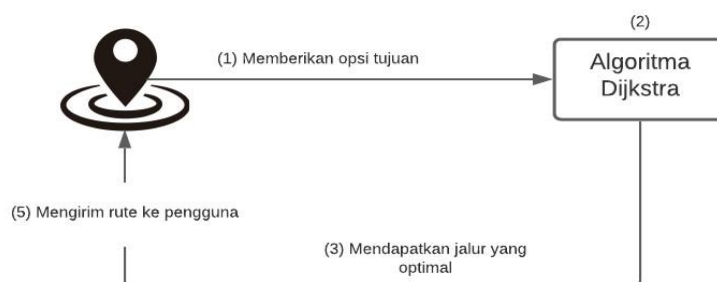
### E. MapBox Gl JS

Mapbox adalah *webservice* yang dapat menyediakan informasi tentang *geocoding* dan dari dua buah *node*, rute terkecil dari satu titik (*node*) awal ke titik (*node*) tujuan atau biasa disebut mencari rute terpendek (*shortest path problem*). Mapbox GL JS merupakan kumpulan *JavaScript* yang menggunakan WebGL sebagai pembuatan peta. Mapbox GL adalah *OpenSource* yang dapat menyesuaikan aplikasi seperti web, selular, dan *dekstop* [12].

## III. METODE

### A. Gambaran Umum Sistem

Sistem yang digunakan dalam penelitian kali ini berfungsi untuk menentukan rute perjalanan ke arah restoran sepi terdekat menggunakan Algoritma Dijkstra. Sistem ini dapat membantu masyarakat agar dapat berbelanja dengan aman tanpa takut terpapar Virus COVID-19. Berikut gambaran sistem :

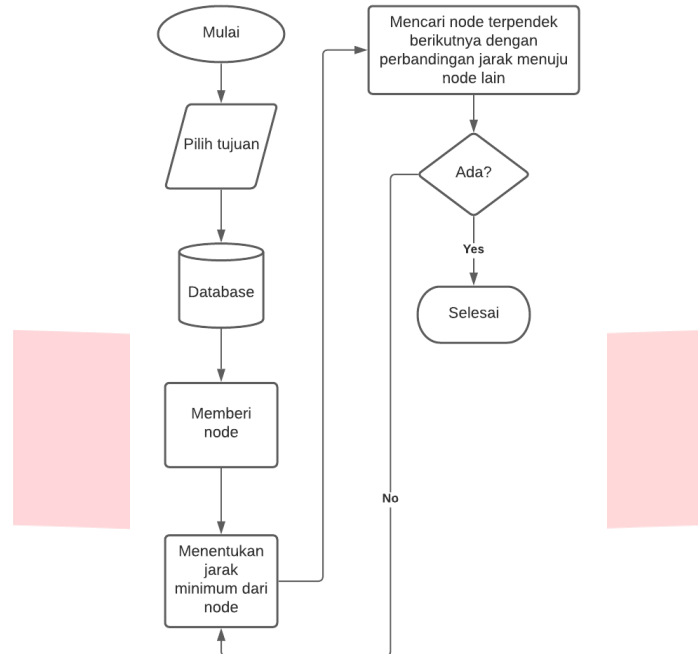


**Gambar 3. 1** Gambaran umum sistem

Percobaan dilakukan dengan membandingkan jalur yang dipilih oleh sistem dengan jalur terpendek

secara manual.

### 3.3.1 Diagram Perancangan Algoritma Dijkstra



Gambar 3. 1 Flowchart Dijkstra

Pada gambar 3.2 dapat dilihat bagaimana proses Algoritma Dijkstra dapat memilih target dari restoran yang ada. Berdasarkan diagram perancangan algoritma Dijkstra, berikut proses yang dilakukan:

3. Setelah mendapatkan *node* awal sistem akan mencari *node* selanjutnya dengan bobot nilai terkecil.
3. Setelah mendapatkan *node* selanjutnya, sistem akan menentukan *node* terpendek yang akan dibandingkan dengan node lainnya
3. Apabila tidak menemukan titik akhir, maka sistem akan kembali pada node sebelumnya dan memilih node ulang.

## IV. HASIL DAN PEMBAHASAN

### A. Implementasi Metode

Percobaan dilakukan dengan membandingkan jalur yang dipilih oleh sistem dengan jalur terpendek secara manual.

Tabel 4. 1 Data Dictionary

id	id_temp	id_tujuan	jarak	latitude	longitude	Variabel
1	1	1	0	-6.9721716742480595	107.63362405825808	A
2	1	1	300	-6.971835939673434	107.6335631140975	B
3	1	1	200	-6.967857923992137	107.63459465624771	C
4	1	1	1500	-6.968408774376329	107.63728822568153	D
5	1	1	1000	-6.965137873805651	107.63573716466193	E
6	1	1	1000	-6.962850824855162	107.63557944277255	F

id	id_temp	id_tujuan	jarak	latitude	longitude	Variabel
7	1	1	700	-6.960670731205224	107.63672416342513	G
8	1	1	300	-6.953486474578419	107.63856000059153	H
9	1	1	500	-6.937611874295368	107.62337669142323	I
10	1	1	500	-6.937839626248906	107.62353301978875	J
11	1	1	400	-6.939188225123269	107.62485951551962	K
12	2	1	0	-6.9721929732446455	107.6335811429158	L
13	2	1	210	-6.9724658742380345	107.63408649967964	M
14	2	1	790	-6.972573324744729	107.63434534164428	N
15	2	1	1500	-6.9728104257742425	107.63617518317618	O
16	2	1	1500	-6.965516574474856	107.63796779226519	P
17	2	1	1000	-6.948099925112626	107.63342754588837	Q
18	2	1	1000	-6.937611874295368	107.62337669142323	R
19	2	1	600	-6.937839626248906	107.62353301978875	S
20	2	1	200	-6.939188225123269	107.62485951551962	T
21	3	1	0	-6.9724658742380345	107.63408649967964	U
22	3	1	210	-6.972573324744729	107.63434534164428	V
23	3	1	790	-6.9728104257742425	107.63617518317618	W
24	3	1	500	-6.965516574474856	107.63796779226519	X
25	3	1	2000	-6.948099925112626	107.63342754588837	Y
26	3	1	1500	-6.937611874295368	107.62337669142323	Z
27	3	1	1500	-6.937839626248906	107.62353301978875	AB
28	3	1	200	-6.939188225123269	107.62485951551962	BC

Keterangan: id\_temp = node

Id\_tujuan = tujuan

Tabel 4. 2 Pengujian Hasil Dari Algoritma

No	Node Awal	Node Tujuan	Jalur dengan jarak terpendek	Jalur yang dipilih oleh sistem	Jarak
1	Telkom University	B	Telkom University-B	Telkom University-B	300
2	B	C	Telkom University-B-C	Telkom University-B-C	200
3	C	D	Telkom University-B-C-D	Telkom University-B-C-D	1500
4	D	E	Telkom University-B-C-D-E	Telkom University-B-C-D-E	1000
5	E	F	Telkom University-B-C-D-E-F	Telkom University-B-C-D-E-F	1000
6	F	G	Telkom University-B-C-D-E-F-G	Telkom University-B-C-D-E-F-G	700
7	G	H	Telkom University-B-C-D-E-F-G-H	Telkom University-B-C-D-E-F-G-H	300
8	H	I	Telkom University-B-C-D-E-F-G-H-I	Telkom University-B-C-D-E-F-G-H-I	500
9	I	J	Telkom University-B-C-D-E-F-G-H-I-J	Telkom University-B-C-D-E-F-G-H-I-J	500
10	J	K	Telkom University-B-C-D-E-F-G-H-I-J-K	Telkom University-B-C-D-E-F-G-H-I-J-K	400

Tabel 4. 3 Pengujian Estimasi Waktu Tempuh

No	Node Awal	Node Tujuan	Waktu tempuh	Jarak
1	Telkom University	1	01:00 Menit	342
2	Telkom University	2	01:00 Menit	766
3	Telkom University	3	02:00 Menit	1059
4	Telkom University	4	02:00 Menit	1059
5	Telkom University	5	02:42 Menit	1162
6	Telkom University	6	03:02 Menit	1400
7	Telkom University	7	05:00 Menit	4100
8	Telkom University	8	06:32 Menit	4400
9	Telkom University	9	06:40 Menit	4700
10	Telkom University	10	10:00 Menit	6400
11	Telkom University	11	07:56 Menit	6500

Keterangan : Node tujuan diambil dari id\_tujuan pada halaman 13-19

Data waktu tempuh diambil pada tanggal 7 Februari 2022 pada pukul 21.39

Dari percobaan tersebut, maka didapatkan hasil yang serupa pada sistem. Maka dari itu, algoritma Dijkstra sudah bekerja dengan baik karena sistem dapat memberikan jalur terpendek.

Rute dengan jarak terpendek akan menjadi pilihan terbaik [10].

$$NK = (ks - \min K) / (\max K - \min K)$$

$$NJ = (js - \min J) / (\max J - \min J)$$

$$HO = (NJ + NK) * 50\%$$

Keterangan : NK = Normalisasi Keramaian

NJ = Normalisasi Jarak

HO = Hasil Optimasi

ks = Keramaian Sekarang

minK = Minimal Keramaian

maxK = Maximal Keramaian

js = Jarak Sekarang

minJ = Minimal Jarak

maxJ= Maximal Jarak

Rumus diatas digunakan untuk mendapatkan nilai Hasil Optimal yang digunakan untuk bobot nilai node. Node terkecil akan digunakan untuk rute rekomendasi.

## V. KESIMPULAN DAN SARAN

### a. Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan pada Tugas Akhir ini, maka didapatkan kesimpulan bahwa:

1. Seluruh fitur yang ada pada web ini telah diuji dan sudah sesuai dengan yang diinginkan. Hasil dari pengujian Alpha sudah sesuai dengan yang diharapkan. Dengan nilai keakurasian 100% pada pengujian Alpha.
2. Website berhasil mengambil data restoran yang sudah menerapkan *Physical Distancing* dengan cara *survey* restoran yang ada di sekitaran area Telkom University.

### b. Saran

Berdasarkan hasil yang diperoleh dari tugas akhir ini, maka ada beberapa hal yang penulis dapat sarankan untuk penelitian selanjutnya. Berikut saran yang dapat diberikan:

1. Web dapat ditambahkan fitur tentang kepadatan restoran secara *real time*.
2. Membandingkan keakuratan dengan menggunakan algoritma lainnya.

## REFERENSI

- [1] F. S. Taufiq Hidayat, "Penentuan Rute Pariwisata Kota Bandung Menggunakan Algoritma Particle Swarm Optimization," vol. 1, p. 6,

2007.  
[2] L. J. E. Dewi, "Pencarian Rute Terpendek Tempat Wisata Di Bali Dengan Menggunakan Algoritma Dijkstra", Seminar Nasional Aplikasi Teknologi Informasi 2010, p. D-47, 2010.

- [3] N. R. Jones, Z. U Qureshi, R. J. Temple, J. P. Larwood, T. Greenhalgh, L. Bourouiba, "Two metres or one: what is the evidence for physical distancing in covid-19?", The BMJ, p. 3, 2020

- [4] V. F. Mukaromah. WHO Gunakan Istilah *Physical Distancing*, Ini Bedanya dengan *Social Distancing*. KOMPAS.com. 2020  
<https://www.kompas.com/tr-en/read/2020/04/01/061500965/who-gunakan-istilah-physical-distancing-ini-bedanya-dengan-social?page=all>

(diakses pada Minggu, 22 November pukul 22:57)

- [5] L. J. E. Dewi, "Pencarian Rute Terpendek Tempat Wisata Di Bali Dengan Menggunakan Algoritma Dijkstra", Seminar Nasional Aplikasi Teknologi Informasi 2010, p. D-47,



- 2010.
- [6] C. Prianto, M. Kusnadi, "Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada *Mobile E-Parking* Berbasis Sistem Informasi", *Jurnal Informatika*, vol.03, no. 03, p 330, 2018.
- [7] R. Saputra. "Sistem Informasi Pencarian Obyek Kota Medan Dengan Algoritma Dijkstra." *Jurnal Matematika*, No. 1, Vol.14, Hal 19-24, April 2011.
- [8] C. Prianto, M. Kusnadi, "Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada *Mobile E-Parking* Berbasis Sistem Informasi", *Jurnal Informatika*, vol.03, no. 03, p 329, 2018.
- [9] L. Marco, C Banabera, "Pewarnaan titik pada korona graf kipas Dengan Graf Kipas Dan Graf Buku Segitiga Dan Graf Buku Segitiga Berorder Sama", *Jurnal Axiomath*, Vol 2, No.2, p. 17, 2020.
- [10] R. Munir, *Matematika Diskrit*, Bandung: Informatika Bandung, 2010
- [11] Z. Majesty, D. W Putri, "Website Organisasi IMMIG Menggunakan Django Framework", *Jurnal Ilmiah*, p.4, 2020
- [12] H. E. Saputra, "Sistem Informasi Geografi Pemetaan Jaringan Pipa Air Berbasis Mapbox GL JS", *Tugas Akhir*, p. 12, 2017