

---

## PENCARIAN TEMPAT IBADAH TERDEKAT MENGGUNAKAN ALGORITMA DIJKSTRA

Bagas Lutfi Ainur Rofiq<sup>1\*</sup>, Herny Februariyanti<sup>2</sup>, Jati Sasongko Wibowo<sup>3</sup>, Zuly Budiarmo<sup>4</sup>  
Program Studi Sistem Informasi, Universitas Stikubank, Semarang<sup>1,2,3,4</sup>  
e-mail: bagaslutfi1997@gmail.com<sup>1</sup>, hernyfeb@edu.unisbank.ac.id<sup>2</sup>,  
jatisw@edu.unisbank.ac.id<sup>3</sup>, 4zulybudiarmo@edu.unisbank.ac.id<sup>4</sup>

### Abstrak

Untuk mengetahui lokasi tempat ibadah biasanya para pendatang dengan bertanya kepada masyarakat setempat. Pencarian jalur terpendek merupakan suatu problematika dalam sistem yang mengacu pada titik awal dan titik akhir. Tidak terkecuali sistem untuk mencari tempat ibadah terdekat. Penelitian ini digunakan untuk melakukan pencarian rute terdekat tempat ibadah dengan menggunakan algoritma dijkstra. Algoritma Dijkstra sebuah algoritma yang dapat melakukan pencarian suatu rute terpendek dari sebuah graf di setiap simpul bobot yang terpendek dengan berdasarkan bobot paling kecil dari sebuah titik ke titik yang lainnya. Algoritma dijkstra juga sangat efisien dibandingkan dengan algoritma yang lainnya dalam melakukan pencarian path terpendek. Tujuan dari penelitian ini adalah memudahkan pengguna dalam melakukan pencarian jalur terpendek menuju tempat ibadah yang diinginkan. Rancangan sistem ini untuk mengetahui lokasi tempat ibadah dan rute terdekat dengan menggunakan algoritma Dijkstra untuk menampilkan peta, serta fitur unggulan dalam pengoperasian aplikasi yang dapat melakukan pencarian lokasi tempat ibadah yang terdekat dari posisi pengguna. Sistem akan memberikan panduan jalan yang harus dilewati oleh pengguna yang dihasilkan dari perhitungan menggunakan algoritma dijkstra. Hasil penelitian pengguna akan mendapatkan rekomendasi tempat ibadah terdekat sesuai Agama yang dipilih, beserta rute yang direkomendasikan oleh sistem. Hasil perhitungan manual dengan perhitungan dari sistem didapatkan perbedaan hasil yaitu dengan selisih 0.01 km.

**Kata Kunci :** *Algoritma Dijkstra*; Jalur Terpendek; Tempat Ibadah

### Abstract

*To find out the location of places of worship usually pendatag by asking the local community. The search for the shortest path is a problem in the system that refers to the starting point and end point. The system is no exception to finding the nearest place of worship. This research uses the dijkstra algorithm to find the nearest route to a place of worship. Dijkstra algorithm is an algorithm that can be used to search for the shortest route a graph at each vertex of the shortest weight based on the smallest weight from one point to another. The Dijkstra algorithm is also very efficient compared to other algorithms in searching for the shortest path. This study aims to make it difficult for users to find a path to get to the place of worship. In the design of this Web-based application to find out the nearest place of worship and the route will be displayed on the map, the excellent feature of this application is the operation to find the location of the nearest place of worship from the user's current position and also this system provides a guide to the road the user is using dijkstra. The results of the user's research will get a recommendation for the nearest place of worship according to the chosen religion and the route recommended by the system. The results of manual calculations with calculations from the system obtained a difference in the results, namely a difference of 0.01 km.*

**Keywords :** *Dijkstra Algorithm, Shortest Path, Place of Worship*

## I. PENDAHULUAN

Teknologi pada masa kini sangat membantu dan mempermudah kegiatan manusia sehari-hari, seperti pencarian data, untuk komunikasi serta pencarian suatu tempat. Pemanfaatan internet dalam pencarian suatu tempat sangat dibutuhkan masyarakat yang kurang paham jalur daerah tertentu.

Tempat ibadah merupakan suatu sarana masyarakat untuk dalam beribadah kepada Tuhan Yang Maha Esa. Sering kali para pendatang ketika mau beribadah tidak tahu keberadaan tempat ibadah tersebut. Untuk mengetahui lokasi tempat ibadah biasanya para pendatang dengan bertanya kepada masyarakat setempat.

Pencarian jalur terpendek merupakan suatu problematika dalam sistem yang mengacu pada titik awal dan titik akhir. Tidak terkecuali sistem untuk mencari tempat ibadah terdekat. Sistem tersebut menggunakan algoritma dijkstra yang memang diperuntukan untuk mencari rute terdekat. Teknologi informasi berbasis WEB merupakan teknologi yang paling berkembang, aplikasi berbasis WEB mempunyai beberapa kelebihan, salah satunya aplikasi berbasis web dapat di akses dengan koneksi internet dan intranet serta ringan dalam penggunaan sumber daya. Sehingga pengguna dapat dengan mudah mengakses data atau informasi dimanapun berada.

Pencarian jalur terpendek pada Sistem Informasi Geografis (SIG) untuk pariwisata menggunakan sebuah algoritma Dijkstra dibuat bertujuan untuk membantu kementerian pariwisata Timor Leste untuk pengembangan industri pariwisata dibuat agar pengguna dapat memperoleh informasi pariwisata dan mengaksesnya dimanapun berada menggunakan internet. Penelitian ini dilakukan pencarian jalur terpendek mulai posisi titik awal lokasi sampai dengan titik akhir lokasi yang diinginkan dengan

menerapkan algoritma Dijkstra menghasilkan keakuratan hasil perhitungan jarak rata-rata sebesar 0.03% [1].

Perhitungan jarak terpendek dari user menuju lokasi SPBU dengan perhitungan yang dilakukan menggunakan algoritma dijktra di dalam sistem dengan cara memilih posisi pengguna pada sebuah peta. Hasil perhitungan akan memberikan rekomendasi jarak terdekat pengguna ke lokasi SPBU, dengan rekomendasi tersebut memberikan efisiensi dalam penggunaan bahan bakar. Nilai kriteria, cost serta *reverse* cost akan mempengaruhi hasil dari menghitung jarak terpendek ddialam pencarian SPBU terdekat. Sedangkan perhitungan jarak dibagi dengan kecepatan akan menghasilkan nilai waktu tempuh [2].

Penelitian dengan penerapan metode Haversine untuk pencarian wisata terdekat yang menghasilkan 2 (dua) rekomendasi lokasi wisata terdekat di kota Padang telah dilakukan. Penelitian menghasilkan detail informasi tempat wisata meliputi nama wisata, jarak tempuh lokasi wisata dari pengguna, alamat, nomor telepon, informasi hotel dan restoran terdekat lokasi wisata, serta deskripsi singkat tentang objek wisata tersebut sangat membantu wisatawan jika ingin berkunjung ke Kota Padang. Implementasi perhitungan algoritma Haversine secara manual dan secara sistem didapatkan hasil yang sama[3].

Penelitian Pencarian Taman Wi-Fi Gratis Kota Semarang telah dilakukan dengan menggunakan Algoritma Dijkstra. Tujuan dari penerapan algoritma ini untuk membantu memberikan informasi kepada user tentang lokasi taman wifi gratis di kota Semarang dan penerapan algoritma dijkstra sangat membantu dalam penentuan rute terpendek serta sangat efisien. Dengan algoritma dijkstra maka akan dihasilkan bobot paling rendah dari rute yang akan dilalui.[4]

Penelitian yang akan dilakukan dengan menggunakan algoritma dijkstra untuk Pencarian Tempat Ibadah dilengkapi dengan menu pilihan untuk tempat ibadah yang diinginkan. Pengguna sistem dapat menentukan atau memilih jenis tempat ibadah yang diinginkan, kemudian sistem akan memberikan rekomendasi beberapa tempat ibadah sesuai yang diinginkan serta memberikan informasi jarak terdekat dari pencarian yang diinginkan. Sistem akan menampilkan peta, aplikasi ini mempunyai fitur utama yang disediakan sistem adalah dari posisi dimana pengguna berada dapat mencari lokasi terdekat saat itu sertadalam pengoperasiannya bahwa pengguna dapat mencari lokasi tempat ibadah terdekat dari posisi pengguna berada saat itu serta adanya panduan jalan yang akan dilewati pengguna.

Menggunakan Algoritma Dijkstra serta memanfaatkan fasilitas Google Maps API, maka sistem dapat menampilkan jalur terpendek yang akan digunakan oleh pengguna menuju ke tempat ibadah yang diinginkan

## II. TINJAUAN PUSTAKA

### 2.1 Algoritma Dijkstra

Algoritma Dijkstra ditemukan oleh Edsger Wybe Dijkstra pada tahun 1959. Algoritma Dijkstra merupakan algoritma yang dapat menyelesaikan permasalahan dalam pencarian jalur terpendek dari suatu graf disetiap simpul bobot paling pendek dengan berdasar pada bobot paling kecil dari suatu titik menuju ke titik lainnya. Algoritma Dijkstra ini akan mencari path paling pendek serta dibandingkan dengan algoritma *Wharsall* maka algoritma Dijkstra dianggap lebih efisien. Algoritma *Dijkstra* digunakan dalam melakukan pencarian lintasan terpendek pada graf berarah. Akan tetapi algoritma Dijkstra dapat juga digunakan untuk graf yang tdk berarah. Algoritma Dijkstra menggunakan prinsip greedy.[5] Prinsip greedy pada algoritma djikstra

bahwa disetiap langkah kita memilih sisi yang berbobot paling rendah (minimum) dan akan dimasukkan kedalam himpunan solusi.

Pertama kali akan ditentukan titik mana yang akan menjadi node awal, kemudian beri bobot jarak pada *node* terdekat satu per satu, Selanjutnya algoritma akan melakukan pengembangan pencarian dari satu titik menuju ke titik lainnya serta menuju ke titik selanjutnya setahap demi tahap. Urutan logika dari Algoritma Dijkstra sebagai berikut [6] :

1. Beri penilaian bobot (jarak) untuk setiap titik terhadap titik lainnya, kemudian pada *node* awal diberikan nilai 0 dan nilai tak hingga terhadap *node* lainnya.
2. Langkah kedua *Set* semua *node* yang belum terjamah dan set *node* awal sebagai *node* keberangkatan.
3. Selanjutnya dari *node* keberangkatan tersebut, pertimbangkan *node nearest* (tetangga) yang belum terjamah, kemudian hitung jarak dari titik keberangkatan
4. Kemudian tandai *node* yang sudah terjamah sebagai *node* terjamah setelah selesai dipertimbangkan setiap jarak terhadap *node nearest* (tetangga),. Selanjutnya *Node* terjamah tidak akan lagi dicek, jarak yang tersimpan merupakan jarak terakhir serta merupakan jarak yang paling terendah bobotnya.
5. Kemudian *Set node* yang belum terjamah dengan jarak terkecil (dari *node* keberangkatan) yang selanjutnya disebut sebagai *node* keberangkatan, kemudian dilanjutkan kembali ke langkah 3.

## 2.2 Jalur Terpendek (*Shortest Path Problem*)

Proses perhitungan jalur (rute) terpendek merupakan proses melakukan pencarian jarak terpendek atau mencari biaya terkecil dari suatu jalur (rute) dari node awal menuju ke node tujuan atau node akhir dari sebuah jaringan. Untuk melakukan proses perhitungan jalur terpendek meliputi 2 (dua) jenis proses, yaitu proses pemeriksaan node lalu pemberian label. Sedangkan yang dimaksud dengan metode pemberian label adalah metode untuk memberikan suatu identifikasi terhadap masing-masing node dalam suatu jaringan. Di sebagian besar algoritma perhitungan jalur terpendek, biasanya terdiri dari 3 (tiga) label informasi yang dikelola untuk setiap node  $i$  pada proses pemberian label yaitu: *parent node*  $p(i)$ , label jarak  $d(i)$ , status node  $S(i)$  [2]. Sudah banyak penelitian terkait algoritma yang dapat digunakan dalam melakukan pencarian jalur terpendek. Contoh algoritma yang dapat digunakan dalam penyelesaian menentukan jalur terpendek terdiri dari dijkstra's algorithm, floyd-warshall algorithm, a\* search algorithm, dan bellmanford's algorithm [7].

## 2.3 Google Maps API

Dalam menciptakan suatu rancangan sistem aplikasi yang terkait dengan pemetaan atau petunjuk arah suatu lokasi Google Maps telah memberikan paket yang lengkap untuk kebutuhan tersebut. Dengan memanfaatkan layanan Maps API (Application Programming Interface) banyak sekali fitur-fitur yang telah disediakan oleh Google Maps membuat kita bisa berkreasi. Dengan adanya layanan yang disediakan dan banyaknya fitur yang disediakan tersebut akan sangat menghemat waktu bagi perancang sistem aplikasi, hal ini dimungkinkan karena perancang sistem tidak perlu lagi membuat peta dari awal.

sebagai pengguna Google Maps API kita dapat menentukan tampilan apa saja yang berhak menentukan tampilan apa saja di Maps, serta dapat menentukan lokasi sesuai yang diinginkan. Dijelaskan juga bahwa Google Maps merupakan salah satu fasilitas yang telah disediakan Google dengan adanya layanan pemetaan daerah. Pemetaan tersebut dilengkapi dengan berbagai kemampuan dan mudah sangat mudah untuk digunakan. Kelengkapan pendukung lainnya dari peta yang tersedia tersebut meliputi layanan informasi jalan, lokasi, bisnis, jasa, layanan public, dan masih banyak layanan yang disediakan [8]. Google Maps juga merupakan suatu library yang berbentuk *JavaScript*. Untuk dapat menampilkan Google Maps di halaman web atau blog sangat mudah dilakukan, hanya saja diperlukan pengetahuan mengenai HTML dan *JavaScript*, selanjutnya harus ada juga koneksi Internet. Penggunaan Google Maps sangat menghemat waktu dan biaya dalam mengembangkan aplikasi digital sehingga kita hanya focus pada data-data yang ingin ditampilkan. Kita dapat sangat menghemat waktu, dikarenakan kita hanya perlu membuat suatu data atau atribut serta membuat peta yang akan ditampilkan merupakan milik Google sehingga tidak dipusingkan dengan membuat peta baru dalam pengembangan dari suatu lokasi

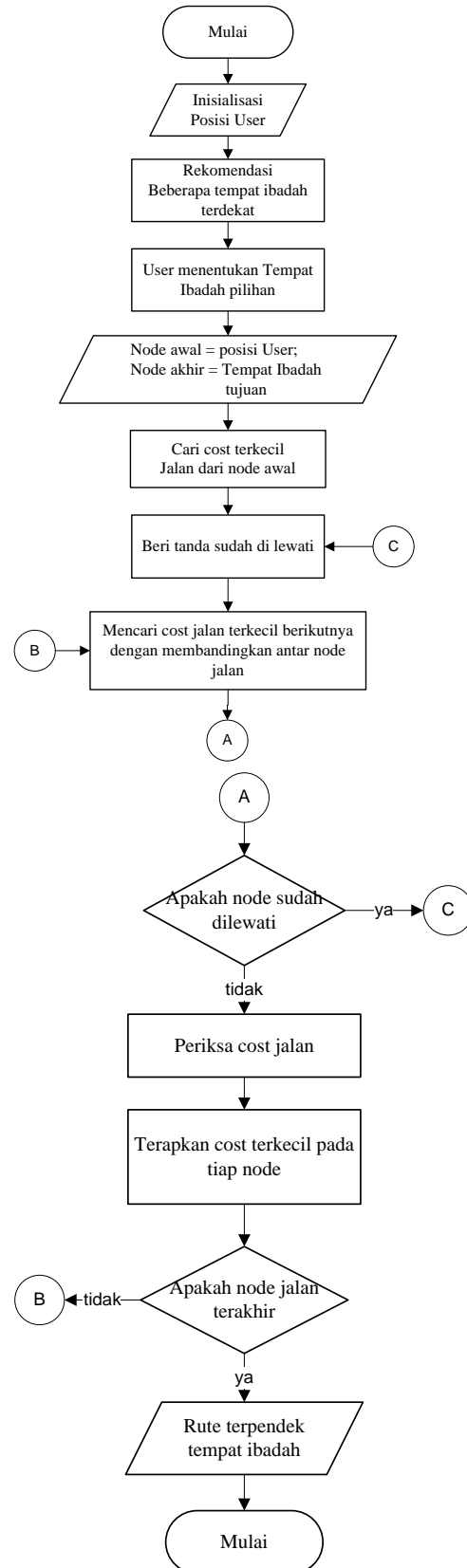
Google maps secara online memberikan layanan peta yang disediakan oleh Google dalam bentuk tampilan peta yang secara gratis dapat dimanfaatkan dengan menggunakan teknologi foto satelit sehingga planet bumi dapat terlihat secara landscape. Google maps memberikan fasilitas diantaranya adalah menjelajah peta serta mencari lokasi tertentu, seperti pencarian restoran, hotel serta menghitung jalur (rute) dalam berkendara Google maps dengan menggunakan bahasa pemrograman HTML, javascript dan bahasa pemrograman lainnya dibuat kombinasi dari gambar peta,

database, serta objek-objek secara interaktif.[9]

### III. METODOLOGI PENELITIAN

Penelitian ini dimulai dengan pengumpulan data yaitu data koordinat tempat ibadah yang berada dikota Semarang. Data lain yang dibutuhkan adalah data jalan, data alamat lengkap, nomor telepon (jika ada), foto atau gambar tempat ibadah, serta deskripsi dari tempat ibadah. Data tersebut difungsikan untuk ditampilkan kepada user atau pengguna yang mencari tempat ibadah. Untuk data koordinat tempat ibadah, dilakukan dengan menggunakan google maps, sehingga harus mendukung GPS. Sementara untuk data pendukung lainnya dilakukan dengan mewawancarai pengurus tempat Ibadah.

Data koordinat yang telah didapat melalui google maps, secara spesifiknya digunakan untuk mengetahui letak tempat ibadah didalam website, serta digunakan untuk menghitung jarak antara posisi awal pengguna dengan letak semua tempat ibadah. Sehingga nantinya didapatkan jarak paling dekat antara pengguna dan tempat ibadah. Perhitungan yang digunakan untuk mencari jarak terdekat tersebut menggunakan algoritma dijkstra. Detail proses penentuan jalur terpendek menuju Tempat Ibadah menggunakan Algoritma Dijkstra dapat dilihat pada gambar 1



**Gambar 1** Flowchart penentuan jalur terpendek Tempat Ibadah  
Sumber : Hasil Penelitian (2019)

Pada tahap pengembangan sistem penelitian menggunakan metode pengembangan sistem *waterfall*. Pendekatan pada metode *waterfall* dilakukan pendekatan secara sistematis dan berurutan dalam membangun sebuah sistem. Dengan menggunakannya metode pengembangan sistem berbasis *waterfall*, maka peneliti akan menerapkan langkah-langkah dalam pengembangan sistem, antara lain [10]:

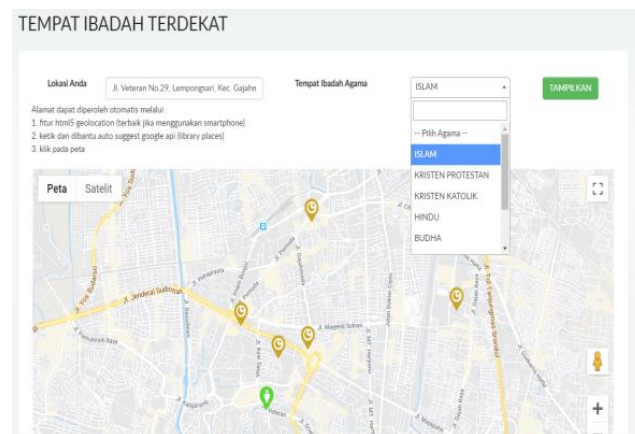
1. Analisa  
Langkah ini merupakan analisa terhadap masalah yang terjadi. Sehingga akan di dapat data mengenai kebutuhan sistem. Pada tahap analisa dilakukan survey untuk mendapatkan data-data seperti lokasi tempat ibadah, serta mendapatkan peta lokasi dengan menggunakan fasilitas Google Maps
2. Design  
Tahap yang kedua adalah tahap perancangan sistem atau mendesain sistem. Tujuan dari tahap design ini memberikan gambaran tahapan proses yang akan di kerjakan dan bentuk tampilan sistemnya. Pada tahap ini dilakukan proses desain pemetaan lokasi tempat ibadah serta menentukan simpul-simpul koordinat dari tempat ibadah. Selain itu juga dibuat rancangan untuk aplikasinya menggunakan tools UML.
3. Pembuatan Aplikasi  
Tahap ini merupakan tahap dimana pembuatan perangkat lunak melalui serangkaian program atau unit program yang di harapkan dapat memecah masalah yang sedang di teliti. Pada tahap ini dilakukan pembuatan program dengan menerapkan algoritma Dijkstra untuk dapat memberikan rekomendasi tempat ibadah terdekat.
4. Pemeliharaan  
Merupakan tahap akhir dari metode *waterfall*. Sistem sudah mulai di

implementasikan. Pemeliharaan yang dilakukan mencakup perbaikan terhadap berbagai *error* yang di temukan secara *minor*, termasuk perbaikan saat implementasi dan persiapan pengembangan sistem, serta pemeliharaan terhadap program yang berjalan.

#### IV. HASIL DAN PEMBAHASAN

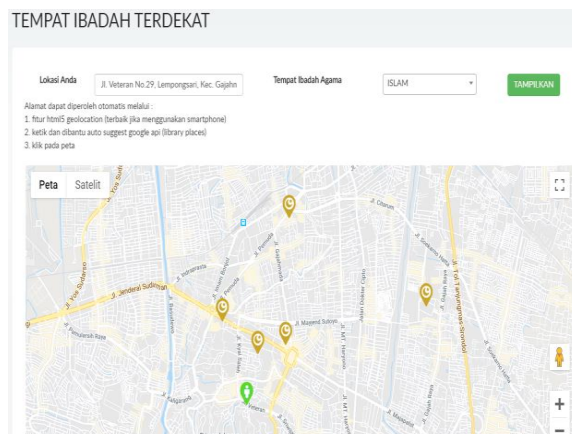
##### 4.1 Hasil

Hasil penelitian ini adalah merekomendasikan tempat Ibadah terdekat dari posisi pengguna saat ini. Proses rekomendasi dimulai dari pengguna memilih jenis Agama sesuai keinginan yang terdiri dari Agama Islam, Kristen Khatolik, Kristen Protestan, Hindu dan Budha. Sebagai contoh Agama yang dipilih pengguna adalah Agama Islam yang diperlihatkan seperti Gambar 2



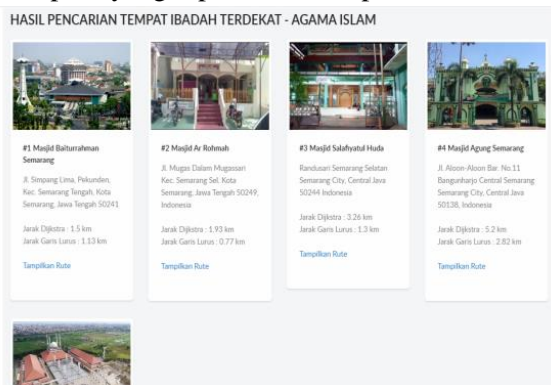
**Gambar 2** Pemilihan jenis Agama Islam  
Sumber : Hasil Penelitian (2019)

Proses selanjutnya adalah memilih lokasi pengguna yang diinginkan. Untuk memilih lokasi pengguna, pengguna dapat mengklik satu kali pada gambar peta *google map*. Sebagai contoh lokasi pengguna yang dipilih berada pada Alamat Jl. Veteran Lemponsari, Kec. Gajahmungkur, kota Semarang, Jawa Tengah 50231, Indonesia. Lokasi pengguna yang dipilih diperlihatkan seperti Gambar 3



**Gambar 3** Posisi Pengguna  
Sumber : Hasil Penelitian (2019)

Selanjutnya Sistem akan menghitung jarak dari lokasi pengguna ke tempat ibadah dengan algoritma dijkstra. Hasil rekomendasi akan menampilkan tempat dengan nomer urut yang sesuai jarak yang paling terdekat dari lokasi pengguna yang dipilih yang diperlihatkan seperti Gambar 4

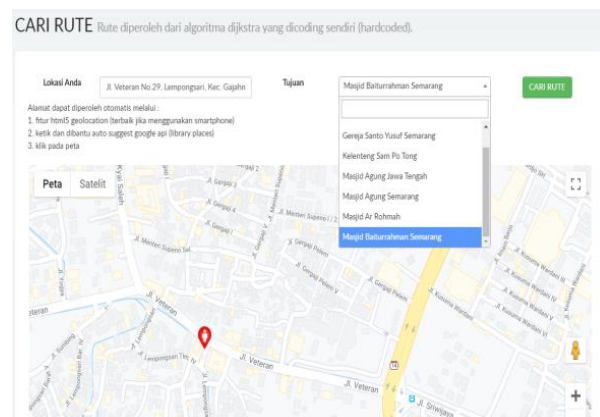


**Gambar 4** Hasil Rekomendasi  
Sumber : Hasil Penelitian (2019)

Pada Gambar 4 Masjid yang terdekat dengan pengguna adalah masjid baiturrahman dengan jarak 1.5 km, yang terdekat selanjutnya Masjid Ar Rahmah dengan jarak 1.93 km. hasil rekomendasi ini sistem akan menampilkan tempat ibadah dengan nomor urut sesuai dengan jarak terdekat sama pengguna.

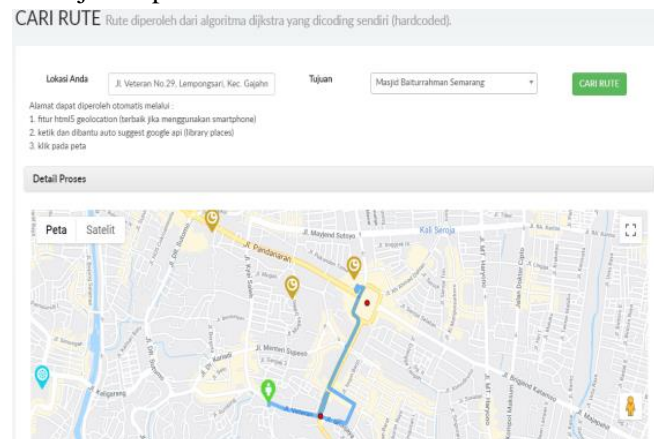
Proses selanjutnya adalah mencari rute yang direkomendasikan oleh sistem. Pertama pengguna memilih menu cari rute, pengguna memilih lokasi yang diinginkan,

setelah itu pengguna memilih tempat tujuan yang diinginkan. Cari Rute di perlihatkan Pada Gambar 5



**Gambar 5** Cari Rute  
Sumber : Hasil Penelitian (2019)

Selanjutnya sistem akan menampilkan jalur yang telah direkomendasikan. Berikut hasil Cari Rute di tunjukan pada Gambar 6

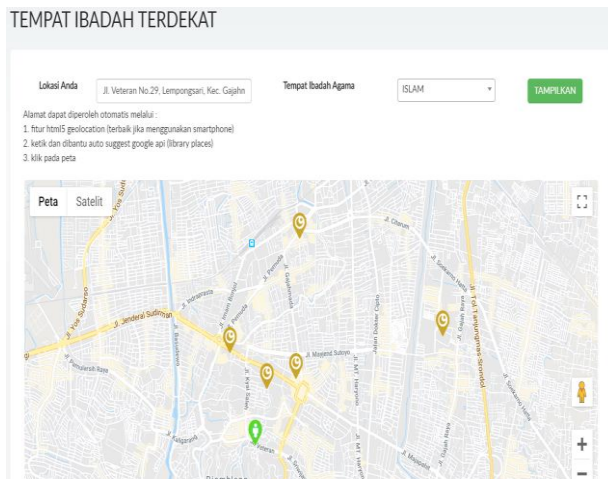


**Gambar 6** Hasil Cari Rute  
Sumber : Hasil Penelitian (2019)

#### 4.2 Pembahasan

Proses perhitungan jarak pengguna ke tempat ibadah dan dapat merekomendasikan Rute terdekat dengan menggunakan algoritma dijkstra sebagai berikut:

Pada penelitian ini diberikan sebuah contoh kasus detail untuk perhitungan jalur terdekat pada tempat ibadah. Langkah pertama dengan menentukan posisi pengguna dan posisi tempat ibadah terdekat yang dapat dilihat pada gambar 7.



**Gambar 7** Posisi Pengguna dengan Tempat Ibadah Terdekat  
Sumber : Hasil Penelitian (2019)

Langkah selanjutnya adalah membandingkan posisi geometry dari pengguna sistem dengan lokasi tempat ibadah untuk mendapat jarak. Nilai perhitungan hasil perbandingan jarak dari penelitian yang dilakukan dapat dilihat pada tabel 1. Hasil dari perbandingan yang telah dilakukan, selanjutnya akan dihitung jalur (rute) terpendek. Perhitungan jalur terpendek ini dengan cara menghitung panjang dari masing-masing jalan (rute). Pada penelitian ini, setiap persimpangan jalan merupakan sebuah vertex.

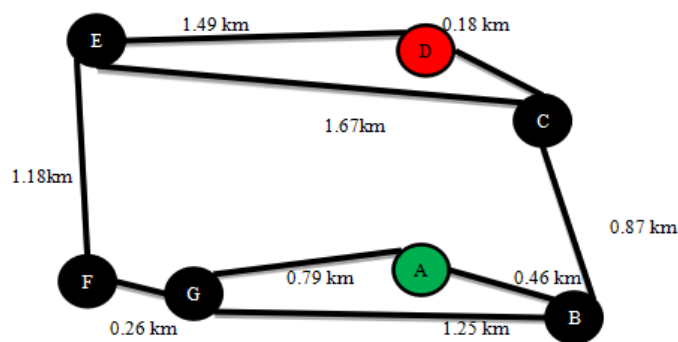
Kemudian dilanjutkan dengan melakukan perhitungan menggunakan algoritma Dijkstra untuk mendapatkan jalur (rute) terpendek. Perhitungan ini direpresentasikan melalui sebuah *graph*. Jalur (jalan) antara Pengguna sistem dengan Tempat Ibadah yang dituju akan digambarkan dengan menggunakan *edge*, sedangkan setiap persimpangan akan digambarkan dengan *vertex*.

**Tabel 1** Hasil Perbandingan Jarak Geometry

No	Nama	Tempat Ibadah		Pengguna		
		Latitudo	Longtitude	Latitudo	Longtitude	Jarak
1	Masjid Ar Rohmah	-6.990163317095965	110.41733443737703	-6.996651229676784	110.1536569595337	1.28 km
2	Masjid Baiturrahman	-6.988930683388142	110.422006854742	-6.996651229676784	110.1536569595337	1.13 km
3	Masjid Salafiyatul Huda	-6.98588502903951	110.412833738327	-6.996651229676784	110.1536569595337	2.81 km
4	Masjid Agung Semarang	-6.972466858223718	110.42260766029358	-6.996651229676784	110.41536569595337	3.66 km
5	Masjid Agung Jawa Tengah	-6.9839026	110.4459023	-6.996651229676784	110.41536569595337	0.75 km

Sumber : Hasil Penelitian (2019)

Notasi *graph* hasil representasi pencarian jalur terpendek tempat ibadah dapat dilihat pada Gambar 8



**Gambar 8** Notasi Graph pencarian jalur terdekat

Sumber : Hasil Penelitian (2019)

Pada Gambar 8 Titik A berwarna hijau merupakan pengguna dan Titik D berwarna merah tujuan. Penentuan jalur terpendek pada gambar 8 dengan



menggunakan algoritma Dijkstra menghasilkan rute terpendek  $A > B > C > D$ . untuk detail Perhitungannya dapat dilihat pada Tabel 2.

**Tabel 2** Hasil Perhitungan Dijkstra

No	Vertex Sumber	Jarak Dengan Vertex Lain	
1	A	A – B	0.46
		A – G	0.79
2	B	B – C	$0.87+0.46=1.33$
		B – G	$1.25+0.46=1.71$
3	C	C – D	$0.18+1.33=1.51$
		C – E	$1.67+1.33=3$
4	D	D – E	$1.49+1.51=3$
5	E	E – F	$1.18+3=4.18$
6	F	F – G	$0.26+4.18=4.44$

Sumber : Hasil Penelitian (2019)

## V. KESIMPULAN

Dari apa yang telah diuraikan diatas maka dapat ditarik kesimpulan sebagai berikut :

1. Pada Sistem ini pengguna akan mendapatkan rekomendasi Tempat Ibadah Terdekat sesuai Agama yang dipilih, beserta Rute yang direkomendasikan oleh sistem.
2. Hasil perhitungan manual dengan perhitungan Sistem Pencarian Tempat Terdekat untuk merekomendasikan Tempat Terdekat Menggunakan Algoritma Dijkstra didapatkan hasil yang tidak jauh berbeda

## VI. SARAN

Berikut ini merupakan saran dari penulis terhadap pengembangan Sistem lebih lanjut :

1. Sistem informasi pencarian tempat ibadah untuk merekomendasikan tempat terdekat menggunakan Algoritma Dijkstra dapat dikembangkan tidak hanya tempat

ibadah saja, bisa ditambah dengan tempat wisata.

2. Membandingkan dengan metode pengukuran jarak lainnya.

## VII. DAFTAR PUSTAKA

- [1] A. Gusmão and S. H. Pramono, "Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra," vol. 7, no. 2, pp. 125–130, 2013.
- [2] W. E. Y. Retnani, D. Istiadi, and A. Roqib, "Pencarian SPBU Terdekat dan Penentuan Jarak Terpendek Menggunakan Algoritma DIJKSTRA (Studi Kasus di Kabupaten Jember)," *J. Nas. Tek. Elektro*, vol. 4, no. 1, p. 89, 2015, doi: 10.25077/jnte.v4n1.132.2015.
- [3] E. Putri and H. Februariyanti, "Sistem Rekomendasi Tempat Wisata Kota Padang Dengan Haversine," *Proceeding SENDIU (Seminar Nas. Multi Disiplin Ilmu) 2020 Univ. Stikubank Semarang*, pp. 106–114, 2020, [Online]. Available: <https://www.unisbank.ac.id/ojs/index.php/sendiu/article/view/7968>.
- [4] C. Skolastika and H. Februariyanti, "Pencarian Jalur Taman Wifi Gratis Kota Semarang Menggunakan Algoritma Dijkstra," *Informations Technol. Informatics*, vol. 9, pp. 0–3, 2013.
- [5] B. Junanda, D. Kurniadi, and Y. Huda, "Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra pada Sistem Informasi Geografis Pemetaan Stasiun Pengisian Bahan Bakar Umum," *J. Vokasional Tek. Elektron. Inform.*, vol. 4, no. 1, pp. 1–8, 2016.
- [6] Ferdiansyah and R. Ahmad, "Penerapan Algoritma Dijkstra untuk Menentukan Rute Terpendek

- Pembacaan Water Meter Induk PDAM Tirta Kerta Raharja Kabupaten Tangerang,” *J. TICOM*, vol. 2, no. 1, pp. 51–57, 2013.
- [7] S. Sanan, L. Jain2;, and KapporBharti, “Shortest Path Algorithm Comparison,” *Int. J. Appl. or Innov. Eng. Manag.*, vol. 2, no. 7, pp. 316–320, 2011, [Online]. Available: <http://rebustechologies.com/shortest-path-algorithm-comparison/>.
- [8] M. A. Husaini and W. Dwi P, “Sistem Informasi Geografis (Sig) Pemetaan Sekolah Berbasis Web Di Kecamatan Wonodadi Kabupaten Blitar,” *ANTIVIRUS J. Ilm. Tek. Inform.*, vol. 11, no. 1, pp. 50–64, 2017, doi: 10.30957/antivirus.v11i1.198.
- [9] G. an A. ddiw inoto; H. W. intolo; D. N. Ugraheny, “Pemanfaatan Direction Api ( Application Programming Interface ) Pada Layanan Google Map Untuk Pencarian,” *Compiler*, vol. 1, pp. 21–28, 2012, [Online]. Available: <http://ejournals.stta.ac.id/index.php/compiler/article/view/14/14>.
- [10] A. Rifai and Y. P. Yuniar, “Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Ujian Pada SMK Indonesia Global Berbasis Web,” *J. Khatulistiwa Inform.*, vol. 7, no. 1, pp. 1–6, 2019, doi: 10.31294/jki.v7i1.64.