

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam penelitian pencarian rute terpendek dengan menggunakan Algoritma Dijkstra ini akan digunakan beberapa tinjauan studi yang nantinya mendukung penelitian yang akan dilakukan seperti yang ada pada tabel 2.1

Tabel 2.1 Perbandingan Metode Penelitian

Peneliti, Tahun	Objek/Data	Metode	Teknologi	Hasil
Benyamin Jago Belalawe, M.Suyanto, Amir Fatah Sofyan, 2012	Jalur Wisata Terpendek Dinas Pariwisata Kota Kupang	<i>DFS (Depth-First Search)</i>	Berbasis WEB	Menggunakan metode Forward Chaining yang Digunakan untuk pengambilan keputusan jalur terpendek.
Dwi Ardana, Ragil Saputra, 2016	Pencarian Rute Bus Trans Semarang	Algoritma Dijkstra	Berbasis WEB	Jarak dihitung dengan metode <i>Haversaine</i> dan node-node rute akan ditampilkan pada peta.
Fahri Hikmawan Fuady, 2017	Rute Terpendek Bengkel Mobil	Algoritma Dijkstra	Android	Jalur tependek bengkel mobil di Yogyakarta dengan jarak dihitung dengan metode <i>Harvesine</i> .
Helga Aditya Rizqi Geovani, 2016	Mengetahui Lokasi Tempat Ibadah Umat Muslim Di Kota Malang	Algoritma Dijkstra	<i>Mobile Phone</i>	Dibangun menggunakan <i>IDE Eclipse</i> dan <i>Android SDK</i> sebagai <i>Development tools</i> . Output dari penelitian berupa peta.

Tabel 2.1 Perbandingan Metode Penelitian (Lanjutan)

Shaga Bogas Priatmoko,	Jalur Terdekat Dan Rekomendasi Objek Pariwisata Di Pulau Bali	Algoritma Dijkstra	Berbasis WEB	Selain rute terpendek dari lokasi awal ke lokasi akhir, aplikasi ini juga terdapat tombol untuk mencari rute rekomendasi beserta jarak antar lokasi pariwisata.
Novitasari, 2019	Rute Terpendek Tempat Wisata Di Gunung Kidul	Algoritma Dijkstra	Berbasis WEB	Lokasi awal bersifat dinamis dengan jarak dihitung dengan menggunakan metode <i>euclidean</i> dan output berupa peta.

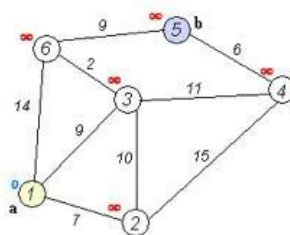
Dalam penelitian ini penulis melakukan pencarian rute terpendek tempat wisata di Gunung kidul menggunakan metode Algoritma *Dijkstra* yang akan diimplementasikan ke dalam aplikasi berbasis web dengan titik awal dan nama tempat wisata diinputkan oleh wisatawan.

2.2 Dasar Teori

2.2.1 Algoritma Dijkstra

Menurut Dwi Ardana dan Ragil Saputra (2016), Algoritma Dijkstra adalah sebuah algoritma greedy yang dipakai dalam memecahkan permasalahan jarak terpendek untuk sebuah graf berarah dengan bobot-bobot sisi (edge) yang bernilai tak negatif. Ide dasar algoritma Dijkstra sendiri ialah pencarian nilai cost yang terdekat dengan tujuan yang berfungsi pada sebuah graf berbobot, sehingga dapat membantu memberikan pilihan jalur. Pada Algoritma Dijkstra, node digunakan karena Algoritma Dijkstra menggunakan graph berarah untuk penentuan rute lintasan terpendek. Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka Algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik. Pada

Gambar 2.1 disajikan contoh graf dengan bobotnya dalam menentukan jalur menggunakan Algoritma Dijkstra (Dwi Ardana dan Ragil Saputra, 2016).



Gambar 2.1 Contoh Menemukan Jalur Menggunakan Algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Algoritma Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Urutan logika dari Algoritma Dijkstra sebagai berikut :

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai takhingga terhadap node lain (belum terisi).
2. *Set* semua node belum terjamah dan *set* node awal sebagai node keberangkatan.
3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai node terjamah. Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. *Set* node belum terjamah dengan jarak terkecil (dari node keberangkatan) sebagai node keberangkatan selanjutnya dan lanjutkan dengan kembali ke *step* 3.

2.2.2 Metode Euclidean

Menurut Rahayu Ningati (2014), Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam Euclidean *space*. Euclidean *space* diperkenalkan oleh seorang matematika Wadani Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini biasanya diterapkan pada 2 dimensi dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Rumus Euclidean tertulis pada persamaan (1) (Rahmad Ramdhani, 2016).

$$n = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Keterangan persamaan (1) :

n : jarak

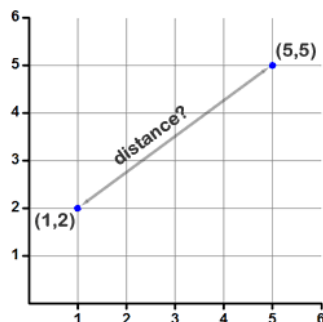
x_1 : kordinat latitude titik 1

x_2 : kordinat latitude titik 2

y_1 : kordinat longitude titik 1

y_2 : kordinat longitude titik 2

Misalkan titik pertama mempunyai kordinat (1,2). Titik kedua ada di kordinat (5,5). Caranya adalah kurangkan setiap kordinat titik kedua dengan titik yang pertama. Yaitu, (5-1,5-2) sehingga menjadi (4,3). Kemudian pangkatkan masing masing sehingga memperoleh (16,9). Kemudian tambahkan semuanya sehingga memperoleh nilai $16+9 = 25$. Hasil ini kemudian diakarkan menjadi 5. Sehingga jarak euclideannya adalah 5. Gambar kurva yang dihasilkan dari contoh diatas disajikan dalam gambar 2.2



Gambar 2.2 Kurva Euclidean Distance

Sehingga untuk mencari jarak antar koordinat dari persamaan rumus (1) diatas dapat diimplementasikan menjadi :

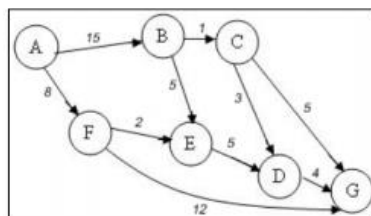
$$\text{Jarak} = \sqrt{(\text{Lat}_1 - \text{Lat}_2)^2 + (\text{Long}_1 - \text{Long}_2)^2}$$

2.2.3 Rute Terpendek

Rute terpendek diartikan sebagai lintasan yang mempunyai biaya terkecil suatu rute dari node awal ke node tujuan dalam sebuah jaringan (Siswanto, 2013:383).

Pada proses penghitungan rute terpendek terdapat dua macam proses yaitu proses pemberian label dan proses pemeriksaan node. Metode pemberian label adalah metode untuk memberikan identifikasi pada setiap node dalam jaringan.

Proses pemberian label berjalan seiring dengan proses scanning (pemeriksaan). Proses pemeriksaan node adalah proses membandingkan jarak antara node awal s dengan node i melalui node j sebagai node lain dalam suatu jaringan.



Gambar 2.3 Contoh Graph Jalan

Pada Gambar 2.3, node A akan dianggap sebagai node awal dan node G dianggap sebagai *node* tujuan. *Node* A mempunyai label status r (permanen), label jarak $d(s) = 0$, dan

label parent $p(s) = 0$, oleh karena itu node A dianggap sebagai node awal. Node B dan *node* F mempunyai label status t (temporal) yang berarti node tersebut telah melalui proses pemberian label tetapi belum melalui proses pemeriksaan. Node C, D, E dan G mempunyai label status u (unreached), label jaraknya $d(i) = \infty$, dan label parent $p(i) = \text{null}$, karena node-node tersebut belum melalui proses pemberian label dan proses pemeriksaan. Pada proses pemeriksaan node B dan node F, akan dipilih node dengan nilai bobot yang terkecil yaitu node F, oleh karena itu, maka label status node F, $s(b)$, akan berubah menjadi r , label parent, $p(b)$, menjadi A, dan label jaraknya, $d(b)$ menjadi 8. Proses ini akan terus berlangsung sampai node tujuan (Rahayu Ningati, 2014).

2.2.4 Google Maps

Adalah layanan aplikasi peta online yang disediakan oleh Google secara gratis. Pada Google Maps tersebut dilihat informasi geografis pada hampir semua permukaan di bumi kecuali daerah kutub utara dan selatan. Layanan ini di buat dengan interaktif, karena di dalamnya peta dapat di geser sesuai keinginan pengguna, mengubah level *zoom*, serta mengubah tampilan jenis peta.

Google Maps API memungkinkan pengembang untuk mengintegrasikan Google Maps ke dalam aplikasi yang dibuatnya. Dengan menggunakan Google Maps API memungkinkan untuk menanamkan situs Google Maps ke situs eksternal, dan situs tersebut dapat dilakukan *Overlay*.

Meskipun pada awalnya hanya JavaScript API, Google Maps API diperluas untuk menyertakan API untuk Adobe Flash, layanan untuk mengambil gambar peta statis dan layanan web untuk melakukan geocoding, menghasilkan petunjuk rute.

Menurut Nurul Azmi (2013), Beberapa tujuan dari penggunaan Google Maps API adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain

sebaginya. Hampir semua hal yang berhubungan dengan peta dapat memanfaatkan Google Maps.