

## **BAB 2**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1. Tinjauan Pustaka**

Dalam melaksanakan penelitian ini, diambil beberapa referensi sebagai dasar pelaksanaan penelitian, ada banyak tesis yang mengangkat penelitian tentang pencarian rute terdekat menggunakan algoritma Dijkstra.

I Nyoman Prama Pradnyana (2010), melakukan penelitian tentang Pencarian Rute Terpendek Tempat Penting Melalui Mobile GMaps dengan Menggunakan Algoritma Dijkstra. Dalam jurnal tersebut dibahas mengenai bagaimana penerapan algoritma Dijkstra dalam menghubungkan dan menampilkan rute-rute yang ditempuh serta cost atau beban biaya minimum, dari suatu titik/vertex/node awal atau disebut juga source node hingga node akhir (tujuan) melalui node-node lain selain kedua node tersebut dalam suatu graf.

Stevian Suryo Saputro (2009), Implementasi algoritma Dijkstra pada pencarian rute wisata di kota Manado. Aplikasi ini dapat mencari rute terpendek dari satu lokasi menuju lokasi lain sehingga dapat meminimalisir biaya perjalanan para wisatawan. Aplikasi dapat menampilkan posisi user, hotel dan lokasi-lokasi wisata serta menampilkan rute menuju hotel dan lokasi wisata tertentu.

Ifatul Faizah (2010), membuat aplikasi Penentuan Rute Perjalanan Wisata Di Malang. Aplikasi tersebut menggunakan algoritma Dijkstra untuk pencarian

dan penentuan rutenya. Aplikasi yang dibuat berbasis desktop dan berjalan pada sistem operasi windows. Hasil yang ditampilkan pada aplikasi tersebut berupa peta dengan rute yang sudah cari.

Fuady, Fahri Hikmawan (2017) Aplikasi Pencarian Rute Terpendek Bengkel Mobil Menggunakan Algoritma Dijkstra. Aplikasi ini bertujuan untuk menghasilkan suatu aplikasi pencari jalur terpendek lokasi bengkel mobil resmi dan tidak resmi di Yogyakarta dengan menerapkan algoritma Dijkstra.

**Tabel 2.1 TINJAUAN PUSTAKA**

<b>Penulis</b>	<b>Obyek</b>	<b>Metode</b>	<b>Hasil</b>
I Nyoman Prama Pradnyana	Pencarian Rute Terpendek Tempat Penting Melalui.	Algoritma Dijkstra	Menampilkan rute-rute yang ditempuh serta cost atau beban biaya minimum.
Stevian Suryo Saputro	pencarian rute wisata di kota Manado.	Algoritma Dijkstra	Menampilkan rute terpendek lokasi hotel dan wisata di Manado
Ifatul Faizah	Penentuan Rute Perjalanan Wisata Di Malang	Algoritma Dijkstra	Menampilkan peta dengan rute yang sudah cari berbasis desktop
Fuady, Fahri Hikmawan	Pencarian Rute Terpendek Bengkel Mobil di Yogyakarta.	Algoritma Dijkstra	SIG lokasi bengkel mobil di Yogyakarta
Al Amin Ali Imron (Diajukan)	Pencarian rute terdekat lokasi wisata di Kabupaten Bantul Yogyakarta	Algoritma Dijkstra	Menampilkan 2 pilihan rute terpendek lokasi wisata Bantul Yogyakarta

## 2.2. DASAR TEORI

### 2.2.1. Algoritma Dijkstra

Pencarian rute terpendek atau *shortest path* termasuk dalam teori graf dimana teori tersebut untuk menentukan rute yang akan dilalui dengan membandingkan simpul atau *node*. Ada berbagai algoritma yang dapat digunakan salah satunya adalah algoritma Dijkstra. Algoritma ini ditemukan oleh seorang ilmuwan komputer berkebangsaan belanda yang bernama Edsger Dijkstra. Algoritma Dijkstra digunakan untuk menentukan jarak terpendek pada sebuah graf berarah.

Sebagai contoh untuk menyelesaikan pertama-tama tentukan titik mana yang akan menjadikan node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu persatu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap inilah urutan logika dari algoritma Dijkstra :

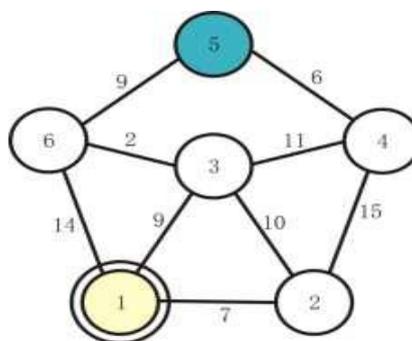
1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum Terjamah” dan set node awal sebagai “Node keberangkatan”
3. Dari no keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C

melewati B menjadi  $6+2=8$ . 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. (Wira setiawan, 2015).

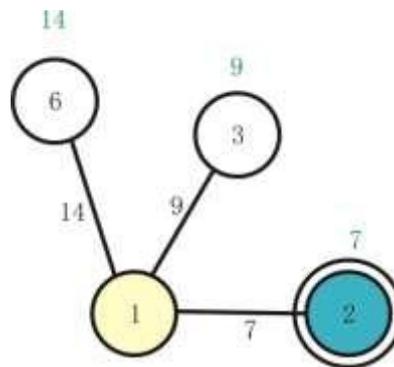
Di bawah ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

1. Node awal 1, Node tujuan 5. Setiap edge yang terhubung antar node telah diberi nilai.



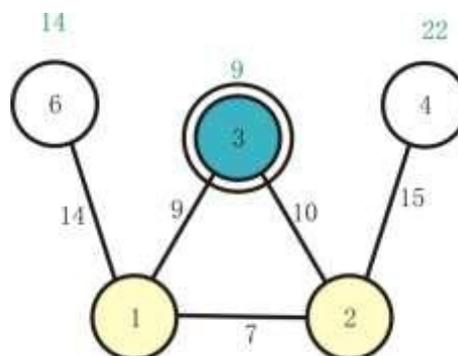
Gambar 2.1 Contoh kasus Dijkstra - Langkah 1

2. Dijkstra melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node 1), dan hasil yang didapat adalah node 2 karena bobot nilai node 2 paling kecil dibandingkan nilai pada node lain, nilai = 7 ( $0+7$ ).



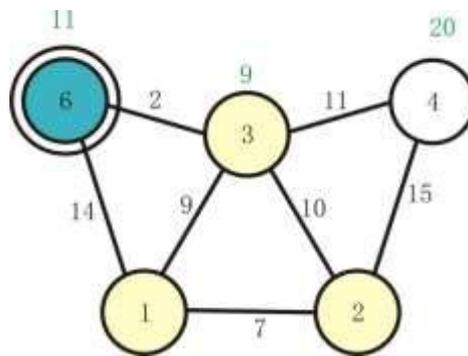
Gambar 2.2 Contoh kasus Dijkstra - Langkah 2

3. Node 2 diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node yang telah terjamah. Dan kalkulasi Dijkstra menunjukkan bahwa node 3 yang menjadi node keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai 9 ( $0+9$ ).



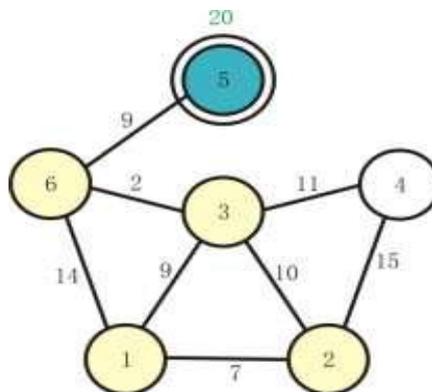
Gambar 2.3 Contoh kasus Dijkstra - Langkah 3

4. Perhitungan berlanjut dengan node 3 ditandai menjadi node yang telah terjamah. Dari semua node tetangga belum terjamah yang terhubung langsung dengan node terjamah, node selanjutnya yang ditandai menjadi node terjamah adalah node 6 karena nilai bobot yang terkecil, nilai 11 ( $9+2$ ).



Gambar 2.4 Contoh kasus Dijkstra - Langkah 4

5. Node 6 menjadi node terjamah, Dijkstra melakukan kalkulasi kembali, dan menemukan bahwa node 5 (node tujuan) telah tercapai lewat node 6. Jalur terpendeknya adalah 1-3-6-5, dan nilai bobot yang didapat adalah 20 ( $11+9$ ). Bila node tujuan telah tercapai maka kalkulasi Dijkstra dinyatakan selesai.



Gambar 2.5 Contoh kasus Dijkstra - Langkah 5

### **2.2.2. Mobile Web**

Mobile web merupakan sebuah kumpulan halaman html dengan berbasis browser yang dapat diakses dengan menggunakan perangkat *portable* seperti *smartphone* dan *gadget / tablet* melalui jaringan internet atau telekomunikasi berupa 3G, 4G, atau wifi. Mobile web ini dirancang untuk menampilkan sebuah text atau konten data gambar, audio dan video atau penggabungan dari beberapa konten yang berada pada layar sebuah *smartphone* dan tentu juga di pengaruhi oleh keterbasan ukuran layar dan jenis layar sentuhnya.

Di dalam mobile web mempunyai prinsip dalam sistemnya yakni *develope once run everywhere* yang dapat diartikan bahwa mobile web ini beroperasi dalam lintas platform untuk sekali pengembangan.

### **2.2.3. Sistem Informasi Geografis (SIG)**

Menurut Aronoff tahun 1997 dalam Prahastha tahun 2001 SIG adalah sistem yang berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografi. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek-objek dan fenomena dimana lokasi geografis merupakan karakteristik yang penting atau kritis untuk dianalisis. Dengan demikian, SIG merupakan sistem komputer yang memiliki empat kemampuan berikut dalam menangani data bereferensi geografi:

- a. Masukan
- b. Manajemen data (penyimpanan dan pemanggilan data)
- c. Analisis dan manipulasi data
- d. Keluaran.

Secara umum SIG dapat diartikan sebagai sistem informasi yang berbasis komputer dalam menyimpan, mengolah, menganalisis, dan menampilkan data.

Sistem Informasi Geografis (SIG) apabila dipisah merupakan gabungan dari 3 kata yaitu:

- Sistem adalah suatu kesatuan komponen atau variabel yang terorganisir secara terpadu, saling berinteraksi, saling bergantung satu sama lain untuk mendapatkan suatu hasil.
- Informasi adalah data yang berformat dan terorganisasi dengan baik agar mudah dianalisis atau diproses.
- Geografis adalah menunjukkan keterkaitan data dengan lokasi yang diketahui dan dapat dihitung berdasarkan koordinat geografis.

Berdasarkan pengertian diatas dapat dikatakan bahwa SIG dirancang untuk membentuk suatu data yang terorganisasi dari berbagai data keruangan dan atribut yang mempunyai "Geo Code" dalam suatu basis data agar dapat dengan mudah dimanfaatkan dan dianalisis, hal ini dikemukakan oleh team pelatihan SIG (BP2SIG Unnes, 2006:5).

#### 2.2.4. PHP

PHP adalah singkatan dari *PHP: Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: *PHP: Hypertext Preprocessor*.

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>.

PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser (client)*.

#### 2.2.5. JAVA

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus dirancang untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa *platform* sistem operasi yang berbeda, Java dikenal pula dengan slogannya, “Tulis sekali, jalankan di manapun”.

Menurut Rickyanto (2003). Java adalah suatu teknologi di dunia *software* komputer. selain merupakan bahasa pemrograman, java juga merupakan *platform*. Java merupakan teknologi dimana teknologi tersebut mencakup java sebagai bahasa pemrograman yang memiliki *sintaks* dan aturan pemrograman sendiri. Java juga mencakup sebagai *platform* dimana teknologi ini memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemrograman java.

#### **2.2.6. MATERIALIZE**

Materialize adalah sebuah framework intuitif yang mirip dengan Bootstrap dan foundation yang menawarkan komponen UI. Namun, berbeda fungsi karena Bootstrap dan Foundation berbasis mobile pertama kali sedangkan Materialize mengikuti filosofi Material Design dari Google. Materialize adalah pustaka komponen UI yang dibuat dengan CSS, JavaScript, dan HTML. Mewujudkan komponen UI yang dapat digunakan membantu dalam membangun halaman web yang menarik, konsisten, fungsional, dan responsif. Aplikasi web yang mengikuti prinsip desain web modern seperti portabilitas browser, independensi perangkat, dan degradasi yang anggun. Materialize telah membangun perancangan responsif sehingga situs web yang dibuat menggunakan Materialize akan mendesain ulang sendiri sesuai ukuran perangkat, sehingga situs web dapat memuat ukuran layar apapun. (Anirudh Prabhu dan Aravind Shenoy, 2016).

### 2.2.7. JQUERY

JQuery merupakan suatu **framework** (library) Javascript yang menekankan bagaimana interaksi antara **Javascript** dan **HTML**. JQuery pertama kali dirilis pada tahun 2006 oleh John Resig. Pada perkembangannya JQuery tidak sekedar sebagai framework Javascript, namun memiliki kehandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak developer web menggunakannya. JQuery memiliki slogan “**Write less, do more**” yang kurang lebih maksudnya adalah kesederhanaan dalam penulisan code, tapi dengan hasil yang lebih banyak.