

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Sebelumnya**

Penelitian yang dilakukan oleh penulis tentunya tidak terlepas dari hasil penelitian yang sudah ada terdahulu dengan metode dan kasus permasalahan yang hampir sama, yang kemudian dijadikan penulis sebagai studi pustaka, berikut beberapa penelitian sebelumnya.

Dalam jurnal yang berjudul “Algoritma Dijkstra untuk Pendistribusian *Carica Nida Food Wonosobo*” yang di tulis oleh[2] penulis menggunakan algoritma dijkstra dalam penentuan rute terpendek, penelitian ini merupakan pembaharuan dari penelitian yang sudah pernah dilakukan sebelumnya. Pembaharuan yang dilakukan penulis adalah menggunakan algoritma yang berbeda. Pembaharuan itu dilakukan karena dalam penelitian sebelumnya proses pencariannya memerlukan waktu yang sangat lama. Pada penelitian ini menghasilkan rute jarak tempuh terdekat sebesar 20,51.

Dalam jurnal “Implementasi algoritma dijkstra untuk analisa rute transportasi umum trans jogja berbasis andorid” yang di tulis[3] menggunakan algoritma dijkstra. Data yang di pakai dalam penelitian ini yaitu halte dan rute perjalanan bus. Pengembangan aplikasi ini menggunakan android studio dan untuk penyimpanan data base menggunakan SQLite. Pada penelitian ini penulis juga membandingkan algoritma dijkstra dengan algoritma *Ant Colony* yang diperoleh hasil perhitungan dijkstra lebih cepat dibandingkan *Ant Colony* yaitu sebesar 0,0037 dan paling besar 0,0076. Sedang kan *Ant Colony* 0,0105 dan 0,0246.

Pada penelitian yang berjudul “Penentuan Rute Terpendek Pendistribusian Naskah Ujian Nasional Menggunakan Algoritma Dijkstra” yang ditulis oleh[4] menggunakan algoritma dijkstra. Data yang digunakan berupa nama, alamat, lokasi, dan titik koordinat

(*latitude, longitude*) sekolah SMA/SMK di kota Binjai. Dalam penentuan titik koordinat menggunakan *one touch location* dan *google maps*. Pada penelitian ini dilakukan uji validasi dengan menggunakan metode *black box*. Algoritma ini dapat menyelesaikan masalah distribusi naskah ujian di kota Binjai.

Dalam jurnal[5] yang berjudul “Peta Pendistribusian Hasil Pertanian Dan Perikanan Di Kabupaten Klaten Berbasis Android Menggunakan Algoritma Dijkstra” data yang digunakan yaitu data primer yg di dapat dari wawancara dan observasi dan data sekunder berasal dari studi literatur. Penelitian ini berbasis android dan menggunakan algoritma dijkstra, penentuan titik koordinat menggunakan *google maps*.

Dalam jurnal yang berjudul[6] “Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra” . Algoritma ini dipakai untuk penentuan jarak titik TPS ke TPA, yang kemudian diurutkan dengan perhitungan dijkstra dan didapatkan rute dari urutan titik titik tersebut. Pada implementasinya menggunakan Php dan mysql untuk basisdata. Penentuan titik menggunakan *MapInfo Professional 7.0*. Pada penelitian ini menggunakan 10 sampel data TPS.

Dalam jurnal yang berjudul[7] “Analisis Algoritma *Dijkstra* dan Algoritma *Bellman-Ford* sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran”. Penelitian ini membandingkan dua algoritma yaitu *Dijkstra* dan *Bellman-Ford* di kecamatan Praya Kota. Dalam penelitian ini menghasilkan kesimpulan algoritma *Dijkstra* memiliki keunggulan dalam kompleksitas waktu, dan algoritma *Bellman-Ford* memiliki keunggulan bisa menghitung sisi negatif.

Tabel 2. 1 Penelitian Terdahulu

No	Penulis dan Tahun	Judul	Metode	Isi dan Hasil
1	Sulaiman, Hamdun Yuliani, Yuri Fitri, Evita Herlinawati, Nuraeni  Tahun 2020	Algoritma Dijkstra untuk Pendistribusian <i>Carica Nida Food</i> Wonosobo	Algoritma Dijkstra	Penelitian ini merupakan pembaharuan dari penelitian yang sudah pernah dilakukan sebelumnya. Pembaharuan yang dilakukan penulis adalah menggunakan algoritma yang berbeda. Pembaharuan itu dilakukan karena dalam penelitian sebelumnya proses pencariannya memerlukan waktu yang sangat lama. Penelitian ini menghasilkan rute jarak tempuh terdekat sebesar 20,51.
2.	Yudhana, Anton Azhar, Ahmad  Tahun 2019	Implementasi algoritma dijkstra untuk analisa rute transportasi umum trans jogja berbasis andorid.	Algoritma Dijkstra	Pada penelitian ini penulis membangun aplikasi berbasis android. Data yang digunakan adalah halte dan rute perjalanan bus. Pada penelitian ini penulis juga membandingkan algoritma <i>dijkstra</i> dengan algoritma <i>Ant Colony</i> yang diperoleh hasil perhitungan <i>dijkstra</i> lebih cepat dibanding <i>Ant Colony</i> yaitu sebesar 0,0037 dan paling besar 0,0076. Sedangkan <i>Ant Colony</i> 0,0105 dan 0,0246.

No	Penulis dan Tahun	Judul	Metode	Isi dan Hasil
3.	Syahputra, Siswan Tahun 2017	Penentuan Rute Terpendek Pendistribusian Naskah Ujian Nasional menggunakan Algoritma Dijkstra.	Algoritma Dijkstra	Data yang digunakan berupa nama, alamat, lokasi, dan titik koordinat ( <i>latitude, longitude</i> ) sekolah SMA/SMK di kota Binjai. Dalam penentuan titik koordinat menggunakan <i>one touch location</i> dan <i>google maps</i> . Pada penelitian ini dilakukan uji validasi dengan menggunakan metode <i>black box</i> .
4.	Hidayat, Syams Kurniawan Setyawan, Doni Tahun 2018	Peta Pendistribusian Hasil Pertanian Dan Perikanan Di Kabupaten Klaten Berbasis Android Menggunakan Algoritma Dijkstra	Algoritma Dijkstra	Penelitian ini menggunakan konsep <i>Location Based Service (LBS)</i> berbasis android, data yang digunakan adalah data primer dan data sekunder yang di dapat melalui observasi dan studi literatur.

No	Penulis dan Tahun	Judul	Metode	Isi dan Hasil
5.	Andayani, Sri Perwitasari Endah Wulan  Tahun 2014	Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra	Algoritma Dijkstra	Pada implementasinya menggunakan Php dan mysql untuk basisdata. Penentuan titik menggunakan <i>MapInfo Professional 7.0</i> . Pada penelitian ini menggunakan 10 sampel data TPS. Dalam penentuan rute terpendek di sajikan dalam bentuk matrik sebelum diolah dengan algoritma dijkstra.
6.	S. Hamdi dan Prihandoko  Tahun 2018	Analisis Algoritma <i>Dijkstra</i> dan Algoritma <i>Bellman-Ford</i> sebagai Penentuan Jalur Terpedek Menuju Lokasi Kebakaran.	Algoritma Dijkstra dan Algoritma Bellman-Ford	Penelitian ini membandingkan dua algoritma yaitu <i>Dijkstra</i> dan <i>Bellman-Ford</i> di kecamatan Praya Kota. Dalam penelitian ini menghasilkan kesimpulan algoritma Dijkstra memiliki keunggulan dalam kompleksitas waktu dan algoritma <i>Bellman-Ford</i> memiliki keunggulan bisa menghitung sisi negatif.

## **2.2 Dasar Teori**

### **2.2.1 *Traveling Salesman Problem***

*Traveling Salesman Problem* (TSP) adalah masalah yang solusinya telah banyak memunculkan matematikawan selama bertahun-tahun. Sampai dengan saat ini belum ada solusi yang memuaskan para ahli matematika tersebut. Matematika yang terkait dengan TSP ini dikembangkan sekitar tahun 1800-an oleh dua orang ahli matematika, Sir William Rowan Hamilton, berkebangsaan Irlandia dan Thomas Penyngton Kirkman, kerkebangsaan Inggris. Hamilton adalah pencipta permainan Icosian pada tahun 1857, dimana pemain harus menyelesaikan perjalanan dari 20 titik hanya menggunakan jalur-jalur tertentu. Namun, bentuk umum TSP pertama kali dipelajari oleh Karl Menger di Wina dan Harvard pada akhir 1920-an hingga awal 1930-an[8]. TSP merupakan pencarian jarak terdekat, waktu tercepat dengan metode yang dapat meminimalisir biaya distribusi.

### **2.2.2 *Reseller***

*Reseller* merupakan sebuah penjual yang menjual barang milik penjual lain. Sehingga *reseller* sendiri mampu membantu memasarkan penjualan dari penjual lain. Sistem yang diterapkan *reseller* dalam ilmu manajemen termasuk sebagai strategi distribusi tak langsung (*indirect*). Distribusi tak langsung sendiri memiliki pengertian penyaluran atau penjualan barang dari produsen kepada konsumen melalui perantara yang dilakukan oleh agen, makelar atau *reseller*[9].

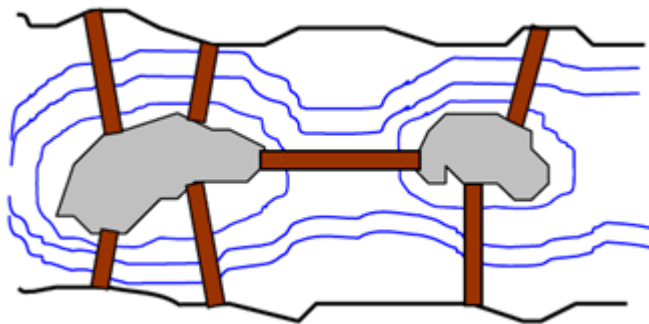
### **2.2.3 Rute Terpendek**

Pada dasarnya permasalahan pencarian jalur terpendek antar kota/lokasi merupakan pencarian jalur terpendek antar titik yang telah diketahui koordinatnya[10]. Dengan mengetahui konsep pencarian jalur

terpendek antar titik, untuk selanjutnya dapat diterapkan pada pencarian jalur terpendek pada berbagai kota/lokasi yang ingin diketahui.

#### 2.2.4 Teori Graph

Teori *graph* pertama kali muncul pada tahun 1736. Ketika itu permasalahan yang akan diselesaikan adalah mencari suatu rute agar dapat melewati ketujuh Jembatan *Könsigsberg* (Gambar 2.1) tepat satu kali dan kemudian kembali lagi ke titik semula Teori graf didefinisikan sebagai pasangan himpunan  $(V,E)$  yang ditulis dengan notasi  $G=(V,E)$ . Dimana  $V$  adalah himpunan tidak-kosong dari simpul- simpul (vertex) dan  $E$  adalah himpunan sisi (edge) yang menghubungkan sepasang simpul[11].



Gambar 2. 1 Jembatan *Konigsberg*

#### 2.2.5 Google Maps

*Google Maps* adalah jasa peta gratis dan *online* disediakan oleh google yang dapat ditemukan di <http://maps.google.com>. Pada situs tersebut kita dapat melihat informasi geografis pada hampir semua wilayah di muka bumi. Layanan ini interaktif, karena di dalamnya peta dapat digeser sesuai keinginan pengguna, mengubah tingkat zoom, serta mengubah tampilan peta. *Google Maps* juga menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia, serta menawarkan rute perjalanan. *Google Maps* dibuat dengan menggunakan kombinasi dari gambar peta, database, serta objek-objek interaktif yang dibuat dengan bahasa pemrograman HTML, *Javascript*, dan AJAX, serta beberapa

bahasa pemrograman lainnya. Seluruh citra yang ada diintegrasikan ke dalam suatu database pada google server. Bagian-bagian gambar peta yang merupakan gabungan dari gambar-gambar yang berukuran 256 x 256 pixel[12].

### 2.2.6 Algoritma Dijkstra

Algoritma Dijkstra adalah sebuah algoritma *greedy*. Algoritma ini pertama kali dikemukakan oleh Edsger W. Dijkstra pada tahun 1959 dan telah secara luas digunakan dalam menentukan rute tersingkat atau jalur terpendek berdasarkan kriteria tertentu yang digunakan sebagai batasan[13]. Algoritma Dijkstra sering dipakai dalam memecahkan permasalahan jarak terpendek untuk sebuah graf berarah dengan bobot-bobot sisi (*edge*) yang bernilai tak negatif. Algoritma dijkstra akan mencari jalur terpendek mulai dari titik awal sampai titik tujuan, sehingga dapat membantu memberikan pilihan jalur. Algoritma Dijkstra merepresentasikan titik dan simpul ke dalam graph berarah untuk penentuan rute lintasan terpendek. Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Titik menggambarkan Lokasi dan garis menggambarkan jalan, maka Algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.

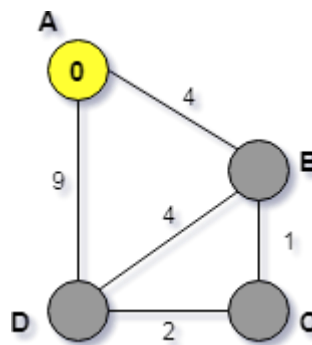
Langkah-langkah dalam menentukan lintasan terpendek pada algoritma Dijkstra yaitu:

1. Tentukan node mana yang akan menjadi node awal. Beri nilai bobot (jarak) dari node awal untuk setiap node ke node lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga untuk node lain yang belum dilewati(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.

4. Setelah selesai menghitung setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node terjamah”. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru. 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 ulangi seperti langkah seperti nomor tiga.



Gambar 2. 2 Contoh Representasi Graph

Sebagai contoh hitung lah jarak dari titik A menuju titik D dari gambar 2.2 diatas. Dalam pemecalahan masalah tersebut rute yang dapat dilalui dari A menuju D adalah:

1. A ke D = 9
2. A ke D melalui B = 8
3. A ke D melalui B dan C = 7

Setelah di dapat jumlah bobot jarak dari semua rute tersebut kemudian cari bobot minimalnya. Dalam contoh tersebut maka di dapat bobot minimal titik A menuju titik D adalah 7.