

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka merupakan acuan utama pada penelitian ini berupa beberapa studi yang pernah dilakukan yang berkaitan dengan penelitian ini.

Tabel 2.1 Tinjauan Pustaka

Parameter Penulis	Objek	Model	Metode
Bambang Yuwono, Agus Sasmito Aribowo, Siswanto Budi Wardoyo (2009)	Pencarian jalur terpendek jalan protocol di yogyakarta	Menggunakan teknologi web	Algoritma koloni semut
I Wayan Eka Swastikayana (2011)	pemetaan pariwisata kabupaten Gianyar	Menggunakan bahasa pemrograman Google Map API, Java Script, dan HTML.	Metode <i>Waterfall</i>
Rizky Yusaputra (2013)	pencarian rute terpendek lokasi fasilitas umum	Menggunakan teknologi mobile (android)	Algoritma <i>Floyd warshall</i>
Benny Bagiarta (2016)	Sistem Identifikasi Lokasi Penyedia Donor Darah	Menggunakan Teknologi LBS	Metode <i>AHP</i>
Mohamad Arifin (2016)	Rancang Bangun SIG Wisata Alam di Kabupaten Kaimana	Menggunakan teknologi AJAX	
I Made Fajar Ananda Kusuma (2017)	Pemetaan Wilayah Asrama Mahasiswa Daerah di Yogyakarta	Menggunakan teknologi GIS	Algoritma <i>Dijkstra</i>

Dalam usulan penelitian ini yang membedakan antara usulan dengan yang lainnya selain objek (asrama mahasiswa daerah) adalah akan dibuat system pencarian, jarak, dan waktu tempuh. Perhitungan rute terpendek menggunakan Algoritma *Dijkstra* sehingga memudahkan dalam mencari lokasi objek.

2.2 Dasar Teori

2.2.1 Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, sifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan (Joperson Hutahean, 2014: 13)

2.2.2 Geografi

Istilah ini digunakan karena GIS dibangun berdasarkan pada 'geografi' atau 'spasial'. Object ini mengarah pada spesifikasi lokasi dalam suatu space. Objek bisa berupa fisik, budaya, atau ekonomi alamiah. Penampakan tersebut ditampilkan pada suatu peta untuk memberikan gambaran yang representatif dari spasial suatu objek sesuai dengan 10 kenyataannya dibumi. Simbol, warna dan gaya garis digunakan untuk mewakili setiap spasial

yang berbeda pada peta dua dimensi. (I Wayan Eka Swastikayana, 2011).

2.2.3 Sistem Informasi geografis (SIG)

Sistem Informasi Geografis (SIG) atau *Geographic Information System* (GIS) adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, menganalisa, mengatur dan menampilkan seluruh jenis data geografis (Edy Irwansyah, 2013).

2.2.4 Peta

Peta merupakan penyajian grafis dari permukaan bumi dalam skala tertentu dan digambarkan pada bidang datar melalui sistem proyeksi peta dengan menggunakan simbol-simbol tertentu sebagai perwakilan dari objek-objek spasial di permukaan bumi. Secara singkat Prihandito (1988) mendefinisikan peta merupakan penyajian grafis dari bentuk ruang dan hubungan keruangan antara berbagai perwujudan yang di wakili.

2.2.5 Pemetaan

Pemetaan merupakan Proses pengukuran ,perhitungan dan penggambaran permukaan bumi dengan menggunakan cara atau metode tertentu sehingga didapatkan hasil berupa *Softcopy* dan *Hardcopy*

2.2.6 Google Maps

Google Maps adalah sebuah jasa peta globe virtual gratis dan *online* disediakan oleh Google dapat ditemukan di <http://maps.google.com> (I Wayan Eka Swastikayana, 2011). Ia menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia dan baru-baru ini, Bulan, dan juga menawarkan perencana rute dan pencari letak bisnis di U.S., Kanada, Jepang, Hong Kong, Cina, UK, Irlandia (hanya pusat kota) dan beberapa bagian Eropa. Google Maps masih berada dalam tahap beta.

2.2.7 Google Maps API

Google Maps API adalah suatu *library* yang berbentuk *JavaScript*. Cara membuat *Google Maps* untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi Internet yang sangat stabil. Dengan menggunakan *Google Maps API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia (I Wayan Eka Swastikayana, 2011).

Kode program dasar:

```

<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      html, body { height: 100%; margin: 0; padding: 0; }
      #map { height: 100%; }
    </style>
  </head>
  <body>
    <div id="map"></div>
    <script type="text/javascript">

var map;
function initMap() {
  map = new google.maps.Map(document.getElementById('map'), {
    center: {lat: -34.397, lng: 150.644},
    zoom: 8
  });
}

    </script>
    <script async defer
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&
callback=initMap">
    </script>
  </body>
</html>

```

Gambar 2.1 Contoh Program Dasar Google Maps API

2.2.8 Algoritma *Dijkstra*

Pencarian rute terpendek termasuk ke dalam materi teori graf. Algoritma yang sangat terkenal untuk menyelesaikan persoalan ini adalah algoritma *Dijkstra*. Algoritma ini ditemukan oleh seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger *Dijkstra*. "*Dijkstra*" diartikan sebagai "algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah" (Siswanto, 2013: 384). Cara kerja algoritma *Dijkstra* memakai strategi greedy, dimana pada setiap langkah dipilih sisi

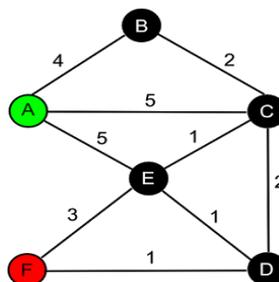
dengan bobot terkecil yang menghubungkan sebuah simpul lain yang belum terpilih. Algoritma *Dijkstra* membutuhkan parameter tempat asal, dan tempat tujuan.

2.2.9 Penerapan Algoritma *Dijkstra*

Algoritma *Dijkstra* adalah salah satu metode untuk memecahkan masalah pencarian rute terpendek. Algoritma ini biasanya diterapkan pada sebuah aplikasi pencari rute jalan yang terdekat dari suatu daerah ke daerah lainnya. Untuk bisa menerapkan algoritma ini dibutuhkan beberapa data yang harus disiapkan, yaitu :

1. Beberapa Titik/simpul/daerah, titik/simpul/daerah yang bisa dijangkau secara langsung, dan juga jarak antara mereka.
2. Titik/simpul/daerah awal.
3. Titik/simpul/daerah tujuan.

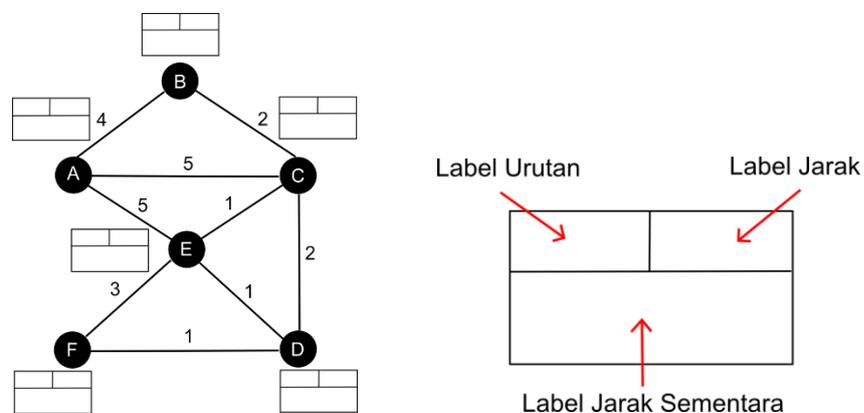
Jika dicontohkan dengan gambar grafik akan seperti ini :



Gambar 2.2 Graph Algoritma *Dijkstra*

Titik **A** adalah titik awal dan titik **F** adalah titik tujuan. Kemudian kita akan mencari rute manakah yang harus dilewati dan memiliki total jarak yang paling dekat. Untuk bisa mendapatkan rute itu, maka grafik diatas ditambahkan beberapa kotak untuk mengisi beberapa label. Seperti ini :

Tabel 2.2 Tabel jarak sementara



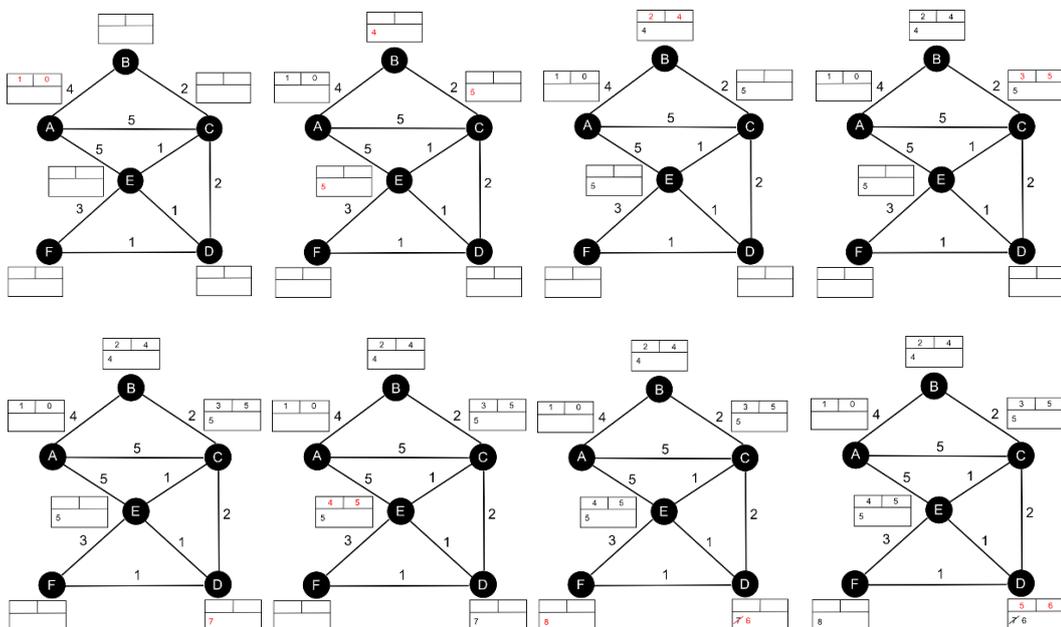
Setelah itu ada beberapa langkah yang harus dilakukan, yaitu :

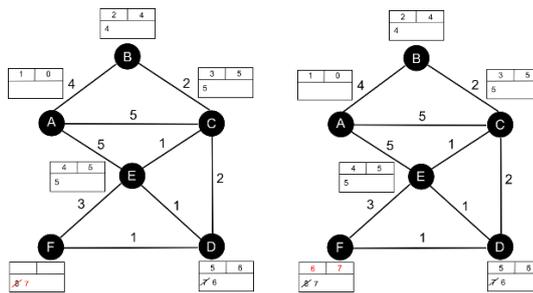
1. Mengisi kotak label pada titik awal dengan *label urutan* 1 dan *label jarak* 0.
2. Menetapkan *label jarak sementara* untuk semua titik yang dapat dihubungi langsung dari awal.
3. Pilih titik dengan *label jarak sementara* terkecil dan menuliskan nilainya di *label jarak*, serta tambahkan *label urutan*-nya.
4. Masukkan *label jarak sementara* pada setiap titik yang belum memiliki *label urutan* dan *label jarak* dan dapat dihubungi

langsung dari titik yang baru saja ditulis *label jarak* dan *label urutan*-nya. nilainya diisi dengan total dari *label jarak* dari titik sebelumnya dan jarak dari titik tersebut. Jika *label jarak sementara* di titik tersebut sudah memiliki nilai, maka harus diganti hanya jika nilai yang baru lebih kecil.

5. Pilih titik dengan *label jarak sementara* terkecil dan menggunakan *label jarak sementara*-nya sebagai *label jarak* dari titik tersebut, serta tambahkan *label urutan*-nya.
6. Ulangi langkah 4 dan 5 hingga titik tujuan memiliki *label jarak* dan *label urutan*.

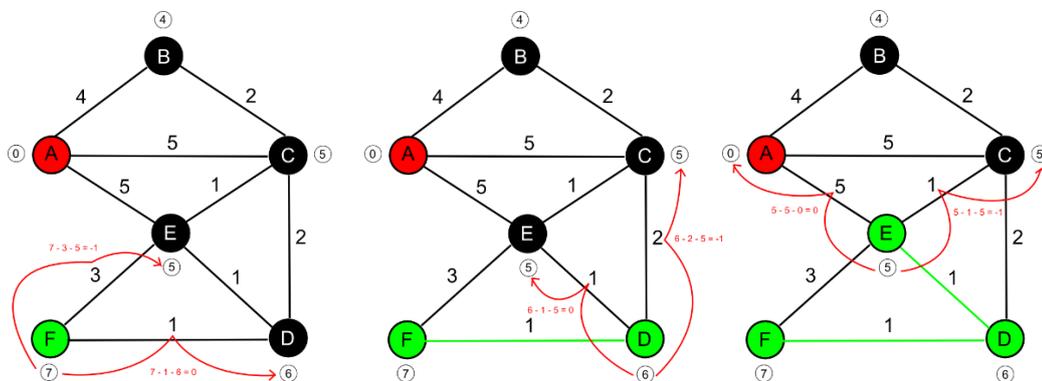
Maka dalam perhitungannya akan tampak seperti gambar dibawah :





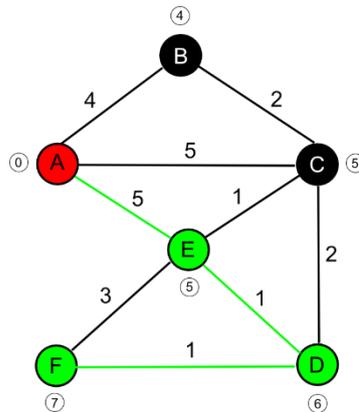
Gambar 2.3 Perhitungan jarak sementara

Setelah menghitung semua jarak yang ada pada simpul maka akan ditentukan jalur terpendek dari *graph* diatas. Dengan mengurangi jarak antar poin dengan jarak sementara simpul jika nilai samadengan 0 maka jalur dipilih, jika nilai tidak sama dengan 0 maka jalur diabaikan.



Gambar 2.4 Menentukan jarak terpendek

Dengan begitu diketahui rute yang harus dilewati dan memiliki jarak terpendek dari titik **A** menuju titik **F** adalah : **A -> E -> D -> F**



Gambar 2.5 Jalur terpendek algoritma *Dijkstra*

2.2.10 Basis Data

Basis data adalah penggunaan bersama dari data yang terhubung secara logis dan deskripsi dari data, yang dirancang untuk keperluan informasi. (I Wayan Eka Swastikayana, 2011) Integrasi logis dari catatan-catatan banyak file disebut sebagai konsep basis data. Tujuan dari konsep basis data yaitu untuk meminimumkan pengulangan data dan mencapai independensi data.

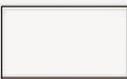
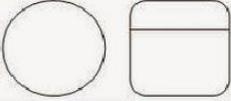
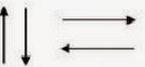
2.2.11 DFD

DFD (*Data Flow Diagram*) merupakan diagram yang menggunakan notasi-notasi untuk menggambarkan arus data sistem secara logika. DFD menggambarkan komponen – komponen sebuah sistem. Keuntungan menggunakan DFD adalah memudahkan pemakai yang kurang menguasai komputer untuk

mengerti sistem yang akan dikembangkan. (I Wayan Eka Swastikayana, 2011)

DFD merupakan dokumentasi grafik yang menggunakan simbol penomoran di dalam mengilustrasikan arus data yang saling berhubungan diantara pemrosesan data untuk diubah menjadi informasi. Dengan demikian proses di dalam DFD adalah ditekankan pada informasi data.

Tabel 2.3 DFD (*Data Flow Diagram*)

Simbol	Keterangan
	<i>External Entity</i> , merupakan kesatuan di lingkungan luar sistem yang bisa berupa orang, organisasi atau sistem lain.
	<i>Process</i> , merupakan proses seperti perhitungan aritmatik penulisan suatu formula atau pembuatan laporan
	<i>Data Store</i> (Simpan Data), dapat berupa suatu file atau database pada sistem komputer atau catatan manual
	<i>Data Flow</i> (arus data), arus data ini mengalir diantara proses, simpan data dan kesatuan luar

2.2.12 MySQL

SQL (*Structured Query Language*) merupakan sebuah bahasa *relational* yang berisi pernyataan yang digunakan untuk memasukkan, mengubah, menghapus, memilih dan melindungi data (Prihatna, 2005). MySQL merupakan sebuah perangkat lunak atau *software* sistem manajemen basis data SQL atau DBMS Multithread dan multi *user*. MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam database untuk

pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis.

2.2.13 Flowchart

Flowchart adalah kumpulan perintah yang ditujukan untuk memberi gambaran atau aliran dari satu scene ke scene lain. Subrutin adalah kumpulan perintah yang ditujukan untuk menangani suatu tindakan dengan tujuan untuk memudahkan pembuatan program mengingat subrutin bisa dipanggil berkali-kali dalam suatu program.(Martin, 1975). Simbul subrutin terlihat dibawah ini:

Tabel 2.4 Flowchart

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Pemulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda