**BAB II**

**LANDASAN TEORI**

1. **Tinjauan Studi**

Ada beberapa penelitian tentang Sistem Informasi Geografis yang pernah dilakukan yaitu :

1. Penelitian yang dilakukan oleh Stevian Suryo Saputro (2013)“Perancangan Aplikasi Gis Pencarian Rute Terpendek Peta Wisata Di Kota Manado Berbasis Mobile Web Dengan Algoritma *Dijkstra*”**.** Dengan tujuan untuk menghasilkan sebuah Aplikasi SIG (Sistem Informasi Geografis) berbasis Mobile web yang dapat menampilkan rute pengaksesan tempat-tempat wisata di kota Manado, dengan menerapkan dan mengimplementasikan algoritma *Dijkstra*, sehingga para wisatawan khususnya para wisatawan lokal yang berasal dari luar daerah Manado dapat dengan mudah dan cepat dalam pengaksesan tempat-tempat wisata tersebut.
2. Penelitian yang dilakukan oleh Shaga Bogas Priatmoko (2014) dengan judul “Algoritma *Dijkstra* Untuk Pencarian Jalur Terdekat Dan Rekomendasi Objek Wisata di Pulau Bali”**.**Dengan tujuan Merancang aplikasi pencarian jalur terdekat pariwisata dan rekomendasi objek pariwisata di pulau bali dengan menggunakan algoritma *dijkstra*. sehingga waktu tempuh dapat dioptimalkan dan dapat mengetahui objek wisata lainnya yang dilalui jalur tersebut.
3. Penelitian yang dilakukan oleh Apri Triansyah (2014) dengan judul “Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatra Bagian Selatan” dengan tujuan memberikan solusi dalam pemilihan lintasan terpendek pada jalan darat antara kota kota di sumatera bagian selatan meliputi provinsi Jambi, Sumatera Selatan, Bengkulu dan Lampung, mempercepat dalam mencari solusi lintasan terpendek pada jalan darat antara kotakota di sumatera bagian selatan.Dengan menerapkan alogaritma dijkstra untuk mencari lintasan terdekat dapat membantu para pengguna jalan, *traveling salesman*, perusahaan yang bergerak dibidang pariwisata dan angkutan antar provinsi, instansi pemerintah dan lain sebagainya terutama bagi yang membutuhkan informasi tentang lintasan terdekat.
4. **Tinjauan Pustaka**
   * 1. **Pemadam Kebakaran**

Pemadam Kebakaran adalah satuan yang dilatih dan bertugas untuk menanggulangi kebakaran. Pemadam Kebakaran mempunyai dua fungsi utama. Pertama, fungsi pencegahan kebakaran, yaitu mengantisipasi dan melakukan usaha preventif agar tidak terjadi atau mengurangi serta meminimkan resiko terjadinya kebakaran. Kedua, fungsi penanggulangan kebakaran, yaitu segala upaya dan tindakan penyelamatan pada saat terjadinya musibah kebakaran secara efektif. Selain itu Badan ini juga berperan besar dalam meningkatkan kesadaran dan kewaspadaan masyarakat terhadap bahaya kebakaran dan pentingnya upaya pencegahan dan penanggulangannya dilaksanakan secara terpadu. Pada kondisi darurat atau menanggapi suatu kebakaran, kendaraan ini wajib diberi laluan dan jalan di lalulintas agar sampai di lokasi dengan cepat. Kenderaan Pemadam Kebakaran merupakan kendaraan gawat darurat pemadam kebakaran. Pada kondisi darurat atau menanggapi suatu kebakaran, kendaraan ini akan membunyikan sirene dan menyalakan lampu-lampu darurat yang umumnya berwarna merah atau biru maupun kuning, jika pengemudi melihat ini di jalan raya atau lalulintas, maka seluruh kendaraan wajib memberi laluan atau minggir untuk memprioritaskan tugas penyelamatan nyawa tersebut. Dan jika ada pengemudi yang mengabaikan, membiarkan, atau mengganggu perjalanan kendaraan darurat yang sedang menjalankan tugas, maka itu merupakan tindakan pelanggaran lalu lintas dan sangat dilarang dalam peraturan lalu-lintas Indonesia maupun seluruh dunia (Indra Bastian, 2005 : 293). Permasalahan yang dihadapi pemadam kebakaran antara lain yaitu menentukan rute tercepat untuk menuju ke lokasi kebakaran, kondisi lingkungan masyarakat yang kurang berpartisipasi dalam membantu pihak pemadam kebakaran. (Lahmudin J. Hinelo Pimpinan UPTD Pemadam Kebakaran Kota Gorontalo)

* + 1. **Sistem Informasi Geografis**

SIG (Sistem Informasi Geografis) atau dikenal pula dengan GIS *(Geographical Information System)* merupakan suatu istilah dalam bidang pemetaan yang memiliki ruang lingkup mengenai bagaimana suatu sistem dapat menghubungkan objek geografis dengan informasinya. GIS merupakan akronim dari :

1. *Geographic*

Istilah ini digunakan karena GIS dibangun berdasarkan pada ‘geografi’ atau spasial. Objek ini mengarah pada spesifikasi lokasi dalam suatu *space*. Objek bias, budaya atau ekonomi alamiah. Penampakan tersebut ditampilkan pada suatu peta untuk memberikan gambaran yang representasif dari spasial suatu objek sesuai dengan kenyataannya dibumi. Simbol, warna dan gaya garis digunakan untuk mewakili setiap spasial yang berbeda pada peta dua dimensi.

1. *Information*

Informasi berasal dari pengolahan sejumlah data. Dalam GIS informasi memiliki volume terbesar. Setiap geografi memiliki *setting* data tersendiri karena tidak sepenuhnya data yang ada dapat terwakili dalam peta. Jadi, semua data harus diasosiasikan dengan objek spasial yang dapat membuat peta menjadi *intelligent.* Ketika data tersebut diasosiasikan dengan permukaan geografis yang refresentasif, data tersebut mampu memberikan informasi dengan hanya mengklik mouse pada objek. Perlu diingat bahwa semua informasi adalah data tapi tidak semua data merupakan informasi.

1. *System*

Pengetian suatu sistem adalah kumpulan elemen-elemen yang saling berintegrasi dan berintegrasi dalam lingkungan yang dinamis untuk mencapai tujuan tertentu. (R. Rizky Ajie Oktora, 2014 : II-1)

1. **Konsep Dasar Sistem Informasi Geografis**

Sistem Informasi Geografis (SIG) atau juga dikenal sebagai *Geographic Information System* (GIS) pertama pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan geografis. 40 tahun kemudian GIS berkembang tidak hanya bertujuan untuk menyelesaikan permasalahan geografi saja tetapi sudah menambah ke berbagai bidang seperti analisis penyakit epidemic (demam berdarah) dan analisis kejahatan (kerusuhan) termasuk analisis kepariwisataan.

Kemampuan dasar dari SIG adalah mengintegrasikan berbagai operasi basis data seperti *query*, menganalisisnya serta menampilkannya dalam bentuk pemetaaan berdasarkan letak geografisnya. Inilah yang membedakan SIG dengan sistem informasi lain. (R. Rizky Ajie Oktora, 2014 : II-2)

1. **Definisi Sistem Informasi Geografis**

Istilah *Geography* digunakan karena SIG dibangun berdasarkan pada geografi atau spasial. Objek ini mengarah pada spesifikasi lokasi dalam suatu *space.* *Geographic Information System* (GIS) merupakan sistem komputer yang berbasis pada sistem informasi yang digunakan untuk memberikan bentuk digital dan analisis terhadap permukaan geografi bumi.

Geografi adalah informasi mengenal permukaan bumi dan semua obyek yang berada diatasnya, sedangkan sistem informasi geografis (SIG) atau dalam bahasa inggris disebut *Geographic Information System* (GIS) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). (R. Rizky Ajie Oktora, 2014 : II-2)

1. **Manfaat Sistem Informasi Geografis**

Fungsi SIG adalah meningkatkan kemampuan menganalisis informasi spasial karena terpadu untuk perencaaan dan pengambilan keputusan. SIG dapat memberikan informasi kepada pengambil keputusan untuk analisis dan penerapan *database* keruangan.

SIG mampu memberikan kemudahan-kemudahan yang diinginkan. Dengan SIG kita akan dimudahkan dalam melihat fenomena kebumian dengan perspektif yang lebih baik. SIG mampu mengakomodasi penyimpanan, pemrosesan, dan penanyangan data spasial digital bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta bahkan data statistik. SIG juga mengakomodasi dinamika data, pemuktahiran data yang akan menjadi lebih mudah. (R. Rizky Ajie Oktora, 2014 : II-3)

1. **Komponen Utama Sistem Informasi Geografis**

Dalam merancang SIG dibutuhkan 3 komponen utama sistem komputer, data *geospasial* serta pengguna. Ketiganya saling berhubungan seperti pada gambar berikut :



**Gambar 2.1** Komponen SIG

Sistem komputer terdiri dari *hardware* dan *software,* komponen pada *software* terdiri dari *program, database,* dan *Graphical User Interface (GUI).* Dimana perlu diketahui bahwa bagian *GUI* merupakan tampilan dari program yang berinteraksi langsung dengan pengguna. Dalam berkomunikasi dan mendapatkan informasi, *GUI*  menjebatani program kompleks dan kumpulan informasi dalam *database* yang ingin diakses dengan kemampuan seorang penggguna yang awam.

Sedangkan *hardware* merupakan perangkat elektronik atau juga dapat disebut dengan *platform* dimana program dan *database* berjalan. *Hardware* dapat berupa komputer atau perangkat-perangkat elektronik bersifat *mobile* seperti alat *GPS,PDA* ataupun *smartphone.*

Data *geospatial* mengandung rujukan geografi secara langsung seperti *latitude* (garis lintang), *longitude* (garis bujur), atau sebuah rujukan implisit seperti sebuah alamat, kode pos, dan lain-lain. Pada aplikasi yang kompleks, rujukan geografi mempunyai sebuah proses yang otomatis yang disebut *geocoding,* digunakan untuk menciptakan rujukan geografi explicit dari implisit atau gambaran seperti sebuah alamat.

Kumpulan dari data geospasial dihubungkan pada suatu sistem komputer. Sistem ini dapat mengenal informasi yang terkandung pada data geospasial dan mengidentifikasi informasi yang dibutuhkan oleh pengguna. Lalu kebutuhan pengguna dapat disesuaikan dengan data yang tersedia. Maka dapat dimunculkan data *geospatial* yang berhubungan.

Apabila kita melihat secara perspektif global maka ketiga data tersebut berinteraksi pada suatu sistem yang dimiliki aturan atau prosedur tertentu dalam berhubungan satu sama lain. (R. Rizky Ajie Oktora, 2014 : II-3)

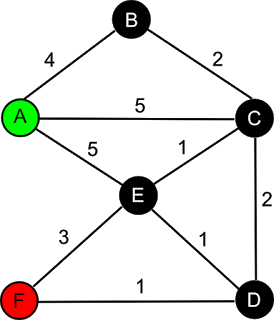
* + 1. **Algortima *Dijkstra***

Pencarian lintasan terpendek (*Shortest Path*) yaitu pencarian lintasan yang memiliki bobot minimum. Salah satu algoritma untuk memecahkan permasalahan lintasan terpendek yaitu algoritma *Dijkstra*.Algoritma ini diberi nama sesuai nama penemunya, Edsger Wybe Dijkstra. Algortima *Dijkstra* mencari lintasan terpendek dalam sejumlah langkah. Algoritma ini pada setiap langkah akan memilih sisi yang berbobot minimum dan memasukannya kedalam himpunan solusi. Algoritma *Dijkstra* menggunakan strategi *Greedy* dalam mencari lintasan terpendek. Strategi *Greedy* adalah strategi yang memecahkan masalah langkah demi langkah, pada setiap langkah mengambil pilihan yang terbaik yang dapat diperoleh dengan berharap bahwa pemilihan solusi optimum lokal pada setiap langkah akan mencapai solusi optimum global. Dengan demikian algoritma Dijkstra adalah sebagai berikut : “Pada setiap langkah, ambil sisi yang berbobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan sebuah simpul lain yang belum terpilih. Lintasan dari simpul asal ke simpul yang baru haruslah merupakan lintasan terpendek diantara semua lintasannya ke simpul-simpul yang belum terpilih”. (Ginanjar, 2010 : 21)

Untuk bisa menerapkan algoritma ini dibutuhkan beberapa data yang harus disiapkan, yaitu :

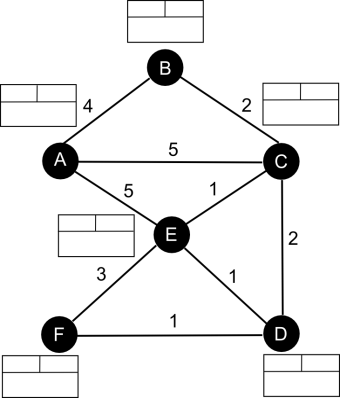
1. Beberapa Titik/simpul/daerah, titik/simpul/daerah yang bisa dijangkau secara langsung, dan juga jarak antara mereka.
2. Titik/simpul/daerah awal.
3. Titik/simpul/daerah tujuan.

Jika dicontohkan dengan gambar grafik akan seperti ini :

[](http://1.bp.blogspot.com/-UqUxYQowrek/UPZ6BBp86wI/AAAAAAAAACo/_FxUrsKZeng/s1600/Djikstar_01.png)

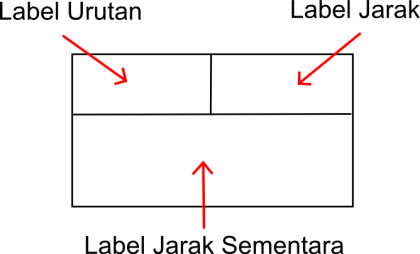
**Gambar 2.2** Titik **A** adalah titik awal dan titik **F** adalah titik tujuan.

Kemudian kita akan mencari rute manakah yang harus dilewati dan memilik total jarak yang paling dekat. Untuk bisa mendapatkan rute itu, maka grafik diatas ditambahkan beberapa kotak untuk mengisi beberapa label. Seperti ini :

[](http://1.bp.blogspot.com/-t4ofxDMy4hs/UPaBzif20VI/AAAAAAAAAC4/dTO4KBjaeJo/s1600/Djikstar_02.png)

**Gambar 2.3** Menambahkan beberapa kotak untuk mengisi beberapa label.

Penjelasannya adalah :

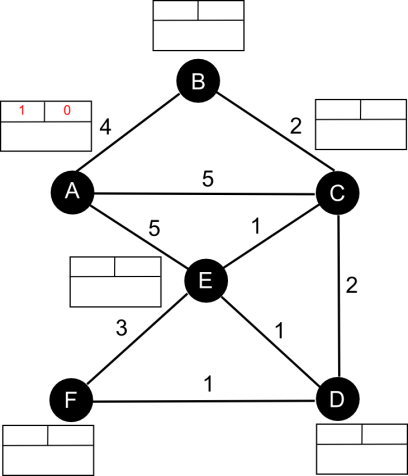
[](http://2.bp.blogspot.com/-XHT6TpeqCD8/UPaCFzVypvI/AAAAAAAAADA/Ciq-mMBrEVc/s1600/Djikstar_03.png)

**Gambar 2.4** Label urutan, label jarak dan label jarak sementara

Setelah itu ada beberapa langkah yang harus dilakukan, yaitu :

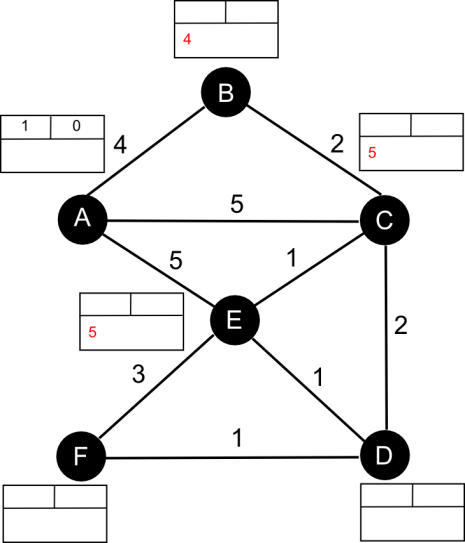
1. Mengisi kotak label pada titik awal dengan label urutan 1 dan label jarak 0.
2. Menetapkan label jarak sementara untuk semua titik yang dapat dihubungi langsung dari awal.
3. Pilih titik dengan label jarak sementara terkecil dan menuliskan nilainya di label jarak, serta tambahkan label urutan-nya.
4. Masukan 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 pada langkah pertama adalah Mengisi kotak label pada titik awal dengan label urutan 1 dan label jarak 0.

[](http://3.bp.blogspot.com/-S-qNKtFX1nE/UPdkW18ioZI/AAAAAAAAADQ/vyzKWPbEHwQ/s1600/Djikstar_04.png)

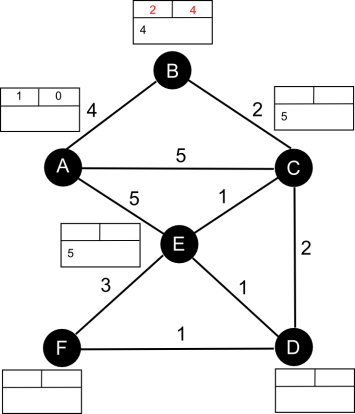
**Gambar 2.5** Mengisi kotak label pada titik awal dengan label urutan 1 dan label jarak 0.

Kemudian mengisi label jarak sementara titik yang dapat dihubungi langsung dari titik **A** yakni titik **B**, **C**, dan **E**.

[](http://2.bp.blogspot.com/-z7O3SYCygCA/UPdmq1Go8oI/AAAAAAAAADg/ZrM6XKQJ6MY/s1600/Djikstar_05.png)

**Gambar 2.6** Mengisi label jarak sementara

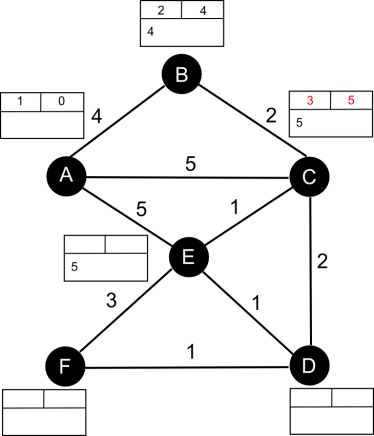
Maka yang terpilih adalah titik **B** karena memiliki label jarak sementara terkecil, dan mengisi nilai label jarak-nya sama dengan label jarak sementara serta memberikan label urutan-nya.

[](http://1.bp.blogspot.com/-TAYsUecgI88/UPdo0h9bzDI/AAAAAAAAADw/DVOnZcp2lxU/s1600/Djikstar_06.png)

**Gambar 2.7** Titik **B** yang terpilih

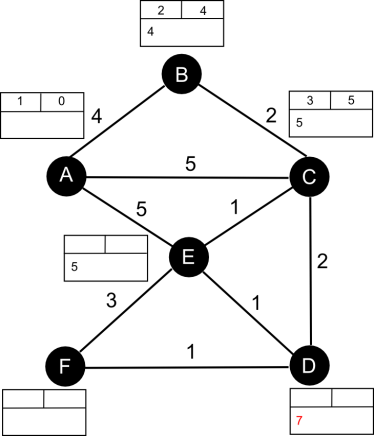
Selanjutnya mengisi label jarak sementara titik yang belum memiliki label jarak dan dapat dihubungi langsung dari titik **B** yaknihanyatitik **C.** Label jarak sementara titik **C** diisi dengan total jarak dari titik **A** sampai ke titik **C** yang melalui titik **B**, yakni **4 + 2 = 6**. Namun sebelumnya nilai label jarak sementara-nya titik **C** sudah ada dan lebih kecil **(5),** jadi label jarak sementara-nya tidak diganti dan tetap bernilai **5.**

Langkah selanjutnya adalah memilih label jarak sementara terkecil. Karena titik **E** dan titik **C** memiliki label jarak sementara yang sama yakni **5**, maka bisa memilih salah satu dari kedua titik tersebut. Misalkan titik **C** yang dipilih, maka berikan label jarak dan label urutan-nya.

[](http://4.bp.blogspot.com/-eyJCKynOE7s/UPd4G_9kt5I/AAAAAAAAAEQ/vadHgE0iclM/s1600/Djikstar_07.png)

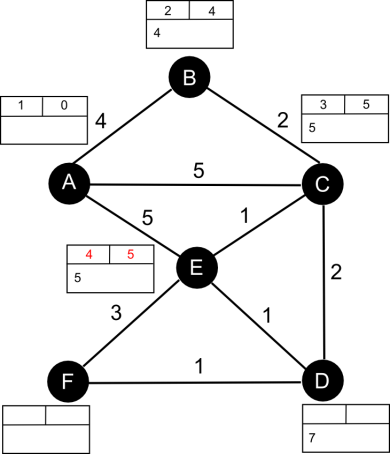
**Gambar 2.8** Memberikan label jarak dan label urutan pada titik **C**

Kemudian titik yang dapat dihubungi secara langsung dari titik **C** dan belum memiliki label jarak adalah titik **E** dan **D**. Titik **E => 5 + 1 = 6**, lebih besar jika dibandingkan dengan nilai label jarak sementara yang dimiliki oleh titik **E** sebelumnya **(5)**, maka nilai **6** diabaikan dan tetap diisi **5**. Titik **D => 5 + 2 = 7**, maka langsung saja label jarak sementara titik **D** diisi dengan **7**.

[](http://4.bp.blogspot.com/-1cKV5kJYiXU/UPd76QTU1zI/AAAAAAAAAEg/0I7GMrrRMh8/s1600/Djikstar_08.png)

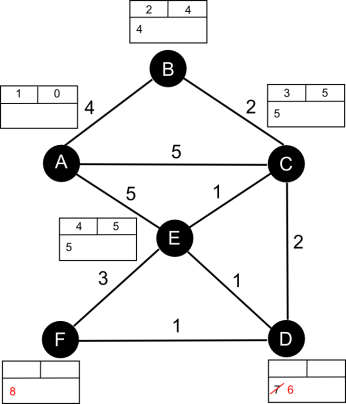
**Gambar 2.9** Label jarak sementara titik **D** diisi dengan **7**.

Selanjutnya titik **E** terpilih karena memiliki label jarak sementara terkecil. Berikan label jarak dan label urutan-nya.

[](http://2.bp.blogspot.com/-49_nSDc2rs0/UPd9F_l5xdI/AAAAAAAAAEw/3q2pk-cWkmk/s1600/Djikstar_09.png)

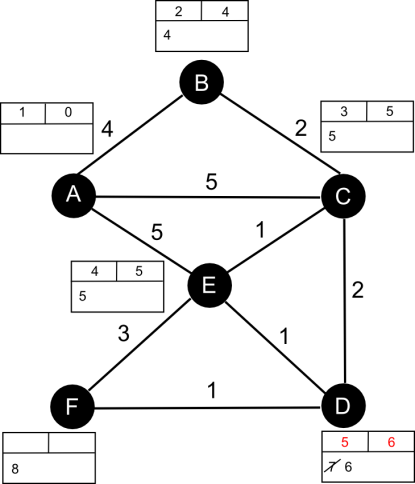
**Gambar 2.10** Memberikan label jarak dan label urutan pada titik **E.**

Dan titik **F** dan **D** adalah titik yang dapat dihubungi secara langsung dari titik **E** dan belum memiliki label jarak. **Titik F => 5 + 3 = 8** dan langsung diisikan kedalam label jarak sementara-nya. sedangkan titik **D => 5 + 1 = 6** dan lebih kecil dari pada nilai sebelumnya yaitu **7,** maka nilai label jarak sementara-nya diganti dengan **6**.

[](http://4.bp.blogspot.com/-Kb__lXI4mos/UPeAq8ia4mI/AAAAAAAAAFA/YnTlS4gwmLA/s1600/Djikstar_10.png)

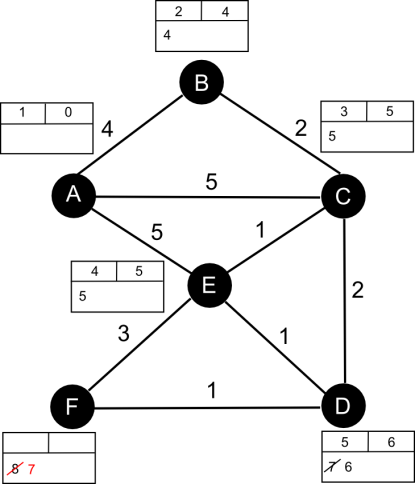
**Gambar 2.11** Nilai label jarak sementara pada titik **D** diganti dengan **6**.

Maka titik **D** terpilih karena memiliki label jarak sementara terkecil. Berikan label jarak dan label urutan-nya.

[](http://3.bp.blogspot.com/--H9K-PNeHg8/UPeBr69Wd9I/AAAAAAAAAFM/Dho8S_W1XIc/s1600/Djikstar_11.png)

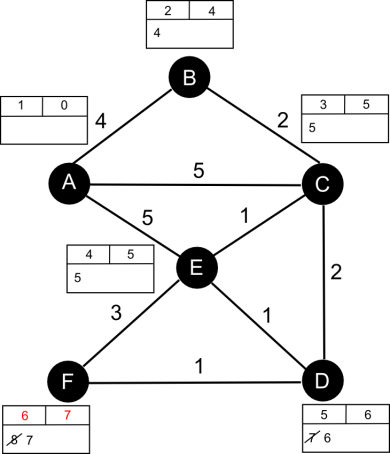
**Gambar 2.12** Memberikan label jarak dan label urutan pada titik **D.**

Titik **F** adalah titik terakhir yang dapat dihubungi secara langsung dari titik **D** dan belum memiliki label jarak serta merupakan titik tujuan. **Titik F => 6 + 1 = 7** dan lebih kecil dari pada nilai sebelumnya yaitu **8,** maka nilai label jarak sementara-nya diganti dengan **7**.

[](http://2.bp.blogspot.com/-g_e3_J_LbVk/UPeDmB-GT7I/AAAAAAAAAFg/OPNUZwWRu6g/s1600/Djikstar_12.png)

**Gambar 2.13** Nilai label jarak sementara pada titik **F** diganti dengan **7**.

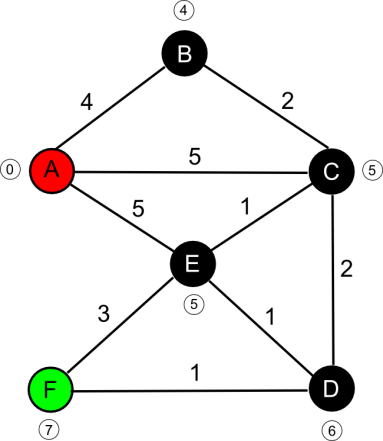
Karena titik **F** adalah satu-satunya titik terakhir yang belum mempunyai label jarak dan label urutan. maka langsung saja berikan nilai label jarak dan label urutan-nya. Dengan begitu titik tujuan sudah memiliki label jarak dan label jarak sementara.

[](http://2.bp.blogspot.com/-qika_ZiP6Ks/UPeFA53LR_I/AAAAAAAAAFw/Af8Vh88_a-4/s1600/Djikstar_13.png)

**Gambar 2.14** Memberikan label jarak dan label urutan pada titik **C**

Cara mengetahui rute yang harus dilewati :

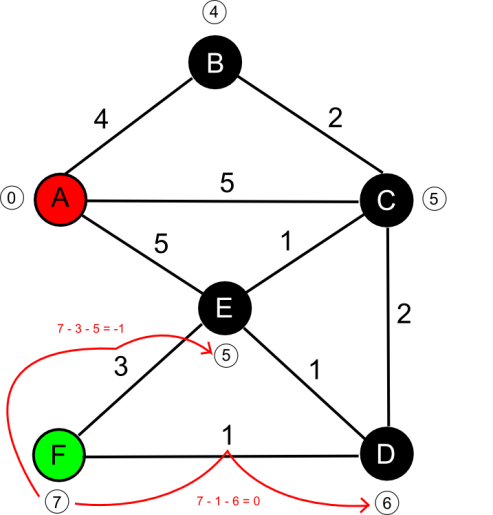
Untuk mengetahui rute manakah yang harus dilewati adalah dengan menelusuri kembali dari titik tujuan ke titik awal. tuliskan label jarak di samping setiap titik.

[](http://4.bp.blogspot.com/-pdsIi4tKi8A/UPeJTG1SQhI/AAAAAAAAAGA/rBeshf1jeu0/s1600/Djikstar_15.png)

**Gambar 2.15** Menuliskan label jarak di samping setiap titik.

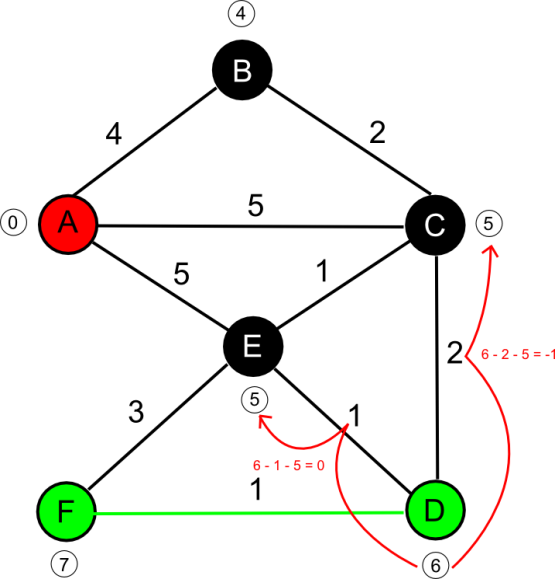
Titik mana sajakah yang dapat dihubungi langsung dari titik **F** ?, Yakni titik **E** dan **D**. maka, untuk menentukan titik manakah yang seharusnya dilewati adalah dengan cara mengurangkan label jarak titik **F** dengan jaraknya ke titik tujuan serta label jarak titik tersebut. jika hasilnya kurang dari **0** maka titik tersebut tidak layak untuk dilewati, dan jika hasilnya lebih dari **0** serta lebih mendekati **0** maka titik tersebut yang seharusnya dilewati.

Langkah pertama :

[](http://2.bp.blogspot.com/-dhOnqbKIF04/UPeMwhKxKgI/AAAAAAAAAGQ/R8cDfQ1T-AA/s1600/Djikstar_16.png)

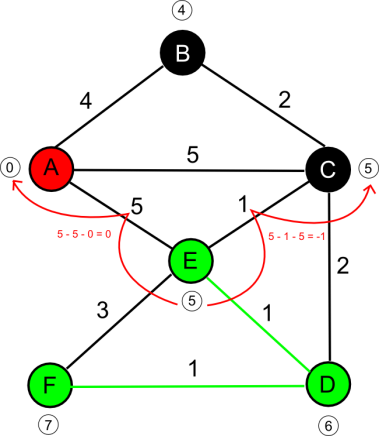
**Gambar 2.16** Langkah Pertama

Langkah kedua :

[](http://2.bp.blogspot.com/-ejY3PTo9eCU/UPeNFPLVJaI/AAAAAAAAAGY/TP4_RBnfy2Q/s1600/Djikstar_17.png)

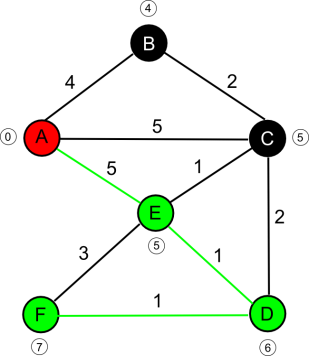
**Gambar 2.17** Langkah Kedua

Langkah ketiga :

[](http://3.bp.blogspot.com/-rB0VuzcNUDM/UPeNN1m85aI/AAAAAAAAAGg/rMODUuX9CQY/s1600/Djikstar_18.png)

**Gambar 2.18** Langkah Ketiga

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

[](http://4.bp.blogspot.com/-eFr7UlcvIUE/UPeNW6snmhI/AAAAAAAAAGo/PqA571A33Jg/s1600/Djikstar_19.png)

**Gambar 2.19** Rute yang harus dilewati dan memiliki rute terpendek.

* + 1. ***Google Maps***

*Google Maps* adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh *Google* lain dapat ditemukan di http://maps.google.com. 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.Melalui fitur *Google Maps*, pengguna internet dapat *browsing* informasi grafis berikut:

1. *Satellite Map*

Pengguna dapat menikmati gambar satelit planet bumi. Pengguna juga dapat menikmati foto satelit lebih detail lengkap dengan cara *zooming* pada bagian peta yang diinginkan.



**Gambar 2.20** *Satellite* *Map*

1. Hasil Pencarian Integrasi

Mencari lokasi, bisnis, peta buatan pengguna dan *real estate*.

1. *Draggable Maps*

Peta digital mapping yang *dragable* (bisa digeser) dengan bantuan mouse.

1. *Terrain Maps* (Peta Topograpi)

*Terrain Maps* menyediakan informasi fitur peta fisik atau peta topograpi yg biasa disediakan buku peta Atlas.



**Gambar 2.21** *Terrain* *Maps*

1. *Earth Map*

*Earth Map* menyediakan informasi peta bumi dimana akan tampak bumi secara utuh dan bila di-zoom akan terlihat awan yang menyelimuti bumi beserta pulau dan lautan yang tampak nyata dari ketinggian.



**Gambar 2.22** *Earth* *Map*

1. *My Location*

Dengan fitur ini pengguna dapat mengetahui letak dimana lokasi dari pengguna tersebut. (R. Rizky Ajie Oktora, 2014 : II-4)

* + 1. ***Google Maps* API**

*Google Maps* API *(application Program Interface)* 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. Dalam pembutan program *Google Map* APImenggunakan urutan sebagai berikut:

* 1. Memasukkan *Maps* API *JavaScript* ke dalam HTML.
  2. Membuat element div dengan nama map\_canvas untuk menampilkan peta.
  3. Membuat beberapa objek literal untuk menyimpan property-properti pada peta.
  4. Menuliskan fungsi *JavaScript* untuk membuat objek peta.
  5. Meng-inisiasi peta dalam tag body HTML dengan event onload.

Pada *Google Maps* APIterdapat 4 jenis pilihan model peta yang disediakan oleh *Google*, diantaranya adalah:

* 1. *ROADMAP*, untuk menampilkan peta biasa 2 dimensi
  2. *SATELLITE*, untuk menampilkan foto satelit
  3. *TERRAIN*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
  4. *HYBRID*, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan dan nama kota). (R. Rizky Ajie Oktora, 2014 : II-6)
     1. **Siklus Hidup Pengembangan Sistem**

Dalam membangun sebuah sistem (dalam hal ini lebih mengacu kepada pengertian aplikasi perangkat lunak) digunakan metode siklus hidup pengembangan sistem (*System Development Life Cycle* atau SDLC). SDLC terdiri dari sejumlah tahapan yang dilaksanakan secara berurutan yaitu :

1. *Analisis*
2. *Design*
3. *Implementasi*

*System Development Life Cycle* atau siklus hidup pengembangan sistem (SDLC), merupakan metode alternative. Metode SDLC mempunyanyi beberapa kelebihan dan kekurangan.

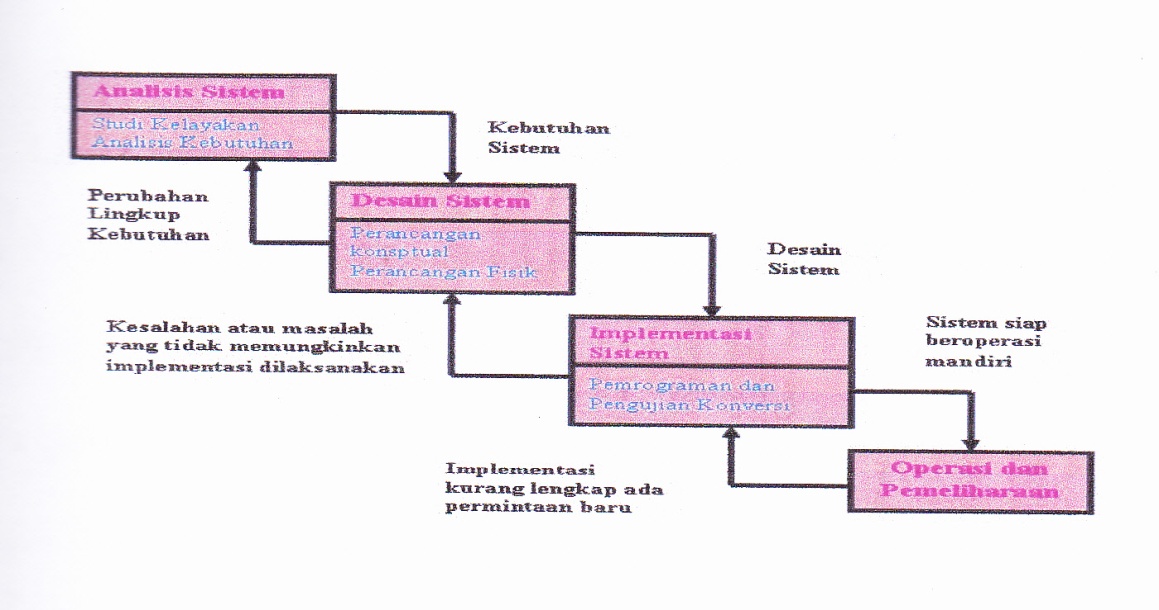
Kelebihan – kelebihan dari metode ini adalah :

1. Menyediakan tahapan yang dapat digunakan sebagai pedoman mengembangkan sistem.
2. Memberikan hasil sistem yang lebih baik karena sistem dianalisis dan dirancang secara keseluruhan sebelum diimplementasikan.

Disamping kelebihan-kelebihan tersebut, SDLC juga mempunyai kekurangan, yang diantaranya adalah :

1. Hasil dari SLDC tergantung dari hasil tahap analisis, sehingga jika terdapat kesalahan analisis, akan terbawa terus.
2. Dibutuhkan waktu yang lama untuk mengembangkannya karena sistem harus dikembangkan sampai selesai semua terlebih dahulu.

Siklus hidup pengembangan sistem dengan langkah-langkah utama adalah sebagai berikut:



**Gambar 2.23** Siklus Hidup Pengembangan Sistem

* + - 1. **Analisis Sistem**

Menurut Jogiyanto (2005:129) analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan–perbaikannya.

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan didalam tahap ini akan menyebabkan juga kesalahan ditahap selanjutnya.

Menurut Abdul Kadir (2003 : 38) analisis sistem mencakup studi kelayakan dan analisis kebutuhan yaitu **:**

1. Studi kelayakan

Studi kelayakan merupakan proses yang mempelajari atau menganalisa permasalahan yang telah ditentukan sesuai dengan tujuan akhir yang akan dicapai. Analisis kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan ini berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat tercapai dengan sumber daya dan dengan memperhatikan kendala yang terdapat pada permasalahan serta dampak terhadap lingkungan sekeliling. Lima macam kelayakan dalam merancang sistem informasi yaitu kelayakan teknik, kelayakan ekonomi, kelayakan operasi, kelayakan hukum dan kelayakan jadwal.

1. Analisis kebutuhan

Analisis kebutuhan merupakan proses untuk menghasilkan spesifikasi kebutuhan. Spesifikasi kebutuhan adalah spesifikasi yang rinci tentang pengolahan data yaitu jumlah data yang harus diproses, waktu pengolahan saat data siap diproses sampai informasi yang dihasilkan. Spesifikasi ini digunakan untuk membuat  kesepakatan dalam pengembangan sistem.

Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut ini.

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.
   * + 1. **Desain Sistem**

Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem (systems design). (Jogiyanto, 2005 : 195)

Tahap desain sistem mempunyai dua maksud atau tujuan utama, yaitu sebagai berikut :

1. Untuk memenuhi kebutuhan kepada pemakai sistem
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya yang terlibat.

Desain sistem dapat dibagi dalam dua bagian, yaitu desain sistem secara umum (*general systems design*) dan desain sistem terinci (*detailed systems design*).

1. **Desain Sistem Secara Umum (*general systems design*)**

Tujuan dari desain sistem secara umum adalah untuk memberikan gambaran secara umum kepada user tentang sistem yang baru, yang mana merupakan persiapan dari desain sistem secara rinci. Desain secara umum dilakukan oleh analisis sistem untuk mengidentifikasikan komponen-komponen sistem informasi yang akan di desain secara rinci oleh pemrograman komputer dan ahli teknik lainnya.

1. **Desain Sistem Terinci (*detailed systems design*)**

* **Desain *Output* Terinci**

Desain output terinci dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk *output-output* dari sistem yang baru. Desain *output* terinci terbagi atas dua, yaitu desain *output* berbentuk laporan di media kertas dan desain *output* dalam bentuk dialog di layar terminal.

1. Desain *Output* dalam bentuk laporan

Desain ini dimaksudkan untuk menghasilkan output dalam bentuk laporan di media kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk table dan berbentuk grafik atau bagan.

1. Desain *Output* dalam bentuk layar terminal

Desain ini merupakan rancang bangun dari percakapan antara pemakai sistem (*user*) dengan komputer. Percakapan ini terdiri dari proses memasukkan data ke sistem, menampilkan *output* informasi kepada *user*, atau keduanya.

* **Desain *Input* Terinci**

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data hasil dari transaksi merupakan masukan untuk sistem informasi. Hasil dari sistem informasi tidak lepas dari data yang dimasukan.

* **Desain Database Terinci**

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan dan disimpan diluar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai penyedia informasi bagi para pemakainya. Penerapan *database* dalam sistem informasi disebut *Database System*.

Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam didalam suatu organisasi. Dengan sistem basis data ini tiap-tiap orang atau bagian dapat memandang *database* dari beberapa sudut pandang yang berbeda. Bagian kredit dapat memandangnya sebagai data penjualan, bagian personalia dapat memandangnya sebagai data karyawan, bagian gudang data dapat memandangnya sebagai data persediaan, semuanya terintegrasi dalam sebuah data yang umum.

* **Desain Teknologi**

Pada menjalankan model, menyimpan dan mengakses data, menghasilkan dan tahap ini kita menentukan teknologi yang akan di pergunakan dalam menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi yang di maksud meliputi :

1. Perangkat keras (*hardware)*, yang terdiri dari alat masukan,alat pemroses,alat *output* dan simpanan luar.
2. Perangkat lunak (*software*), yang terdiri dari perangkat lunak sistem operasi (*operating system*), perangkat lunak bahasa (*language software*) dan perangkat lunak aplikasi (*application software*).
3. Sumber daya manusia (*brainware*), misalnya operator komputer, pemrograman, spesialis telekomunikasi, sistem analis dan lain sebagainya.

Desain teknologi sangat diperlukan pada tahap implementasi dan gpengujian untuk membuktikan bahwa sistem dapat berjalan secara semestinya.

* **Desain Model**

Analisis sistem dapat mendesain model dari sistem informasi yang diusulkan dalam bentuk *physical system dan logical model.* Bagan alir sistem (*systems flowchart*) merupakan alat yang tepat digunakan untuk menggambarkan physical system*.* Simbol-simbol bagan alir sistem ini menunjukkan secara tepat arti fisiknya, seperti simbol terminal, *hard disk,* laporan-laporan. *Logical Model* dari sistem informasi lebih menjelaskan kepada *user* bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja.

*Logical Model* dapat digambarkan dengan menggunakan diagram arus data (*data flow diagram*). Arus dari data di DAD dapat dijelaskan dengan menggunakan kamus data (*data dictionary*). Sketsa dari *physical system* dapat menunjukkan kepada user bagaimana nantinya sistem secara fisik akan diterapkan.

Bagan alir sistem merupakan bagan yang menunjukan arus pekerjaan seara keseluruhan dari sistem. Bagan alir sistem digambar dengan simbol-simbol berikut :

**Tabel 2.1** Simbol Bagan Alir Sistem

|  |  |  |
| --- | --- | --- |
| NAMA SIMBOL | SIMBOL | KETERANGAN |
| Simbol Terminal  Simbol Dokumen  Simbol Kegiatan Manual  Symbol Simpanan *Offline*  Simbol Kartu Plong  Simbol Proses |  | Menunjukan permulaan atau akhir dari suatu program  Menunjukan dokumen input dan output baik itu proses manual, mekanik, atau komputer  Menunjukan pekerjaan manual  Menunjukan file non-komputer yang diarsip urut angka (*numerical*), huruf (*alphabetical*), atau tanggal (*chronological*).  Menunjukan input dan output yang menggunakan kartu plong (*punched card*).  Menunjukan kegiatan proses dari operasi program komputer |

|  |  |  |
| --- | --- | --- |
| NAMA SIMBOL | SIMBOL | KETERANGAN |
| Simbol Operasi Luar  Simbol Pengurutan *Offline*  Simbol Pita Maknetik  Simbol Hard disk  Simbol Diskette  Simbol Drum Maknetik  Simbol Pita Kertas Berlubang  Simbol *Keyboard*  Simbol *Display*  Simbol Pita Kontrol |  | Menunjukan operasi yang dilakukan diluar proses operasi komputer  Menunjukan proses urut data di luar proses komputer  Menunjukkan *input* dan *output* menggunakan pita *magnetic*  Menunjukkan *input* dan *output* menggunakan *harddisk*  Menunjukkan *input* dan *output* menggunakan *diskette*  Menunjukkan *input* dan *output* menggunakan drum magnetik  Menunjukkan *input* dan *output* menggunakan pita kertas berlubang  Menunjukkan *input* yang menggunakan *on-line keyboard*  Menunjukkan *output* yang di tampilkan di monitor  Menunjukkan penggunaan pita kontrol (*control tape)* dalam *batch control* total untuk pencocokan di proses *batch processing* |

|  |  |  |
| --- | --- | --- |
| NAMA SIMBOL | SIMBOL | KETERANGAN |
| Simbol Hubungan Komunikasi  Simbol Garis Alir  Simbol Penjelasan  Simbol Penghubung |  | Menunjukkan proses transmisi data melalui channel komunikasi  Menunjukkan arus dari proses  Menunjukkan penjelasan dari suatu proses  Menunjukkan penghubung ke halaman yang masih sama atau ke halaman yang lain |

Sumber : Jogiyanto, 2005 : 802

Untuk mempermudah gambaran suatu sistem yang ada atau sistem yang baru yang akan di kembangkan secara logika tanpa memperhatikan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik di mana data tersebut akan di simpan, maka di gunakan Diagram Arus Data atau *Data Flow Diagram* (DFD). Dalam menggambarkan sistem perlu dilakukan pembentukan simbol, berikut ini simbol-simbol yang sering di gunakan dalam DAD :

1. *External entity* (Kesatuan luar) atau *boundary* (batas sistem)

Seiap sistem pasti mempunyai batas sistem (*boundary)* yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* an dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity)* merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luarnya yang akan memberikan *input* serta menerima *output* dari sistem. (Jogiyanto, HM, 2005 : 701)

**Gambar 2.24** Notasi kesatuan luar di DAD

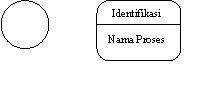
1. *Data Flow* (arus data)

Arus data ini menunjukkan arus atau aliran data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. (Jogiyanto, HM. 2005 : 701)

**Gambar 2.25** Notasi Arus Data di DAD

1. *Process* (proses)

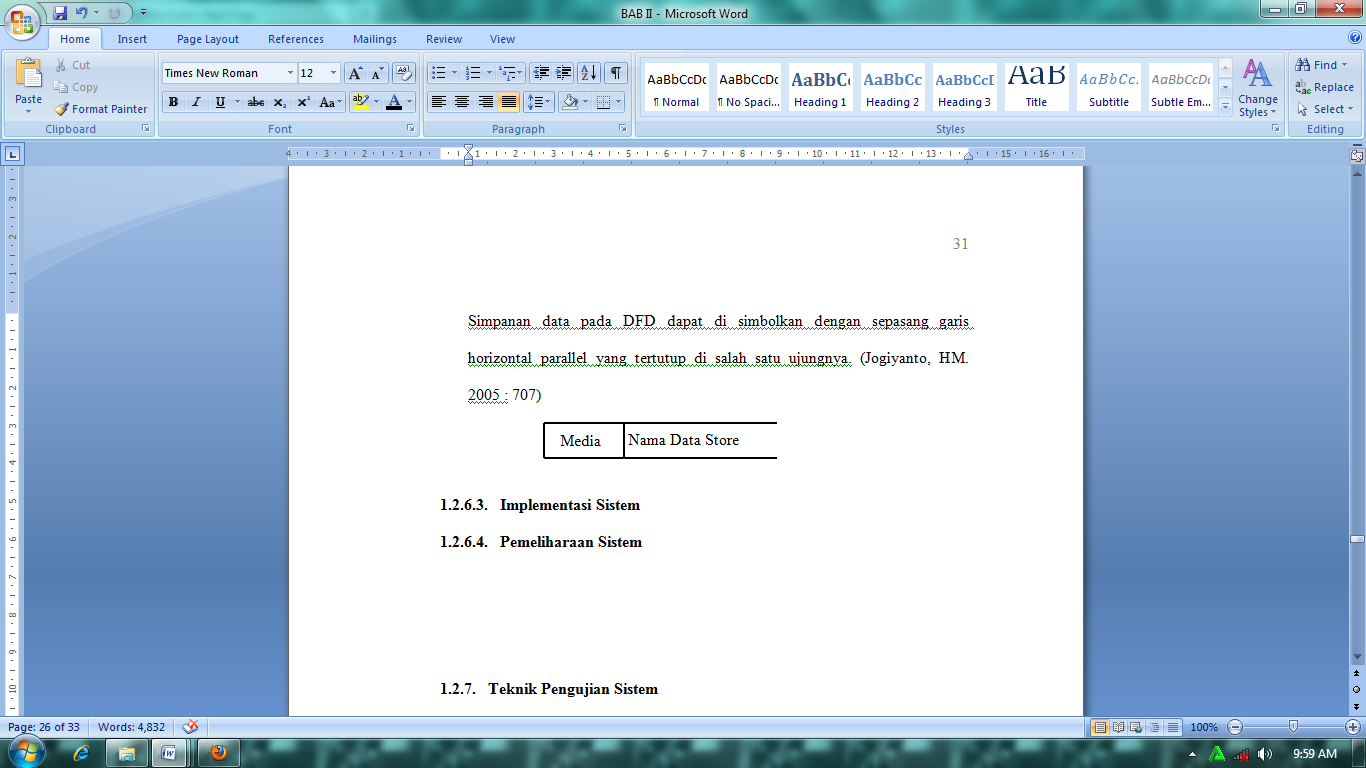
Suatu proses adalah kegiatan atau kerja yang di lakukan orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. (Jogiyanto, HM. 2005 : 705)

[](http://fairuzelsaid.files.wordpress.com/2010/01/image004.gif)

**Gambar 2.26** Notasi Proses di DAD

1. *Data Store* (simpanan data)

Simpanan data pada DFD dapat di simbolkan dengan sepasang garis horizontal parallel yang tertutup di salah satu ujungnya. (Jogiyanto, HM. 2005 : 707)



**Gambar 2.27** Notasi Simpanan Data di DAD

* + - 1. **Implementasi Sistem**

Menurut Abdul Kadir dalam bukunya “Perancangan Sistem Informasi” Andi , Yogyakarta, 2003 mengungkapkan : “Implementasi adalah kegiatan yang dilakukan untuk menguji data dan menerapkan sistem yang diperoleh dari kegiatan seleksi”.

Implementasi merupakan salah satu unsur pertahapan dari keseluruhan pembangunan sistem komputerisasi, dan unsur yang harus dipertimbangkan dalam pembangunan sistem komputerisasi yaitu masalah perangkat lunak (*software*), karena perangkat lunak yang digunakan haruslah sesuai dengan masalah yang akan diselesaikan, disamping masalah perangkat keras (*hardware*) itu sendiri.

Menurut Whitten, et al, 2004 Implementasi Sistem mempunyai 4 tahap, yaitu :

1. Membuat dan menguji basis data & jaringan.

Penerapan sistem yang baru atau perbaikan sistem dibuat pada basis data dan jaringan yang telah ada. Jika penerapan sistem yang baru memerlukan basis data dan jaringan yang baru atau dimodifikasi, maka sistem yang baru ini biasanya harus diimplementasikan sebelum pemasangan program komputer.

1. Membuat dan menguji program.

Merupakan tahap pertama untuk siklus pengembangan sistem yang spesifik bagi programer. Bertujuan untuk mengembangkan rencana yang lebih rinci dalam pengembangan dan pengujian program komputer yang baru.

1. Memasang dan menguji sistem baru.

Tahap ini dilakukan untuk menyakinkan bahwa kebutuhan integrasi sistem baru terpenuhi.

1. Mengirim sistem baru kedalam sistem operasi.

Tujuan tahap ini adalah untuk mengubah secara perlahan – lahan sistem lama menjadi sistem baru sehingga perlu dilakukan pemasangan basis data yang akan digunakan pada sistem baru.

* + - 1. **Pemeliharaan Sistem**

Pemeliharaan Sistem adalah suatu kombinasi dari berbagai tindakan yang dilakukan untuk menjaga suatu sistem dalam, atau memperbaikinya sampai, suatu kondisi yang bisa diterima. Pada bulan April 1970 didefinisikan sebuah istilah untuk Teknologi Pemeliharaan yang mencakup pengertian yang lebih luas dari pada pengertian Pemeliharaan diatas. Istilah ini adalah Teroteknologi.

Sistem perlu dipelihara karena beberapa hal, yaitu :

1. Sistem memiliki kesalahan yang dulunya belum terdeteksi, sehingga kesalahan- kesalahan sistem perlu diperbaiki.
2. Sistem mengalami perubahan-perubahan karena permintaan baru dari pemakai sistem.
3. Sistem mengalami perubahan karena perubahan lingkungan luar(perubahan bisnis)
4. Sistem terinfeksi
5. Sistem berkas corrupt
6. Perangkat keras melemah
7. Sistem perlu ditingkatkan

Tujuan dari pemeliharaan sistem:

1. Untuk memperpanjang usia kegunaan aset dari sistem tersebut. Hal ini terutama penting dinegara berkembang karena kurangnya sumber daya modal untuk penggantian. Dinegara-negara maju kadang-kadang lebih menguntungkan untuk ‘mengganti’ daripada ‘memelihara’.
2. Untuk menjamin ketersediaan optimum peralatan
3. Untuk menjamin kesiapan operasional dari seluruh peralatan yang diperlukan dalam keadaan darurat setiap waktu.
4. Untuk menjamin keselamatan orang yang menggunakan sarana tersebut

Jenis Pemeliharaan Sistem :

1. Pemeliharaan Korektif

Pemeliharaan korektif adalah bagian pemeliharaan sistem yang tidak begitu tinggi nilainya dan lebih membebani, karena pemeliharaan ini mengkoreksi kesalahan-kesahan yang ditemukan pada saat sistem berjalan.

1. Pemeliharaan Adaptif

Pemeliharaan adaptif dilakukan untuk menyesuaikan perubahan dalam lingkungan data atau pemrosesan dan memenuhi persyaratan pemakai baru. Lingkungan tempat sistem beroperasi adalah dinamik, dengan demikian, sistem harus terus merespon perubahan persyaratan pemakai. Misalnya, Undang-Undang Perpajakan yang baru mungkin memerlukan suatu perubahan dalam kalkulasi pembayaran bersih. Umumnya pemeliharaan adatif ini baik dan tidak dapat dihindari.

1. Pemeliharaan Perfektif

Pemeliharaan penyempurnaan mempertinggi cara kerja atau maintainabilitas (kemampuan untuk dipelihara). Tindakan ini juga memungkinkan sistem untuk memenuhi persyaratan pemakai yang sebelumnya tidak dikenal.  
Ketika membuat perubahan substansial modul apapun, petugas pemeliharaan juga menggunakan kesempatan untuk mengupgrade kode, mengganti cabang-cabang yang kadaluwarsa, memperbaiki kecerobohan, dan mengembangkan dokumentasi.  
Sebagai contoh, kegiatan pemeliharaan ini dapat berbentuk perekayasaan ulang atau restrukturisasi perangkat lunak, penulisan ulang dokumentasi, pengubahan format dan isi laporan, penentuan logika pemrosesan yang lebih efisien, dan pengembangan efisiensi pengoperasian perangkat.

1. Pemeliharaan Preventif

Pemeliharaan Preventif terdiri atas inspeksi periodik dan pemeriksaan sistem untuk mengungkap dan mengantisipasi permasalahan. Karena personil pemeliharaan sistem bekerja dalam sistem ini, mereka seringkali menemukan cacat-cacat (bukan kesalahan yang sebenarnya) yang menandakan permasalahan potensial. Sementara tidak memerlukan tindakan segera, cacat ini bila tidak dikoreksi di tingkat awal, jelas sekali akan mempengaruhi baik fungsi sistem maupun kemampuan untuk memeliharanya dalam waktu dekat.

* + 1. **Teknik Pengujian Sistem**
       1. ***White Box Testing***

Pengujian *White Box* adalah metode pengujian yang menggunakan struktur kontrol desain prosedur untuk memperoleh *test case*. Dengan menggunakan metode *white box*, perekayasa sistem dapat melakukan *test case* yang memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi *true and false,* mengeksekusi semua loop pada batasan mereka dan pada batas operasional mereka, dan menggunakan struktur data internal untuk menjamin validitasnya. Pengujian *Basis Path* adalah teknik pengujian *White Box* yang di usulkan pertama kali oleh Tom McCabe. Metode *Basis Path*  ini memungkinkan desainer *test case* mengukur kompleksitas logis dari desain prosedural dan menggunakannya sebagai pedoman untuk menentukan basis set dari jalur eksekusi. (Roger S. Pressman, 2002 : 536).



**Gambar 2.28** Bagan Alir

Bagan alir digunakan untuk menggambarkan struktur kontrol program dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain procedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut kedalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan didalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut *simpul* grafik alir, mempresentasikan satu atau lebih statemen procedural. Urutan kotak proses dan permata keputusan dapat memetakan simpul tunggal. Anak panah disebut *edges* atau *links*, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. *Edge* harus berhenti pada suatu simpul,meskipun bila simpul tersebut tidak merepresentasikan statemen procedural. (Roger S. Pressman, 2002 : 536).



**Gambar 2.29** Grafik Alir

*Kompleksitas siklomatis* adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Bila metrics ini digunakan dalam konteks metode pengujian *basis path*, maka nilai yang terhitung untuk komplesitas siklomatis menentukan jumlah jalur independen. Jalur independen adalah jalur yang melalui program yang mengintroduksi sedikitnya satu rangkaian statemen proses baru atau suatu kondisi baru.bila dinyatakan dengan terminology grafik alir, jalur independen harus bergerak sepanjang paling tidak satu *edge* yang tidak dilewatkan sebelum jalur tersebut ditentukan. Sebagai contoh, serangkaian jalur independen untuk grafik alir yang di tunjukkan pada gambar 2.29 adalah :

Jalur 1 : 1 – 11

Jalur 2 : 1 – 2 – 3 – 10 – 1 – 11

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Jalur 1, 2, 3, dan 4 yang ditentukan diatas terdiri dari sebuah *basis set* untuk grafik alir pada gambar 2.29 Bagaimana kita tahu banyaknya jalur yang dicari? Komputasi kompleksitas siklomatis memberikan jawaban. Fondasi Kompleksitas siklomatis adalah tori grafik, dan member kita metrics perangkat lunak yang sangat berguna. Kompleksitas dihitung dalam salah satu dari tiga cara berikut :

1. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis
2. Kompleksitas siklomatis, *V(G),* untuk grafik alir *G* ditentukan sebagai *V(G) = E – N + 2* di mana *E*  adalah jumlah *edge* grafik alir dan *N* adalah jumlah simpul grafik alir.
3. Kompleksitas siklomatis, *V(G),* untuk grafik alir *G* juga ditentukan sebagai *V(G) =* P + 1, di mana P adalah jumlah predikat yang diisikan dalam grafik alir G.

Pada gambar 2.29 grafik alir, kompleksitas siklomatis dapat dihitung dengan menggunakan masing-masing dari algoritma yang ditulis di atas :

1. Grafik alir mempunyai 4 region.
2. *V(G)* = 11 edge – 9 simpul + 2 = 4.
3. *V(G)* = 3 simpul yang diperkirakan + 1 = 4.

Dengan demikian, kompleksitas siklomatis dari grafik alir pada gambar 2.29 adalah 4. Yang lebih penting, nilai untuk *V(G)* member kita batas atas untuk jumlah jalur independen yang membentuk *basis set,* dan implikasinya batas atas.

* + - 1. ***Black Box Testing***

Menurut Ladjamudin (2006 : 379 ) Pengujian black box berfokus pada persyaratan fungsional perangkat lunak. Konsep pengujian black box (kotak hitam) digunakan untuk mempresentasikan sistem yang cara kerja didalamnya tidak tersedia untuk diinspeksi karena item-item yang diuji dianggap gelap karena logic nya tidak diketahui,yang diketahui hanya apa yang masuk dan apa yang keluar dari kotak hitam. Tehnik pengujian yang termasuk dalam kategori pengujian *blackbox* diantaranya *Graph-bases-testing, equivalence partitioning, comparison testing, artogonal array testing.*

* + 1. **Perangkat Lunak Pendukung**

Penelitian ini mengggunakan beberapa perangkat lunak pendukung, antara lain : PHP (*Personal Home Page*), *JavaScript* dan *MySQL*

* 1. **Kerangka Pemikiran**

**Peluang**

Algoritma *Dijkstra* menggunakan strategi *Greedy* dalam mencari lintasan terpendek di mana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih

**Masalah**

1. Bagaimana cara merekayasa sebuah sistem informasi geografis yang dapat menentukan rute terpendek pemadam kebakaran dengan menggunakan metode *dijkstra* ?
2. Bagaimana hasil penerapan metode *djikstra* dalam penentuan rute terpendek pemadam kebakaran?

Menerapkan algoritma *Dijkstra* dalam Penentuan Rute Terpendek Pemadam Kebakaran

**Solusi**

**Analisis Sistem**

* Sistem Berjalan
* Sistem Diusulkan
* *PHP*
* *Java Script*
* *MySql*
* Desain Model
* Desain *User Interface*
* Desain *Input*
* Desain *Proses*
* Desain *Output*
* Desain *Database*
* Desain Teknologi

**Pembangunan Sistem**

**Desain Sistem**

**Pengujian Sistem**

* *White Box*
* *Black Box*

**Implementasi Sistem**

UPTD Pemadam Kebakaran Kota Gorontalo

**Tujuan**

1. Untuk merekayasa sebuah sistem informasi geografis penentuan rute terpendek pemadam kebakaran dengan menggunakan metode *djikstra*.
2. Untuk menerapkan metode *dijkstra* dalam penentuan rute terpendek pemadam kebakaran.

**Gambar 2.30** Bagan Kerangka Pemikiran