

SISTEM PENUNJANG KEPUTUSAN PENCARIAN JARAK TERPENDEK MENUJU RUMAH SAKIT DAN PUSKESMAS DENGAN METODE DIJKSTRA

Hari Toha Hidayat

Program Studi Teknik Multimedia dan Jaringan Komputer,
Politeknik Negeri Lhokseumawe
Jl. Medan – Banda Aceh Km 280.3, Buketrata, Lhokseumawe
Email: haritoha@pnl.ac.id

Abstrak

Bagi warga pendatang sangat penting untuk mengetahui dimana saja letak rumah sakit dan puskesmas serta harus melewati jalur mana yang lebih dekat dari tempat mereka berada. Adapun tujuan di buatnya penelitian ini adalah sebagai berikut mempermudah masyarakat untuk mencari informasi rute rumah sakit dan puskesmas terdekat. DSS Merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Dijkstra Algorithms (Shortest Path Algorithms) adalah algoritma untuk menemukan jarak terpendek dari suatu *vertex* ke *vertex* yang lainnya pada suatu *graph* yang berbobot. Dari contoh data yang tersedia didapatkan jarak antara A menuju ke RSUD, dimana jarak yang ditempuh jika melalui B adalah 11 Km, jarak A yang ditempuh menuju RSUD melalui B ke D adalah 10 Km, jarak yang ditempuh A melalui D ke F adalah 4 Km, sementara jarak yang ditempuh A jika melalui F untuk menuju RSUD 2 Km. Maka jarak tempuh terdekat adalah melalui F.

Kata kunci: dijkstra, DSS, jarak

Latar Belakang Masalah

Keberadaan rumah sakit dan puskesmas sangatlah penting karena dapat membantu mengobati orang yang sedang sakit serta memberikan pelayanan Unit Gawat Darurat (UGD) selama 24 jam. Khususnya warga pendatang yang masih belum mengetahui dimana saja letak rumah sakit dan puskesmas serta harus melewati jalur mana yang lebih dekat dari tempat mereka berada. Pada saat mereka mengalami keadaan yang sangat genting, misalnya saja salah satu keluarga mereka mengalami kecelakaan, sementara mereka baru beberapa hari berada di Lhokseumawe, maka mereka kesulitan dalam menentukan harus dibawa ke mana dan melewati jalur mana yang lebih dekat dari tempat kejadian, agar orang tersebut segera mendapatkan pertolongan medis untuk memperkecil resiko yang tidak diinginkan.

Di era yang serba modern saat ini sudah sangat penting suatu aplikasi yang bisa menunjukkan lokasi terdekat dari rumah sakit maupun puskesmas agar memberikan kemudahan bagi masyarakat yang mencari informasi. Hal ini juga menuntut masyarakat untuk tidak gagap dalam teknologi sehingga bisa memanfaatkan aplikasi yang telah ada.

Dengan adanya aplikasi yang memberikan petunjuk untuk menuju rumah sakit dan puskesmas akan membuat

masyarakat tidak kebingungan dalam mencari rute yang tercepat dan terdekat untuk ditempuh.

Dari permasalahan tersebut penulis ingin membuat suatu aplikasi sistem pendukung keputusan pencarian jarak terpendek menuju rumah sakit dan puskesmas dengan metode djijkstra.

Tujuan Penelitian

Adapun tujuan di buatnya penelitian ini adalah sebagai berikut mempermudah masyarakat untuk mencari informasi rute rumah sakit dan puskesmas terdekat. Sehingga bisa membantu pendatang dalam mencari posisi dan rute rumah sakit maupun puskesmas terdekat dari tempat tinggalnya.

Kajian Pustaka

Menurut [1] Universitas Pendidikan Ganesha, Bali merupakan tempat tujuan wisata yang banyak diminati wisatawan domestik maupun mancanegara. Keindahan alam, keunikan budaya, keamanan, kenyamanan, keramah-tamahan yang dimiliki Bali merupakan faktor penyebab kenapa orang ingin menghabiskan liburan di Bali. Untuk menuju ke tempat wisata, ada beberapa rute yang bisa ditempuh. Wisatawan pastinya menginginkan jalur yang paling efisien untuk menuju tempat wisata tujuan sehingga dapat menghemat waktu dan biaya. Tentunya masih banyak para

wisatawan tidak mengetahui jalur-jalur untuk mengakses tempat wisata di Bali. Untuk mengatasi hal tersebut saya melakukan penelitian dengan judul "*pencarian rute terpendek tempat wisata di bali dengan menggunakan algoritma djikstra*".

Penelitian yang dilakukan [2], membuat sistem yang menggunakan *algoritma djikstra* yang dapat menemukan jalur tercepat dan terpendek dengan menyertakan faktor kecepatan dan waktu tempuh perjalanan, ruang lingkup yang luas dan bersifat *online*. Adapun penggunaan *algoritma djikstra* karena algoritma ini dipastikan menemukan solusi terbaik dan memiliki kompleksitas yang lebih sedikit jika dibandingkan dengan algoritma sejenis seperti *algoritma Bellman Ford dan Floyd Warshall*. Sistem ini memberikan keluaran berupa jalur tercepat dan terpendek dari tempat asal menuju tempat tujuan yang diinputkan oleh pengguna. Jalur tercepat dan terpendek tersebut dilengkapi dengan total jarak tempuh, waktu tempuh, serta kecepatan rata-rata.

Penelitian yang dilakukan [3], membuat sistem pencarian jalur terpendek dan rekomendasi objek wisata, yang diharapkan dapat membantu menentukan jalur obyek wisata lain yang dapat dijadikan untuk mengatur jadwal dari berwisata ataupun dapat digunakan menjadi bahan pertimbangan untuk menentukan alternative lokasi obyek wisata yang satu arah atau yang lokasinya berdekatan, sehingga dapat menghemat biaya dan waktu. Dari penjelasan di atas penelitian ini perlu adanya suatu cara dalam

menyelesaikan masalah, dari beberapa cara yang ada yang sesuai untuk pencarian jalur terpendek adalah dengan menggunakan algoritma dijkstra.

Menurut [4], Algoritma Dijkstra merupakan algoritma yang dapat digunakan dalam pencarian lintasan terpendek, di mana memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Pada setiap iterasi, jarak titik yang diketahui (dari titik awal) diperbarui bila ternyata didapat titik yang baru yang memberikan jarak terpendek. Pembuatan program Implementasi Algoritma Dijkstra Dalam Pencarian Lintasan Terpendek Lokasi Rumah Sakit, Hotel dan Terminal Kota Malang Berbasis Web diawali dengan perencanaan pembuatan program selanjutnya dilakukan pembangunan web dan diakhiri dengan publikasi web. Algoritma Dijkstra diimplementasikan ke dalam sebuah web dengan mengkonversi prosedur algoritma Dijkstra menjadi *script* program, kemudian disertakan dalam web. Titik-titik yang digunakan dalam program implementasi algoritma Dijkstra adalah nama rumah sakit, hotel dan terminal yang terletak di pusat kota Malang. Program yang dihasilkan disimpan dalam sebuah folder dengan nama Dijkstra yang ditempatkan dalam server web untuk dapat dijalankan.

Metodologi Penelitian

A. Sistem Pendukung Keputusan

Menurut [5], DSS Merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. DSS biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. DSS yang seperti itu disebut aplikasi DSS. Aplikasi DSS digunakan dalam pengambilan keputusan. DSS tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

Menurut [5] ditinjau dari tingkat teknologinya, DSS dibagi menjadi 3, yaitu:

1. SPK spesifik

SPK spesifik bertujuan membantu memecahkan suatu masalah dengan karakteristik tertentu. Misalnya, SPK penentu harga satuan barang.

2. Pembangkit SPK

Suatu software yang khusus digunakan untuk membangun dan mengembangkan SPK. Pembangkit SPK akan memudahkan perancang dalam membangun SPK spesifik.

3. Perlengkapan SPK

Berupa software dan hardware yang digunakan atau mendukung pembangunan SPK spesifik maupun pembangkit SPK.

Menurut [5] berdasarkan tingkat dukungannya, DSS dibagi menjadi 6 yaitu:

1. *Retrieve Information Elements*

Inilah dukungan terendah yang bisa diberikan oleh DSS, yakni berupa akses selektif terhadap informasi.

2. *Analyze Entre File*

Dalam tahapan ini, diberi akses untuk melihat dan menganalisis file secara lengkap.

3. *Prepare Reports from Multiple File*

Dukungan seperti ini cenderung dibutuhkan mengingat para manajer berhubungan dengan banyak aktivitas dalam satu momen tertentu.

4. *Estimate Decision Consequences*

Dalam tahapan ini, dimungkinkan untuk melihat dampak dari setiap keputusan yang mungkin diambil.

5. *Propose Decision*

Dukungan di tahapan ini sedikit lebih maju lagi. Suatu alternatif keputusan bisa disodorkan untuk dipertimbangkan.

6. *Make Decision*

Ini adalah jenis dukungan yang sangat diharapkan dari DSS. Tahapan ini akan memberikan sebuah keputusan yang tinggal menunggu legitimasi untuk dijalankan.

Tujuan dari DSS adalah [5]:

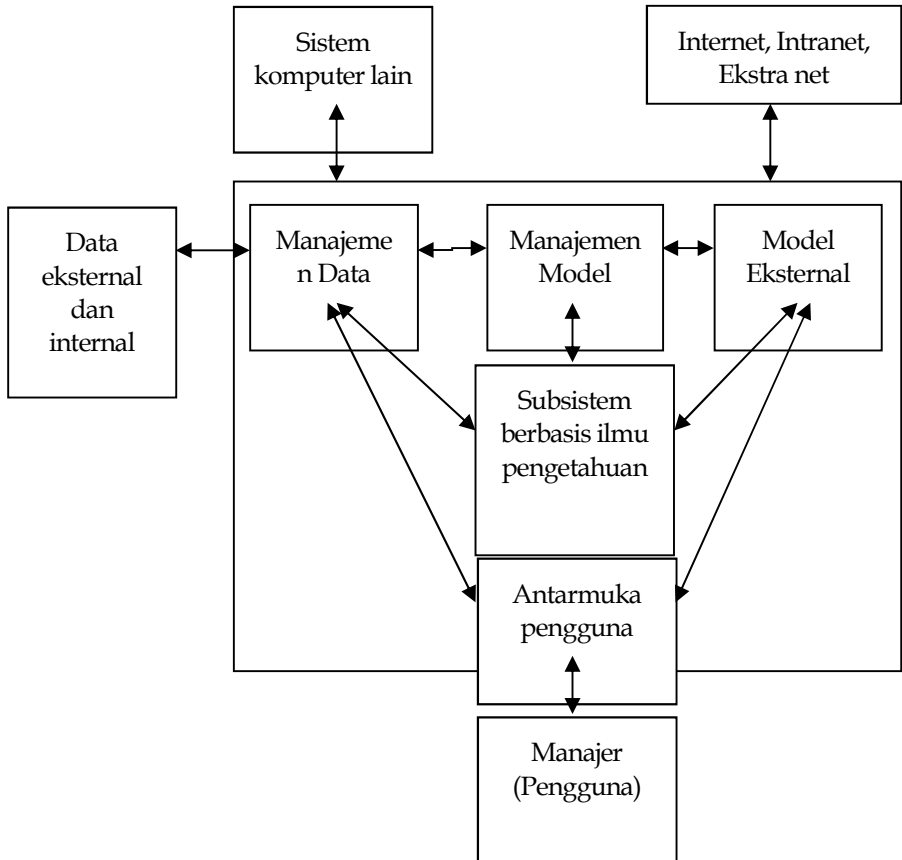
1. Membantu dalam pengambilan keputusan atas masalah semiterstruktur.
2. Memberikan dukungan atas pertimbangan dan bukannya dimaksudkan untuk mengganti fungsi manajer.
3. Meningkatkan efektivitas keputusan yang diambil lebih dari pada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya yang rendah.
5. Peningkatan produktivitas. Membantu satu kelompok pengambil keputusan, terutama para pakar, bisa sangat mahal. Pendukung komputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada di berbagai lokasi yang berbeda-beda

(menghemat biaya perjalanan). Selain itu, produktivitas staf pendukung (misalnya analis keuangan dan hukum) bisa ditingkatkan. Produktivitas juga bisa ditingkatkan menggunakan peralatan optimalisasi yang menentukan cara terbaik untuk menjalankan sebuah bisnis.

6. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat. Sebagai contoh, semakin banyak data yang diakses, makin banyak juga alternatif yang bisa dievaluasi. Analisis risiko bisa dilakukan dengan cepat dan pandangan dari para pakar (beberapa dari mereka berada di lokasi yang jauh) bisa dikumpulkan dengan cepat dan dengan biaya yang lebih rendah. Keahlian bahkan bisa diambil langsung dari sebuah sistem komputer melalui metode kecerdasan tiruan. Dengan komputer, para pengambil keputusan bisa melakukan simulasi yang kompleks, memeriksa banyak skenario yang memungkinkan, dan menilai berbagai pengaruh secara cepat dan ekonomis. Semua kapabilitas tersebut mengarah kepada keputusan yang lebih baik.
7. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan. Tekanan persaingan menyebabkan tugas pengambil keputusan menjadi sulit. Persaingan didasarkan tidak hanya pada harga, tetapi juga pada kualitas, kecepatan, kustomasi produk, dan dukungan

pelanggan. Organisasi harus mampu secara sering dan cepat mengubah model operasi, merekayasa ulang proses dan struktur, memberdayakan karyawan, serta berinovasi. Teknologi pengambilan keputusan bisa menciptakan pemberdayaan yang signifikan dengan cara memperbolehkan seseorang untuk membuat keputusan yang baik secara cepat, bahkan jika mereka memiliki pengetahuan yang kurang.

8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. Menurut Simon (1977), otak manusia memiliki kemampuan yang terbatas untuk memproses dan menyimpan informasi. Orang-orang kadang sulit mengingat dan menggunakan sebuah informasi dengan cara yang bebas dari kesalahan.



. Gambar 1. Skema komponen DSS

Kriteria atau ciri-ciri dari keputusan adalah:

1. Banyak pilihan/alternative.
2. Ada kendala atau syarat.
3. Mengikuti suatu pola/model tingkah laku, baik yang terstruktur maupun tidak terstruktur.

4. Banyak input/variabel
5. Ada faktor risiko.
6. Dibutuhkan kecepatan, ketepatan, dan keakuratan

B. Algoritma Dijkstra

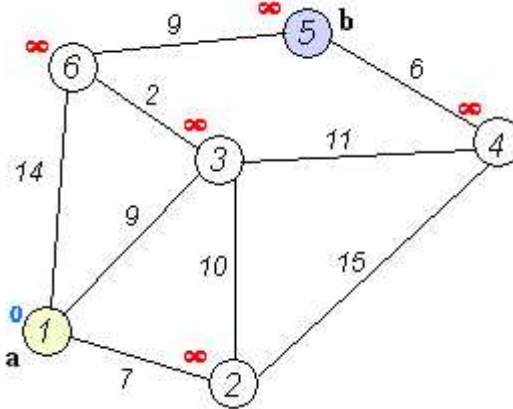
Menurut [6], Algoritma Dijkstra dinamakan sesuai dengan nama penemunya, seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger Dijkstra, adalah algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Dijkstra Algorithms (Shortest Path Algorithms) adalah algoritma untuk menemukan jarak terpendek dari suatu *vertex* ke *vertex* yang lainnya pada suatu *graph* yang berbobot, dimana jarak antar *vertex* adalah bobot dari tiap *edge* pada *graph* tersebut. Algoritma Dijkstra mencari jarak terpendek untuk tiap *vertex* dari suatu *subgraph* yang berbobot. Algoritma Dijkstra mencari jarak terpendek dari *node* asal ke *vertex* terdekatnya, kemudian ke *vertex* kedua, dan seterusnya. Secara umum, sebelum dilakukan iterasi, algoritma sudah mengidentifikasi jarak terdekat dari $i-1$ *vertex* terdekatnya. Selama seluruh *edge* berbobot tertentu yang (positif), maka *vertex* terdekat berikutnya dari *node* asal dapat ditemukan selama *vertex* berdekatan dengan *vertex* T_i . Kumpulan *vertex* yang berdekatan dengan *vertex* T_i dapat dikatakan sebagai "*fringe vertices*". *Vertex* inilah yang

merupakan kandidat dari algoritma dijkstra untuk memilih *vertex* berikutnya dari *node* asal. Algoritma Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi. Sifatnya sederhana dan lempang (*straightforward*).

Ada beberapa kasus pencarian lintasan terpendek yang diselesaikan menggunakan algoritma Dijkstra, yaitu:

- a. Pencarian lintasan terpendek antara dua buah simpul tertentu (*a pair shortest path*),
- b. Pencarian lintasan terpendek antara semua pasangan simpul (*all pairs shortest path*).
- c. Pencarian lintasan terpendek dari simpul tertentu ke semua simpul yang lain (*single-source shortest path*).
- d. Pencarian lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermediate shortest path*)

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.



Gambar 2. 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, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma Dijkstra:

- a. 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)
- b. Set semua node "Belum terjamah" dan set node awal sebagai "Node keberangkatan"
- c. Dari node 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.

- d. 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.
- e. Set "Node belum terjamah" dengan jarak terkecil (dari node keberangkatan) sebagai "Node Keberangkatan" selanjutnya dan lanjutkan dengan kembali ke step 3.

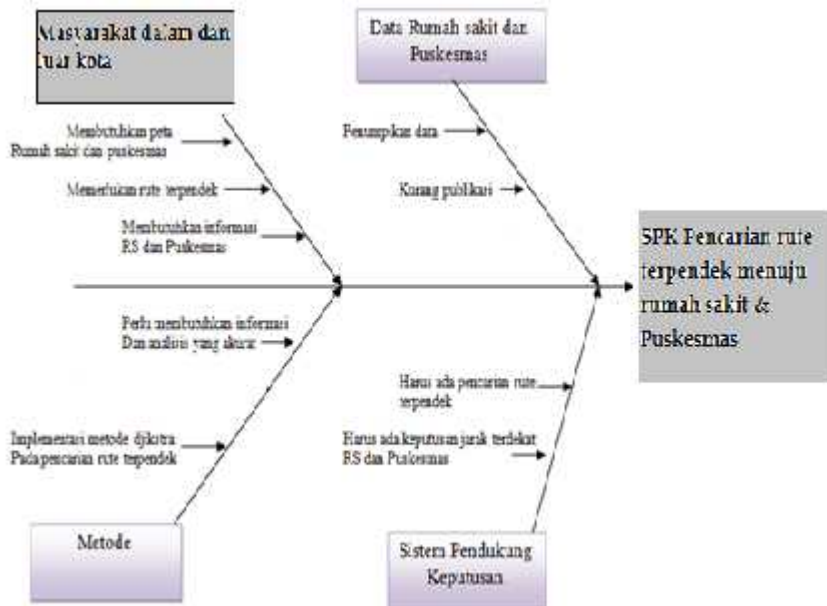
Analisis dan Perancangan

1. Analisis Masalah

Bagi warga pendatang, masih ada yang belum mengetahui dimana saja letak Rumah sakit dan harus melewati jalur mana yang lebih dekat dari tempat mereka berada. Pada saat mereka mengalami keadaan yang sangat genting, misalnya saja salah satu keluarga mereka mengalami

kecelakaan, sementara mereka baru beberapa hari berada di kota yang dikunjungi, maka mereka kesulitan dalam menentukan harus dibawa ke mana dan melewati jalur mana yang lebih dekat dari tempat kejadian, agar orang tersebut segera mendapatkan pertolongan medis untuk memperkecil resiko yang tidak diinginkan.

Mengidentifikasi masalah tersebut dapat digunakan *Diagram Ishikawa*. Diagram Ishikawa adalah sebuah alat grafis yang digunakan untuk mengeksplorasi dan menampilkan pendapat tentang komponen inti suatu kondisi di dalam organisasi. Diagram ini juga dapat menyusuri sumber-sumber penyebab atas suatu masalah. Oleh sebab itu, diagram Ishikawa sering disebut diagram sebab-akibat atau diagram tulang ikan (*fishbone diagram*), karena secara kasat mata digaram ini menyerupai tulang ikan. Identifikasi terhadap permasalahan melalui diagram ini akan membantu menganalisis kebutuhan sistem yang akan dikembangkan.



Gambar 3. Analisis masalah

Diagram Ishikawa di atas menunjukkan masalah utama berada pada kotak segi empat paling kanan (kepala ikan), sedangkan kategori penyebab utama ditunjukkan oleh kotak segi empat yang dihubungkan oleh sebuah garis ke tulang utama (garis horizontal yang terhubung ke kepala ikan). Selanjutnya dari kotak penyebab utama (diwakili oleh garis panah yang mengarah ke tulang kategori penyebab utama) ditampilkan masalah-masalah yang terdapat / terjadi pada penyebab utama. Sehingga secara keseluruhan, diagram tersebut telah menggambarkan sebab-akibat.

Setelah diidentifikasi dengan diagram Ishikawa, selanjutnya dapat digambarkan lebih spesifik ke diagram konteks, yang terdiri dari suatu proses yang menggambarkan ruang lingkup sistem sehingga memberikan gambaran umum tentang keseluruhan sistem.

2. Perancangan Sistem Pendukung Keputusan

Sistem pendukung keputusan untuk menentukan jarak terdekat rumah sakit dan puskesmas menggunakan metode djikstra yang sangat membantu masyarakat dan sekitarnya dalam mencari rumah sakit dan puskesmas terdekat . Dalam perancangan sistem pendukung keputusan ini dibutuhkan antara lain:

1. Model

Model sistem pendukung keputusan untuk menentukan jarak terdekat rumah sakit dan puskesmas menggunakan model matematika atau kuantitatif yaitu dengan menggunakan metode djikstra.

2. Data

Data yang diperlukan oleh sistem adalah nama rumah sakit dan nama puskesmas, dan jalan.

3. Proses - proses

Untuk menggambarkan proses – proses yang terjadi pada sistem ini perlu dibuat data flow diagram (DFD) atau diagram alir data (DAD).

4. Antar muka pengguna

Sistem ini memerlukan antar muka pengguna sehingga pengguna dapat berinteraksi dengan sistem, interaksi dapat berbentuk menampilkan output kepada pengguna.

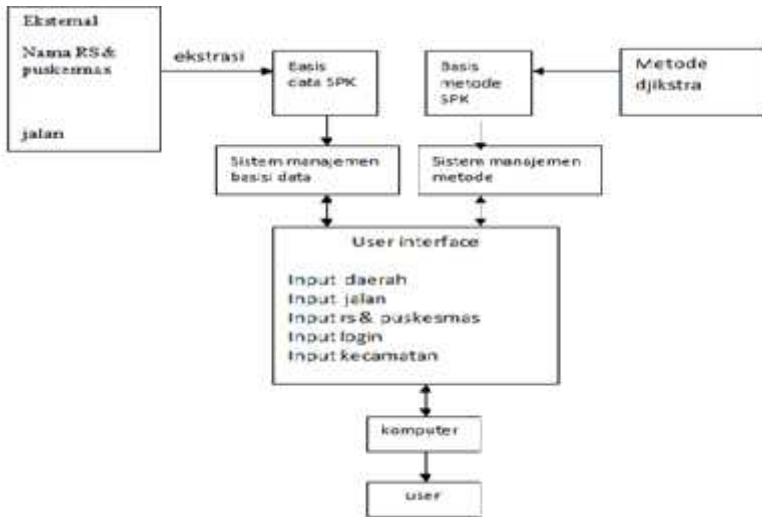
Adapun sumber data yang digunakan dalam sistem pendukung keputusan untuk menentukan jarak terdekat rumah sakit dan puskesmas sebagai berikut:

1. Data internal

Data internal adalah data yang berasal dari dalam organisasi yang diperlukan untuk mendukung sistem pendukung keputusan. Adapun yang tergolong dalam data internal adalah data nama rumah sakit dan puskesmas, nama jalan.

2. Ekstraksi data

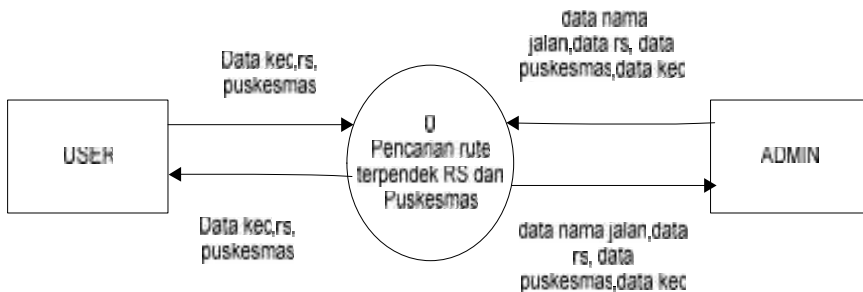
Berdasarkan data yang masuk maka disusun basis data sistem pendukung keputusan sehingga dengan tersedianya basis data tersebut dapat dilakukan pengelolaan data. Adapun sistem pendukung keputusan yang dibangun seperti terlihat pada Gambar 3.



Gambar 3. Rancangan sistem pendukung keputusan

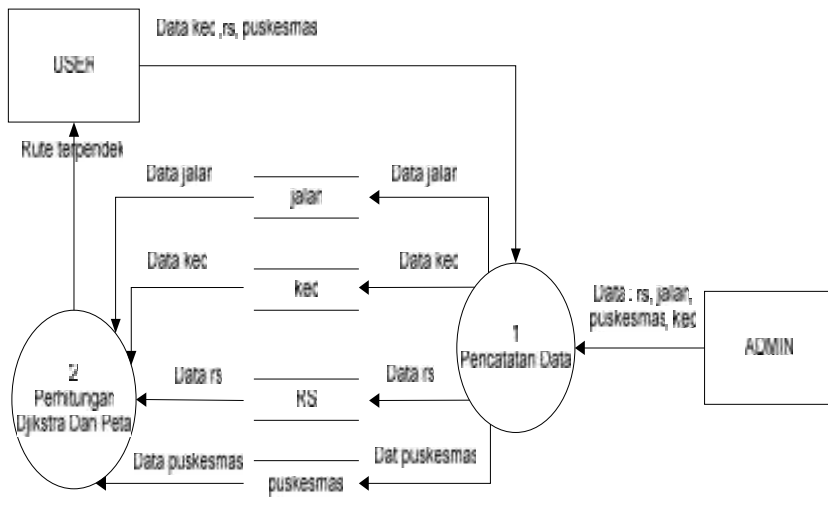
3. Perancangan Aplikasi

Data Flow Diagram (DFD) atau diagram aliran data adalah model proses yang digunakan untuk menggambarkan aliran data melalui sebuah sistem dan tugas atau pengolahan data yang dilakukan oleh sistem seperti pada gambar 4.



Gambar 4. Diagram konteks aplikasi spk rute terdekat Rs dan puskesmas

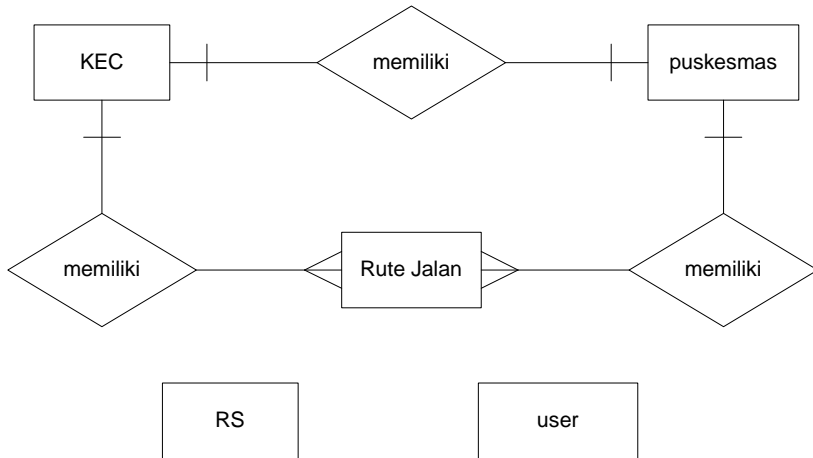
DFD level 1



Gambar 5. DFD level 1 spk rute terdekat RS dan Puskesmas

Entity Relationship Diagram (ERD) akan menjelaskan hubungan antar entitas dalam sebuah sistem. Maka sebelum

merancang basisdata, disusun terlebih dahulu model ERD. Adapun rancangan ERD seperti pada gambar 6.



Gambar 6. ERD spk rute terpendek RS dan Puskesmas

Hasil dan Pembahasan

Pengujian dilakukan dengan mengambil sampel 6 puskesmas dan satu rumah sakit. Dimana kedua puskesmas tersebut memiliki tujuan menuju ke rumah sakit. Rumah sakit yang digunakan disini adalah rumah sakit umum daerah (RSUD). Pada tabel 1 merupakan pencarian manual yang digunakan sebagai dasar acuan untuk menentukan nilai rute terpendek.

Tabel 1 acuan perhitungan rute terpendek dengan metode dijksra
Sesuai node asal dan node tujuan dari sebuah rute

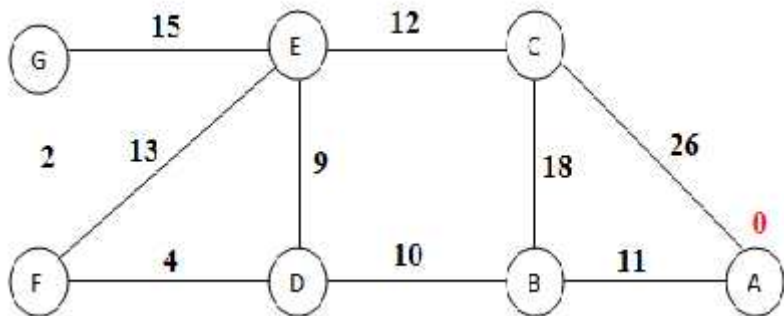
Lokasi	Puskes A	Puskes B	Puskes C	Puskes D	Puskes. E	Puskes. F	RSUD G
Jalan A	0	11	26	21	30	25	27
Jalan B	11	0	18	10	19	14	16
Jalan C	26	18	0	21	12	24	27
Jalan D	21	10	21	0	9	4	6
Jalan E	30	19	12	9	0	13	15
Jalan F	25	14	24	4	13	0	2

--	--	--	--	--	--	--	--

Kecamatan A

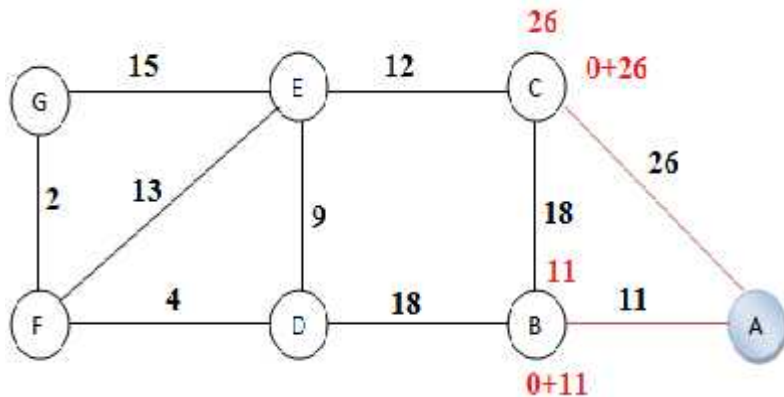
Berdasarkan tabel analisis, yang akan diselesaikan dijkstra apabila verteks asal ialah jalan A dan verteks tujuan ialah RSUD, adalah sebagai berikut :

- Titik A merupakan node awal keberangkatan dan verteks tujuan adalah ke titik G. Beri nilai titik awal dengan nilai 0.



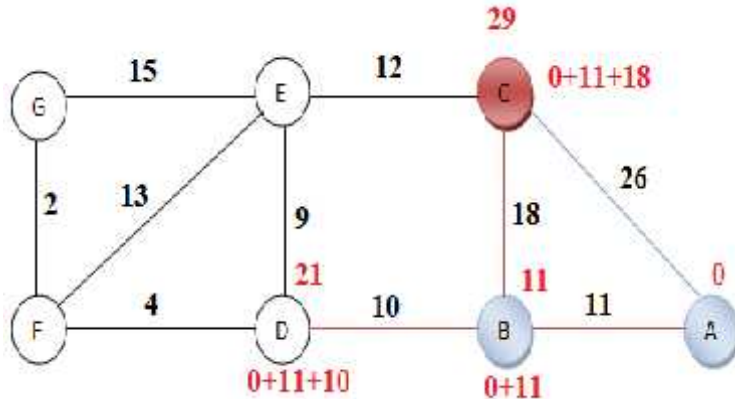
Gambar 7. Rute terpendek langkah pertama

- Dijkstra akan melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node A) , dan hasil yang di dapat adalah node B karena bobot nilai node B paling kecil di dibandingkan dengan nilai node yang lain, nilai = 11 (0+11).



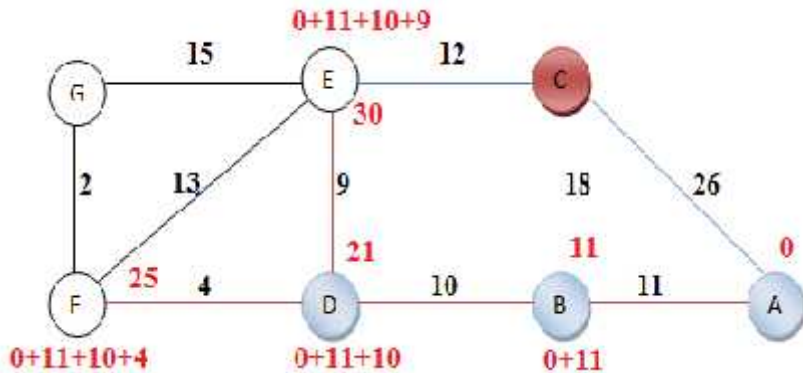
Gambar 8. Rute terpendek langkah kedua

- c. Node B diset menjadi node keberangkatan dan di tandai sebagai node yang sudah terjamah. Dijkstra akan melakukan kalkulasi kembali terhadap node - node tetangga yang terhubung langsung dengan node yang telah terjamah. Dan kalkulasi dijkstra menunjukkan bahwa node D yang menjadi node keberangkatan selanjutnya karena mempunyai nilai bobot yang paling kecil yaitu 10 ($0+10$). Total node yang telah terjamah adalah A - B - D dengan nilai 21 ($0+11+10$).



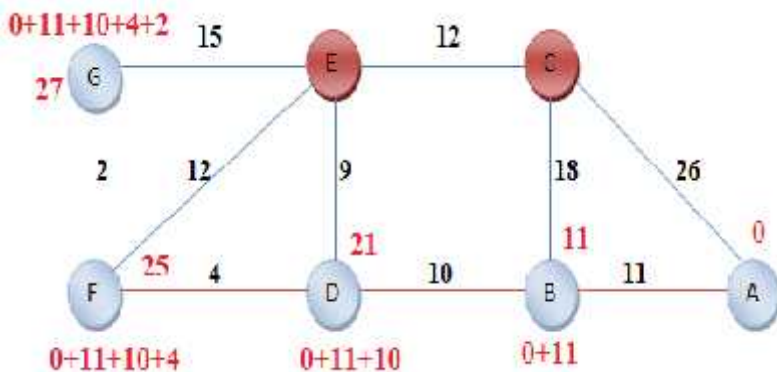
Gambar 9. Rute terpendek langkah ketiga

- d. Perhitungan berlanjut dengan node D di tandai sebagai node yang telah terjamah. Dijkstra akan melakukan kalkulasi kembali terhadap node - node tetangga yang terhubung langsung dengan node yang telah terjamah. Dan kalkulasi dijkstra menunjukan bahwa node F yang menjadi node keberangkatan selanjutnya karena mempunyai nilai bobot yang paling kecil yaitu 4 ($0+4$). Total node yang telah terjamah adalah A - B - D - F dengan nilai 25 ($0+11+10+4$).



Gambar 10. Rute terpendek langkah keempat

- e. Node F menjadi node yang telah terjamah, Dijkstra akan melakukan kalkulasi kembali dan menemukan bahwa node G (node tujuan) telah tercapai lewat node F. Jalur terpendeknya adalah A - B - D - F - G, dengan nilai bobot yang di dapat adalah 27 ($0+11+10+4+2$). Bila node tujuan telah tercapai maka kalkulasi djikstra dinyatakan selesai.



Gambar 11. Rute terpendek langkah kelima

Algoritma Dijkstra mencari jarak terpendek dari verteks asal ke verteks terdekatnya, kemudian ke verteks berikutnya, dan seterusnya dengan ketentuan :

$$d_i(\text{baru}) = \min \{ d_i(\text{lama}) , d_j + m_{ji} \}$$

Maka program akan berhenti karena semua node / verteks sudah terpilih. Dan menghasilkan jalur terpendek dari verteks A ke verteks G, sehingga akan didapat :

$$A \rightarrow G : A-B-D-F-G = 27$$

Atau pada dunia nyatanya, rute terpendek dari kecamatan A (verteks A) menuju ke RSUD (verteks G) adalah melewati kecamatan B, D, F hingga sampai RSUD dengan total jarak tempuh adalah 27 km.

Adapun bentuk aplikasi seperti terlihat pada gambar 12 berikut:



Rumah Sakit
Pudoesmas

Selamat Datang di Aplikasi Pencarian rute terpendek

Dari	Tujuan	Jarak
A	B	11 Km
B	D	10 Km
D	F	4 Km
F	RSUD	2 Km

Rute terdekat antara kecamatan A dengan RSUD melewati Kecamatan B — D — F — RSUD dengan jarak total 27 KM

Gambar 12. Aplikasi SPK pencarian rute terpendek Puskesmas dan Rumah Sakit

Berdasarkan contoh data yang tersedia didapatkan jarak antara A menuju ke RSUD, dimana jarak yang ditempuh jika melalui B adalah 11 Km, jarak A yang ditempuh menuju RSUD melalui B ke D adalah 10 Km, jarak yang ditempuh A melalui D ke F adalah 4 Km, sementara jarak yang ditempuh A jika melalui F untuk menuju RSUD 2 Km. Maka jarak tempuh terdekat adalah melalui F untuk menuju RSUD.

Kesimpulan

Aplikasi yang digunakan dalam pencarian rute terpendek ini sangatlah membantu masyarakat dalam mencari rute untuk menuju ke rumah sakit dan puskesmas. Adapun hasil perhitungan secara manual dan menggunakan aplikasi dengan metode dijkstra tidak ditemukan perbedaan. Aplikasi ini juga bisa memberikan pilihan kepada pengguna dalam mencari rute terdekat yang diinginkan.

Saran

Aplikasi ini akan semakin baik bila didukung dengan menggunakan sistem informasi geografis sehingga bisa membantu pengguna dalam melihat secara langsung lokasi yang akan dituju melalui peta.

Referensi

- [1] Dewi Luh Joni Erawati, 2010, "Pencarian Rute Terpendek Tempat Wisata Di Bali Dengan Menggunakan Algoritma Dijkstra", Seminar Nasional Aplikasi Teknologi Informasi (SNATI), hal D46-D49, ISSN 1907-5022, Yogyakarta,
<http://www.jurnal.uui.ac.id/index.php/Snati/article/viewFile/1926/1701>, diakses tanggal 06-03-2016

- [2] Fauzi Imron, 2011, "Penggunaan Algoritma Dijkstra Dalam Pencarian Rute Tercepat dan Rute Terpendek", Skripsi UIN Syarif Hidayatullah, Jakarta, <http://repository.uinjkt.ac.id/dspace/bitstream/123456789/2696/1/IMRON%20FAUZI-FST.pdf>, diakses tanggal 06-03-2016
- [3] Priatmoko, 2012, "Algoritma Dijkstra Untuk Pencarian Jalur Terdekat dan Rekomendasi Objek Pariwisata di Pulau Bali", Skripsi Universitas Dian Nuswantoro, Semarang, http://eprints.dinus.ac.id/13345/1/jurnal_13933.pdf, diakses tanggal 06-03-2016
- [4] Sholichin R dkk, 2011, "Implementasi Algoritma Dijkstra Dalam Pencarian Lintasan Terpendek Lokasi Rumah Sakit, Hotel dan Terminal Kota Malang Berbasis Web", Skripsi Universitas Negeri Malang, Malang, <http://jurnal-online.um.ac.id/data/artikel/artikelC7BAA81E70E4BA519A5B64FEA7B04B3B.pdf>, diakses tanggal 06-03-2016
- [5] Turban E dkk, 2007, "*Decision Support System and Intelligent System*", Andi Publisher, Yogyakarta
- [6] Pugas Okta D dkk, 2011, "Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra dan Astar (A*) pada SIG Berbasis Web untuk Pemetaan Pariwisata Kota Sawahlunto", Jurnal Transmisi, ISSN 1411-0814, hal 27-32, Undip Semarang, <http://ejournal.undip.ac.id/index.php/transmisi/article/viewFile/3632/pdf>, diakses tanggal 06-03-2016