

APLIKASI PENENTUAN RUTE RUMAH SAKIT TERDEKAT MENGUNAKAN ALGORITMA DIJKSTRA

Jenni Veronika Ginting¹, Ertina Sabarita Barus²

¹Manajemen Informatika

²Sistem Informasi

STMIK Kristen Neuman Indonesia Medan, Jl. Jamin Ginting Km.10.2 Medan, Sumatera Utara
20141

veronikarossy11@gmail.com¹, baruschild2@gmail.com²

Abstrak

Penghitungan rute terpendek memegang peranan penting dalam kehidupan sehari-hari karena harus dilakukan dalam waktu singkat dan pada saat itu juga agar segera dapat diketahui rute mana yang paling pendek untuk dilewati. Rumah sakit adalah tempat yang penting jika ada seseorang yang membutuhkan pertolongan medis. Letak rumah sakit di kota Kisaran berbeda-beda sehingga jika ingin menuju ke rumah sakit ada banyak pilihan rute yang harus dipilih, sesuai dengan kebutuhan dari pasien. Aplikasi penentuan rute terpendek rumah sakit dengan algoritma dijkstra yang akan dibangun berbasis web dengan script php dan database mysql. Aplikasi penentuan rute rumah sakit terdekat menggunakan parameter jarak objek dengan rumah sakit tujuan. Dengan menentukan lokasi objek dan rumah sakit yang dituju sebagai input. Jalur menuju rumah sakit terdekat sebagai output. Algoritma dijkstra akan melakukan penghitungan untuk mencari rute terpendek dari titik awal ke titik akhir. Sehingga dalam keadaan darurat ketika pasien membutuhkan pertolongan medis dengan melewati rute terpendek dapat membuat mobilitas sehari-hari menjadi lebih efisien.

Kata Kunci: Rute Terpendek, Algoritma Dijkstra, Rumah Sakit.

1. PENDAHULUAN

Waktu dapat dikaitkan dengan jarak tempuh, semakin pendek jarak tempuh maka semakin singkat waktu yang di butuhkan untuk menempuh jarak tersebut. Penghitungan rute terpendek memegang peranan penting dalam kehidupan sehari-hari karena harus dilakukan dalam waktu singkat dan pada saat itu juga agar segera dapat diketahui rute mana yang paling pendek untuk dilewati. Dengan melewati rute terpendek dapat membuat mobilitas sehari-hari menjadi lebih efisien. Untuk mencapai suatu tempat yang dituju terkadang seseorang tidak mengetahui terdapat jalur terdekat untuk mencapai tujuannya. Jika seseorang mengetahui jalur terdekat tersebut akan dapat mempersingkat waktu tempuh yang di milikinya.

Pada penelitian ini, di ambil contoh kasus rumah sakit yang ada di kota kisaran. Rumah sakit merupakan tempat yang penting jika ada seseorang yang membutuhkan pertolongan medis. Dalam keadaan darurat ketika pasien membutuhkan pertolongan medis dengan cepat, dibutuhkan jalur terpendek untuk mencapai rumah sakit.

Letak rumah sakit dikota Kisaran berbeda-beda sehingga jika ingin menuju ke rumah sakit

ada banyak pilihan rute yang harus dipilih, sesuai dengan kebutuhan dari pasien. Jarak dari tempat kejadian menuju rumah sakit memiliki perbedaan waktu tempuh. Terkadang ada orang yang mengabaikan waktu tempuh untuk menuju rumah sakit dengan jarak terdekat. Maka dari itu penulis merancang aplikasi penentuan rute terpendek rumah sakit dengan algoritma *dijkstra* dan aplikasi dibuat berbasis web. Algoritma ini akan melakukan penghitungan untuk mencari rute terpendek dari titik awal ke titik akhir

II. TEORI

a. Algoritma Dijkstra

Algoritma adalah urutan langkah-langkah untuk memecahkan suatu masalah." Algoritma adalah jantung ilmu komputer atau informatika. Banyak cabang dari ilmu komputer yang diacu dalam termologi algoritma, misalnya algoritma perutean (*routing*), algoritma *brensenham* untuk menggambar garis lurus, algoritma *knuth-morris-pratt* untuk mencari suatu pola didalam suatu teks.[4]

Algoritma adalah sebuah prosedur komputasi yang mentransformasikan sejumlah *input* menjadi sejumlah *output*. Sebuah algoritma dikatakan "benar (*correct*)" jika untuk setiap *inputnya*

menghasilkan *output* yang benar pula. Dalam hal ini algoritma dapat digunakan sebagai metode untuk mengetahui langkah-langkah secara urut untuk mencapai tujuan. Setiap algoritma memiliki perbedaan dalam mencapai tujuan.

Algoritma Dijkstra menurut penemunya seorang ilmuwan komputer, (Edsger Dijkstra), adalah sebuah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif. Dalam jurnal “Jurnal Informatika JUITA” [1]-[4]

Ada beberapa algoritma yang dapat digunakan untuk menentukan jalur terpendek. Salah satunya menggunakan algoritma *dijkstra*. 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.

Misalkan:

$$V(G) = \{v_1, v_2, \dots, v_n\}$$

L = Himpunan titik-titik $V(G)$ yang sudah terpilih dalam jalur path terpendek.

$D(j)$ = Jumlah bobot path terkecil dari v_1 ke v_j .

$w(i,j)$ = Bobot garis dari titik v_i ke v_j .

$w^*(1,j)$ = Jumlah bobot path terkecil dari v_1 ke v_j

Secara formal, algoritma *Dijkstra* untuk mencari path terpendek adalah sebagai berikut:

1. $L = \{ \}$;
 $V = \{v_2, v_3, \dots, v_n\}$.
2. Untuk $i = 2, \dots, n$, lakukan $D(i) = w(1, i)$
3. Selama $v_n \in L$ lakukan:
 - a. Pilih titik $v_k \in V - L$ dengan $D(k)$ terkecil. $L = L \cup \{v_k\}$
 - b. Untuk setiap $v_j \in V - L$ lakukan: Jika $D(j) > D(k) + W(k,j)$ maka ganti $D(j)$ dengan $D(k) + W(k,j)$
4. Untuk setiap $v_j \in V$, $w^*(1, j) = D(j)$

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.

III. METODOLOGI

Adapun kerangka kerja penelitian ini dapat digambarkan sebagai berikut:

1. Studi Literatur
Pada tahap ini dilakukan pencarian teori yang dari berbagai referensi seperti buku dan jurnal.
2. Pengumpulan Data

Proses pengumpulan data dilakukan dengan melakukan wawancara ke pihak yang terkait yaitu Dinas Kesehatan Kota Kisaran dan observasi untuk melakukan pengamatan secara langsung dalam pencarian data yang diperlukan, yaitu data tentang letak geografis rumah sakit yang ada di kota kisaran.

3. Analisis Sistem
Pada tahap ini dilakukan identifikasi masalah pada sistem yang sedang berjalan dan dilakukan analisa kebutuhan sistem.
4. Perancangan Sistem
Pada tahap ini dilakukan perancangan model menggunakan UML, perancangan *user interface*, perancangan *input* dan perancangan *output*.
5. Penulisan Kode Program dan Pengujian
Pada tahapan ini akan dilakukan *coding* program menggunakan Php, *database* Mysql dan editor *Dreamweaver CS3*. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat menggunakan *blackbox testing*. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki[5][6]

IV. HASIL DAN PEMBAHASAN

a. Analisa Metode Dijkstra

Ada beberapa algoritma yang dapat digunakan untuk menentukan jalur terpendek. Salah satunya menggunakan algoritma *dijkstra*. Algoritma ini telah secara luas digunakan dalam menentukan rute tersingkat atau jalur terpendek berdasarkan kriteria tertentu yang digunakan sebagai batasan.

Misalkan:

$$V(G) = \{v_1, v_2, \dots, v_n\}$$

L = Himpunan titik-titik $V(G)$ yang sudah terpilih dalam jalur path terpendek.

$D(j)$ = Jumlah bobot path terkecil dari v_1 ke v_j .

$w(i,j)$ = Bobot garis dari titik v_i ke v_j .

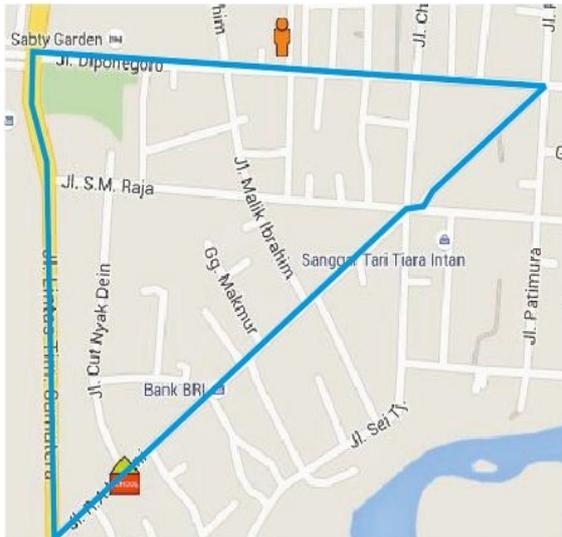
$w^*(1,j)$ = Jumlah bobot path terkecil dari v_1 ke v_j

Secara formal, algoritma *Dijkstra* untuk mencari path terpendek adalah sebagai berikut:

1. $L = \{ \}$;
 $V = \{v_2, v_3, \dots, v_n\}$.
2. Untuk $i = 2, \dots, n$, lakukan $D(i) = w(1, i)$
3. Selama $v_n \in L$ lakukan:
 - a. Pilih titik $v_k \in V - L$ dengan $D(k)$ terkecil. $L = L \cup \{v_k\}$
 - b. Untuk setiap $v_j \in V - L$ lakukan: Jika $D(j) > D(k) + W(k,j)$ maka ganti $D(j)$ dengan $D(k) + W(k,j)$
4. Untuk setiap $v_j \in V$, $w^*(1, j) = D(j)$

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari

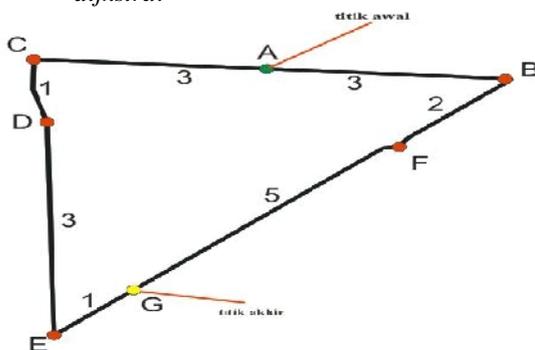
satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma *dijkstra* melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik. Berikut adalah contoh kasus penentuan rute terdekat menuju R.S.U Wira Husada di Kota Kisaran.



Gambar 1. Contoh Rute Menuju R.S.U Wira Husada

Langka-langkah dalam penerapan algoritma *dijkstra* dalam menentukan rute terdekat adalah sebagai berikut:

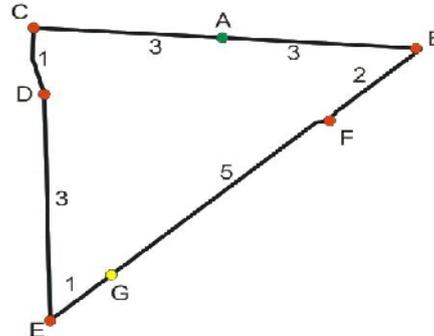
1. Pertama menentukan titik node dari setiap sudut yang ada.
2. tentukan titik mana yang akan menjadi node awal.
3. Tentukan titik yang menjadi node akhir.
4. lalu beri bobot jarak pada node pertama ke node terdekat satu per satu.
5. Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma *dijkstra*:



Gambar 2. Contoh Kasus Algoritma *Dijkstra*

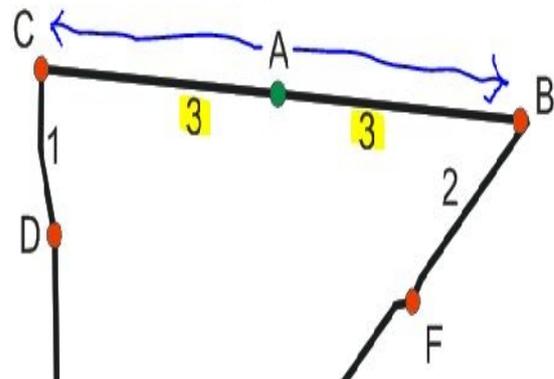
Dibawah ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

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



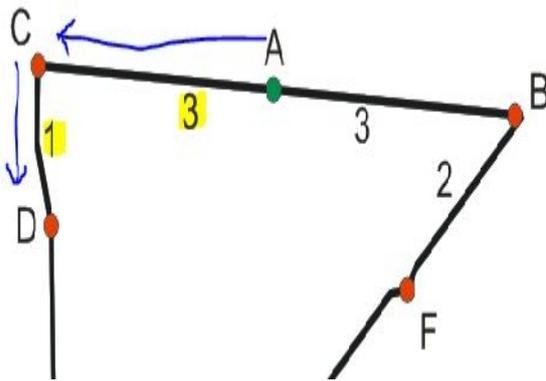
Gambar 3. Contoh Kasus Algoritma *Dijkstra* – Langkah 1

2. *Dijkstra* melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node A), dan hasil yang didapat adalah node C dan B karena bobot nilai node C dan B memiliki nilai yang sama, jadi node tetangga selanjutnya yang menentukan jarak terdekatnya.



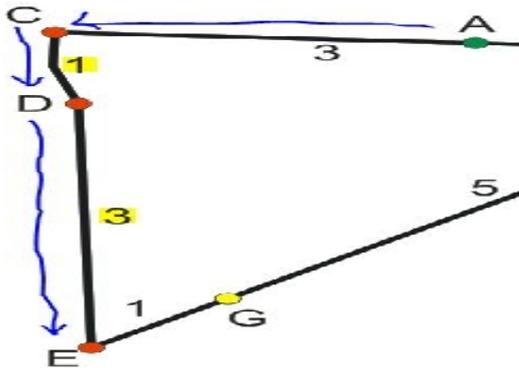
Gambar 4. Contoh Kasus Algoritma *Dijkstra* – Langkah 2

3. Node C dan B diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node yang telah terjamah. Dan kalkulasi *dijkstra* menunjukkan bahwa node D yang menjadi node keberangkatan selanjutnya, karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai 4 ($3+1$). Sedangkan node F tidak termasuk rute terdekat karena bobot nilai 5 ($3+2$), bobot nilai lebih besar di bandingkan node D.



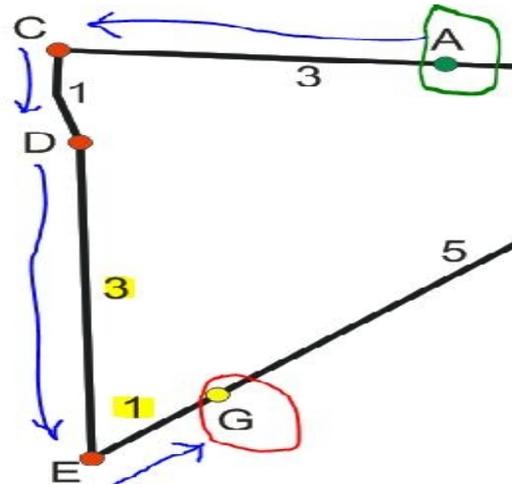
Gambar 5. Contoh Kasus Algoritma Dijkstra – Langkah 3

4. Perhitungan berlanjut dengan node D ditandai menjadi node yang telah terjamah. Dari semua node tetangga belum terjamah yang terhubung langsung dengan node terjamah, node selanjutnya yang ditandai menjadi node terjamah adalah node E karena node E tidak terhubung dengan node yang lain secara otomatis node D akan terhubung langsung dengan node E, nilai $7 (4 + 3)$.



Gambar 6. Contoh Kasus Algoritma Dijkstra – Langkah 4

5. Node E menjadi node terjamah, dijkstra melakukan kalkulasi kembali, dan menemukan bahwa node G (node tujuan) telah tercapai lewat node E. Jadi Jalur terpendeknya adalah A-C-D-E-G, dan nilai bobot yang didapat adalah $8 (17+1)$. Bila node tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai.



Gambar 7. Contoh Kasus Algoritma Dijkstra – Langkah 5

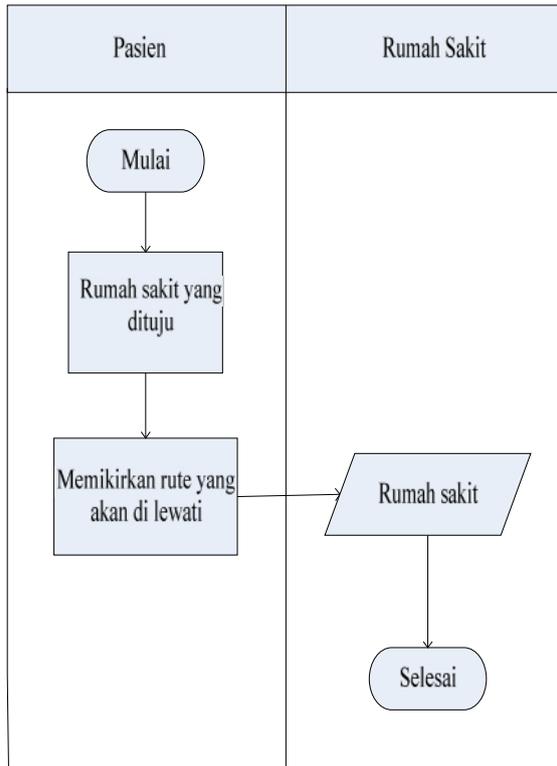
Dalam pembuatan suatu program aplikasi yang diinginkan harus dilakukan analisa permasalahan yang ada pada sebuah program aplikasi tersebut, agar program aplikasi tersebut dapat berjalan sesuai dengan keinginan pengguna dan program atau aplikasi yang dibuat bekerja secara efektif dan efisien. Dalam menganalisa permasalahan pada program atau aplikasi yang ada kegiatan yang akan penulis lakukan diantaranya melakukan analisa rute menuju rumah sakit yang ada di Kota Kisaran. Menganalisa rute bertujuan untuk mengetahui rute terpendek agar dalam membuat program aplikasi dapat di duat sesuai kebutuhan pengguna yaitu menunjukkan rute terdekat menuju rumah sakit yang ada di Kota Kisaran.

b. Analisa Sistem Yang Ada

Penghitungan rute terpendek memegang peranan penting dalam kehidupan sehari-hari karena harus dilakukan dalam waktu singkat dan pada saat itu juga agar segera dapat diketahui rute mana yang paling pendek untuk dilewati. Dengan melewati rute terpendek dapat membuat mobilitas sehari-hari menjadi lebih efisien. Untuk mencapai suatu tempat yang dituju terkadang seseorang tidak mengetahui terdapat jalur terdekat untuk mencapai tujuannya. Jika seseorang mengetahui jalur terdekat tersebut akan dapat mempersingkat waktu tempuh yang di milikinya.

Berikut adalah analisa sistem yang ada:

1. Pasien memulai dengan menentukan rumah sakit yang akan dituju.
2. Selanjutnya pasien akan memikirkan dan menentukan rute yang akan di lewati.
3. Dan sampailah pasien pada rumah sakit yang dituju.



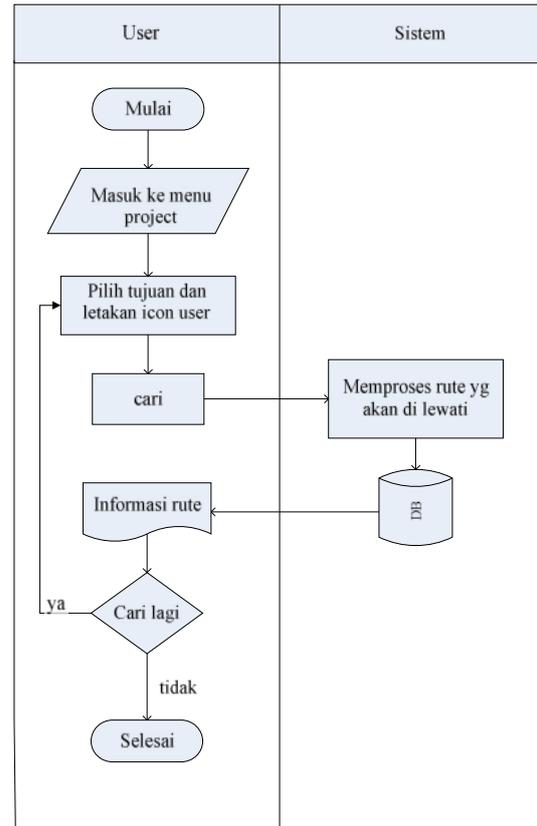
Gambar 8. Analisa Sistem yang Ada

c. Perancangan Sistem

Perancangan sistem dibuat untuk memperbaiki sistem yang sedang berjalan saat ini dan mempermudah dalam pembuatan program aplikasi yang akan dibuat. Dengan merancang sistem yang diusulkan akan terlihat perbedaan antara sistem yang lama dengan sistem yang baru, dimana sistem yang lama tidak begitu efektif dan efisien.

Berikut adalah analisa sistem yang ada:

1. User memulai dengan masuk ke menu *project*.
2. Pilih rumah sakit tujuan dan letakan *icon user* pada tampilan peta kota Kisaran.
3. Klik tombol cari untuk mendapatkan rute menuju rumah sakit yang telah di pilih.
4. Sistem akan memproses mengambil rute pada *data base*.
5. Selanjutnya rute menuju rumah sakit akan di tampilkan pada program.
6. Jika ingin mengganti rumah sakit tujuan, kembali pilih rumah sakit tujuan dan letakan *icon user*, kemudian klik tombol cari.
7. Jika tidak, ikuti rute yang tampil pada program.

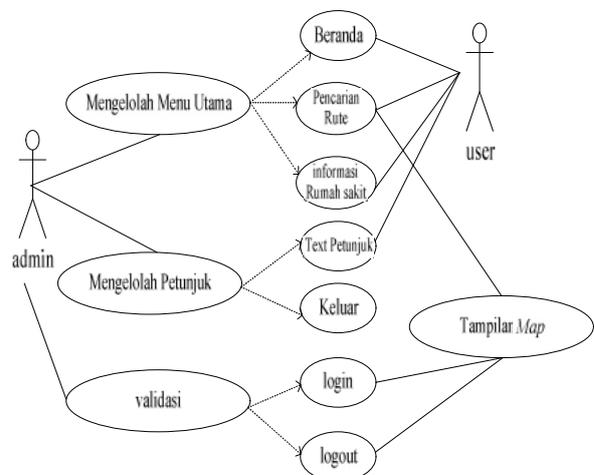


Gambar 9. Perancangan Sistem yang Diusulkan

d. Use case Diagram

Use case atau diagram *Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antar satu atau lebih actor dengan sistem informasi yang akan di buat.[2]-[5]

Berikut adalah diagram *Use case* dari sistem aplikasi penentuan rute terdekat.



Gambar 10. Use Case Diagram

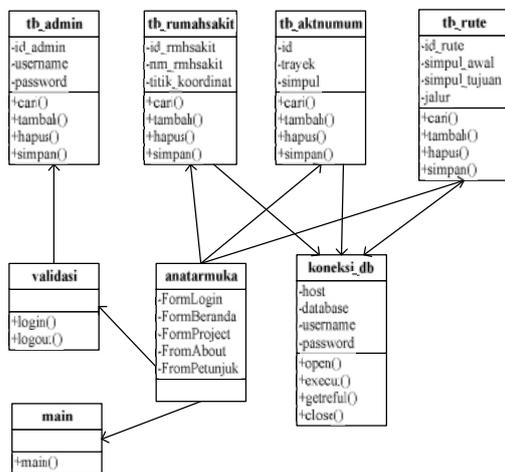
Berikut adalah deskripsi pendefinisian aktor pada Pengolahan sistem aplikasi:

Tabel 1 Defenisi Aktor

No	Aktor	Deskripsi
1	Admin	Orang yang dapat mengelola data dan memiliki hak ases sepenuhnya pada program aplikasi
2	User	Orang yang menggunakan sistem aplikasi yang telah dibuat.

e. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian dari kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah gambaran dari *Class Diagram*:



Gambar 11. Class Diagram

f. Hasil

Hasil Implementasi ini dilakukan agar pemakai sistem dapat mengerti tentang bagaimana cara bekerja dari sistem ini. Berikut adalah implementasi dari sistem penentuan rute rumah sakit terdekat:

1. Halaman Utama Website

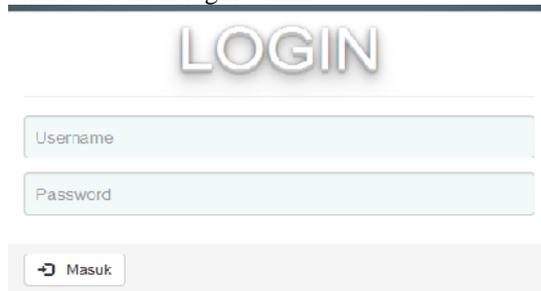
Halaman utama *interface* dari *website* Aplikasi Penentuan Rute Rumah Sakit Terdekat Menggunakan Algoritma *Dijkstra* di Kota Kisaran berisi tentang halaman selamat datang. Berikut adalah tampilan halaman utama *user*



Gambar 12. Halaman Utama

2. Halaman Login Admin

Halaman login admin adalah terdapat pada halaman yang berisi login admin untuk masuk ke halaman administrator agar dapat mengakses seluruh pengelolaan data-data yg dibutuhkan, berikut halaman login admin:



Gambar 13. Halaman Login Admin

3. Halaman Admin

Berisi menu pengolahan data admin dan pengolahan data rumah sakit.



Gambar 14. Halaman Admin

4. Halaman Data Admin

Halaman data admin berisi tentang input data admin, tambah, edit, dan hapus data admin.



Gambar 15. Halaman Data Admin



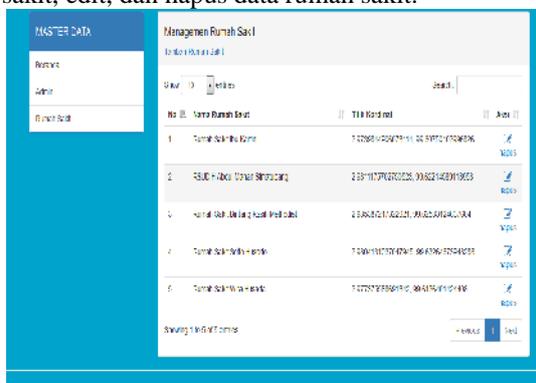
Gambar 16 Halaman Tambah Admin



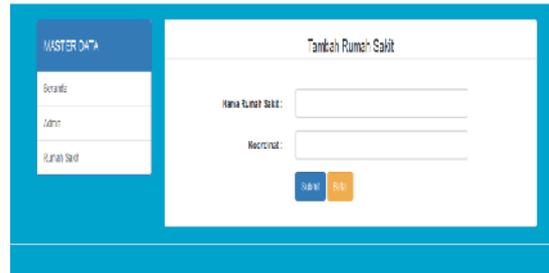
Gambar 17. Halaman Edit Admin

5. Halaman Data Rumah Sakit

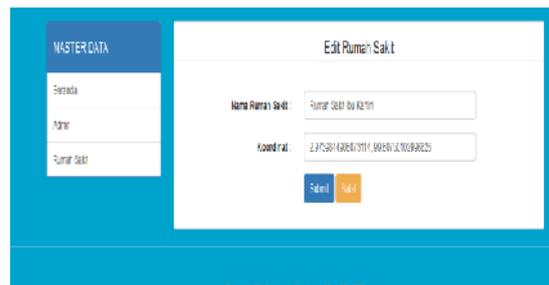
Pada halaman ini membahas tentang input data rumah sakit, dan terdapat tambah rumah sakit, edit, dan hapus data rumah sakit.



Gambar 18. Halaman Data Rumah Sakit



Gambar 19. Halaman Tambah Rumah Sakit



Gambar 20. Halaman Edit Rumah Sakit

g. Pengujian Sistem

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian adalah menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal yaitu mampu memproses kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri. Adapun kriteria pengujian tersebut dimaksudkan untuk mengetahui kebenaran perangkat lunak dan kesesuaian dengan spesifikasinya dinyatakan berhasil jika aplikasi mampu menjalankan fungsi-fungsi berdasarkan spesifikasi secara benar. Sebaliknya pengujian dinyatakan gagal jika terdapat fungsi dari perangkat lunak dalam spesifikasi yang tidak dapat dijalankan pada proses pengujian. Pengujian yang telah dilakukan merupakan proses yang terdapat dalam sistem Perancangan Aplikasi Penentuan Rute Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra di Kota Kisaran. Setelah dilakukan pengujian, maka berdasarkan hasil dari pengujian dapat diambil kesimpulan bahwa aplikasi yang telah dibuat dapat digunakan dengan baik.

V. KESIMPULAN

Dari penelitian yang telah dilakukan terdapat beberapa kesimpulan yang dapat di ambil, diantaranya:

1. Aplikasi Penentuan Rute Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra di Kota Kisaran di buat untuk membantu mobilitas sehari-hari menjadi lebih efektif dan efisien dengan cara menampilkan rute terdeka

untuk mencapai tujuan sehingga dapat mempersingkat waktu tempuh.

2. Metode *algoritma dijkstra* digunakan dalam menentukan rute terdekat. Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya
3. Sistem aplikasi yang dirancang juga memiliki pembatasan hak akses bagisetiap user dengan tujuan untuk meningkatkan *internal control* dan mencegah pihak-pihak yang tidak berwenang dalam mengakses data.

VI. REFERENSI

- [1] Abdulloh, Rohi(2015). “*Web programming is easy*”. Jakarta:KompasGramedia.
- [2] A.S, Rossa dan Salahuddin, M. (2013). “*Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*”. Bandung: Informatika.
- [3] Aswati, Safrian dan Kartika, widya (2014).“Rancang Bangun Sistem Informasi Data Karyawan Kantor Notaris/PPAT Mulia Ginting Suka”, JURTEKSI ROYAL Vol 1 No.1 pp.1.
- [4] Ghofur Wibowo, Abdul dan Purwo Wicaksono, Agung (2012). “Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Rumah Sakit di Purbalingga dengan Metod eAlgoritma Dijkstra”, JUITA Vol 2 No.1 pp.21.
- [5] Hutahaeen, Jeperson (2014). *Konsep Sistem Informasi*. Yogyakarta: Deepublish
- [6] Mujiatik, Hanik (2014). “Analisis dan Perancangan Sistem Informas Stok Obat pada Apotek Arjowinangun” ,Sentral Penelitian Engineering dan EdukasiVol 11 No.2 pp.24.
- [7] Peranginangi, Kasiman (2006). *Aplikasi Web dengan PHP dan Mysql*. Yogyakarta: Penerbit Andi Offset.