

PENENTUAN JARAK EFISIEN PENGANTARAN PASIEN OLEH AMBULANCE KE RSUD KARAWANG DENGAN ALGORITME DIJKSTRA

Masdui Kartasasmita¹, Arif Budimansyah Purba², Wawan Kusdiawan³

¹masdwi96@gmail.com, ²arifbudimansyahpurba@gmail.com,

³wawankusdiawan8@gmail.com

¹²³Teknik Informatika, STMIK Kharisma Karawang

Abstrak

Banyak jalan yang menghubungkan dari berbagai puskesmas yang ada di Karawang menuju ke RSUD Karawang dengan jarak tempuh yang berbeda-beda. Penelitian ini bertujuan untuk memberikan solusi kepada pihak ambulance untuk menentukan rute dalam pengantaran pasien ke RSUD Karawang dengan menggunakan algoritma Dijkstra. Algoritma ini digunakan dalam pencarian rute terpendek yang diharapkan dapat menjadi rute terefisien untuk mencapai tujuan dari lokasi yang diinginkan. Algoritma 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. Algoritme Dijkstra dapat diimplementasikan/digunakan sebagai alternatif dalam penentuan jarak efisien suatu daerah ke daerah yang lain dalam hal ini adalah penentuan jarak efisien pengantaran pasien oleh ambulance ke RSUD Karawang.

Kata Kunci: Efisien, Algoritma Dijkstra, Rute Terpendek, Graf Berarah, Ambulance.

1. Pendahuluan

Perkembangan teknologi informasi memberikan pengaruh diberbagai bidang, mulai dari dunia pendidikan, kesehatan, perdagangan, pemerintahan, dan sebagainya. Dimana secara keseluruhan digunakan untuk meningkatkan efisiensi, efektifitas, dan keakuratan untuk mendukung aktivitas-aktivitas bisnis [1].

Ambulance adalah unit transportasi medis yang didesain khusus yang berbeda dengan moda transportasi lainnya. Ambulance gawat darurat di design agar dapat menangani pasien gawat darurat, memberikan pertolongan pertama dan melakukan perawatan insntif selama perjalanan menuju rumah sakit rujukan. Ambulance gawat darurat juga harus memenuhi aspek higiene dan ergonomic. Komponen Ambulance/EMS (*Emergency Medical Services*) merupakan layanan perawatan yang bersifat *out-of-hospital* yang disediakan oleh rumah sakit serta menyediakan transportasi bagi pasien yang memerlukan perawatan medis ke rumah sakit [2].

Seseorang dengan permintaan darurat dalam medis harus segera memerlukan pertolongan yang efektif dengan adanya kecepatan tindakan dari tim medis begitu juga dengan perlunya kecepatan ambulance menuju lokasi permintaan medis dan membawanya ke rumah sakit dengan efisiensi waktu yang singkat bisa menolong banyak orang yang sedang membutuhkan bantuan medis secara darurat. Dewasa ini kebutuhan akan ambulance sangat meningkat karena masyarakat sudah banyak yang mempunyai alat komunikasi untuk memanggil ambulance, dengan kebutuhan yang sangat banyak dan ketepatan waktu pihak ambulance untuk menjemput pasien dari jarak tempuh, waktu kondisi jalanan dan onkos biaya menjadi perhatian utama dalam pengaplikasiannya. Pada penelitian sebelumnya tentang untuk minimasi ongkos distribusi, masalah yang berkaitan dengan pendistribusian sampah diantaranya membuat keputusan-keputusan mengenai rute pengambilan sampah. Pemilihan rute kendaraan akan menentukan total jarak perjalanan armada. Sehingga rute yang optimal adalah rute yang memenuhi karakteristik permasalahan penentuan rute pengambilan sampah [3].

Dalam dunia industri, salah satu informasi yang dibutuhkan adalah informasi rute dalam distribusi barang. Permasalahan distribusi merupakan salah satu faktor yang penting dalam mempengaruhi peningkatan pendapatan. Berdasarkan penelitian para ahli, menyatakan bahwa biaya distribusi rata-rata sebesar 16% dari harga jual barang yang dihasilkan. Ini berarti bahwa perlu adanya metode yang digunakan untuk mengurangi biaya distribusi barang [4].

Android sebagai sistem operasi pada perangkat smartphone yang banyak digunakan memiliki berbagai macam layanan yang memudahkan pengguna. Salah satu layanan yang sering digunakan adalah layanan Global Positioning System (GPS) untuk menentukan lokasi dan arah yang dituju. Maka dari hal ini lah penulis mengajukan penelitian untuk mengembangkan suatu aplikasi berbasis

Android yang mampu untuk mencari jalur terpendek menuju koordinat yang telah ditentukan dengan mengimplementasikan metode A-star dan dengan menggunakan metode Software Development Life Cycle (SDLC) Waterfall [5].

2. Metode

2.1 Alat Penelitian

Alat yang digunakan dalam penelitian ini meliputi perangkat keras dan perangkat lunak.

Tabel 1. Rincian Project Planning Phase

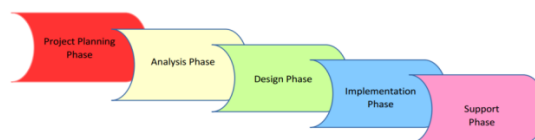
No.	Perangkat Keras	Spesifikasi
1.	<i>Laptop</i> (Sistem Komputer)	Suatu alat yang terdiri atas satu perangkat yang memiliki papan tombol (keyboard), layar, tampilan, dan <i>microprocessor</i> dengan spesifikasi <i>processor</i> CPU minimal 1.67 Ghz.
2.	Printer	Alat yang digunakan untuk mencetak tampilan monitor ke kertas dan informasi yang dicetak dapat berupa teks dan gambar.

Tabel 2. Rincian Tools Yang Digunakan

No.	Perangkat Lunak	Fungsi
1.	Ubuntu 14.04 LTS 32bit	Sistem Operasi Linux
2.	<i>Eclipse</i>	Tools yang digunakan untuk membuat program android
3.	<i>OpenOffice Writer</i>	Teks <i>editor</i> yang digunakan untuk merepresentasikan tentang penelitian.
4.	<i>OpenOffice Impress</i>	<i>Software</i> yang digunakan untuk mempresentasikan tentang penelitian.
5.	<i>Java</i>	Bahasa pemrograman yang didesain untuk mengembangkan perangkat mobile.
6.	<i>Android Virtual</i>	<i>Software</i> yang digunakan untuk menguji tampilan android mobile.
7.	<i>SDK</i>	Paket bundel yang di sediakan android developer
8.	<i>Dia Diagram</i>	<i>Software</i> yang digunakan untuk pemodelan sistem.

2.2 Metode SDLC Waterfall

Metode penelitian yang digunakan dalam membangun sistem adalah metode SDLC Waterfall yaitu suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematis dan sekuensial yang mulai pada Project planning phase, Analysis phase, Design phase, Implementation phase, dan Support Phase yang ditunjukkan pada Gambar 1.



Gambar 1. The Waterfall Approach to The SDLC

▪ Project Planning Phase

Tahap perencanaan adalah proses dasar memahami mengapa sistem manajemen pengelolaan data pemeriksaan sampel dan inventaris harus dibangun dan menentukan bagaimana membangun sistem tersebut.

Tabel 3. Tabel Deskripsi Project Planing Phase

No. Tahapan	Deskripsi
1. Identifikasi Masalah	Banyaknya warga yang membutuhkan penanganan cepat tenaga ahli medis untuk keadaan darurat
2. Pengumpulan data	Teknik melakukan pengumpulan data meliputi Wawancara dan Studi Literatur

3. Analisis Teori	Menganalisis sistem dengan menentukan graf terdekat dalam suatu jalur metode pengembangan sistem SDLC <i>Waterfall</i> .
4. Pembuatan Jadwal	Membuat Jadwal perencanaan pelaksanaan penelitian
5. Mencari Solusi	Bagaimana membangun aplikasi pencarian rute terdekat untuk ambulance dengan metode Dijkstra

▪ **Analysis Phase**

- 1) Analisis sistem berjalan dengan menggunakan grafis dan narasi.
- 2) Analisis sistem ajuan dengan menggunakan Object Oriented Analysis (OOA).

Tahapan dari analisis tersebut yaitu :

- 1) System Activities (Actor Description and Use Case Description, Use Case Diagram, Scenario Use Case).
- 2) Class Diagram (Class Definition, Class Relation).
- 3) Object Interaction (Sequence Diagram).
- 4) Object Behavior (Activity Diagram).

▪ **Design Pashe**

Tahap desain akan menerjemahkansyarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Dalam tahapan ini desain yang dilakukan oleh peneliti adalah pendesainan berbasis Object Oriented Design (OOD) terdiri dari :

- 1) Desain Basis data.
 - a. Rancangan tabel.
 - b. Normalisasi.
 - c. Relasi antar tabel.
- 2) Desain Proses.

Rancangan logika pemrosesan data yang akan digunakan akan disajikan menggunakan flow chart.
- 3) Desain Antarmuka.

Rancangan tampilan masukan dan keluaran yang akan dioperasikan oleh user.

▪ **Implementation Phase**

Tahapan implementasi merupakan tahap pembuatan program termasuk penulisan kode program, pengetesan program secara terus menerus untuk mencari kesalahan yang terjadi pada program yang dibuat, kemudian meletakkan sistem untuk di operasikan.

Penulisan kode program dilakukan dengan teknik Object Oriented Programming (OOP) secara modular sehingga akan mempermudah dalam memodifikasi kode yang sudah ada. Objek yang baru dapat dibuat tanpa mengubah kode yang sudah ada. Dapat menyederhanakan kompleksitas dengan memungkinkan programmer untuk mendefinisikan sebuah sistem besar dan kompleks menggunakan set yang lebih kecil dari objek yang saling terkait.

Pada Tahapan ini dilakukan beberapa tahapan, antara lain :

- 1) Instalasi Sistem
- 2) Pengujian Terhadap Sistem
Pengujian sistem dilakukan dengan dua tahap yaitu pengujian white box dan black box.

3. Hasil dan Pembahasan

3.1 Project Planning Phase

Tahapan *Project Planning Phase* menghasilkan rinciansetiap aktivitas yang dilakukan.

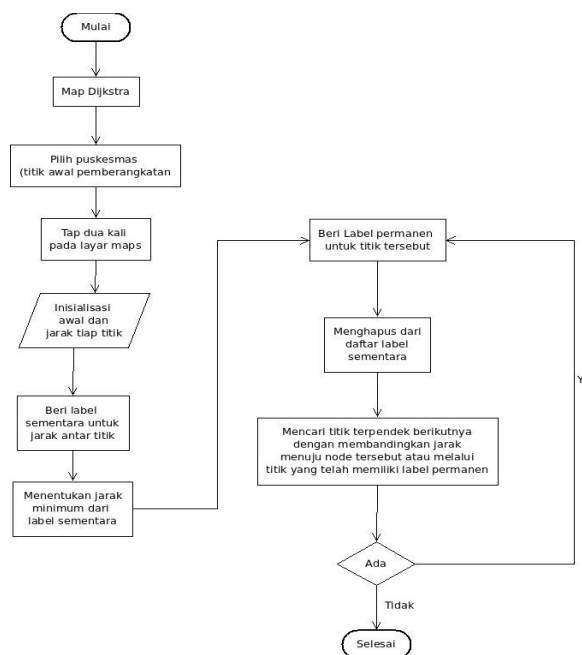
Tabel 4. Tabel Hasil Tahap Perencanaan

No	Tahapan	Hasil
1.	Identifikasi Masalah	Aplikasi penentuan jarak efisien pengantar pasien oleh ambulance dengan metode dijkstra berbasis android menentukan rute yang efisien untuk supir ambulance mengantarkan pasien.
2.	Pengumpulan data	Hasil pengamatan dilapangan wawancara terhadap pengendara dan mengambil data yang terdahulu tentang kemacetan di dinas perhubungan karawang.



- | | | |
|----|--------------------------|--|
| 3. | Analisis Teori | Konsep pencarian rute terefisien menggunakan algoritma dijkstra dan pengembangan aplikasi dengan menggunakan metode <i>SDLC waterfall</i> . |
| 4. | Pembuatan Jadwal | Penelitian dilakukan di RSUD karawang dengan mewawancarai supir ambulance, mengamati jalan yang akan di lalui ambulance dari bulan Februari sampai Juli 2017. |
| 5. | Mencari solusi | Dapat mencari rute yang efisien untuk pengantaran pasien dari puskesmas menuju RSUD karawang. |
| 6. | Mendefinisikan kebutuhan | Perangkat Lunak : <i>Linux ubuntu, eclipse, ADT, API, sqllite, LibreOffice Writer, LibreOffice Impress, Diagram</i> . Perangkat Keras (<i>Laptop dan Printer</i>). |

3.2 Analysis Phase



Gambar 2. Flowchat Proses Algoritma Dijkstra

1. STUDI KASUS : Ambulance puskesmas Klari mengantarkan pasien ke RSUD Karawang Langkah pertama menentukan jalur-jalur terpendek dalam rute

Keterangan inisialisasi

N adalah titik awal

V adalah vertex/node

Jarak awal yang belum terkena kemacetan

$N \rightarrow 2 = 2198.5596m$

$2 \rightarrow 3 = 2301.0205m$

$2 \rightarrow 4 = 2199.4595m$

$4 \rightarrow 3 = 1910.1381m$

$3 \rightarrow 5 = 2772.4222m$

$3 \rightarrow 6 = 4055.5867m$

$5 \rightarrow 6 = 310.0448m$

Node 2 ke 3 terjadi kemacetan 2 menit

Node 2 ke 4 terjadi kemacetan 1 menit

Node 4 ke 3 terjadi kemacetan 2 menit

Node 4 ke 5 terjadi kemacetan 2 menit

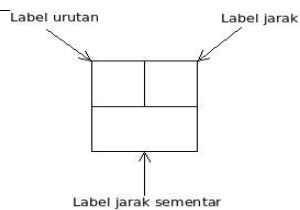
Node 3 ke 6 terjadi kemacetan 1 menit

rata-rata kecepatan ambulance 60 km/jam.

$N \rightarrow 2 = 2198.5596 \text{ m}$
 karna dari node N hanya ada satu jalur yaitu node 2
 $2 \rightarrow 3 = 2301.0205\text{m} + 2000\text{m} = 4301.0205\text{m}$
 $2 \rightarrow 4 = 2199.4595\text{m} + 1000\text{m} = 3199.4595\text{m}$
 Node terpendek dari 2 adalah 4 maka 4 yang dipilih
 $4 \rightarrow 3 = 1910.1381\text{m} + 2000\text{m} = 3910.1381\text{m}$
 $4 \rightarrow 5 = 3895.7723\text{m} + 2000\text{m} = 5895.7723\text{m}$
 Node terpendek dari 4 adalah 3 maka 3 yang dipilih
 $3 \rightarrow 5 = 2772.4222\text{m}$
 $3 \rightarrow 6 = 4055.5867\text{m} + 1000\text{m} = 5055.5867\text{m}$
 Node terpendek dari 3 adalah 5 maka 5 yang dipilih
 $5 \rightarrow 6 = 310.0448\text{m}$

Tabel 5. Tabel Hasil Perhitungan Jarak Klari

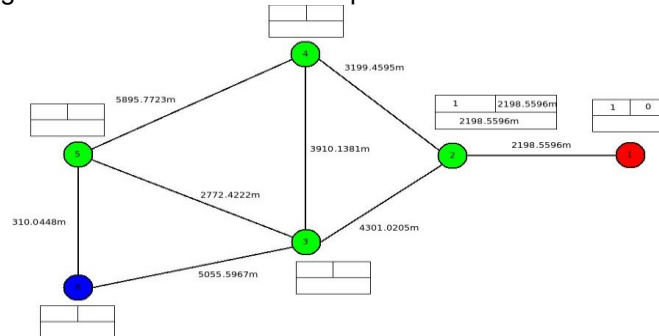
$N \rightarrow 2$	$2 \rightarrow 3$	$2 \rightarrow 4$	$4 \rightarrow 3$	$4 \rightarrow 5$	$3 \rightarrow 5$	$3 \rightarrow 6$	$5 \rightarrow 6$	Jumlah
2198.55	2301.0205+20	2199.4595+10	1910.138	3895.772	2772.42	4055.586	310.04	27643.00
96	00=	00=	1+	3+	22	7+	48	37
	4301.0205	3199.4595	2000=	2000=		1000=		
			3910.138	5895.772		5055.586		
			1	3		7		



Gambar 3. Tabel Simulasi

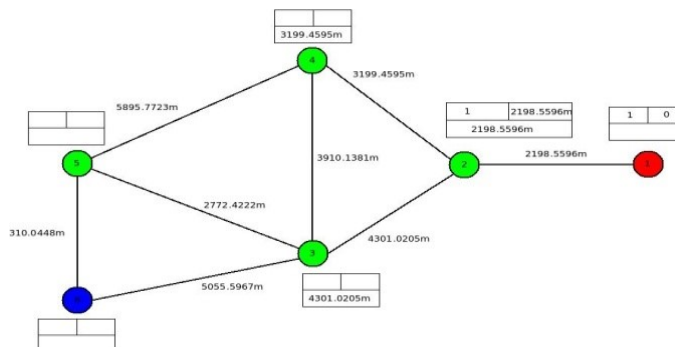
Cara perhitungan Algoritma Dijkstra

Langkah pertama mengisi label urutan Node/Vertex pada titik awal



Gambar 4. Proses Perhitungan Algoritma Dijkstra Klari1

Kemudian mengisi label jarak sementara titik yang dapat dihubungi langsung dari titik 2 yakni titik 3 dan 4.

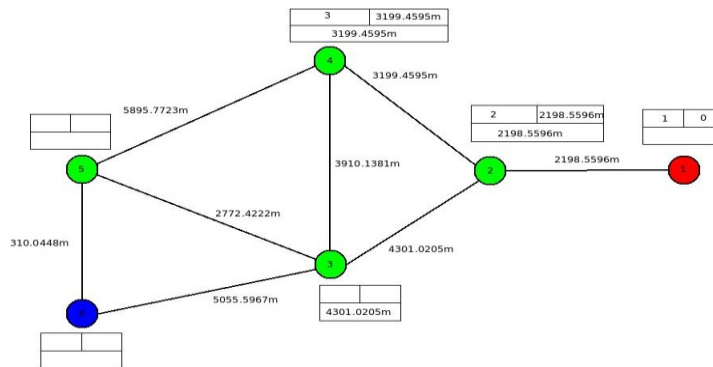


Gambar 5. Proses Perhitungan Algoritma Dijkstra Klari2

Maka yang terpilih adalah titik 4 karena memiliki label jarak sementara terkecil, dan mengisi nilai label

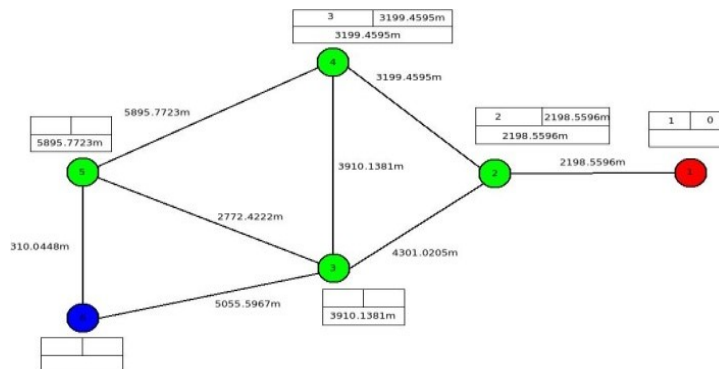


jarak-nya sama dengan label jarak sementara serta memberikan label urutan-nya.



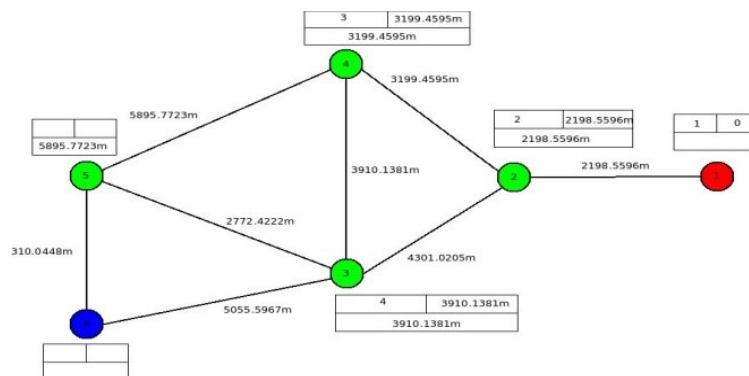
Gambar 6. Proses Perhitungan Algoritma Dijkstra Klari3

Selanjutnya mengisi label jarak sementara titik yang belum memiliki label jarak dan dapat dihubungi langsung dari titik 4 yakni titik 3 dan 5. Label jarak sementara titik 3 diisi dengan jarak dari titik 4 ke titik 3.



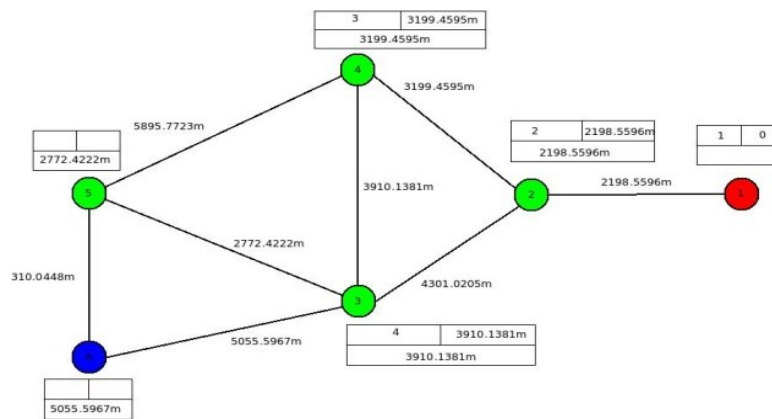
Gambar 7. Proses Perhitungan Algoritma Dijkstra Klari4

Maka yang terpilih adalah titik 3 karena memiliki label jarak sementara terkecil, dan mengisi nilai label jarak-nya sama dengan label jarak sementara serta memberikan label urutan-nya.



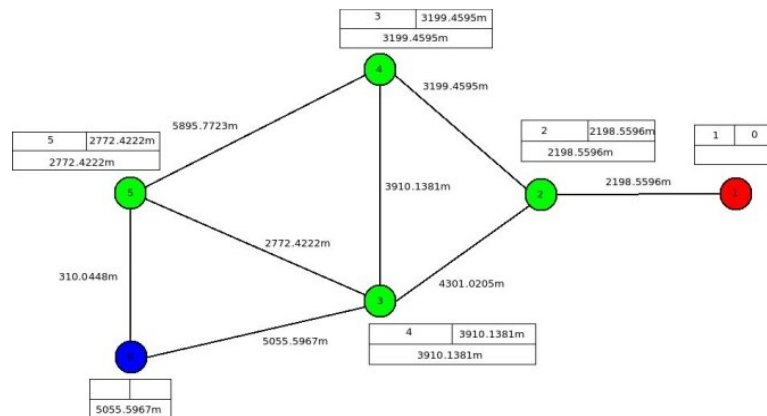
Gambar 8. Proses Perhitungan Algoritma Dijkstra Klari5

Selanjutnya mengisi label jarak sementara titik yang belum memiliki label jarak dan dapat dihubungi langsung dari titik 3 yakni titik 5 dan 6. Label jarak sementara titik 5 diisi dengan jarak dari titik 3 ke titik 5.



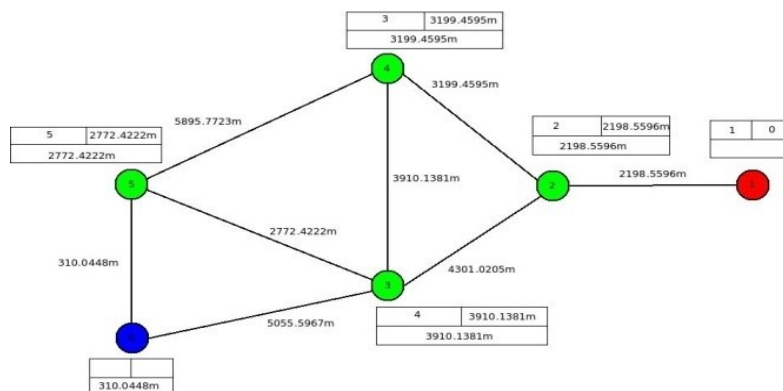
Gambar 9. Proses Perhitungan Algoritma Dijkstra Klari6

Maka yang terpilih adalah titik 5 karena memiliki label jarak sementara terkecil, dan mengisi nilai label jarak-nya sama dengan label jarak sementara serta memberikan label urutan-nya.



Gambar 10. Proses Perhitungan Algoritma Dijkstra Klari7

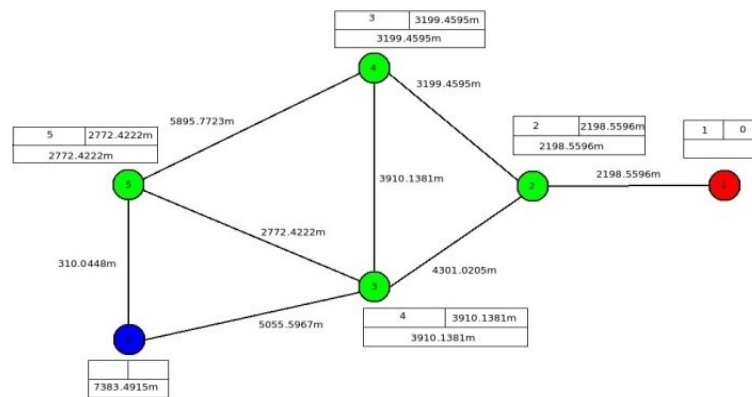
Selanjutnya mengisi label jarak sementara titik yang belum memiliki label jarak dan dapat dihubungi langsung dari titik 5 yakni titik 6. Label jarak sementara titik 6 diisi dengan jarak dari titik 5 ke titik 6.



Gambar 11. Proses Perhitungan Algoritma Dijkstra Klari8

- Kemudian untuk tujuan akhir kita gabungkan dari semua titik
- 2→4→5→6 (3199.4595 + 5895.7723 + 310.0448 = 9405.2766)
 - 2→4→3→5→6 (3199.4595 + 3910.1381 + 2772.4222 + 310.0448 = 10192.0646)
 - 2→4→3→6 (3199.4595 + 3910.1381 + 5055.5867 = 12165.1843)
 - 2→3→5→6 (4301.0205 + 2772.4222 + 310.0488 = 7383.4915)
 - 2→3→6 (4301.0205 + 5055.5867 = 9356.6072)





Gambar 12. Proses Perhitungan Algoritma Dijkstra Klari9

maka jarak titik akhir 6 adalah 7383.4915 karna jarak terkecil/terpendek.

Lalu menghitung terbalik untuk menentukan rute terefisien di mulai dari perhitungan titik akhir sampai titik awal.

$6 \rightarrow 5 \rightarrow 4 \rightarrow 2$ adalah $(7383.4915 - 310.0448 - 5895.7723 - 3199.4595 = -2021.7851)$

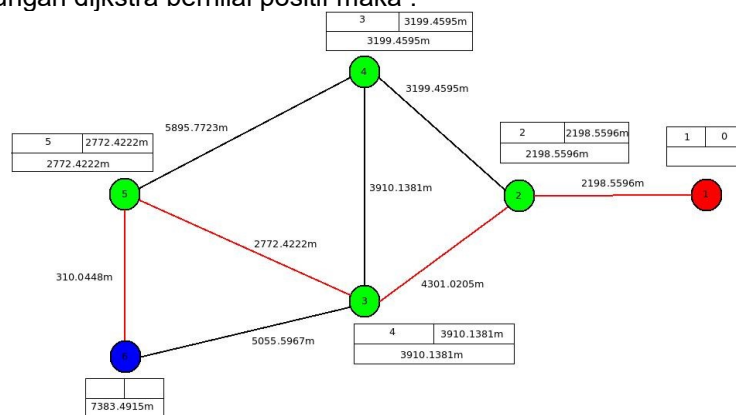
$6 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 2$ adalah $(7383.4915 - 310.0448 - 2772.4222 - 3910.1381 - 3199.4595 = -2808.5731)$

$6 \rightarrow 3 \rightarrow 4 \rightarrow 2$ adalah $(7383.4915 - 5055.5867 - 3910.1381 - 3199.4595 = -4781.6928)$

$6 \rightarrow 5 \rightarrow 3 \rightarrow 2$ adalah $(7383.4915 - 310.0448 - 2772.4222 - 4301.0205 = 0)$

$6 \rightarrow 3 \rightarrow 2$ adalah $(7383.4915 - 5055.5867 - 4301.0205 = -1973.1157)$

karna dalam perhitungan dijkstra bernilai positif maka :



Gambar 12. Proses Perhitungan Algoritma Dijkstra Klari10

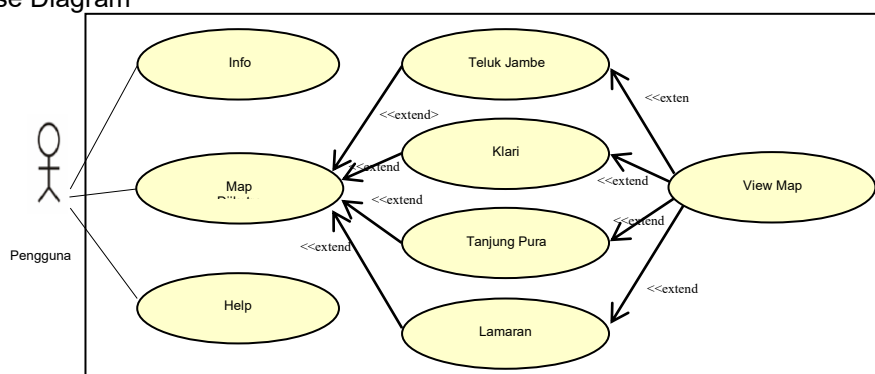
Rute terefisien dari puskesmas klari(1) menuju RSUD Karawang (5) adalah melalui titik $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6$.

1) System Activities

a. Actor Description

Aktor pada aplikasi ini terdiri dari satu aktor yang disebut pengguna.

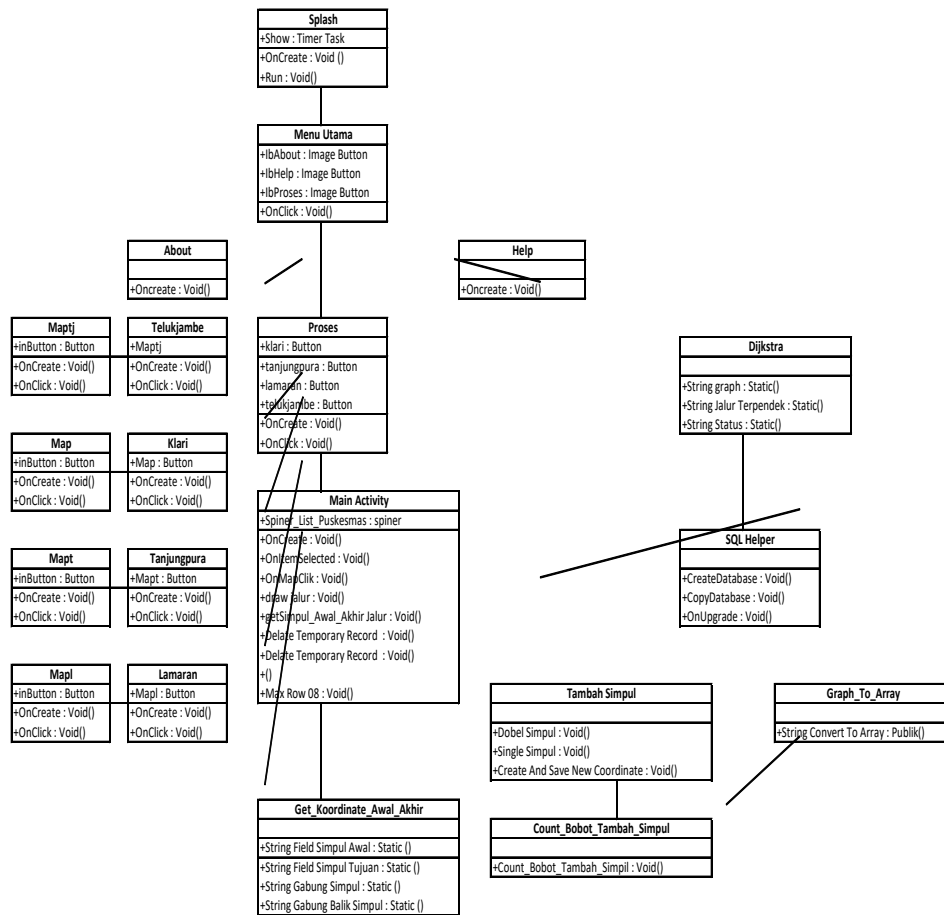
b. Use Case Diagram



Gambar 13. Use Case

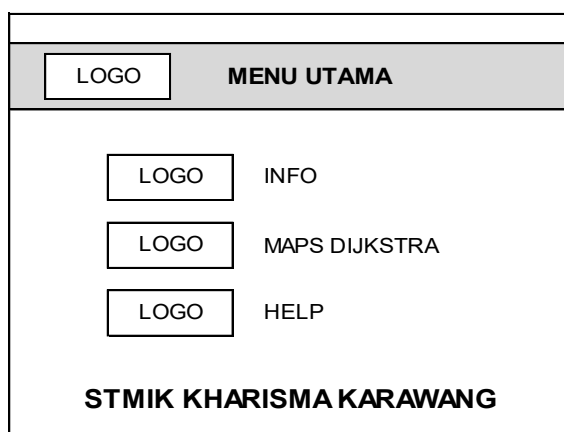


2) Class Diagram



3.3 Desain Phase

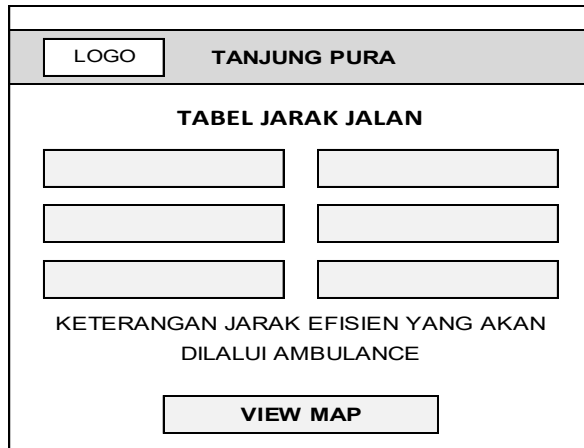
1. Desain Halaman Utama



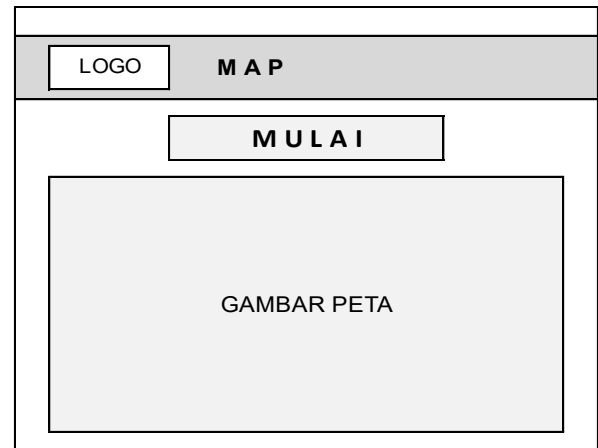
2. Desain Halaman Maps Dijkstra



3. Desain Halaman Tanjung Pura



4. Desain Halaman View Map



3.4 Implementation Phase

1. Antarmuka Halaman Utama



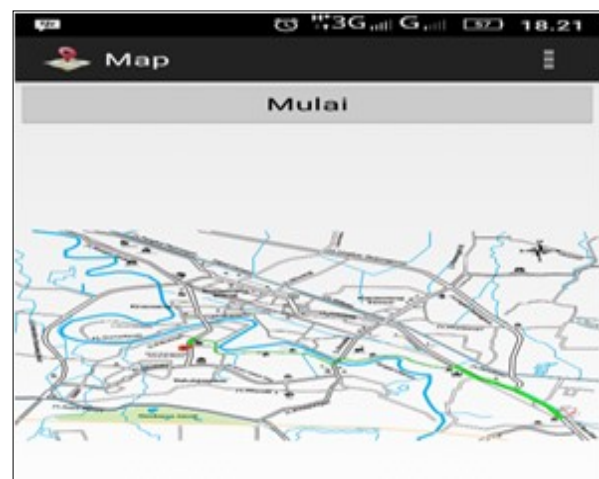
2. Antarmuka Maps Dijkstra



3. Antarmuka Tanjung Pura



4. Antarmuka View Maps



4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan dari hasil analisis, desain, implementasi dan pengujian dari Aplikasi, maka penulis mengambil kesimpulan diantaranya :

- 1) Jarak efisien yang didapatkan setelah implementasi dan pengujian aplikasi adalah sebagai berikut :
Jarak efisien pengantaran pasien oleh ambulan menuju RSUD Karawang adalah
 - a. Dari puskesmas Klari didapatkan rute (jalur graph) 1→2→3→5→6 dengan total jarak 9582.0511 M.
 - b. Dari puskesmas Telukjambe didapatkan rute (jalur graph) 1→3→4→6 dengan total jarak 3382.7962 M.
 - c. Dari puskesmas Tanjungpura didapatkan rute (jalur graph) 1→2→3→5→7 dengan total jarak 4231.8159 M.
 - d. Dari puskesmas Lamaran didapatkan rute (jalur graph) 1→2→4.
- 2) Algoritme Dijkstra dapat diimplementasikan/digunakan sebagai alternatif dalam penentuan jarak efisien suatu daerah kedaerah yang lain dalam hal ini adalah penentuan jarak efisien pengantaran pasien oleh ambulan ke RSUD Karawang.
- 3) Algoritme Dijkstra dapat divisualisasikan menggunakan bantuan aplikasi berbasis android dengan menambahkan peta/map sebagai ilustrasi dan menggambarkan rute (jalur graph) terefisien.

4.2 Saran

Sistem aplikasi yang dibuat belum bisa mendeteksi kemacetan secara mendadak sehingga perlu dikembangkan lagi menjadi sistem yang dapat mendeteksi kemacetan tersebut tanpa bantuan data yang sudah ada (data hasil dari perhitungan dinas perhubungan kabupaten Karawang).

Daftar Pustaka

- [1] Aris Puji Widodo (2007), Simulasi Lintasan Jalur Terpendek Algoritma Dijkstra Berbasis Extensible Markup Language(XML) Brassard and Bratley (1988), Dijkstra's Algorithm [pp. 87-92].
- [2] Amponsah, S.K, Amoako, G., Darkwah, K.F. and Agyeman, E. (2010). *Location of Ambulance Emergency Medical Service in The Kumasi Metropolis*, Ghana, *African Journal of Mathematics and Computer Science Researc*, Vol.41(1), pp.1826.
- [3] Andayani, Perwitasari (2014), Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra. ISBN: 979-26-0276-3.
- [4] Isnaeni Setiyadi, Teguh Bharata Adji, Noor Akhmad Setiawan, (2015) Optimalisasi algoritma Dijkstra Dalam Menghadapi Perbedaan Bobot Jalur pada Waktu yang Berbeda, ISSN : 2302-3805.
- [5] Satzinger, John W., Jackson, Robert B., Burd, Stephen D. 2010. *System Analysis and Design in a Changing World, Fourth Edition, Thomson Course Technology, Canada*.
- [6] Ahuja, Ravinda K., Thomas L. Magnanti, James B. Orlin. (1993), *Netw Flows*, Prentice Hall.
- [7] Kartika Gunandi, Yulia (2002), Perencanaan Rute Perjalanan di Jawa Timur dengan Dukungan GIS Menggunakan Metode Dijkstra's.
- [8] Muh. Yamin, Moh. Bandrigo Talai (2015), Aplikasi Pencarian Jalur Terpendek pada Rumah Sakit Umum Bahteramas Menggunakan Algoritma A* (A-STAR).
- [9] Purwati, Firnawati, & Willy (2013), Penerapan Algoritme A*(A STAR) dalam Optimasi Penentuan Haltetransmusi Di Palembang Berbasis Android.
- [10] Nazruddin Safaat (2011). *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Penerbit Informatika, Bandung.