

**PERANCANGAN APLIKASI PENCARIAN RUTE UNTUK MOBIL PEMADAM
KEBAKARAN BERBASIS ANDROID DENGAN MENGGUNAKAN ALOGRITMA
ANT COLONY OPTIMIZATION (Studi Kasus : Kota Bandung)**

***DESIGN AND IMPLEMENTATION OF APPLICATION ROUTE SEARCH ANDROID
BASED FOR FIRE DEPARTMENT USING ANT COLONY OPTIMIZATION
ALGORITHM (Case Study : Bandung)***

Izzat Arramsyah¹ , Budhi Irawani² , Casi Setianingsih³

Prodi S1 Sistem Komputer, Fakultas Teknik, Universitas Telkom

¹arramsyah@gmail.com, ² budhiirawan@telkomuniversity.ac.id, ³setiacasie@telkomuniversity.ac.id

Abstrak

Dengan status kota metropolitan, Bandung tentu memiliki banyak permasalahan yang dihadapi. Salah satu permasalahan yang sering terjadi adalah kebakaran di daerah padat penduduk. Ketika kebakaran terjadi pihak pemadam kebakaran harus cepat menuju lokasi kebakaran. Namun permasalahannya terkadang tidak diketahui situasi jalan yang akan dilewati. Dengan permasalahan yang ada, maka diperlukan sebuah solusi bagi pemadam kebakaran yaitu dengan sebuah aplikasi android pencarian rute optimal untuk mencapai lokasi.

Dalam penentuan rute optimal pada aplikasi digunakan metode *simple additive weighting* (SAW) untuk melakukan perhitungan banyak kriteria seperti jarak, lokasi kemacetan atau daerah padat dan sumber air yang bisa dimanfaatkan untuk memadamkan api dan algoritma *ant colony optimization* sebagai pencarian rute. Dengan aplikasi ini petugas pemadam kebakaran akan dimudahkan dalam memilih rute optimal sehingga dapat meminimalisir terjadinya kejadian yang tidak diinginkan.

Dari hasil pengujian yang telah dilakukan, algoritma *ant colony optimization* dapat diimplementasikan untuk melakukan pencarian rute optimal pada sistem operasi android. Pemadam kebakaran selain mengetahui rute optimal juga dapat mengetahui informasi kemacetan serta sumber air yang bisa digunakan.

Kata Kunci : Android, Ant Colony Optimization, Kota Bandung.

Abstract

As a metropolitan city, Bandung certainly has many problems. The problem that often occurs in a densely populated areas like in Bandung is a fire. One of the causes is electrical short-circuited, natural disasters, human negligence and many other causes.

When a fire occurs, the firefighters had quickly to get to location of fire. However sometimes the firefighter don't know the situation of street will be passed. It could be the road is jammed or others obstacle. This should be avoided by firefighters and firefighter have to find other freeway road to get to fire location in right time.

With that problems, needed a solution that is by an application to find the optimal route to get to the fire location. To find the optimal route for firefighters, many things are taken to be calculated. One of them is distance to location, choose the right route to avoid congestion and find water sources such as hydrant. The application is made on android operating system which is the most widely used.

From the result of the testing that has been done, ant colony optimization can be implemented to find optimal route on the android operation system. The system also use simple additive weighting method that used to calculate accumulation from the rate processing from every alternatives based on distance, congestion and water sources criteria to get the weight of a road.

Keyword : Android, Ant Colony Optimization, Bandung

1. Pendahuluan

1.1 Latar Belakang

Kebakaran merupakan salah satu masalah yang sering terjadi di kota besar dan padat penduduk seperti di kota Bandung. Kebakaran ini dapat terjadi dimana saja dan kapan saja tanpa kenal waktu. Kebakaran ini dapat terjadi dimana saja dan kapan saja tanpa kenal waktu. Banyak hal-hal yang menjadi penyebab terjadinya kebakaran ini misalnya, faktor kelalaian manusia, faktor teknis seperti hubungan arus pendek atau peningkatan suhu panas atau adanya api terbuka

Saat terjadi kebakaran, petugas pemadam harus cepat mencapai lokasi guna meminimalisir resiko yang akan ditimbulkan. Selama ini, petugas pemadam kebakaran hanya menggunakan bantuan GPS dan mengandalkan perkiraan mereka untuk menentukan jalan menuju lokasi. Pemadam kebakaran harus mempertimbangkan sendiri jalan mana yang harus dilewati. Namun cara ini kurang efektif. Bisa saja jalan yang dilalui adalah jalan yang sering terjadi kemacetan atau daerah padat yang justru akan menghambat

Untuk mengatasi masalah tersebut maka diperlukan sebuah aplikasi yang mampu mempermudah petugas pemadam kebakaran untuk mencapai lokasi. aplikasi yang dibangun merupakan aplikasi berbasis android berupa peta digital dengan fungsi utama melakukan pencarian rute optimal dan memberikan informasi jalan di kota Bandung dan sekitarnya serta sumber air yang dapat digunakan untuk memadamkan api.

Dengan adanya aplikasi ini, diharapkan akan mempermudah petugas pemadam kebakaran untuk menentukan jalan yang akan dilalui ketika terjadi kebakaran. Dengan memanfaatkan perangkat mobil berbasis *Android*, petugas akan dengan mudah mendapatkan informasi mengenai jalur yang akan dilalui

1.2 Rumusan Masalah

Adapun perumusan masalah yang dibahas dalam Tugas Akhir ini adalah :

1. Bagaimana aplikasi dapat mampu memberikan petunjuk jalan menuju lokasi tujuan dan memberikan informasi jalan di kota Bandung serta memberikan informasi lokasi sumber air seperti hydrant atau sungai.
2. Bagaimana merancang dan mengimplementasikan metode *Simple Additive Weighting* dalam melakukan perhitungan kriteria jarak, kemacetan dan posisi hydrant.
3. Bagaimana mengimplementasikan algoritma *Ant Colony Optimization* untuk menentukan rute optimal.

1.3 Tujuan

Adapun Tujuan dari Tugas Akhir ini adalah :

1. Membuat aplikasi pencarian rute optimal untuk pemadam kebakaran berbasis android.
2. Mengimplementasikan algoritma *Ant Colony Optimization* dalam penentuan rute optimal untuk pemadam kebakaran berbasis *smartphone android*.

1.4 Batasan Masalah

Adapun batasan masalah dalam sistem yang akan dibuat adalah sebagai berikut :

1. Aplikasi yang dibangun adalah aplikasi *mobile* berbasis android dengan minimum spesifikasi sistem operasi perangkat mobile Android 4.1 Jelly Bean.
2. Aplikasi ini tidak terhubung dengan internet (Offline)
3. Aplikasi ini memiliki fitur utama menampilkan peta dan informasi jalan kota Bandung, melakukan pencarian rute serta memberikan informasi posisi hydrant.
4. Metode yang digunakan untuk menentukan bobot jalan adalah metode *Simple Additive Weighting*.
5. Algoritma yang digunakan dalam penentuan rute optimal menggunakan algoritma *Ant Colony System* yang merupakan bagian dari algoritma *Ant Colony Optimization*.
6. Peta yang digunakan adalah peta *OpenStreetMap*.
7. Pemetaan jalan menggunakan jalan utama yang bisa dilalui oleh mobil pemadam kebakaran.
8. Pemetaan jalur dilakukan di kota Bandung dan terbatas pada daerah-daerah penunjang di sekitarnya.
9. Lokasi kemacetan menggunakan data statistik dari dinas perhubungan.
10. Aplikasi tidak bisa melakukan pembaharuan ketika ada nama jalan yang berganti atau jalur yang ditutup karena kondisi tertentu saat itu.
11. Data kemacetan yang digunakan pada aplikasi didapatkan dari hasil survey dan penelitian yang dilakukan oleh Dinas Perhubungan Kota Bandung.

2. Dasar Teori

2.1 Openstreetmap

OpenStreetMap merupakan sebuah peta dunia digital gratis yang dapat diubah sesuai kebutuhan (editable), dan dirilis dengan lisensi terbuka (open source). Open street map mengizinkan anda untuk menampilkan, mengedit, dan menggunakan data geografis berupa kolaborasi jalan dari berbagai belahan bumi. Layanan ini dapat diakses melalui <http://www.openstreetmap.org>. [10]

2.2 Java OpenStreetMap

Java OpenStreetMap (JOSM) merupakan sebuah perangkat lunak editor peta untuk data OpenStreetMap (OSM). JOSM ini memiliki fungsi utama melakukan membuat, melakukan perubahan atau pembaharuan data OSM. Selain itu, JOSM juga bisa digunakan untuk membuat data peta baru sesuai dengan kebutuhan pengguna. [1]

2.3 Global Positioning System

GPS adalah system satelit navigasi dan penentuan posisi menggunakan satelit. Nama formalnya adalah NAVSTAR GPS, kependekan dari Navigation Satellite Timing and Ranging Global Positioning System. Sistem yang dapat digunakan oleh banyak orang sekaligus dalam segala cuaca ini, di desain untuk memberi posisi, kecepatan tiga dimensi yang teliti, dan informasi mengenai waktu secara kontinyu di seluruh dunia. [2]

2.4 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux. Sistem operasi ini menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi sendiri. Pada awalnya, Google.inc membeli Android.inc yang merupakan pendatang baru untuk menciptakan perangkat lunak mereka sendiri untuk perangkat ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorium dari 34 perusahaan perangkat keras, perangkat lunak dan telekomunikasi termasuk Google.inc, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia. [3]

2.5 Metode SAW

Metode SAW sering dikenal dengan istilah metode penjumlahan terbobot. Konsep dasar dari SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. [4]

2.6 Algoritma Ant Colony Optimization

Algoritma ACO terinspirasi dari perilaku sosial koloni semut dimana seekor semut dapat menjangkau sumber makanan dengan rute terdekat dari sarangnya dengan memanfaatkan material kimia yang disebut pheromone yang dilepaskannya pada saat berjalan, pheromone tersebut akan menarik perhatian semut lain untuk mengikuti suatu rute. [5]

Pada algoritma *Ant Colony Optimization* digunakan istilah *Pheromone* yang menunjukkan intensitas jejak pada suatu edge. Setiap semut akan meninggalkan jejak setiap mereka melewati jalur. Jejak ini yang dinamakan *Pheromone*. Semakin banyak jumlah pheromone yang ada pada suatu rute, semakin potensial rute tersebut untuk diikuti oleh semut-semut lainnya. [6]

2.7 Ant Colony System

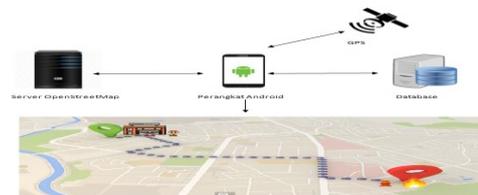
Algoritma Ant Colony System (ACS) merupakan pengembangan dari AS selanjutnya, setelah beberapa algoritma diatas. Algoritma ini tersusun atas sejumlah m semut yang bekerjasama dan berkomunikasi secara tidak langsung melalui komunikasi Pheromone. Secara informal, ACS bekerja sebagai berikut: pertama kali, sejumlah m semut ditempatkan pada sejumlah n titik berdasarkan beberapa aturan inialisasi (misalnya, secara acak). Setiap semut membuat sebuah tour (yaitu, sebuah solusi TSP yang mungkin) dengan menerapkan sebuah aturan transisi status secara berulang kali. Selagi membangun tournya, setiap semut juga memodifikasi jumlah Pheromone pada edge-edge yang 33 dikunjunginya dengan menerapkan aturan pembaruan Pheromone lokal yang telah disebutkan tadi. Setelah semua semut mengakhiri tour mereka, jumlah Pheromone yang ada pada edge-edge dimodifikasi kembali (dengan menerapkan aturan pembaruan Pheromone global). Seperti yang terjadi pada Ant system, dalam membuat tour, semut 'dipandu' oleh informasi heuristic (mereka lebih memilih edge-edge yang pendek) dan oleh informasi Pheromone. Sebuah edge dengan jumlah Pheromone yang tinggi merupakan pilihan yang sangat diinginkan. Kedua aturan pembaruan Pheromone itu dirancang agar semut cenderung untuk memberi lebih banyak Pheromone pada edge-edge yang harus mereka lewati. Berikutnya akan

dibahas mengenai tiga karakteristik utama dari ACS, yaitu aturan transisi status, aturan pembaharuan Pheromone global, dan aturan pembaharuan Pheromone lokal. [7]

3. Pembahasan

3.1 Gambaran Umum Sistem

Pada bagian ini akan dijelaskan mengenai Gambaran umum sistem yang digunakan untuk menggambarkan rancangan sistem yang dibangun secara keseluruhan. Sistem ini dioperasikan oleh pengguna khususnya petugas pemadam kebakaran. Aplikasi ini digunakan untuk menentukan rute terbaik menuju lokasi kebakaran berdasarkan beberapa pertimbangan seperti posisi hydrat, jarak dan kemacetan. Sehingga petugas pemadam kebakaran dapat melalui jalan yang tepat.



Gambar 3. 1 Gambaran Umum Sistem

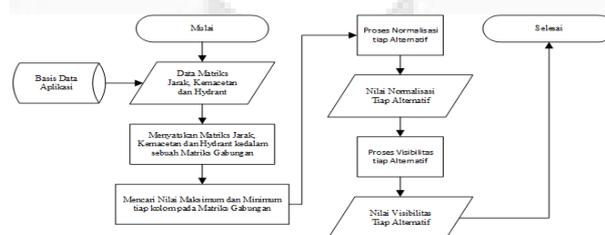
Pada gambar 3.1 menggambarkan mengenai gambaran umum sistem yang digunakan pada perancangan aplikasi. Pada proses awal, aplikasi harus mengaktifkan GPS untuk mendapatkan posisi pengguna dan internet untuk mengunduh peta dari server openstreetmap untuk mendapatkan data peta.

Perangkat android juga terhubung dengan database yang digunakan untuk menyimpan data hydrant dan kemacetan. Setelah mendapatkan data peta dan mengaktifkan GPS, aplikasi siap untuk melakukan pencarian rute.

3.2 Perancangan Sistem

3.2.1 Perancangan Metode SAW

Metode SAW merupakan metode pengambilan keputusan dengan banyak kriteria. Pada aplikasi ini, SAW digunakan untuk melakukan pengolahan data dengan kriteria yaitu Jarak, Hydrant dan Kemacetan. Masing-masing kriteria terdiri matriks yang digambarkan dengan hubungan antar node. Dengan menggunakan metode SAW ini matriks Hydrant, matriks Hydrant dan matriks Kemacetan akan menghasilkan matriks akhir berupa matriks visibilitas. Matriks visibilitas ini merupakan matriks yang digunakan untuk per-rankingan sehingga dapat diketahui alternatif pilihan mana yang paling optimal. Berikut ini adalah diagram alur dari SAW.



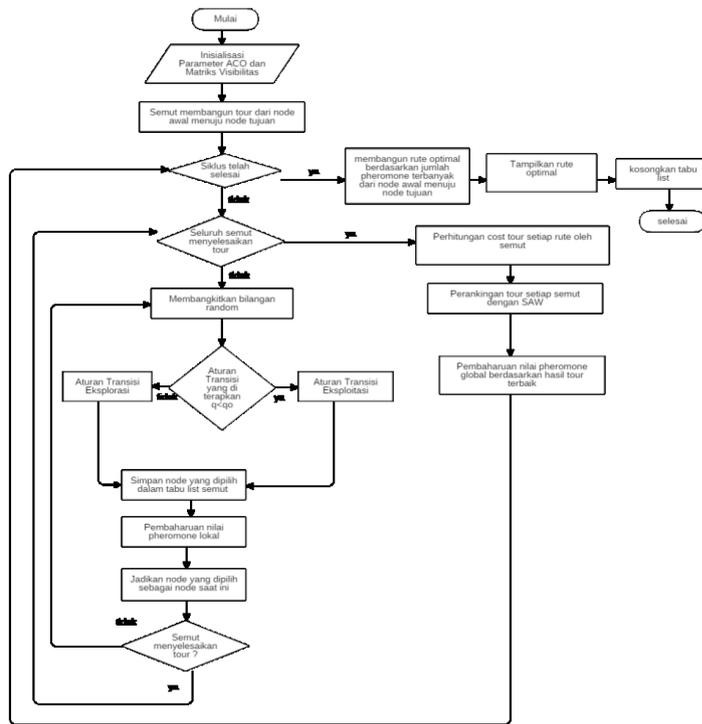
Gambar 3. 2 Diagram Alur SAW

Pada gambar 3.2 menjelaskan cara kerja metode SAW pada aplikasi yang dibuat. Matriks jarak, hydrant dan kemacetan didapatkan dari basisdata aplikasi. Matriks jarak, hydrant dan kemacetan yang terbentuk selanjutnya dilakukan normalisasi. Untuk melakukan proses normalisasi diperlukan penentuan kriteria *benefit* (Keuntungan) atau kriteria *cost* (Kerugian) pada masing-masing kriteria. Matriks normalisasi yang terbentuk selanjutnya akan akan dijadikan satu matriks yaitu matriks akhir preferensi. Matriks inilah yang digunakan untuk pengambilan keputusan. Nilai akhir preferensi terbaik merupakan pilihan alternatif yang terbaik pula.

3.2.2 Perancangan Algoritma Ant Colony System

Pada aplikasi ini, algoritma *Ant Colony Optimization* digunakan sebagai penentuan rute optimal. Matriks akhir preferensi yang didapat setelah melalui proses oleh metode SAW selanjutnya digunakan pada

algoritma sebagai matriks visibilitas. Matriks visibilitas ini merupakan matriks yang menggambarkan kualitas suatu sisi (*Edge*).

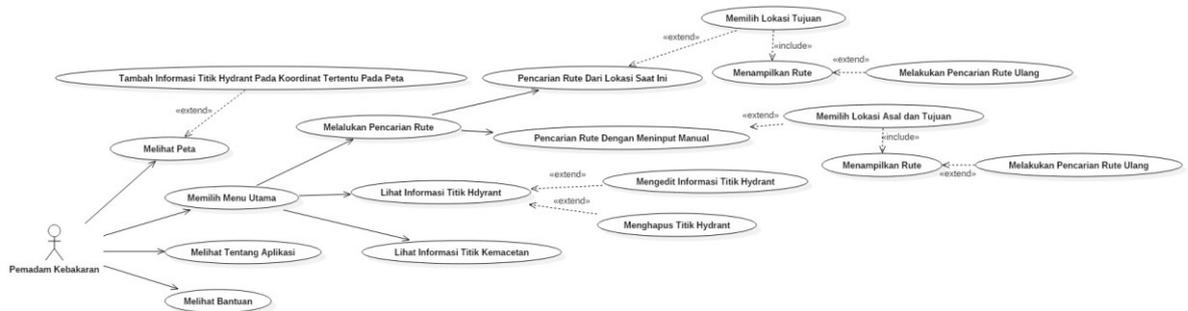


Gambar 3. 3 Diagram Alur Algoritma Ant Colony System

Pada gambar 3.3 menggambarkan dalam menentukan pengambilan keputusan node yang dipilih pada algoritma *Ant Colony System* memiliki dua cara yaitu aturan transisi eksplorasi dan eksploitasi. Penentuan aturan yang dipilih berdasarkan bilangan random yang di bangkitkan. Jika digunakan aturan transisi eksplorasi, dalam penentuan node selanjutnya dipilih berdasarkan nilai maksimum dari perhitungan aturan transisi eksplorasi. Dengan kata lain, semut akan memanfaatkan pengetahuan tentang heuristik jarak dan nilai pheromone antara node tersebut. Jika digunakan aturan eksploitasi maka node selanjutnya ditentukan berdasarkan probabilitas semut pada satu titik menuju titik berikutnya. Selanjutnya dilakukan pembaharuan lokal pada sisi yang dilewati dan simpan node yang dipilih ke dalam tabu list. Perulangan dilakukan sampai seluruh semut menyelesaikan tour atau telah mencapai titik tujuan. Kemudian dibandingkan masing-masing tour semut dan dicari rute paling optimal pada siklus tersebut. Rute semut yang paling optimal tersebut selanjutnya dilakukan pembaharuan global pada setiap sisi rute yang dilalui. Pengosongan tabu list dilakukan untuk digunakan kembali oleh semut pada siklus berikutnya.

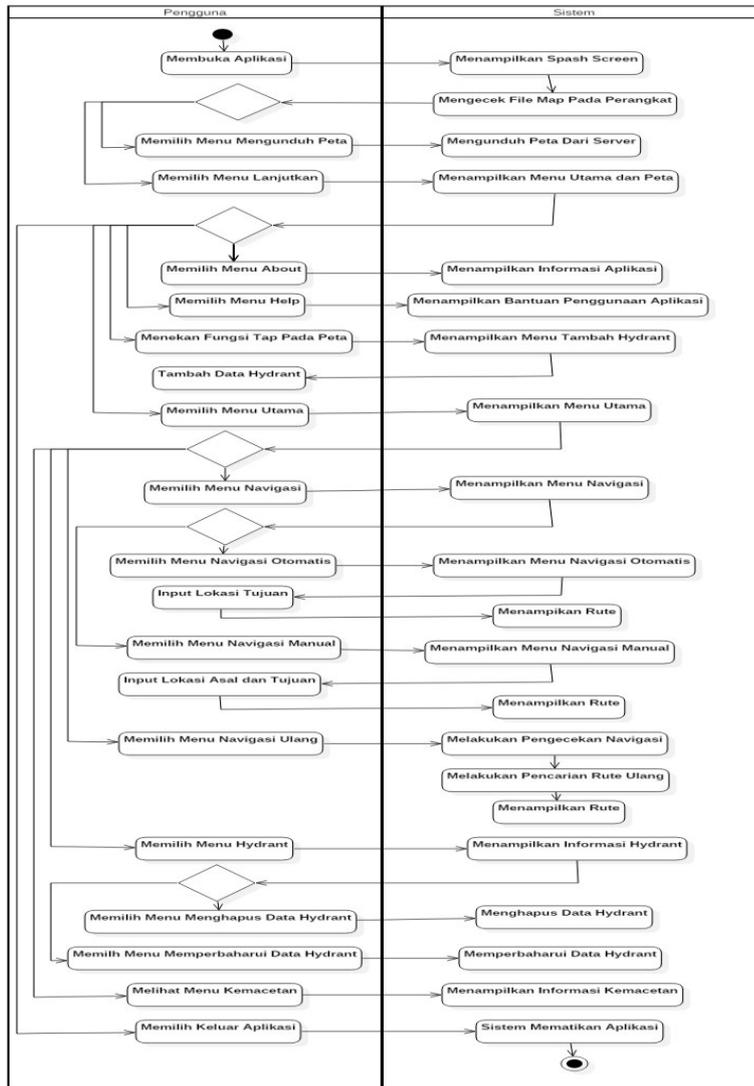
3.3 Pemodelan Sistem

3.3.1 Use Case Diagram



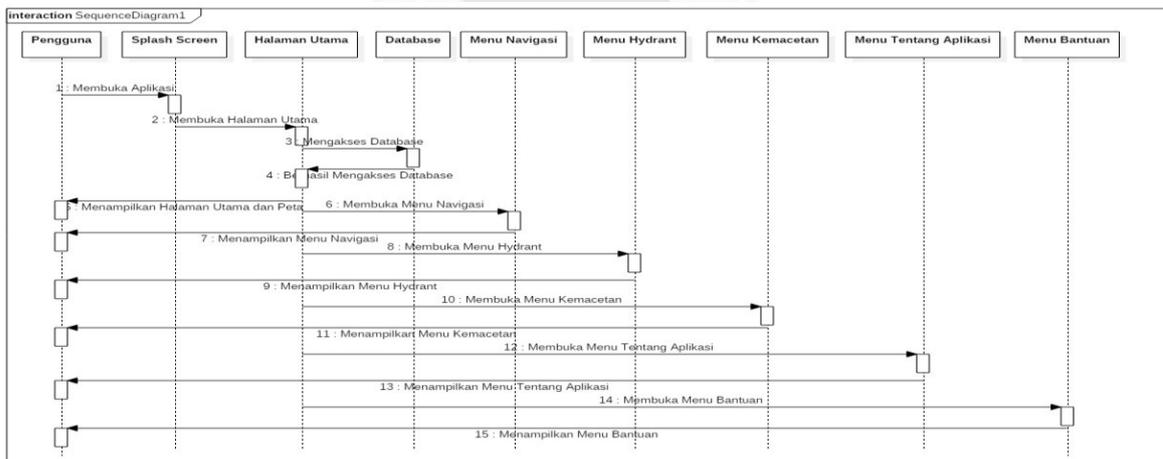
Gambar 3.4 Use Case Diagram

3.3.2 Activity Diagram

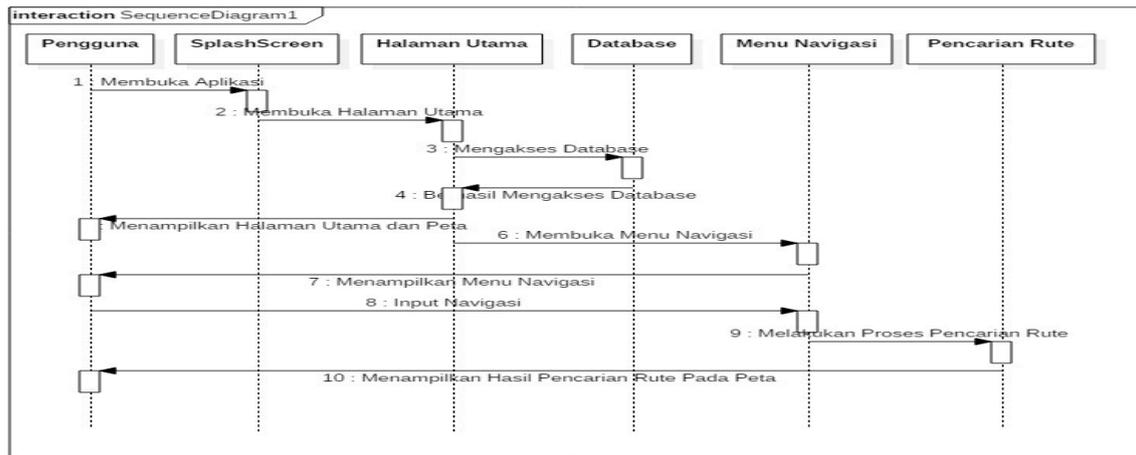


Gambar 3.5 Activity Diagram

3.3.3 Sequence Diagram



Gambar 3.6 Sequence Diagram Halaman Utama



Gambar 3.7 Sequence Diagram Menu Navigasi

4. Hasil

Pengujian aplikasi dilakukan untuk mengetahui waktu yang diperlukan sistem untuk melakukan perhitungan pencarian rute menggunakan algoritma *Ant Colony System* serta penggunaan sumber daya *Memory* dan *CPU* pada perangkat saat melakukan perhitungan. Pada pengujian ini digunakan jumlah node sebanyak 660 node yang menggambarkan persimpangan dan tikungan pada jalan kota Bandung.

Tabel 4.1 Spesifikasi Perangkat Pengujian Aplikasi

No	Nama Perangkat	Versi Android	Prosesor (GHz)	RAM (GB)
1	Perangkat 1	Kitkat (4.4)	1.5 Dual-Core	1,5
2	Perangkat 2	Lolipop (5.0)	2.3 Quad-Core	4
3	Perangkat 3	Lolipop (5.1)	1.3 Quad-Core	2

Pengujian ini dilakukan pada tiga rute berbeda dilakukan saat pengguna menggunakan fitur navigasi. Pada fitur ini proses pencarian rute menggunakan algoritma *Ant Colony System* dilakukan. Pengujian ini dilakukan pada masing-masing rute dan setiap komponen uji dilakukan pengujian sebanyak lima kali.

Tabel 4.2 Hasil Pengujian Waktu Perhitungan Algoritma ACS Perangkat 1

Rute	Komponen Uji	Pengujian					Rata-Rata
		1	2	3	4	5	
1	Waktu Perhitungan ACO (s)	4.34	4.34	4.22	3.73	4.18	4.162
	Penggunaan <i>Memory</i> (MB)	78.8	74.62	80.72	78.08	76.04	77.652
	Penggunaan CPU (%)	48.5	40.2	42.1	44.8	48.6	44.84
2	Waktu Perhitungan ACO (s)	4.206	4.26	4.82	4.61	4.75	4.5292
	Penggunaan <i>Memory</i> (MB)	73.68	80.19	78.88	77.42	76.53	77.34
	Penggunaan CPU (%)	48.6	48.57	50.00	48.38	47.65	48.64
3	Waktu Perhitungan ACO (s)	3.88	3.83	4.01	3.79	3.76	3.86
	Penggunaan <i>Memory</i> (MB)	80.78	76.3	78.84	80.85	81.13	79.58
	Penggunaan CPU (%)	58.6	60.54	46.5	54.65	48.6	53.778

Pada tabel 4.2 merupakan hasil pengujian perhitungan algoritma ACS pada perangkat 1. Waktu yang diperlukan oleh perangkat 1 memiliki rata-rata 4.1837 detik pada setiap rute dan Penggunaan *Memory* rata-rata sebesar 78.19 MB dan penggunaan CPU rata-rata sebesar 49.086 %.

Tabel 4.1 Hasil Pengujian Waktu Perhitungan Algoritma ACS Perangkat 2

Rute	Komponen Uji	Pengujian					Rata-Rata
		1	2	3	4	5	
1	Waktu Perhitungan ACO (s)	36.09	42.07	38.72	39.15	43.05	39.816
	Penggunaan <i>Memory</i> (MB)	126.9	129.7	129.9	128.4	131.4	129.208
	Penggunaan CPU (%)	28.17	30.28	31	29.11	30.77	29.866

2	Waktu Perhitungan ACO (s)	42.1	44.6	45.1	44.48	45.40	44.336
	Penggunaan <i>Memory</i> (MB)	130.6	131.9	132.2	128.8	129.7	130.64
	Penggunaan CPU (%)	30.49	31.33	31.91	31.36	31.51	31.32
3	Waktu Perhitungan ACO (s)	39.34	44.07	41.42	41.62	39.61	41.212
	Penggunaan <i>Memory</i> (MB)	138.8	134.5	132.4	133.6	134.1	134.68
	Penggunaan CPU (%)	32.13	33.04	32.51	32.28	31.11	32.214

Pada tabel 4.3 merupakan hasil pengujian perhitungan algoritma ACS pada perangkat 2. Waktu yang diperlukan oleh perangkat 2 memiliki rata-rata 41.788 detik pada setiap rute dan Penggunaan Memory rata-rata sebesar 131.50 MB dan penggunaan CPU rata-rata sebesar 31.133 %.

Tabel 4. 2 Hasil Pengujian Waktu Perhitungan Algoritma ACS Perangkat 2

Rute	Komponen Uji	Pengujian					Rata-Rata
		1	2	3	4	5	
1	Waktu Perhitungan ACO (s)	52.2	52.5	52.26	52.54	52.54	52.408
	Penggunaan <i>Memory</i> (MB)	92.01	90.68	90.58	94.26	94.5	92.406
	Penggunaan CPU (%)	51.80	51.58	52.69	54.69	52.15	52.582
2	Waktu Perhitungan ACO (s)	54.62	54.29	54.24	53.98	54.28	54.282
	Penggunaan <i>Memory</i> (MB)	102.7	102.5	100.2	101.2	98.58	101.036
	Penggunaan CPU (%)	47.88	47.68	46.55	46.18	47.68	47.594
3	Waktu Perhitungan ACO (s)	62.2	62.5	62.58	63.01	62.5	62.558
	Penggunaan <i>Memory</i> (MB)	105.8	105.6	104.2	100.3	102.5	103.68
	Penggunaan CPU (%)	42.87	42.26	48.58	46.52	47.56	45.492

Pada tabel 4.3 merupakan hasil pengujian perhitungan algoritma ACS pada perangkat 3. Waktu yang diperlukan oleh perangkat 2 memiliki rata-rata 56.416 detik pada setiap rute dan Penggunaan Memory rata-rata sebesar 68.707 MB dan penggunaan CPU rata-rata sebesar 48.456 %.

5. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan pada bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut :

1. Aplikasi mampu melakukan pencarian rute optimal berdasarkan jalan dengankriteria jarak, kemacetan dan posisi hydrant.
2. Algoritma Ant Colony Optimization dapat di implementasikan dalam penentuan rute optimal pada perangkat *mobile* berbasis sistem operasi *android*.
3. Dari hasil pengujian aplikasi, didapatkan perangkat dengan waktu perhitungan adalah perangkat 1 dengan waktu rata-rata 0.412 detik untuk perhitungan metode Simple Additive Weighting dan 4.1837 detik untuk perhitungan algoritma *Ant Colony System*.

DAFTAR PUSTAKA

- [1] Riyanto. 2010. Membuat Sendiri Aplikasi Mobile GIS. Yogyakarta : Andi
- [2] Tanoe, Andre. 2009. GPS bagi pemula, dasar dasar pemakaian sehari hari.
- [3] Safaat, Nazruddin. 2012. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung : Informatika.
- [4] Kusumadewi, Sri dkk 2006, Fuzzy Multi-Attribute Decision Making FuzzyMADM) ,Graha Ilmu, Yogyakarta.
- [5] Ahmed Al-Ani. “**Feature Subset Selection using Ant Colony Optimization**”. International Journal of Computational Intelligence, Winter 2006.
- [6] Andrea Roli. “**Ant Colony Optimization**”. Aironews Vol.7 no.3 (Pages1-3), Autumn 2002.
- [7] Agus Leksono.(2009) “**Algoritma Ant Colony Optimization (ACO) Untuk Menyelesaikan Traveling Salesman Problem (TSP)**”. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Diponegoro.