ISSN cetak 2087-1716
ISSN online 2548-7779

ILKOM Jurnal Ilmiah Volume 10 Nomor 1 April 2018

RANCANG BANGUN APLIKASI *MOBILE* UNTUK MENENTUKAN SOLUSI OPTIMAL PENCARIAN RUTE TERBAIK MENGGUNAKAN ALGORITMA *ANT COLONY OPTIMIZATION*

Budhi Irawan[1], Casi Setianingsih[2] , Izzat Arramsyah[3]
[1]budhiirawan@telkomuniversity.ac.id, [2] setiacasie@telkomuniversity.ac.id, [3]arramsyah@gmail.com
[1,2,3] Prodi Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

Abstrak

Dampak dari musibah kebakaran bisa ditekan jika petugas dan kendaraan Damkar (pemadam kebakaran) bisa bekerja cepat. Selain faktor keterlambatan laporan kepada Damkar dan kondisi jalan Kota Bandung yang macet ikut memperlambat laju petugas. Proses pengisian tangki air menjadi masalah utama, selain terbatas keberadaannya juga kualitas semburannya rendah sehingga petugas harus mencari sumber air selain hydrant. Dari permasalahan tersebut maka dibutuhkan suatu alat bantu yang praktis berupa aplikasi dengan memanfaatkan perangkat *smartphone* yang dapat membantu mencarikan solusi optimal guna mendapatkan rute perjalanan petugas Damkar dalam upaya menjangkau lokasi kebakaran dan mendapat sumber air didalam mendukung tugasnya memadamkan api di lokasi kebakaran. Adapun guna menentukan rute optimal sesuai kebutuhan diatas maka dipilih algoritma ACO (*Ant Colony Optimization*) dan Metode SAW (*Simple Additive Weighting*) yang diimplementasikan pada aplikasi *mobile* yang dibangun. Sehingga dengan aplikasi ini dapat membantu para petugas Damkar didalam menjalankan tugasnya terutama mendapatkan rute jalan yang optimal beserta sumber air yang diperlukan.

Kata kunci : *Simple Additive Weighting, Ant Colony Optimization, Aplikasi Mobile*

1. Pendahuluan

Jumlah peredaran kendaraan di Kota Bandung setiap hari kini semakin padat. Simpul kemacetan terus bertambah di sejumlah ruas jalan. Kondisi seperti itu salah satunya menghambat laju kendaraan Damkar di kota berpenduduk cukup padat ini. Kasus kebakaran yang paling banyak biasanya api membakar kawasan permukiman padat dan separuh total kasus kebakaran lainnya menimpa pabrik, bangunan umum, dan juga gardu listrik. Umumnya penyebab kebakaran diduga karena arus pendek listrik. Besarnya dampak musibah kebakaran bisa ditekan jika petugas dan kendaraan Damkar bisa bekerja dengan relatif cepat. Selain faktor keterlambatan laporan warga ke pihak Damkar dan kondisi jalan didalam kota yang kian macet maka ikut memperlambat laju petugas dari markas ke lokasi kejadian.. Pada jam lalu lintas sibuk dari pagi hingga petang, pengemudi harus menemukan jalan alternatif hingga terpaksa melawan arus. Selain itu yang menjadi masalah juga adalah proses pengisian tangki air yang mana, dari beberapa hydrant yang ada, hanya ada sedikit yang dapat berfungsi dan menyembur kuat untuk mengisi tangki dengan cepat. Apabila semburannya kecil maka petugas harus mencari sungai atau hydrant milik instansi atau pabrik terdekat.

Algoritma ACO diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut. Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang menuju ke sumber makanan dan kembali lagi. Pada saat semut berjalan, semut meninggalkan sebuah informasi yang disebut *pheromone* di tempat yang dilaluinya dan menandai rute tersebut. *Pheromone* digunakan sebagai komunikasi antar semut pada saat membangun rute[1, 2, 3, 4]. Sedangkan metode SAW adalah salah satu metode yang dapat digunakan untuk menyelesaikan masalah MADM *(Multi-Attribute Decision Making).* Metode ini mencari alternatif terbaik, dengan mengevaluasi alternatif terhadap sekumpulan atribut atau kriteria[1, 4, 5, 6].

Dari permasalahan diatas maka dibutuhkan suatu alat bantu yang dapat membantu mencarikan solusi optimal guna mendapatkan rute terbaik perjalanan petugas Damkar dalam upaya menjangkau lokasi kebakaran dan mendapat sumber air didalam mendukung tugasnya memadamkan api, dalam hal ini diimplementasikan berupa aplikasi dengan memanfaatkan perangkat *smartphone*. Adapun guna menentukan rute optimal sesuai kebutuhan diatas maka dipilih algoritma ACO dan metoda SAW yang diimplementasikan pada aplikasi *mobile* yang dibangun.

2. Metode

2.1. Metode SAW

ISSN cetak 2087-1716
ISSN online 2548-7779

Metode SAW sering dikenal dengan istilah metode penjumlahan terbobot. Konsep dasar dari medote SAW ini adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode ini membutuhkan proses normalisasi matriks keputusan ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada [8].

$$r_{ij} = \begin{cases} \dfrac{x_{ij}}{Max\, x_{ij}} & \text{Jika j adalah atribut keberuntungan } \textit{(benefit)} \\ \dfrac{Min\, x_{ij}}{x_{ij}} & \text{Jika j adalah atribut biaya } \textit{(cost)} \end{cases} \quad (1)$$

dimana :

r_{ij} = rating kinerja ternormalisasi dari alternatif A_i pada atribut Cj
di mana i = 1, 2, ... , m dan j = 1, 2, ..., n.

Nilai preferensi untuk setiap alternatif diberikan sebagai :

$$V_i = \sum_{j=1}^{n} w_j r_{ij} \quad (2)$$

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih. Langkah-langkah dari metode SAW adalah :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan,
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria,
3. Membuat matriks keputusan berdasarkan kriteria (C), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R,
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A) sebagai solusi [8].

2.2. Algoritma ACO

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 [9].

Pada algoritma ACO digunakan istilah *pheromone* yang menunjukan intenstias jejak pada suatu *edge*. Setiap semut akan meninggalkan jejak setiap mereka melewati jalur. Semakin banyak jumlah *pheromone* yang ada pada suatu rute, semakin potensial rute tersebut untuk diikuti oleh semut-semut lainnya [10].

2.3. Algoritma ACS *(Ant Colony System)*

Algoritma ACS adalah bagian dari pengembangan Algoritma *Ant System*. Algoritma tersebut disusun dengan sekumpulan *m* semut yang bekerjasama dan melakukan komunikasi secara tidak langsung melalui komunikasi *pheromone*. Adapun algoritma ACS berjalan sebagai berikut yaitu pertama-tama sejumlah m semut ditempatkan pada sejumlah n titik berdasar kepada beberapa aturan inisialisasi. Setiap semut akan membuat sebuah solusi *(tour)* dengan cara menerapkan sebuah aturan transisi status secara berulang kali. Selama membangun *tour*nya, setiap semut memodifikasi jumlah *pheromone* pada *edge-edge* yang akan dikunjunginya dengan menerapkan aturan pembaruan *pheromone* lokal. Setelah semua semut mengakhiri *tour*, jumlah *pheromone* yang ada pada *edge-edge* dimodifikasi kembali dengan cara menerapkan aturan pembaruan *pheromone* global. Seperti yang terjadi pada AS, dalam membuat *tour*, semut dipandu oleh adanya informasi *heuristic* yaitu mereka lebih memilih *edge-edge* yang pendek dan informasi *pheromone*. Sebuah *edge* dengan jumlah *pheromone* yang tinggi merupakan suatu pilihan yang sangat diinginkan. Kedua aturan pembaruan *pheromone* itu dirancang agar supaya semut cenderung untuk memberi lebih banyak *pheromone* pada *edge-edge*

ISSN cetak 2087-1716
ISSN online 2548-7779

yang harus mereka lewati. Karakteristik utama dari ACS diantara yaitu aturan transisi status, aturan pembaharuan *pheromone* global, dan aturan pembaharuan *pheromone* lokal [11].

2.3.1. Aturan Transisi Status

Aturan dari transisi status yang berlaku pada ACS yang ditunjukkan pada persamaan (3) yang mana semut *k* yang berada di titik *r*, akan pemilihan titik berikutnya *s*, menurut persamaan sebagai berikut :

$$s = \begin{cases} \arg\max_{uc J_r^k} \{\tau_{ru}][n_{ru}]^{\beta}\} & \text{Jika } q \leq q_0 \text{ (eksploitasi)} \\ J, & \text{Jika tidak (eksplorasi)} \end{cases} \quad (3)$$

Dimana q adalah bilangan random dalam [0,1], q_0 (0 1) ≤ q_0 ≤ adalah sebuah parameter pembanding bilangan random, dan J adalah probabilitas dari semut k pada titik r yang akan melakukan pemilihan untuk menuju kepada titik s yang ditunjukan pada persamaan (4) berikut ini.

$$P_{rs}^k = \begin{cases} \frac{[\tau_{rs}]^{\alpha}[n_{rs}]^{\beta}}{\sum_{uc J_r^k} \tau_{ru}][n_{ru}]^{\beta}} & \text{Untuk s } \in \text{ j}_\text{r}{}^\text{k} \\ 0 & \text{Untuk s lainnya} \end{cases} \quad (4)$$

Dimana τ_{rs} adalah jumlah *pheromone* yang terdapat pada *edge* antara titik r dan titik *s* dan η_{rs} adalah *visiblilty*. α merupakan suatu parameter yang melakukan pengontrolan terhadap bobot (*weight*) relatif dari *pheromone* dan β merupakan suatu parameter pengendali dari jarak ($\alpha > 0$ dan $\beta > 0$). Pada persamaan (3), *pheromone* pada *edge* (r,s) dikalikan dengan suatu nilai *visibility* yang sesuai (η_{rs}). Dengan menggunakan cara ini maka akan dilakukan proses pemilihan edge yang lebih pendek dan memiliki jumlah *pheromone* yang lebih besar. Dengan istilah lain, Jika $q \leq q_0$ maka semut tersebut akan dapat memanfaatkan pengetahuan *heuristik* mengeai jarak antara titik itu dengan titik-titik lainnya dan juga pengetahuan yang telah diperoleh akan disimpan dalam bentuk *pheromone*. Hal ini mengakibatkan *edge* yang paling baik berdasarkan kepada persamaan (3) akan dipilih. Jika sebaliknya maka sebuah *edge* dan dilakukan pemilihan berdasarkan persamaan (4) [11].

2.3.2. Aturan Pembaharuan Pheromone Lokal

Pada saat akan membangun *tour*, semut mengimplementasikan *local updating rule* berdasarkan persamaan berikut :

$$\tau_{rs} \leftarrow (1 - \xi)\, \tau_{rs} + \xi \tau_o \quad (5)$$

ξ merupakan sebuah parameter evaporasi lokal 0 < ξ < 1. 0 τ merupakan sebuah nilai awal jejak *pheromone*, τ_0 = 1/n C_{nn} dimana n adalah jumlah dari titik dan C_{nn} merupakan panjang sebuah *tour* yang paling baik yang diperoleh dari metode *nearest neighbourhood heuristic*. Persamaan *update pheromone online* ini diimplementasikan pada saat semut membangun *tour*, yaitu ketika akan melewati *edge* dan melakukan pengubahan tingkat *pheromone* pada *edge* (r,s). Tujuan dari kegiatan tersebut adalah guna membantu melewati sebuah *edge*, yang mana *edge* ini akan menjadi kurang diinginkan karena berkurangnya jejak *pheromone* pada *edge* yang bersesuaian tersebut.[11].

2.3.3. Aturan Pembaharuan Pheromone Global

Apabila semua semut sudah menyelesaikan sebuah *tour*, tingkat *pheromone* akan di-*update* dengan mengimplementasikan *global updating rule* berdasar kepada persamaan (6) berikut :

$$\tau_{rs} \leftarrow (1 - \rho)\, \tau_{rs} + \rho \Delta \tau_o \quad (6)$$

ISSN cetak 2087-1716
ISSN online 2548-7779

Dimana :

$\Delta\tau(r,s) = \quad (L_{gb})^{-1}$, jika $(r,s) \in$ rute terbaik keseluruhan

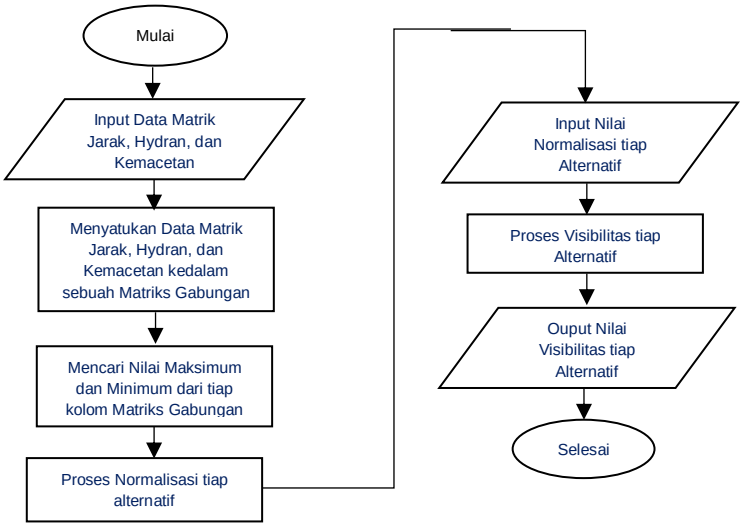
Dimana ρ merupakan sebuah parameter evaporasi global, yang mempunyai nilai $0 < \rho < 1$. $\Delta\tau$ merupakan skala 1 / panjangdari lintasan yang paling baik secara keseluruhan, jika (i,j) merupakan bagian dari panjang lintasan yang paling baik secara keseluruhan.

Pada persamaan dari *update* jejak *pheromone* secara *offline* ini, dilakukan pada bagian akhir dari proses iterasi algoritma, saat dari semua semut sudah menyelesaikan sebuah *tour*. Persamaan diimplementasikan ke *edge* yang akan digunakan semut guna menemukan lintasan keseluruhan yang paling baik sejak awal penelusuran. Tujuan dari proses pemberian nilai ini adalah guna memberikan sejumlah jejak *pheromone* pada lintasan yang paling pendek, dimana *tour* terbaik akan mendapat penguatan. Bersama dengan *pseudo-random proportional rule* dimaksudkan untuk dapat membuat proses pencarian menjadi lebih terarah [11].

3. Hasil dan Pembahasan

3.1. Rancanngan SAW

Metode SAW merupakan metode pengambilan keputusan dengan banyak kriteria. Pada aplikasi ini, metode SAW digunakan untuk melakukan pengolahan data kriteria Jarak, Posisi Hydrant dan Kemacetan. Masing-masing kriteria terdiri matriks yang menggambarkan hubungan antar *node*. B



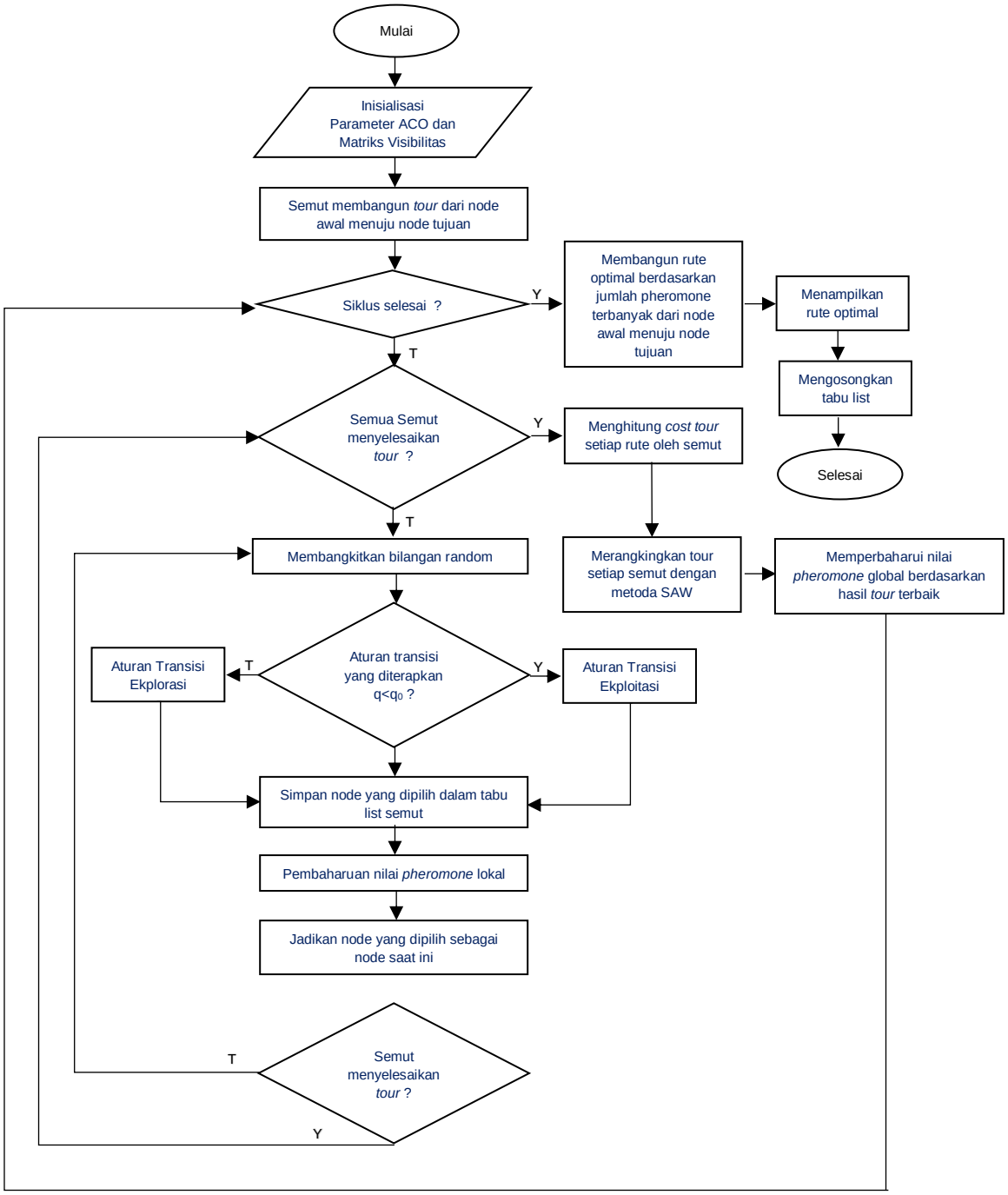
Gambar 1. Diagram Alur SAW

Pada gambar 1 ditunjukan cara kerja metode SAW pada aplikasi yang dibuat. Matriks jarak, posisi hydrant dan kemacetan didapatkan dari basisdata aplikasi. Matriks jarak, posisi hydrant dan kemacetan yang terbentuk selanjutnya dilakukan normalisasi dan untuk melakukan proses normalisasi diperlukan penentuan kriteria *benefit* (Keuntungan) atau kriteria *cost* (Kerugian) pada masing-masing kriteria. Matriks normalisasi yang terbentuk selanjtunya 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. Rancangan ACS

Pada aplikasi ini, algoritma ACS digunakan sebagai penentuan rute optimal. Matriks akhir preferensi yang didapat setelah melalui proses oleh metode SAW selanjtunya digunakan pada algorimta sebagai nilai *Visibilitas*. Nilai *Visibilitas* ini merupakan nilai yang menggambarkan kualitas suatu *edge.*

ISSN cetak 2087-1716
ISSN online 2548-7779



Gambar 2 Diagram Alur Algoritma ACS

Pada gambar 2 merupakan alur kerja dari algoritma *ACS* digunakan pada sistem. Berikut adalah penjelasan dari masing-masing tahap :

1. Menginisialisasi Parameter Algoritma ACS yang meliputi : Intensitas *pheromone* (τ_{ij}), q_0 (Paramter pembanding bilangan random), Parameter pengendali intensitas *pheromone* (α), Parameter pengendali *Visibilitas* (β), Jumlah Semut (m), Paramter penguapan *pheromone* (ρ), Jumlah Siklus Maksimum ($NCmax$)
2. Semut akan ditempatkan di titik awalnya yang kemudian semut akan melakukan *tour*nya dengan memilih *node* selanjutnya. Dalam memilih *node* selanjutnya, semut menggunakan aturan transisi status. Untuk menentukan aturan transisi yang digunakan, oleh karena itu perlu dibangkitkan suatu bilangan secara random (q) antara 0 sampai 1 sebagai acuan untuk memilih aturan transisi yang akan digunakan. Berikutnya bilangan random telah dibangkitkan ini kemudian dibandingkan dengan

ISSN cetak 2087-1716
ISSN online 2548-7779

nilai parameter q_0 yang telah ditetapkan. Penetapan parameter q_0 ini berpengaruh kepada pemilihan aturan transisi yang digunakan. Misal, jika parameter q_0 yang digunakan sebesar 0.9 artinya bahwa semut akan melakukan suatu proses eksploitasi dengan nilai probabilitas sekitar 90% dan eksplorasi sebesar 10%. Setelah didapatkan hasil perbandingan antara nilai q dengan nilai q_0 maka selanjutnya akan di tentukan aturan transisi yang digunakan. Aturan transisi ini berdasarkan persamaan (3) atau persamaan (4) dengan ketentuan :

a) Jika $q \leq q_0$ maka digunakan aturan transisi eksploitasi persamaan (3)
b) Jika $q > q_0$ maka digunakan aturan trasisi eksplorasi dengan perasmaan (4)

3. Setelah didapatkan *node* yang dituju maka selanjutnya dilakukan proses penyimpanan *node* tersebut ke dalam *tabu list. Tabu List* ini digunakan sebagai memori bagi semut untuk menyimpan hasil pencarian rutenya. Selain itu juga mencegah semut untuk mengujungi *node* yang telah dilewati.
4. Pada Algoritma ACS, pembaharuan nilai *pheromone* dilakukan pada sisi yang dilewati semut (Pembaharuan *Pheromone* Lokal). Pembaharuan ini dilakukan setiap semut melewati *node*. Pembaharuan *pheromone* ini menggunakan persamaan (5) dan pembaharuan *pheromone* ini dimaksudkan agar daya tarik pada sisi tersebut menjadi berikurang (*evaporasi* atau penguapan). Sehingga semut lainnya tidak akan melewati sisi tersebut. Sisi yang lebih pendek atau sisi dengan kadar *pheromone* yang banyak akan lebih dipilih untuk dikunjungi semut.
5. Kemudian, setelah didapatkan *node* beriktunya maka menjadikan *node* tersebut sebagai *node* saat ini. Kemudian dilakukan proses pencarian rute dengan mengulangi langkah 2 hingga 4 sampai seluruh semut menyelesaikan *tour*nya atau menemukan *node* tujuan.
6. Setelah seluruh semut m enyelesaikan *tour*nya, setiap *tour* yang dihasilkan kemudian akan dipilih *tour* terbaik dan pemilihan *tour* terbaik ini menggunakan metode perangkingan antar *tour* yang dihasilkan menggunakan metode SAW.
7. Setelah didapatkan *tour* terbaik, kemudian dilakukan pembaharuan *pheromone* global pada isi *tabu list* semut dengan *tour* terbaik tersebut. Pembaharuan *pheromone* global ini menggunakan persamaan (6) dan pembaharuan *pheromone* global ini dimaksudkan agar kadar *pheromone* pada sisi *tour* terbaik semut tersebut mengalami penguatan dan sisi yang tidak menjadi bagian *tour* terbaik mengalami penguapan. Sehingga, pada siklus berikutnya pencarian lebih terarah. Isi *tabu list tour* terbaik tersebut di simpan kedalam *tabu list* rute terbaik yang kemudian dilakukan kembali proses pencarian pada siklus selanjutnya dengan nilai *pheromone* yang telah di perbaharui pada prsoes pembaharuan *pheromone* global.
8. Selanjutnya, isi *tabu list* di kosongkan untuk di isi kembali oleh siklus berikutnya dan proses pencarian *rute* oleh semut dilakukan sampai seluruh siklus selesai. Setelah proses pencarian selesai maka akan terlihat pada sisi *pheromone tour* terbaik akan mengalami penguatan. Jalan atau sisi yang menjadi *tour* terbaik inilah yang akan ditampilkan sebagai hasil pencarian rute optimal.

3.3. Rancangan Area Pemetaan Wilayah

Perancangan wilayah yang diimplementasikan pada aplikasi menggunakan *software* JOSM (*Java OpenStreetMap Editor*).

Gambar 3 Rancangan Area Pemetaan Wilayah Kota Bandung

Pada gambar 3 ditunjukan hasil rancangan jalan utama kota Bandung yang digunakan pada aplikasi ini. Dari perancangan jalan menggunakan *software* JOSM ini bisa didapatkan data tiap *node*. Dimana tiap *node* ini memiliki *id node*, *latitude*, *longitude* dan data *node* ini bisa berupa persimpangan

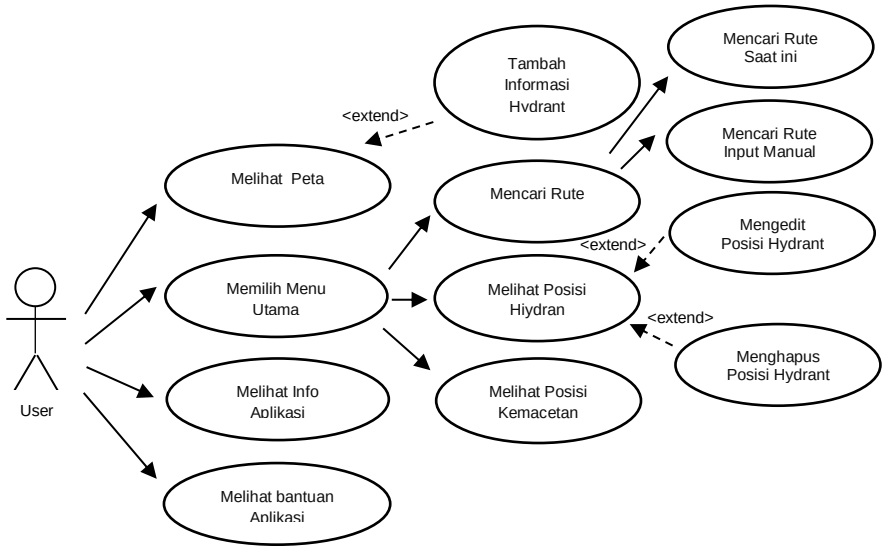
ISSN cetak 2087-1716
ISSN online 2548-7779

jalan atau bagian dari jalan yang berbelok juga selain itu bisa didapatkan juga data jalan yang menyatakan sebuah *edge* atau jalan yang terdiri dari dua *node* atau lebih.

3.4. Perancangan Aplikasi

Pada perancangan aplikasi ini menggunakan pemodelan UML (*Unified Model Languange*) diantaranya berupa *Use Case Diagram* dan *Sequence Diagram.*

3.4.1 Use Case Diagram

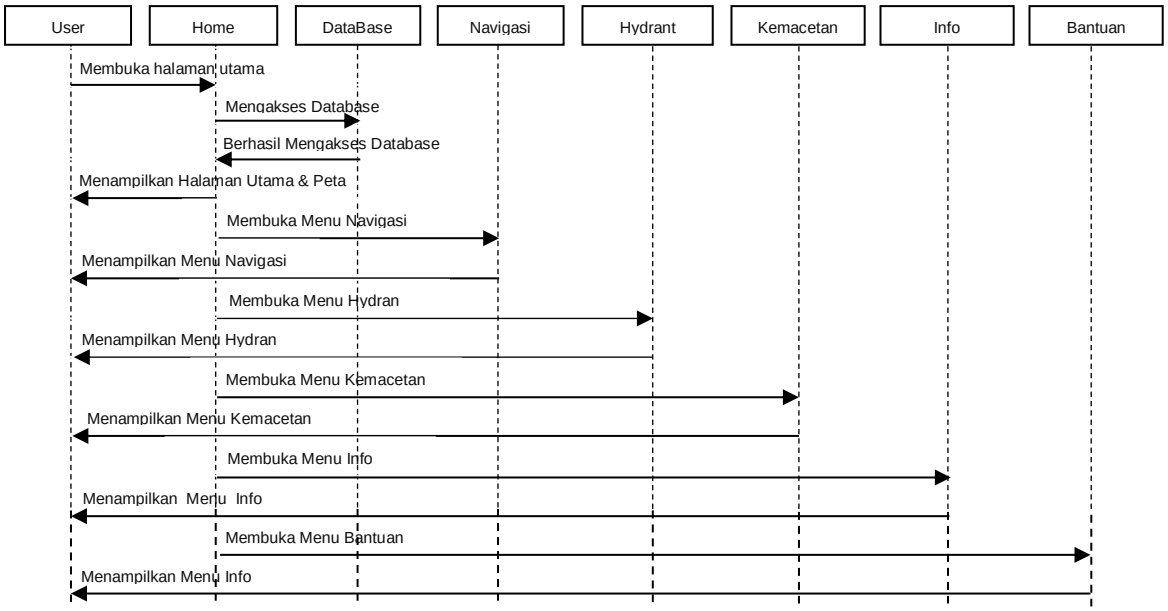


Gambar 4. *Use Case Diagram*

Pada gambar 4 ditunjukan interaksi antara pengguna dan aplikasi yang mana, pengguna dapat melihat peta, melihat peta, mencari rute, melihat posisi hydrant, dan melihat posisi kemacetan.

3.4.2 Sequence Diagram

Pada gambar 5 dibawah ini ditunjukan proses sekuensial yang terjadi pada saat membuka bagian utama aplikasi. Pada proses ini dilakukan beberapa proses navigasi, reroute, pencarian hydrant dan kalkulasi kemacetan.

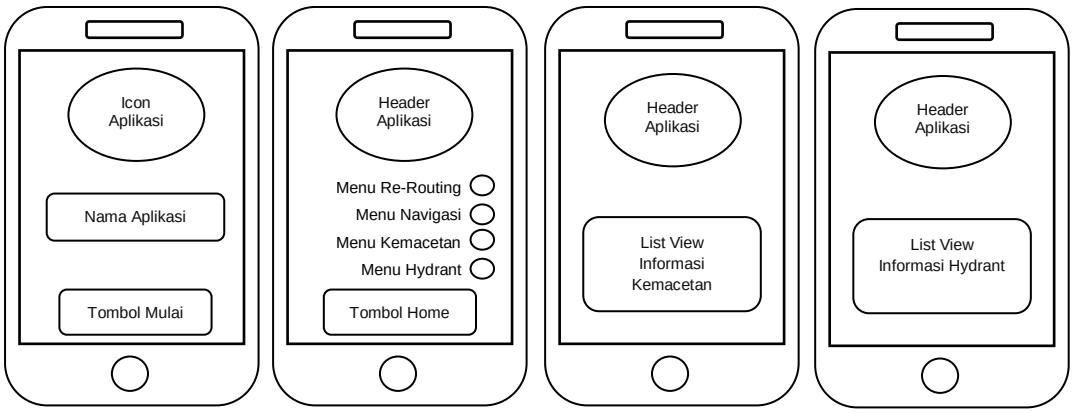


Gambar 5. *Sequence Diagram* Bagian Utama Aplikasi

ISSN cetak 2087-1716
ISSN online 2548-7779

3.4.3. Rancangan Antarmuka Aplikasi

Pada gambar 6 dibawah ini ditunjukan proses rancangan antarmuka utama aplikasi. Pada antarmuka ini ditampilkan layout *property* aplikasi mobile yang meliputi halaman utama beserta halaman-halaman detilnya.

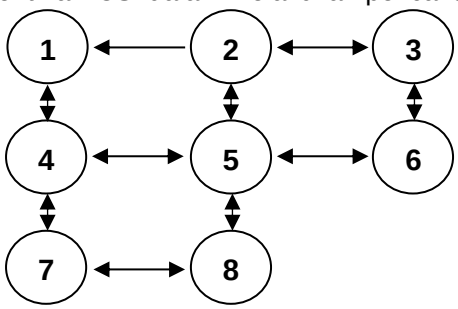


Gambar 6. Rancangan Antarmuka

3.5. Pengujian Sistem

3.5.1. Pengujian Medode SAW dan ACS

Pada pengujian metode SAW dan algortima ACS ini diperlukan sebuah skenario pemetaan jalan yang digunakan oleh metode SAW untuk melakukan bobot jalan dengan perhitungan kriteria jarak, kemacetan dan posisi. Serta algoritma ACS dalam melakukan pencarian rute.



Gambar 8. Skenario Pemetaan Jalan

Keterangan :

1. → : *Edge* (Arah Hubungan) Satu Arah Dari *Node* Awal Ke *Node* Tujuan
2. ↔ : *Edge* (Arah Hubungan) Dua Arah Dari *Node* Awal Ke *Node* Tujuan
3. ○ : *Node* atau Titik Persimpangan

Pada gambar 8 menggambarkan pemetaan jalan yang digunakan untuk pengujian metode SAW dan Algoritma ACS. Proses pemetaan jalan digambarkan dengan *node* dan *edge*.

Tabel 1. Keterangan Pemetaan Jalan

No	Edge	Nama Jalan / Edge	Jarak (KM)	Kemacetan	Hydrant	Satu Arah
1	1 → 2	Jl. Belakang Pasar	0,1838	0,1	0	-1
2	1 → 4	Jl. Gardujati	0,1737	0,1	0	0
3	2 → 3	Jl. Belakang Pasar 1	0,1649	0,1	0	0
4	2 → 5	Jl. Durman	0,1878	0,5	0	-1
5	3 → 6	Jl. Babatan	0,2237	0,1	0	0
6	5 → 4	Jl. Ence Ajis	0,1538	0,8	0	0
7	4 → 7	Jl. Astana Anyar	0,1532	0,1	1	0
8	8 → 5	Jl. Durman 1	0,1577	0,1	0	0
9	6 → 5	Jl. Ence Ajis 1	0,1629	0,1	1	0
10	7 → 8	Jl. Jendral Sudirman	0,1566	0,1	0	0

ISSN cetak 2087-1716
ISSN online 2548-7779

Keterangan :
→ : Menuju Node

Pada Tabel 1 merupakan rincian keterangan pemetaan *node* dan *edge* yang digunakan pada pengujian. Berikut adalah penjelasan dari masing-masing kolom.

1. Kolom Jarak : Menyatakan jarak antar *node*.
2. Kolom kemacetan : Menyatakan bobot kemacetan antar *node*.
3. Kolom Hydrant : Menyatakan posisi hydrant. Jika pada kolom tersebut terdapat nilai 1, maka antara *node* tersebut terdapat hydrant. Sedangkan nilai 0 menyatakan antara *node* tersebut tidak memiliki hydrant.
4. Kolom satu arah : Menyatakan hubungan antar *node* tersebut. Nilai 0 menandakan antara *node* tersebut saling terhubung. Sedangkan nilai -1 menandakan arah dari hubungan satu arah antar *node*.

3.5.2. Perbandingan Hasil Pengujian dengan cara Manual dan Metoda SAW

Hasil perhitungan antara perhitungan dengan cara manual dan perhitungan oleh sistem pada aplikasi memiliki nilai yang sama. Namun pada sistem digunakan pembulatan angka. Dapat disimpulkan bahwa hasil pengujian perhitungan sistem telah berjalan dengan benar.

Tabel 2 Perbandingan Perhitungan Manual Dengan Sistem Metode SAW

No	Alternatif	Perhitungan Manual	Keluaran Sistem	Hasil
1	1 → 4	0.7079	0.708	√
2	2 → 1	0.7312	0.731	√
3	3 → 2	0.7548	0.755	√
4	3 → 6	0.6347	0.635	√
5	4 → 1	0.7079	0.708	√
6	4 → 5	0.5377	0.538	√
7	4 → 7	0.9872	0.987	√
8	5 → 4	0.5377	0.538	√
9	5 → 6	0.9603	0.960	√
10	5 → 2	0.4768	0.477	√
11	5 → 8	0.7747	0.775	√
12	6 → 3	0.6347	0.635	√
13	6 → 5	0.9603	0.960	√
14	7 → 4	0.9872	0.987	√
15	7 → 8	0.7779	0.778	√
16	8 → 5	0.7747	0.775	√
17	8 → 7	0.7779	0.778	√

Keterangan :

1. √ : Nilai Perhitungan Manual Dengan Sistem Memiliki Nilai Yang Sama
2. → : Menuju *Node*

Pada tabel 2 merupakan hasil perhitungan preferensi secara menual dan hasil perhitungan oleh sistem pada aplikasi.

3.5.3. Pengujian Pencarian Rute Algoritma ACS

Pada pengujian ini dilakukan perbandingan perhitungan pencarian rute secara manual dan perhitungan pencarian rute oleh sistem pada aplikasi. Paramater alogoritma ACS yang digunakan pada perhitungan manual dengan sistem memiliki nilai yang sama. Berikut adalah nilai dari setiap parameter yang digunakan adalah $q_0 = 0.4$, $\alpha = 0.1$, $\beta = 1$, $\rho = 0.05$, $m = 8$ dan $NCmax = 1$

Pada pengujian ini , nilai intensitas *pheromone* (τ_{ij}) didapatkan dari perhitungan $\tau_{ij} = 1/(n * Lnn)$. Dimana Lnn merupakan jarak antar *node* dan n adalah jumlah *node*. Nilai visibilitas (η_{ij}) antar *node* didapatkan dari hasil perhitungan oleh metode SAW yaitu nilai *preferensi* tiap *alternative*.

Untuk melakukan perbandingan pencarian rute, diperlukan hasil perhitungan manual dan hasil keluaran pada sistem. Setelah semut menyelesaikan proses pencarian, selanjutnya akan didapatkan rute yang dilalui oleh seluruh semut. Berikut adalah hasil perhitungan pada tiap rute.

ISSN cetak 2087-1716
ISSN online 2548-7779

Tabel 3. Hasil Pencarian Rute Semut Pengujian Rute 1

Semut	Rute	C1 (KM)	C2	C3	Visibilitas
1	1 → 4 → 7 → 8 → 5 → 6 → 3	1.0358	0.6	2	0,9797
2	1 → 4 → 7 → 8 → 5 → 6 → 3	1.0358	0.6	2	0,9797
3	1 → 4 → 7 → 8 → 5 → 6 → 3	1.0358	0.6	2	0,9797
4	1 → 4 → 7 → 8 → 5 → 2 → 3	0.9945	1.0	1	0.7799
5	1 → 4 → 7 → 8 → 5 → 6 → 3	1.0358	0.6	2	0,9797

Keterangan :
C1 : Panjang Rute Yang Dilalui (Kriteria Jarak).
C2 : Jumlah Bobot Kemacetan Yang Dilalui (Kriteria Kemacetan).
C3 : Banyaknya Hydrant Pada Rute Tersebut (Kriteria Hydrant).
→ : Menuju *Node*

Pada tabel 3 merupakan hasil Pencarian Rute Semut pada pengujian rute 1. Dapat diketahui bahwa hasil perhitungan manual algoritma ACS rute 1→ 4→7→ 8→ 5→6→3 dengan panjang rute 1.0358 KM memiliki nilai visibilitas terbaik. Rute dengan nilai visibilitas terbaik adalah rute optimal yang dipilih

Tabel 4. Hasil Pencarian Rute Semut Pengujian Rute 2

Semut	Rute	C1 (KM)	C2	C3	Visibilitas
1	1 → 4 → 7 → 8 → 5	0.6432	0.4	1	0,7558
2	1 → 4 → 7 → 8 → 5	0.6432	0.4	1	0,7558
3	1 → 4 → 7 → 8 → 5	0.6432	0.4	1	0,7558
4	1 → 4 → 5	0.3331	0.9	0	0,6332
5	1 → 4 → 7 → 8 → 5	0.6432	0.4	1	0,7558

Keterangan :
C1 : Panjang Rute Yang Dilalui (Kriteria Jarak).
C2 : Jumlah Bobot Kemacetan Yang Dilalui (Kriteria Kemacetan).
C3 : Banyaknya Hydrant Pada Rute Tersebut (Kriteria Hydrant).
→ : Menuju *Node*.

Pada tabel 4. merupakan hasil Pencarian Rute Semut pada pengujian rute 2. Dapat diketahui bahwa hasil perhitungan manual algoritma ACS dengan rute 1→4→7→8→5 dengan panjang rute 0.632 KM memiliki nilai visibilitas terbaik. Rute dengan nilai visibilitas terbaik adalah rute optimal yang dipilih.

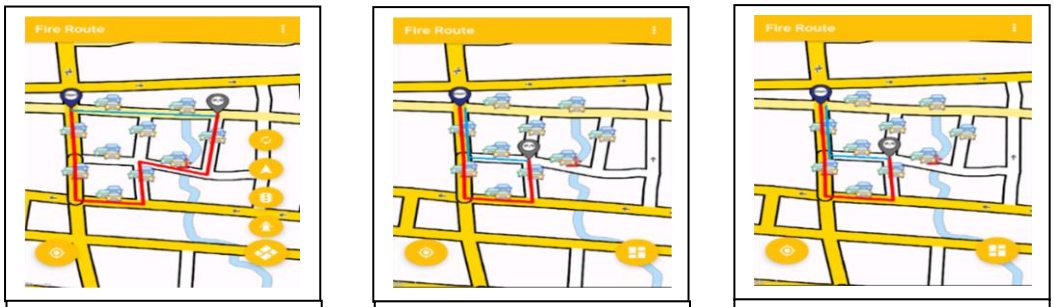
Tabel 5 Hasil Pencarian Rute Semut Pengujian Rute 3

Semut	Rute	C1 (KM)	C2	C3	Visibilitas
1	1 → 4 → 7 → 8	0.4848	0.3	1	1
2	1 → 4 → 7 → 8	0.4848	0.3	1	1
3	1 → 4 → 7 → 8	0.4848	0.3	1	1
4	1 → 4 → 5 → 6 → 3 → 2	Null	Null	Null	Null
5	1 → 4 → 7 → 8	0.4848	0.3	1	1

Keterangan :
C1 : Panjang Rute Yang Dilalui (Kriteria Jarak).
C2 : Jumlah Bobot Kemacetan Yang Dilalui (Kriteria Kemacetan).
C3 : Banyaknya Hydrant Pada Rute Tersebut (Kriteria Hydrant).
→ : Menuju *Node.*

Pada tabel 5 merupakan hasil Pencarian Rute Semut pada pengujian rute 3. Dapat diketahui bahwa hasil perhitungan manual algoritma ACS dengan rute 1→4→7→8 dengan panjang rute 0.4848 Km memiliki nilai visibilitas terbaik. Rute pada semut ke 4 tidak menemui titik tujuan, maka nilai *visibilitas* memiliki nilai *null* yang berarti kosong.

ISSN cetak 2087-1716
ISSN online 2548-7779



(1) Rute 1 (2) Rute 2 (3) Rute 3

Gambar 8. Hasil Pencarian dengan Rute 1, Rute 2 dan Rute 3 visual peta dari aplikasi

4. Kesimpulan dan saran

4.1. Kesimpulan

1. Aplikasi pencarian rute terbaik dengan menggunkan algoritma ACO dan metode SAW, dapat digunakan untuk mencari rute optimal berdasarkan kriteria tertentu (jarak, posisi hydrant, dan kemacetan)
2. Algoritma ACO dapat di implementasikan dalam penentuan rute optimal pada perangkat *smartphone* berbasis *Android*.
3. Hasil Pencarian Rute Semut pada beberapa pengujian rute, dapat diketahui bahwa hasil perhitungan algoritma ACS untuk rute dan panjang rute yang telah diukur memiliki nilai visibilitas terbaik dan rute dengan nilai visibilitas terbaik adalah rute optimal yang dipilih

4.2. Saran

1. Perlu dilakukan pengecekan kembali lajur dan jalur jalan yang ada sehingga didapatkan hasil pencarian rute yang akurat
2. Perlu ditambahan kriteria lain sungai, jalan alternatif dan kepadatan perumahan penduduk guna mendapatkan hasil yang optimal.

Daftar Pustaka

[1] Yuliyani Siyamtining Tyas dan Widodo Prijodiprodjo. (2013). Aplikasi Pencarian Rute Terbaik dengan Metode Ant Colony Optimazation (ACO). IJCCS, Vol.7, No.1, January 2013, pp. 55~64

[2] Dorigo, M., Maniezzo, V. dan Colorni, A. (1996). *Solving Symetric and Asymetric TSPs by Ant Colonies*. IEEE Transaction on Systems, Mana and Cybernetics .

[3] Dorigo, M., dan Gambardela, Gambardella, L.M., (1997), *Ant Colony System:A CooperativeLearning Approach to the Traveling Salesman Problem*, IEEE Transactions on evolutionary computation, vol 1, no 1.

[4] Dorigo, M dan Gambardella, L.M., (1997), *Ant Colonies for the travelling Salesman Problem*, Bio System 43.

[5] Fishburn, P.C., Additive Utilities With Incomplete Product Set: Applications to Priorities and Assigments, Operations Research.

[6] Janko, W., (2005), Multi-Criteria Decision Making: An Aplication Study of ELECTRE & TOPSIS.

[7] Zimmermann, 1991, Fuzzy Sets Theory and Its Applications. Edisi 2. Kluwer Academic Publishers. Massachusetts

[8] Kusumadewi, Sri, Sri Hartati, Agus Harjoko dan Retantyo Wardoyo. (2006), *Fuzzy Multi-Attribute Decision Making (Fuzzy MADM)* ,Graha Ilmu, Yogyakarta.

[9] Ahmed Al-Ani. (2006) "*Feature Subset Selection using Ant Colony Optimization". International Journal of Computational Intelligence*.

[10] Andrea Roli. (2002) "*Ant Colony Optimization*". Aironews Vol.7 no.3 (Pages1-3).

[11] Agus Leksono.(2009) *"Algoritma Ant Colony Optimization (ACO) Untuk Menyelesaikan Traveling Salesman Problem (TSP)"*. Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Diponegro

