



Pencarian Jalur berdasarkan Kepadatan Lalu Lintas Menggunakan Algoritma Koloni Semut

Arna Fariza¹, Arif Basofi², Mochammad Rizki Hidayat³

^{1,2,3}Departemen Teknik Informatika dan Komputer, Politeknik Elektronika Negeri Surabaya

¹arna@pens.ac.id, ²ariv@pens.ac.id

Abstract

Surabaya as a big city with a high traffic density has a complicated problem to solve. The number of road users are increasing, causes higher density. In a dynamic traffic system, the path search based on the shortest distance must take the condition of the density of traffic at a time. Road users are more effective and efficient in determining the path to pass, when considering the density factor. In this research, the development of path searching uses the ant colony algorithm. The ant colony algorithm is applied to the map to find the optimal path based on distance by considering the condition of traffic density at a time. The system built can be a tool for road users to choose the road based on information on density conditions, road directions and paths that can be passed by certain types of vehicles. Based on the experiment results on several source and destination points, the ant colony algorithm managed to search the path well according to the weight of the number of path and the type of vehicle.

Keywords: path searching, traffic density, ant colony algorithm.

Abstrak

Di kota besar seperti Surabaya yang mempunyai kepadatan lalu lintas yang cukup tinggi, merupakan permasalahan yang rumit untuk diselesaikan. Peningkatan jumlah pengguna jalan menyebabkan kepadatan yang semakin tinggi. Pada sistem lalu lintas yang bersifat dinamis, pencarian jalur berdasarkan jarak terpendek harus memperhitungkan kondisi kepadatan lalu lintas pada suatu waktu. Pengguna jalan lebih efektif dan efisien dalam menentukan jalur yang dilewati, apabila mempertimbangkan faktor kepadatan. Pada penelitian ini dilakukan pengembangan pencarian jalur menggunakan algoritma koloni semut (*ant colony algorithm*). Algoritma koloni semut diaplikasikan pada peta untuk menemukan jalur optimal berdasarkan jarak tempuh dengan mempertimbangkan kondisi kepadatan lalu lintas pada suatu waktu. Sistem yang dibangun dapat menjadi alat bantu bagi pengguna jalan untuk memilih jalan yang dilakukan berdasarkan informasi kondisi kepadatan, arah jalan dan serta jalur yang dapat dilalui oleh jenis kendaraan tertentu. Berdasarkan hasil percobaan terhadap beberapa titik asal dan tujuan, algoritma koloni semut berhasil melakukan pencarian jalur dengan baik sesuai dengan bobot jumlah lajur jalan dan jenis kendaraan.

Kata kunci: pencarian jalur, kepadatan lalu lintas, algoritma koloni semut.

1. Pendahuluan

Permasalahan kepadatan lalu lintas di kota besar seperti Surabaya menjadi persoalan yang rumit dengan semakin banyaknya pengguna jalan menggunakan berbagai moda transportasi. Kemacetan menjadi hal yang selalu terjadi terutama di jalur utama, terutama pada jam-jam sibuk seperti saat berangkat kerja di pagi hari dan saat pulang kerja di sore hari. Selain itu, kurang tertibnya pengguna kendaraan, perbaikan sarana umum, kepadatan penduduk sekitar dan kecelakaan juga berpotensi menyebabkan kemacetan.

Aplikasi pencarian jalur yang efektif dan efisien menjadi sangat penting bagi pengguna jalan. Pengguna menginginkan aplikasi yang secara efektif dapat menghindari kemacetan sehingga waktu yang dibutuhkan sampai pada tujuan se-minimal mungkin.

Aplikasi juga diharapkan secara efisiensi menghindari jalur kemacetan untuk menghemat biaya transportasi.

Masalah pencarian jalur terpendek berkaitan dengan alur jaringan yang mencari biaya minimum dari titik sumber ke titik tujuan dalam jaringan[1]. Masalah jalur terpendek dalam jaringan lalu lintas adalah elemen kunci dalam sistem lalu lintas cerdas[2]. Pencarian jalur adalah masalah optimisasi kombinatorial klasik, yang menarik lebih banyak perhatian dari para peneliti karena meningkatnya tingkat kemacetan lalu lintas, terutama di daerah perkotaan.

Masalah jalur terpendek dapat diklasifikasikan ke dalam masalah jalur terpendek statis atau dinamis sesuai karakteristik jaringan[2]. Algoritma Dijkstra[3] dan algoritma Floyd[4] adalah algoritma jalur terpendek statis yang terkenal. Namun, meskipun algoritma klasik ini efektif dalam sistem statis, algoritma tersebut tidak



efisien untuk menentukan jalur terpendek dalam jaringan dinamis[2]. Hal ini karena dalam jaringan dinamis, sub-jalur yang diperoleh adalah jalur terpendek pada suatu waktu dan mungkin bukan jalur terpendek di waktu yang lain. Dalam jaringan lalu lintas, masalah jalur terpendek selalu dinamis[2]. Kondisi lalu lintas selalu berubah waktu (misalnya, beberapa ruas jalan mungkin lebih ramai daripada biasanya selama periode jam sibuk), sehingga pengguna jalan mungkin perlu mengubah jalur terpendek yang telah direncanakan sebelumnya karena perubahan kondisi jalan pada waktu tersebut. Oleh karena itu, masalah jalur terpendek yang dinamis adalah menemukan jalur yang optimal sesuai dengan kondisi lalu lintas dalam waktu nyata.

Algoritma optimisasi heuristik atau cerdas saat ini banyak dikembangkan untuk memecahkan masalah jalur terpendek dinamis dalam jaringan yang kompleks, misalnya dengan menggunakan algoritma genetika[5] atau algoritma koloni semut[6,7]. Algoritma heuristic menunjukkan kinerja yang sangat baik di dalam masalah optimasi yang kompleks.

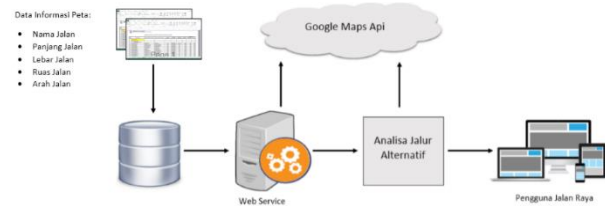
Algoritma koloni semut merupakan metode heuristik yang mengambil perilaku koloni semut dalam pencarian jalur terpendek antara sarang dan sumber makanan yang dikenal sebagai sistem semut[8]. Secara alamiah koloni semut mampu menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilalui. Semakin banyak semut yang melewati suatu lintasan maka akan semakin jelas bekas jejak kakinya, sebaliknya lintasan yang dilalui semut dalam jumlah sedikit semakin lama akan semakin berkurang atau bahkan akan tidak dilewati sama sekali. Prinsip algoritma yang didasarkan pada perilaku koloni semut dalam menemukan jarak perjalanan paling pendek tersebut menunjukkan algoritma koloni sangat tepat digunakan untuk menentukan jalur terpendek.

Pada sistem lalu lintas yang bersifat dinamis, pencarian jalur berdasarkan jarak terpendek harus memperhitungkan kondisi kepadatan lalu lintas pada suatu waktu. Pengguna jalan lebih efektif dan efisien dalam menentukan jalur yang dilewati, apabila mempertimbangkan faktor kepadatan. Pada penelitian ini dilakukan pengembangan pencarian jalur menggunakan algoritma koloni semut. Algoritma koloni semut diaplikasikan pada peta untuk menemukan jalur optimal berdasarkan jarak tempuh dengan mempertimbangkan kondisi kepadatan lalu lintas pada suatu waktu. Sistem yang dibangun dapat menjadi alat bantu bagi pengguna jalan untuk memilih jalan yang dilakukan berdasarkan informasi kondisi kepadatan, arah jalan dan serta jalur yang dapat dilalui oleh jenis kendaraan tertentu.

2. Metode Penelitian

Pada penelitian ini dibangun sebuah sistem *client server* berbasis web. Server basis data menyimpan data

informasi nama jalan, panjang jalan, ruas jalan dan arah jalan yang terhubung dengan Google Maps Api sebagai visualisasi tampilan peta. Pengguna jalan memberikan informasi kepadatan jalan yang disimpan dalam basis data. Server melakukan pemrosesan algoritma koloni semut untuk sebagai pencari jalur berdasarkan data pada basis data dan informasi kepadatan jalan. Selain itu sistem ini dapat melakukan analisa berdasarkan kendaraan yang digunakan oleh pengguna jalan. Diagram sistem pencarian jalur dengan algoritma koloni semut dapat dilihat pada Gambar 1.

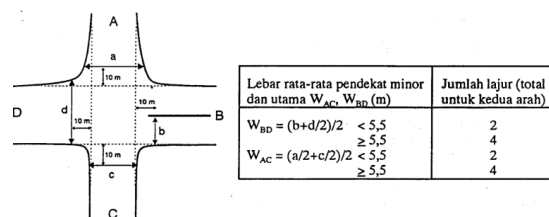


Gambar 1. Diagram sistem pencarian jalur dengan algoritma koloni semut

2.1. Pemodelan Data

Penelitian ini menggunakan data dari Dinas PU Bina Marga dan Pematusan tentang survey jalan di Kota Surabaya. Pada data tersebut terdapat informasi nama jalan, panjang jalan, lebar jalan (baik menggunakan perkerasan jalan maupun tanpa perkerasan jalan). Setiap jalan memiliki pangkal dan ujung jalan yang merupakan informasi jalan yang terhubung dalam jalan tersebut. Pada setiap jalan dalam data juga memiliki klasifikasi fungsi jalan yang memberikan informasi bahwa setiap jalan memiliki golongan dan fungsi masing-masing. Klasifikasi tersebut menggambarkan fungsi jalan pada penggunaannya sesuai dengan bobot kendaraan yang melintasi jalan tersebut.

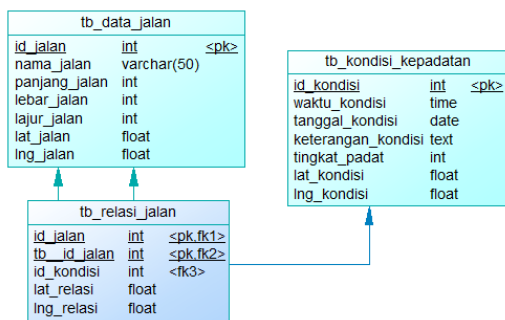
Aplikasi ini juga membutuhkan data arus jalan, lajur jalan dan arah dengan cara melalui pengamatan langsung di lapangan berdasarkan beberapa sampel jalan. Untuk menentukan status jalan untuk kendaraan yang dapat melalui sebuah lajur diperlukan data jumlah lajur yang digunakan sebagai fitur rute jalan. Jumlah lajur jalan pada didapatkan oleh peneliti melalui studi literatur pada Manual Kapasitas Jalan Indonesia (MKJI)[9]. Pada MKJI penentuan jumlah lajur jalan ditentukan dengan menghitung lebar jalan seperti Gambar 2. Jumlah lajur jalan dihitung berdasarkan kondisi jika lebar jalan lebih atau sama dengan dari 5,5 meter maka jumlah lajur jalan diatas 2 maupun lebih. Sedangkan jika lebar jalan kurang dari 5,5 meter maka jumlah lajur adalah dibawah 2.



Gambar 2. Menentukan jumlah lajur[9]

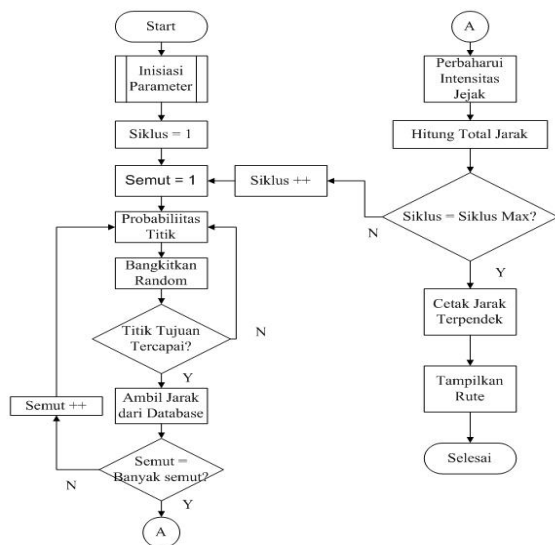
Setelah didapat nama jalan, panjang, lebar, lajur, dan jalan yang terhubung maka ditambahkan data pemetaan berupa lintang (*latitude*) dan bujur (*longitude*) sebagai penentu vertek dan garis dari struktur data graf yang digunakan.

Model data fisik yang digunakan sistem dapat dilihat pada Gambar 3. Basis data terdiri dari tabel data jalan, tabel kondisi kepadatan dan tabel relasi jalan. Tabel Data Jalan memiliki informasi data jalan yang telah di olah dari data jalan Kota Surabaya. Tabel Kondisi Kepadatan menyimpan informasi data kepadatan yang telah dilaporkan oleh pengguna jalan. Sedangkan Tabel Relasi Jalan merupakan relasi dari Tabel Jalan yang melakukan proses rekursif untuk membentuk entitas baru.



Gambar 3. Model data fisik

2.2. Desain Algoritma Koloni Semut



Gambar 4 Desain algoritma koloni semut

Algoritma koloni semut (*ant colony algorithm*) merupakan algoritma metaheuristik. Koloni semut mampu menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilalui. Semakin banyak semut yang melewati suatu lintasan maka akan semakin jelas bekas jejak kakinya, sebaliknya lintasan yang dilalui semut dalam jumlah sedikit semakin lama akan semakin berkurang atau bahkan akan tidak

dilewati sama sekali. Desain algoritma semut yang digunakan pada penelitian ini dapat dilihat pada Gambar 4.

Tahapan dalam mencari jalur berdasarkan kepadatan lalu lintas adalah sebagai berikut :

1. Inisiasi parameter adalah mengisi parameter yang dibutuhkan oleh koloni semut.
 - Intensitas jejak kaki semut dan perubahan (τ_{ij})
 - Banyak titik (n) termasuk koordinat (x, y)
 - Banyak semut (m)
 - Tetapan siklus-semut (Q)
 - Tetapan pengendalian intensitas jejak semut (α)
 - Tetapan pengendalian visibilitas (β)
 - Tetapan penguapan jejak semut (ρ)
 - Menghitung visibilitas antar titik ($\eta_{ij} = \frac{1}{d_{ij}}$)
2. Perhitungan probabilitas adalah bagian dari penyusunan rute kunjungan setiap semut ke setiap titik. Perhitungan probabilitas digunakan sebagai menentukan titik tujuan kunjungan semut. Namun kondisi opsional ada akan di jalankan. Jika dalam jalur yang terhubung mengalami hambatan maka akan memilih opsi jalur lain.

$$\text{Probabilitas } P_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{tabu_k}^n [\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta} \quad (1)$$

3. Perhitungan panjang rute tertutup (L_k) dilakukan sebagai penghitungan akhir dari setiap siklus semut. Perhitungan ini bertujuan untuk menentukan rute terpendek dengan menentukan harga yang paling minimal dari panjang rute.

$$L_k = dtabu_k(n), tabu_k(1) + \sum_{s=1}^{n-1} dtabu_k(s), tabu_k(s + 1) \quad (2)$$

4. Perubahan perubahan harga intensitas jejak kaki semut untuk perulangan semut selanjutnya. Adanya perbedaan yang diakibatkan penguapan dan jumlah semut yang lewat dan adanya kemungkinan perubahan harga intensitas jejak kaki semut antar titik.

Perubahan jejak kaki dihitung dengan persamaan

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

dengan cara mencari $\Delta\tau_{ij}^k = \frac{Q}{L_k}$ pada titik asal dan titik tujuan $\Delta\tau_{ij}^k = 0$. Sedangkan nilai intensitas selanjutnya dihitung dengan persamaan

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta\tau_{ij} \quad (4)$$

3. Hasil dan Pembahasan

3.1. Pencarian Jalur Berdasarkan Jarak Tempuh

Pada uji coba pencarian jalur berdasarkan jarak tempuh, digunakan parameter sebagai berikut :

- Jalan asal : Raya ITS,
- Percobaan : satu kali pencarian, dan
- Kendaraan yang digunakan : Sepeda Motor (MC).

Tabel 1 merupakan hasil pencarian jalur berdasarkan jarak tempuh terhadap 2 sampai dengan >10 jalur. Tabel 1 menunjukkan bahwa dalam satu kali pencarian jalur masih ada pencarian yang belum berhasil. Hal ini dikarenakan penggunaan algoritma koloni semut memiliki pencarian probabilitas yang sangat cocok apabila pencarian jalur dengan bobot yang di tentukan. Namun, pada pengaplikasiannya, algoritma ini masih menggunakan random yang memungkinkan jalur tidak berhasil dicari.

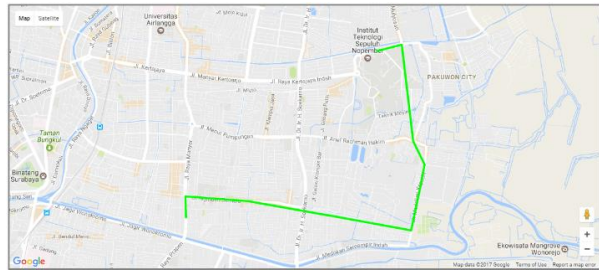
Tabel 1 Hasil pengujian mencari jalur terhadap jarak tempuh

No	Jumlah titik jalan	Rute yang dilalui	Hasil
1	2 jalan	Raya ITS - Raya Kertajaya Indah	Berhasil
2	3 jalan	Raya ITS - Raya Kertajaya Indah - Manyar Kertoarjo	Berhasil
3	4 jalan	Raya ITS - Raya Kertajaya Indah - Manyar Kertoarjo - Kertajaya	Berhasil
4	5 jalan	Raya ITS - Raya Kertajaya Indah - Dharma Husada Indah - Prof. Dr. Mustopo - Karang Menjangan	Berhasil
5	6 jalan	Raya ITS - Raya Kertajaya Indah - Manyar Kertoarjo - Kertajaya - Sulawesi - Raya Gubeng	Berhasil
6	7 jalan	Raya ITS - Kejawan Gebang - Kejawan Keputih - Arief Rachmad Hakim - Menur Pumpungan - Raya Manyar - Bratang Jaya	Berhasil
7	8 jalan	Raya ITS - Raya Kertajaya Indah - Manyar Kertoarjo - Kertajaya - Pucang anom Timur - Ngagel Jaya Timur - Ngagel Jaya Selatan - Raya Ngagel	Berhasil
8	9 jalan	Raya ITS - Raya Kertajaya Indah - Manyar Kertoarjo - Raya Menur - Karang menjangan - Prof. Dr. Mustopo - Gerbong - Indrakila - Pacar Keling	Tidak berhasil
9	10 jalan	Raya ITS - Kejawan Keputih tambak - Kejawan Gebang - Arief Rachman Hakim - Menur Pumpungan - Raya Manyar - Raya nginden - Panjang Jiwo - kedung Baruk	Tidak Berhasil
10	>10 jalan	Raya ITS - Kejawan Keputih tambak - Kejawan Gebang - Arief Rachman Hakim - Menur Pumpungan - Raya Manyar - Raya nginden - Panjang Jiwo - kedung Baruk - kedung asem	Berhasil

3.1. Pengujian Jalur Berdasarkan Kondisi Kepadatan Pada pengujian pencarian jalur berdasarkan kondisi kepadatan digunakan parameter sebagai berikut:

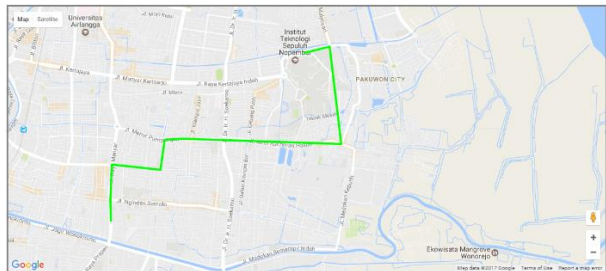
- Jalan asal : Raya ITS
- Jalan tujuan : Raya Nginden
- Kendaraan yang digunakan : mobil (LV)
- Beban kondisi jalan diberikan pada Raya Kertajaya Indah, Manyar Kertoarjo, Raya Menur, dan Raya Manyar.

Hasil pencarian jalur dari Raya ITS menuju ke Jalan Raya Nginden dengan pemberian bobot kondisi jalan di Raya Kertajaya Indah dapat dilihat pada Gambar 5. Hasil pencarian jalur ditandai dengan garis berwarna hijau dengan rute Raya ITS - Kejawan Keputih - Keputih Tambak - Keputih Tegal - Medokan Keputih - Semolowaru - Nginden Semolo - Raya Nginden..



Gambar 5 Hasil pencarian jalur dari Raya ITS ke Raya Nginden dengan pembobotan di jalan Raya Kertajaya Indah.

Hasil dari pencarian jalur dari Raya ITS menuju ke Jalan Raya Nginden dengan pembobotan kepadatan Jalan Kertajaya Indah dan Manyar Kertoarjo dapat dilihat pada Gambar 6. Hasil pencarian jalur ditandai dengan garis berwarna hijau dengan rute Raya ITS - Kejawan Keputih - Keputih Tambak - Arief Rachman Hakim - Menur Pumpungan - Manyar Raya Indah - Manyar Rejo - Raya Manyar - Raya Nginden.

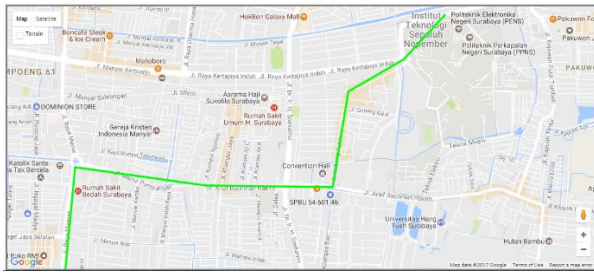


Gambar 6 Hasil pencarian jalur dari Raya ITS ke Raya Nginden dengan pembobotan di jalan Raya Kertajaya Indah dan Manyar Kertoarjo.

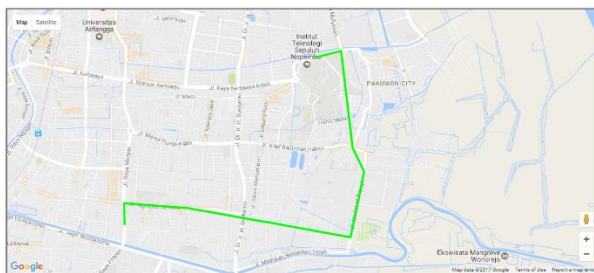
Hasil dari pencarian jalur dari Raya ITS menuju ke Jalan Raya Nginden dengan pembobotan kepadatan Jalan Kertajaya Indah, Manyar Kertoarjo dan Raya Menur dapat dilihat pada Gambar 7. Hasil pencarian jalur ditandai dengan garis berwarna hijau dengan rute Raya ITS - Gebang Lor - Gebang Putih - Arief Rachman Hakim - Menur Pumpungan - Manyar Raya - Raya Nginden.

Hasil dari pencarian jalur dari Raya ITS menuju ke Jalan Raya Nginden dengan pembobotan kepadatan Jalan Kertajaya Indah, Manyar Kertoarjo, Raya Menur dan Raya Manyar dapat dilihat pada Gambar 8. Hasil

pencaharian jalur ditandai dengan garis berwarna hijau dengan rute Raya ITS - Kejawan Keputih - Keputih Tambak - Keputih Tegal - Medokan Keputih - Semolowaru - Nginden Semolo - Raya Nginden.



Gambar 7 Hasil pencarian jalur dari Raya ITS ke Raya Nginden dengan pembobotan di jalan Raya Kertajaya Indah, Manyar Kertoarjo dan Raya Menur



Gambar 8 Hasil pencarian jalur dari Raya ITS ke Raya Nginden dengan pembobotan di jalan Raya Kertajaya Indah, Manyar Kertoarjo, Raya Menur dan Raya Manyar

Pemberian bobot kondisi kepadatan dilakukan pada beberapa titik seperti Tabel 2.

Tabel 2 Hasil pengujian pencarian jalur terhadap kondisi kepadatan

No	Asal	Tujuan	Bobot	Jalur alternatif yang disarankan
1	Raya ITS	Raya Nginden	- Raya Kertajaya Indah	Raya ITS - Kejawan Keputih - Keputih Tambak - Keputih Tegal - Medokan Keputih - Semolowaru - Nginden Semolo - Raya Nginden
2	Raya ITS	Raya Nginden	- Raya Kertajaya Indah - Manyar Kertoarjo	Raya ITS - Kejawan Keputih - Keputih Tambak - Arief Rachman Hakim - Menur Pumpungan - Manyar Raya Indah - Manyar Rejo - Raya Manyar - Raya Nginden
3	Raya ITS	Raya Nginden	- Kertajaya Indah - Manyar Kertoarjo - Raya Menur	Raya ITS - Gebang Lor - Gebang Putih - Arief Rachman Hakim - Menur Pumpungan - Manyar Raya - Raya Nginden
4	Raya ITS	Raya Nginden	- Raya Kertajaya Indah - Manyar Kertoarjo	Raya ITS - Kejawan Keputih - Keputih Tambak - Keputih Tegal - Medokan Keputih - Semolowaru -

- Raya Nginden Semolo -
- Raya Menur Raya Nginden
- Raya Manyar

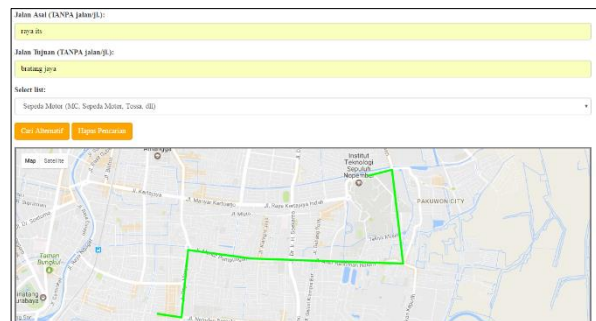
Tabel 2 menunjukkan dalam pencarian jalur, dapat mencari jalur dari jalur utama yang telah dipilih sebelumnya. Pada setiap percobaan pengujian dalam pembobotan memberikan hasil yang berbeda. Hal ini dikarenakan letak dari setiap kondisi kepadatan jalan yang berbeda dan algoritma koloni semut yang berjalan memberikan informasi jalan yang berbeda pula.

3.1. Pencarian Jalur Terhadap Jenis Kendaraan

Pada pengujian jalur terhadap jenis kendaraan, dilakukan pengujian menggunakan sepeda motor (MC), mobil (LV), dan truk (HV). Parameter yang digunakan adalah sebagai berikut:

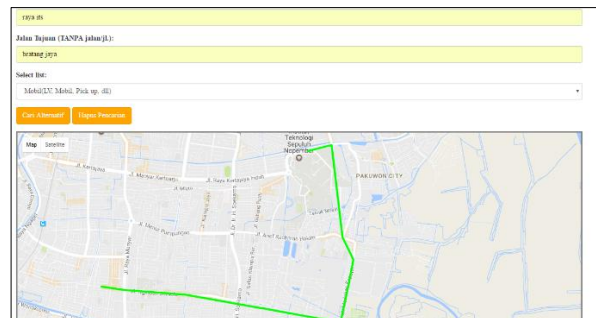
- Jalan asal : Raya ITS
- Jalan Tujuan : Bratang Jaya

Hasil pengujian rute terhadap jenis kendaraan dengan menggunakan sepeda motor (MC) dapat dilihat pada Gambar 9. Pada pencarian jalur dengan menggunakan kendaraan sepeda motor bahwa pencarian tersebut lebih memilih menggunakan rute yang memiliki lajur jalan baik kecil yakni 1 lajur, sedang 2-3 lajur, maupun lebar yakni lebih dari 3 lajur.



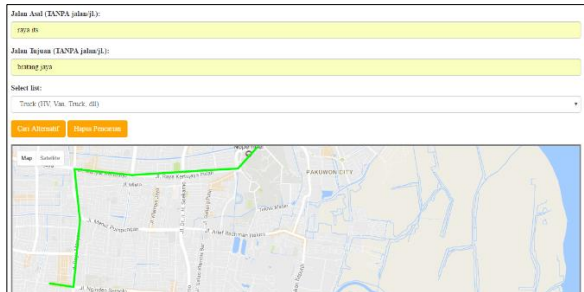
Gambar 9 Hasil pencarian jalur menggunakan sepeda motor

Hasil pengujian rute terhadap jenis kendaraan dengan menggunakan mobil (LV), dapat dilihat pada Gambar 10. Pada pencarian jalur dengan menggunakan kendaraan mobil bahwa pencarian tersebut lebih memilih menggunakan rute yang memiliki lajur jalan baik 2 lajur maupun lebih.



Gambar 10 Hasil pencarian jalur menggunakan mobil

Hasil pengujian rute terhadap jenis kendaraan dengan menggunakan truk (HV). dapat dilihat pada Gambar 11. Pada pencarian jalur dengan menggunakan kendaraan truk bahwa pencarian tersebut lebih memilih menggunakan rute yang memiliki memiliki lajur yang lebar yakni 3 lajur maupun lebih.



Gambar 11 Hasil pencarian jalur menggunakan truk

Aplikasi pencarian jalur yang dibangun ini menggunakan beberapa parameter yang sama dengan Google Maps, yaitu panjang jalan, jumlah jalur, dan jenis kendaraan. Kondisi jalan digunakan pada aplikasi ini berbeda dengan parameter Google Maps, yaitu pada aplikasi ini bobot kondisi jalan ditentukan berdasarkan waktu, sedangkan pada Google Maps, kondisi jalan berdasarkan data lalu lintas *real time* / prakiraan. Sehingga Google Maps memberikan informasi yang lebih riil dari kepadatan lalu lintas, namun informasi yang ditunjukkan tidak sepenuhnya benar karena informasi kondisi lalu lintas tersebut ditampilkan oleh Google berdasarkan banyaknya pemakai smartphone dalam suatu wilayah tertentu yang membutuhkan koneksi internet. Sedangkan aplikasi yang dikembangkan ini menyimpan data kondisi jalan pada jam-jam tertentu berdasarkan survey sehingga proses pencarian jalur tidak membutuhkan koneksi internet.

Algoritma pencarian jalur yang digunakan oleh Google Maps adalah algoritma Dijkstra yang merupakan algoritma *Greedy*. Algoritma *greedy* merupakan algoritma intuitif dan sederhana yang digunakan dalam masalah optimasi. Algoritma ini membuat pilihan optimal pada setiap langkah karena berusaha menemukan cara optimal keseluruhan untuk menyelesaikan seluruh masalah. Namun, dalam banyak kasus, strategi *greedy* tidak menghasilkan solusi yang [10]

optimal, karena membuat keputusan hanya berdasarkan informasi yang dimilikinya pada satu langkah, tanpa memperhatikan masalah keseluruhan. Sedangkan algoritma koloni semut merupakan metode heuristik yang mampu melakukan pencarian jalur terpendek dengan dengan beberapa parameter dengan optimal dan lebih cepat..

4. Kesimpulan

Sistem yang dibangun dapat memberikan informasi pencarian jalur dalam menghindari kepadatan dengan melihat dari kondisi kepadatan, arah jalan, dan lajur yang dapat dilalui. Algoritma koloni semut berhasil melakukan pencarian jalur dengan baik sesuai dengan bobot jumlah lajur jalan dan jenis kendaraan.

Selanjutnya penelitian ini akan dikembangkan dengan melakukan optimasi inialisasi awal algoritma koloni semut untuk mendapatkan rute yang lebih optimal.

Daftar Rujukan

- [1] Ghoseiri, K., and Nadjari, B., An ant colony optimization algorithm for the bi-objective shortest path problem, *Applied soft computing*, 10(4), pp. 1237-1246. doi: <https://doi.org/10.1016/j.asoc.2009.09.014>, 2010.
- [2] Zhang, S., and Zhang, Y., A hybrid genetic and ant colony algorithm for finding the shortest path in dynamic traffic networks, *Automatic Control and Computer Sciences*, 52(1), pp. 67-76. doi: <https://doi.org/10.3103/S014641161801008X>, 2018.
- [3] Dijkstra, E. W., A note on two problems in connection with graphs, *Numer. Math.*, 1(1) , pp. 269–271, 1959.
- [4] Floyd, R. W., Algorithm 97: Shortest path, *Commun. ACM.*, 5(6), p.345, 1959.
- [5] Hanshar, F.T. and Ombuki-Berman, B.M., 2007, Dynamic vehicle routing using genetic algorithms, *Appl. Intell.*, 27(1), pp. 89–99.
- [6] Indrato, I., & Hidayat, T., Pencarian Jalur Terpendek Menggunakan Algoritma Semut. *Proc. Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*. 16 Jun. 2007, Yogyakarta:Indonesia, 2007.
- [7] Lissovoi, A. and Witt, C., Runtime analysis of ant colony optimization on dynamic shortest path problems, *Theor. Comput. Sci.*, 561, pp. 73–85, 2015.
- [8] Dorigo, M., The Ant System: Optimization by a colony of cooperating agents, *IEEE transactions on Systems, Man, and Cybernetics–Part B*, 26(1), pp. 29-41. doi: 10.1109/3477.484436, 1996.
- [9] Direktorat Jenderal Bina Marga Direktorat Bina Jalan Kota (Bnkot), 1997, Manual Kapasitas Jalan Indonesia, diakses di <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbmxtYXRlcmlzaXBpbHNibWVzdGVyNHxneDoyY2VjYmQ2ZWQwYzRlNjll> pada tanggal 31 Mei 2020.