

Implementasi Algoritma Dijkstra dan Algoritma Semut Untuk Analisis Rute Transjogja Berbasis Android

Sunardi¹, Anton Yudhana², Ahmad Azhar Kadim³

Jurusan Teknik Elektro, Fakultas Teknik Industri, Universitas Ahmad Dahlan^{1,2}

Teknik Informatika, Universitas Ahmad Dahlan³

sunardi@mti.uad.ac.id¹, eyudhana@ee.uad.ac.id², azharkadim@yahoo.com³

Article Info

History :

Dikirim 28 Desember 2018

Direvisi 09 Februari 2019

Diterima 12 April 2019

Kata Kunci :

Algoritma Dijkstra

Algoritma Semut

RAM

TransJogja

Yogyakarta

Abstrak

Tujuan dari penelitian ini adalah untuk membandingkan kinerja dari algoritma Dijkstra dan Algoritma Semut (*Ant Colony Optimization*) yang dimana kedua Algoritma tersebut dapat digunakan untuk menentukan jalur terpendek dan tercepat. Algoritma Dijkstra memiliki rata-rata waktu eksekusi tercepat 0,0092 detik sedangkan Algoritma Semut adalah 0,0143 detik, untuk penggunaan RAM Algoritma Dijkstra menggunakan RAM lebih sedikit dibandingkan Algoritma Semut. jalur output dan jarak yang dihasilkan adalah relatif sama sehingga dapat diimplementasikan untuk penentuan rute angkutan umum. Aplikasi penelitian ini akan diterapkan ke TransJogja yang merupakan salah satu transportasi umum di Yogyakarta.

© This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Koresponden:

Ahmad Azhar Kadim,

Program Studi Magister Teknik Informatika

Universitas Ahmad Dahlan,

Jl. Prof. DR. Soepomo Sh, Warungboto, Umbulharjo, Yogyakarta, Indonesia, 55164

Email : azharkadim@yahoo.com

1. PENDAHULUAN

Transportasi umum merupakan salah satu pilihan masyarakat untuk bepergian dari satu tempat ke tempat lainnya. Transportasi umum dapat menjadi solusi untuk mengurangi kemacetan akibat jalan umum yang tidak dapat menampung lagi volume kendaraan. Banyak cara yang ditempuh oleh pemerintah dari mulai membuat peraturan ganjil-genap dan menggunakan kendaraan umum pada hari-hari tertentu. Langkah tersebut belum sesuai harapan diinginkan. Salah satu transportasi umum saat ini yang sering digunakan adalah bus, karena dapat menampung penumpang lebih banyak, contohnya TransJogja.

TransJogja merupakan salah satu transportasi umum yang saat ini diterapkan untuk mengurangi kemacetan. Bus TransJogja sudah menjadi pilihan masyarakat karena biayanya yang relatif murah dibandingkan menggunakan transportasi umum lainnya seperti taxi, taxi online, ojek pangkalan maupun ojek online [1]. TransJogja memiliki beberapa titik halte, namun kurang informatif mengenai posisi halte dan jalur bus TransJogja. Informasi lengkap mengenai TransJogja hanya ada pada halte-halte tertentu. Hal tersebut dapat mengurangi minat penumpang untuk menggunakan TransJogja. TransJogja harus memikirkan cara agar minat pengguna dapat bertambah seperti memadukan TransJogja dengan kemajuan teknologi.

Perkembangan teknologi saat ini berkembang dengan sangat cepat karena kemudahan dalam pengembangan dan terjangkaunya alat-alat yang dibutuhkan. Salah satunya teknologi dalam bidang Mobile seperti smartphone dapat membantu aktifitas manusia untuk berkomunikasi dan urusan pekerjaan [2]. Kemajuan teknologi tersebut dapat dijadikan solusi untuk menanggulangi masalah yang ada pada TransJogja seperti penumpang dapat mengakses dengan mudah informasi secara detail, cepat dan dimana saja. Selain itu dapat memanfaatkannya untuk mengetahui posisi sehingga dapat diprediksi waktu kedatangan bus.

Algoritma Dijkstra merupakan algoritma yang digunakan untuk menentukan jalur terpendek dan tercepat dari satu titik ke titik lainnya dengan cara menghitung segala kemungkinan jalur yang akan dilalui [3]. Algoritma Dijkstra saat ini merupakan algoritma yang dipilih developer aplikasi yang memiliki fitur Navigasi. Algoritma ini diharapkan dapat menentukan rute yang tepat untuk pengguna TransJogja sehingga mengurangi masalah dan dapat menambah minat untuk menggunakan transportasi umum.

Penelitian berjudul “Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra” melibatkan beberapa pertimbangan utama meliputi rute kendaraan dan meminimalisir biaya distribusi, serta dapat memperluas wilayah pengambilan sampah dengan armada yang terbatas [4].

Studi “Finding the Shortest Paths Among Cities in Java Island Using Node Combination Based on Dijkstra Algorithm” mendapatkan jalur terpendek antara kota-kota yang ada di pulau Jawa [5].

Penelitian “Algoritma Dijkstra untuk pemetaan dan menentukan jalur terpendek pariwisata yang ada di Timor Leste berbasis Web” mendapatkan nilai keakuratan jarak rata-rata 0,03%. Hasil pengukuran berupa rute dan waktu tempuh dengan kecepatan rata-rata kendaraan yang bervariasi [6].

Algoritma *Ant Colony Optimization* adalah suatu Algoritma yang dirancang oleh Urszula Boryczka [7]. Algoritma *Ant Colony Optimization* diadopsi dari cara perilaku semut yang dapat menentukan rute terpendek dalam perjalanan dari sarang ke sumber makanan berdasarkan jejak feromon pada lintasan yang telah dilalui.

Perbedaan penelitian ini dengan penelitian sebelumnya adalah memanfaatkan algoritma Dijkstra dan Algoritma Semut sebagai pencari rute Transjogja dan Android sebagai medianya. Pemanfaatan android digunakan sebagai media untuk menampilkan hasil informasi rute yang sudah diproses menggunakan Algoritma serta dengan memanfaatkan fitur GIS diharapkan dapat memberikan informasi kepada pengguna mengenai posisi dari halte dan bus.

2. METODE PENELITIAN

Penelitian ini menggabungkan teknologi Mobile untuk membantu menunjang kualitas pelayanan TransJogja. Penelitian akan membuat aplikasi Mobile berbasis Android yang dapat menampung semua informasi mengenai TransJogja seperti posisi halte, bus dan rute. Adapun untuk ketepatan penentuan rute dari TransJogja menggunakan Algoritma Dijkstra.

Bahan penelitian yang dibutuhkan adalah informasi mengenai TransJogja seperti titik-titik halte yang ada serta rute yang dilewati tiap jalur bus. Alat yang digunakan adalah Laptop Toshiba L745 dengan spesifikasi RAM 8GB, Processor Core i5 dan Windows 10 Pro 64 bit. Kebutuhan perangkat lunak (Software) seperti Android Studio untuk pengembangan aplikasi android, Photoshop untuk mendesain aplikasi yang dibuat, SQLite Desktop untuk menganalisa database internal pada android, Server untuk menampung data-data yang dibutuhkan pada penelitian.

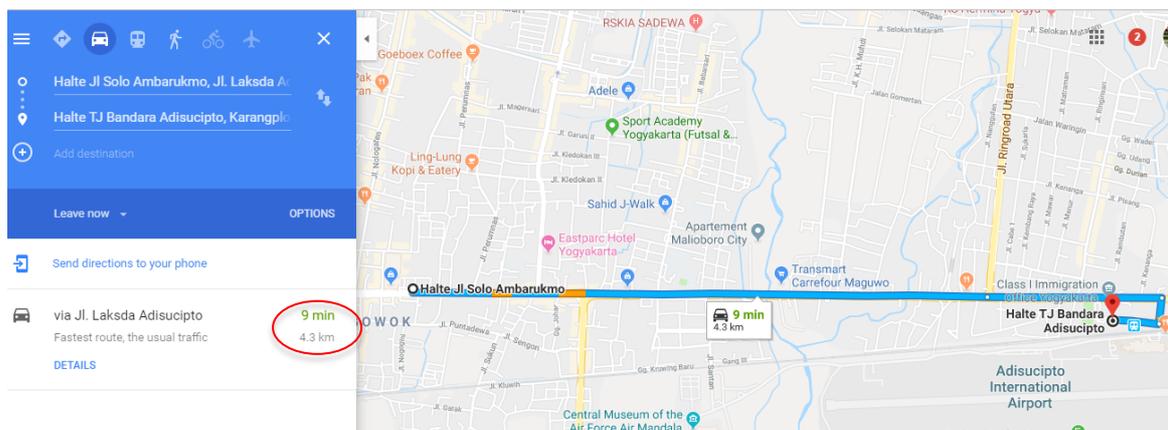
2.1. Pengumpulan Data

Pengumpulan data pada penelitian ini adalah mengambil informasi yang terkait dengan TransJogja seperti Halte, Rute, Trayek dan armada Bus yang dimiliki melalui website resmi dari dinas perhubungan kota Yogyakarta seperti pada gambar 1.



Gambar 1. Website Resmi Dishub Kota Yogyakarta

Setelah mendapatkan data yang terkait, penulis akan mencari titik koordinat dari halte yang ada melalui Google Maps dan menghitung jarak antar halte seperti Gambar 2. Jarak koordinat yang didapatkan akan digunakan dalam proses perhitungan jalur dari Algoritma Dijkstra dan Algoritma Semut.



Gambar 2. Website Resmi Dishub Kota Yogyakarta

2.2. Konsep Teori

2.2.1 Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma yang dipakai untuk memecahkan permasalahan jarak terpendek untuk sebuah graf yang berarah. Nama Dijkstra diambil dari nama penemunya yaitu Edsger Dijkstra yang merupakan seorang ilmuwan computer [8]. Algoritma ini menggunakan prinsip greedy dengan membandingkan setiap bobot minimum yang dilewati kemudian disimpan dalam himpunan. Algoritma Dijkstra populer digunakan untuk menentukan jalur terpendek dan tercepat. Algoritma Dijkstra bekerja dengan cara menghitung semua vertex atau titik yang tersedia. Algoritma Dijkstra dapat menemukan jalur terpendek pada graph yang memiliki vertex dan jarak antar vertex yang memiliki bobot positif [9]. Penerapan algoritma Dijkstra membutuhkan waktu eksekusi yang lebih cepat dibandingkan algoritma lain seperti algoritma Ant Colony sehingga Dijkstra lebih banyak digunakan dalam pencarian jalur optimum [10].

2.2.2 Algoritma Semut

Algoritma Semut (*Ant Colony Optimazition*) dirancang Urszula Boryczka pada tahun 2008. Algoritma semut diadopsi dari cara perilaku semut yang dapat menentukan rute terpendek dalam perjalanan dari sarang ke sumber makanan berdasarkan jejak feronom pada lintasan yang telah dilalui. Semut yang menemukan sumber makanan, akan meninggalkan feronom pada rute saat kembali ke koloninya. Semut lain yang mencium feronom di suatu rute, akan cenderung untuk mengikuti rute tersebut jika kandungan feronom cukup padat. Semakin padat kandungan feronom (tiap kali pulang ke koloninya, semut selalu meninggalkan feronom), maka semakin besar kemungkinan mengikuti rute tersebut. Selain feronom, yang juga mempengaruhi semut dalam memilih rute adalah visibility. Visibility merupakan naluri semut untuk menentukan rute terdekat menuju sumber makanan [11].

2.2.3 Sistem Informasi Geografis (GIS)

Sistem Informasi Geografis (SIG) atau Geographic Information System (GIS) adalah sistem informasi berbasis komputer, yang digunakan untuk memproses data spasial yang ber-georeferensi (berupa detail, fakta, kondisi, dsb) yang disimpan dalam suatu basis data dan berhubungan dengan persoalan serta keadaan dunia nyata (real world). Manfaat SIG secara umum memberikan informasi yang mendekati kondisi dunia nyata, memprediksi suatu hasil dan perencanaan strategis [10].

Studi lain mendefinisikan SIG adalah informasi mengenai permukaan bumi dan semua objek yang berada di atasnya, yang menjadi kerangka bagi pengaturan dan pengorganisasian bagi semua tindakan selanjutnya. Teknologi Sistem Informasi Geografis mengintegrasikan operasi umum database, seperti query dan analisa statistik, dengan kemampuan visualisasi dan analisa yang unik yang dimiliki oleh pemetaan [10].

Penelitian ini menggunakan SIG untuk menentukan posisi dari halte, bus dan pengguna sehingga pengguna dapat mengetahui jarak bus ke halte.

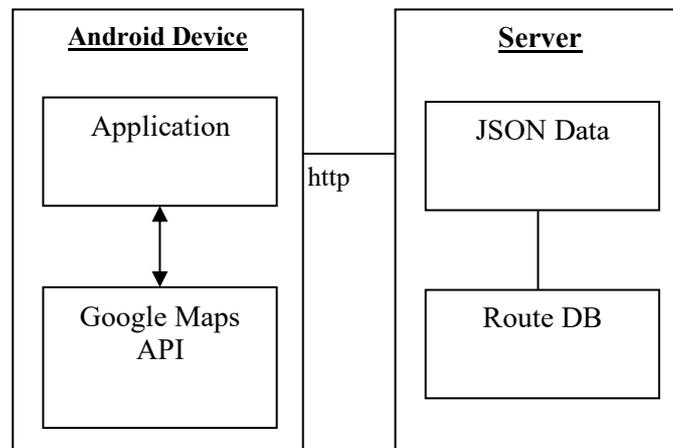
2.2.3 Android

Android merupakan sistem operasi yang berbasis open source dikembangkan oleh Google. Android memberikan kebebasan bagi developer untuk mengembangkan sistem operasi dan aplikasi yang dibuat. Sifat open source Android mendorong pengembang untuk membuat aplikasi dan mengunggahnya ke PlayStore sehingga dapat digunakan oleh pengguna Android lainnya. Aplikasi ini dapat digunakan oleh pengguna dengan mengunduhnya dari Android Market, lalu menginstalnya di ponsel cerdas mereka [12]. Sistem operasi Android merupakan teknologi yang pengguna cukup besar karena banyaknya vendor yang menggunakan sistem operasi ini [13].

Penelitian “Stimulasi Denyut Jantung dengan Pemutar Musik pada Android” memanfaatkan android sebagai media yang digunakan untuk mengukur denyut jantung ketika mendengarkan tempo musik [14].

Penelitian ini menggunakan media Android karena penggunaannya dapat mempermudah pengguna dan android sudah mendukung SIG sehingga penerapan fitur GPS dapat dimanfaatkan pada aplikasi yang akan dikembangkan.

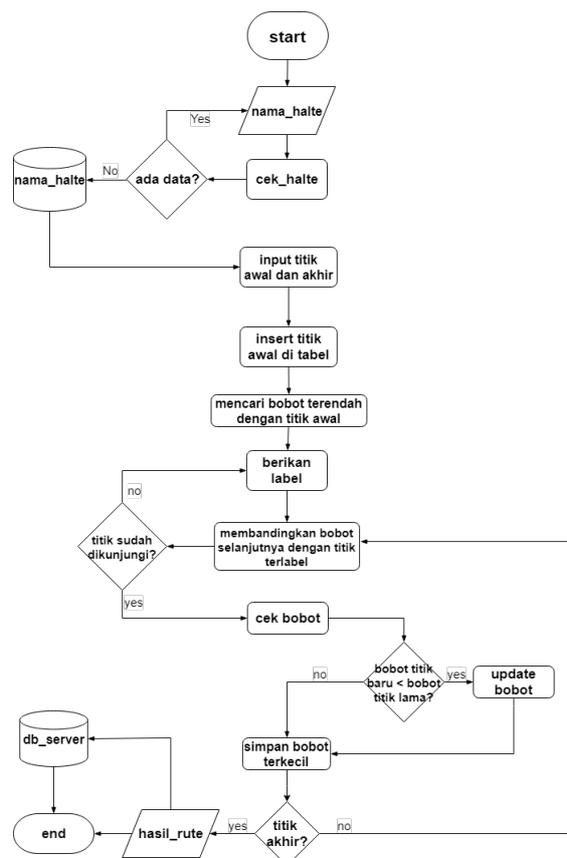
2.2.4 Proses Aplikasi



Gambar 3. Alur Aplikasi

Aplikasi yang dibuat melakukan *request* untuk pencarian rute menggunakan Algoritma *Dijkstra*. Pencarian rute dilakukan di *server* dan dikirimkan kembali ke aplikasi seperti pada Gambar 3.

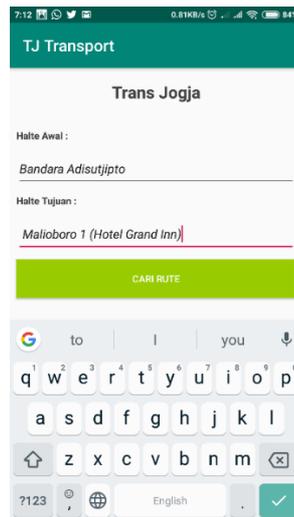
2.2.5 Flowchart



Gambar 4. Flowchart Sistem

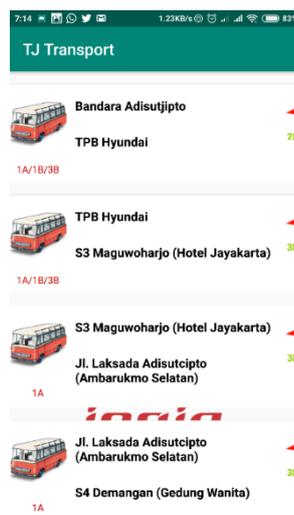
Gambar 4 merupakan flowchart sistem ketika dijalankan. Pada saat aplikasi dijalankan, data nama halte otomatis disinkronasikan dengan SQLite pada aplikasi sesuai yang ada di server. Pengguna menginputkan nama halte sesuai yang ada pada SQLite seperti pada Gambar 1. Setelah pengguna menginputkan titik awal dan akhir maka aplikasi mencari rute yang akan dilalui. Prinsip algoritma Dijkstra yaitu membandingkan semua bobot yang ada dan terhubung dengan titik awal dan akhir. Dijkstra menyimpan bobot yang terkecil sehingga data yang ditampilkan sesuai dengan rute TransJogja yang digunakan.

2.2.6 Implementasi



Gambar 5. Tampilan Input Data

Gambar 5 merupakan proses pengguna transportasi umum Transjogja akan menginputkan data yang diperlukan yaitu titik awal dan titik akhir penumpang. Semua titik awal dan akhir sudah tersimpan kedalam database server dan SQLite. Adapun tujuan disimpan di database server agar jika terjadi perubahan data seperti penambahan rute baru maka SQLite pada aplikasi akan otomatis *synchronize* dengan database yang ada pada server sehingga titik yang diinputkan sesuai dengan data yang diperlukan.



Gambar 6. Tampilan Hasil Pencarian Rute

Gambar 6 merupakan tampilan hasil rute yang dihasilkan dengan Algoritma Dijkstra. Pada tampilan tersebut terdapat informasi halte yang akan dilewati dan bus yang akan digunakan.



Gambar 7. Tampilan *Real Time Location*

Gambar 7 merupakan tampilan jika tombol navigasi dipilih. Pada tampilan ini akan menampilkan posisi dari pengguna aplikasi, halte dan bus Transjogja secara *Real Time* sehingga pengguna transportasi dapat mengetahui posisi bus yang akan digunakan.

3. HASIL DAN PEMBAHASAN

- Trayek 1A**
 Terminal Prambanan – Jl. Raya Yogyakarta Solo – S3. Bandara – Bandara Adisutjipto – Jl. Raya Yogyakarta Solo – S3. Maguwoharjo – Jl. Laksda Adisutjipto – S3. Janti – Jl. Laksda Adisutjipto – S4. Demangan – Jl. Jend Urip Sumoharjo – S4. Galeria – Jl. Jend Sudirman – S4. Tugu – Jl. Margo Utomo – Jl. Kleringan – S3. Jembatan Kewek – Jl. Abubakar Ali – S3. Hotel Garuda – Jl. Maliboro – Jl. Margo Mulyo – S4. Titik 0 Km – Jl. Panembahan Senopati – Jl. Sultan Agung – Jl. Kusumanegara – S4. Gedongkuning – Jl. Gedongkuning – Jl. Janti – S4. Blok O – Ringroad Selatan – S3. Janti – Jl. Laksda Adi Sucipto – S3. Maguwoharjo – Jl. Raya Yogyakarta Solo – S3. Bandara – Bandara Adisutjipto – Jl. Raya Yogyakarta Solo – Terminal Prambanan
- Trayek 1B**
 Bandara Adisutjipto – Jl. Raya Yogyakarta Solo – S3. Maguwoharjo – Jl. Laksda Adi Sucipto – S3. Janti – Ringroad Selatan – S3. Blok O – Jl. Janti – Jl. Gedongkuning – S4. Gedongkuning – Jl. Kusumanegara – Jl. Sultan Agung – Jl. Panembahan Senopati – Jl. KH Ahmad Dahlan – S3. RS PKU Muhammadiyah – Jl. Bhayangkara – Jl. Jogonegaran – Jl. Gandekan Lor – S3. Pasar Kembang – Jl. Jlagran Lor – S4. Badran – Bundaran Samsat – Jl. Tentara Pelajar – S4. Pingit – Jl. Pangeran Diponegoro – Jl. Jend Sudirman – S4. Gramedia – Jl. Cik Di Tiro – Bundaran UGM – Jl. Colombo – S3. Colombo – Jl. Affandi – S4. Demangan – Jl. Laksda Adisutjipto – S3. Babarsari – Jl. Babarsari – S3. Citrouli – Jl. Kledokan Raya – S3. PU Pengairan – Jl. Laksda Adi Sucipto – S3. Maguwoharjo – Jl. Raya Yogyakarta Solo – S3. Bandara – Bandara Adisutjipto
- Trayek 2A**
 Terminal Jombor – S4. Jombor – Ringroad Utara – S4. Monjali – Jl. Nyi Tjondrolukito – Jl. AM Sangaji – Jl. Margo Utomo – Jl. Kleringan – S3. Jembatan Kewek – Jl. Abubakar Ali – S3. Hotel Garuda – Jl. Maliboro – Jl. Margo Mulyo – S4. Titik 0 Km – Jl. Panembahan Senopati – S4. Gondomanan – Jl. Brigjen Katamso – S4. Jukteng Wetan – Jl. Kol. Sugiyono – Jl. Menteri Supeno – S4. XT Square – Jl. Veteran – S4. Warungboto – Jl. Gambiran – S4. Gambiran – Jl. Perintis Kemerdekaan – Jl. Ngeksigondo – S3. Tom Silver – Jl. Gedongkuning – S4. Gedongkuning – Jl. Kusumanegara – S3. Cendana – Jl. Cendana – S4. GOR Among Rogo – Jl. Bung Tarjo – S4. Gayam – Jl. Dr Sutomo – Flyover Lempuyangan – Jl. Atmosukarto – Jl. Yos Sudarso (Bundaran Kridosono) – Jl. Wardhani – Jl. Trimono – S4. Klitren – Jl. Wahidin Sudirohusodo – S4. Galeria – Jl. Jend Sudirman – S4. Gramedia – Jl. Cik Di Tiro – Bundaran UGM – Jl. Colombo – S3. Colombo – Jl. Affandi – S4. Condongcatur – Terminal Condongcatur – S4. Condongcatur – Ringroad Utara – S4. Jombor – Terminal Jombor

Gambar 8. Rute TransJogja

Untuk mendapatkan hasil perbandingan dari algoritma Dijkstra dan Ant Colony Optimization dibutuhkan sebuah graph beserta bobot. Pada penelitian ini digunakan rute dari salah satu transportasi umum di Yogyakarta yaitu Trans Jogja seperti pada gambar 8.

Tabel 1. Pengujian Jarak Dijkstra

| No | Input | | Output | |
|----|--------------------|----------------------|------------|--------------|
| | Titik Awal (Halte) | Titik Tujuan (Halte) | Jarak (KM) | Jumlah titik |
| 1 | Terminal Prambanan | Malioboro | 16 | 13 |
| 2 | Malioboro | Terminal Jombor | 6.2 | 11 |
| 3 | Terminal Jombor | Bandara Adisutjipto | 13 | 8 |

Tabel 2. Pengujian Jarak dengan Algoritma Semut

| No | Input | | Output | |
|----|--------------------|----------------------|------------|--------------|
| | Titik Awal (Halte) | Titik Tujuan (Halte) | Jarak (KM) | Jumlah titik |
| 1 | Terminal Prambanan | Malioboro | 16 | 13 |
| 2 | Malioboro | Terminal Jombor | 6.2 | 11 |
| 3 | Terminal Jombor | Bandara Adisutjipto | 13 | 8 |

Untuk menguji kemampuan dari masing-masing algoritma dalam menentukan jalur terpendek dan jumlah titik yang dilalui, penulis menguji sebanyak tiga kali dengan titik awal dan tujuan berbeda. Pengujian yang dilakukan didapatkan hasil yang sama antara algoritma Dijkstra dan Ant Colony Optimization seperti Tabel 1 dan 2.

Adapun perbandingan waktu eksekusi dari kedua algoritma tersebut dilakukan sebanyak seratus kali pada tiap rute menggunakan fungsi **microtime()** pada php dan dihitung rata-rata kecepatan waktu dan penggunaan memori dapat dilihat pada tabel 3. Hasil yang didapatkan waktu eksekusi Ant Colony Optimazition lebih membutuhkan waktu yang lebih besar dibandingkan Dijkstra hal ini disebabkan karena algoritma Ant Colony Optimazition prinsip kerjanya menyebar semut yang sudah ditentukan jumlahnya untuk menentukan jarak pada graph, sehingga data masukan atau data yang diproses semakin banyak. Berbeda dengan algoritma Dijkstra yang prinsip kerjanya membandingkan bobot tiap jalur yang ada pada graph. Semakin besar waktu eksekusi algoritma maka semakin besar memory yang dibutuhkan oleh algoritma.

Tabel 3. Perbandingan Waktu Eksekusi dan Memori Algoritma Dijkstra dan Semut

| No | Rute | Rata-rata Waktu Proses | | Rata-rata Penggunaan Memori | |
|----|------------------------------------|------------------------|-----------------|-----------------------------|-----------------|
| | | Algoritma Dijkstra | Algoritma Semut | Algoritma Dijkstra | Algoritma Semut |
| 1 | Terminal Prambanan – Malioboro | 0.0125 detik | 0.0229 detik | 387.14 kb | 797.62 kb |
| 2 | Malioboro – Term. Jombor | 0.0107 detik | 0.0143 detik | 387.14 kb | 712.42 kb |
| 3 | Term. Jombor – Bandara Adisutjipto | 0.0092 detik | 0.0158 detik | 387.14 kb | 751.81 kb |

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan didapatkan hasil sebagai berikut :

1. Algoritma *Dijkstra* memerlukan bobot antar titik sehingga dapat mempercepat proses perhitungan tanpa bobot algoritma *Dijkstra* tidak dapat dijalankan, sedangkan algoritma *Ant Colony Optimazition* dihitung berdasarkan jumlah semut ketika melakukan perjalanan.
2. Penggunaan memory pada algoritma *Dijkstra* lebih kecil dibandingkan *Ant Colony Optimazition* sehingga sangat cocok dijalankan pada aplikasi yang menggunakan fitur navigasi penentuan jalur.
3. Dalam penentuan jalur terpendek Algoritma *Dijkstra* dan *Ant Colony Optimazition* menghasilkan *output* jalur dan jarak yang sama.

DAFTAR PUSTAKA

- [1] S. Syukri, "Penerapan Customer Satisfaction Index (CSI) dan Analisis GAP Pada Kualitas Pelayanan Trans Jogja," *J. Ilm. Teknol. Ind.*, vol. 13, no. 1, pp. 103–111, 2014.
- [2] A. Yudhana, M. Dwi, and D. Putra, "Rancang Bangun Sistem Pemantauan Infus Berbasis Android," *TRANSMISI*, vol. 2, no. April, pp. 91–95, 2018.
- [3] A. Ratnasari, F. Ardiani, and F. Nurvita, "Penentuan Jarak Terpendek dan Jarak Terpendek Alternatif Menggunakan Algoritma Dijkstra Serta Estimasi Waktu Tempuh," *Semin. Nas. Teknol. Inf. Komun. Terap.*, pp. 29–34, 2013.
- [4] S. Andayani and E. W. Perwitasari, "Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra," *Semin. Nas. Teknol. Inf. Komun. Terap. 2014*, vol., no., pp. 164–170, 2014.
- [5] B. Amaliah, C. Fatichah, and O. Riptianingdyah, "Finding The Shortest Paths Among Cities in Java Island Using Node Combination Based on Dijkstra Algorithm," *Int. J. Smart Sens. Intell. Syst.*, vol. 9, no. 4, pp. 2219–2236, 2016.
- [6] A. Gusmão, S. H. Pramono, and Sunaryo, "Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra," *J. EECCIS*, vol. 7, no. 2, pp. 125–130, 2013.
- [7] I. M. K. Wicaksana and I. M. Widiartha, "PENERAPAN METODE ANT COLONY OPTIMZATION PADA METODE K-HARMONIC MEANS UNTUK KLASERISASI DATA I Made Kunta Wicaksana , I Made Widiartha Jurusan Ilmu Komputer , Fakultas MIPA , Universitas Udayana , Bali," vol. 5, no. 1, pp. 55–62, 2012.
- [8] Sunardi, A. Yudhana, and A. A. Kadim, "Implementasi Algoritma Dijkstra Dalam Penentuan Jalur dan Pemesanan Online Transportasi Umum Berbasis Android," *Semin. Nas. Teknologi Infromasi dan Komun.*, pp. 1–7, 2017.
- [9] A. G. Wibowo and A. P. Wicaksono, "Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Rumah Sakit di Purbalingga dengan Metode Algoritma Dijkstra," *Juita*, vol. II, no. 1, pp. 21–35, 2012.
- [10] N. Azizah and D. Mahendra, "Geolocation dengan Metode Dijkstra untuk Menentukan Jalur Terpendek Lokasi Peribadatan," *J. Sist. Inf. Bisnis*, vol. 02, pp. 96–103, 2017.
- [11] I. Maryati and H. K. Wibowo, "Optimasi penentuan rute kendaraan pada sistem distribusi barang dengan ant colony optimization," *Semin. Nas. Teknologi Infromasi Komun. Terap. 2012*, vol. 2012, no. Semantik, pp. 163–168, 2012.
- [12] I. Riadi, R. Umar, and A. Firdonsyah, "Identification Of Digital Evidence On Android's Blackberry Messenger Using NIST Mobile Forensic Method," *Int. J. Comput. Sci. Inf. Secur.*, vol. 15, no. 5, pp. 3–8, 2017.
- [13] N. Anwar and I. Riadi, "Analisis Investigasi Forensik WhatsApp Messenger Smartphone Terhadap WhatsApp Berbasis Web," *J. Ilmu Tek. Elektro Komput. dan Inform.*, vol. 3, pp. 1–10, 2017.
- [14] I. Kurnia, R. Ferdiana, and H. Adi, "Stimulasi Denyut Jantung dengan Pemutar Musik pada Android," *J. Sist. Inf. Bisnis*, vol. 03, pp. 205–210, 2014.