PERBANDINGAN ALGORITMA DIJKSTRA DAN ALGORITMA ANT COLONY DALAM PENENTUAN JALUR TRANSPORTASI UMUM

Sunardi, Anton Yudhana, Ahmad Azhar Kadim

Abstract— **The purpose of this research is to compare the working principle and characteristics between Dijkstra algorithm and Ant Colony Optimization in determining the shortest path Dijkstra algorithm requires weight to determine the shortest path, without the weight of Dijkstra, algorithm cannot run , while Ant Colony Optimization algorithm does not require weight at run because on the working principle of this algorithm will produce the output or weight when the ant finishes the journey . Therefore, this algorithm consume more RAM and takes a long time of execution when run than Dijkstra algorithm. The Dijkstra algorithm got the fastest result 0.0045 seconds while on algorithm Ant Colony Optimization 0.0126 sec, for RAM usage result obtained by Dijsktra algorithm stable 387.14 kb different from ACO obtained results vary with the smallest result is 712.42 kb The equation of both of these algorithms are the output path and the resulting distance are the same so that it can be implemented for the determination of public transportation routes. The application of this research will be applied to Transjogja which is one of the public transportations in Yogyakarta.**

Index Terms— **Dijkstra, Ant Colony Optimazition, Transjogja.**

I. PENDAHULUAN

Besarnya volume kendaraan pribadi mengakibatkan jalan umum tidak dapat menampung lagi kendaraan yang melintas mengakibatkan macet yang dapat memakan waktu tempuh. Beberapa Pemerintah daerah mengeluarkan aturan untuk mengurangi kemacetan, salah satunya mengeluarkan aturan plat ganjil/genap yang diterapkan di Jakarta agar sebagian pengguna jalan dapat beralih menggunakan transportasi umum. Transportasi umum merupakan modal tranportasi umum yang digunakan untuk berpindah dari satu tempat ke tempat lain dan memiliki biaya yang relatif murah[1]. Salah satu contoh transportasi yang digunakan adalah bus trans dan mobil angkutan umum yang sudah ditetapkan jalur yang harus dilewati, biasanya ditandai dengan kode pada setiap mobil atau bus[2]. Pengguna transportasi umum kadang perlu untuk transit atau berpindah ke angkutan lain sesuai dengan tujuan, sehingga tidak sedikit pengguna yang salah terutama pengunjung baru pada suatu kota.

Sunardi, Teknik Elektro Universitas Ahmad Dahlan, Yogyakarta, Indonesia (e-mail: sunardi@mti.uad.ac.id).
Anton Yudhana, Teknik Elektro Universitas Ahmad Dahlan, Yogyakarta, Indonesia (e-mail: eyudhana@ee.uad.ac.id).
Ahmad Azhar Kadim, Magister Teknik Informatika Universitas Ahmad Dahlan, Yogyakarta, Indonesia (corresponding author telpon: 082188933534; e-mail: azharkadim@yahoo.com).

Kurangnya informasi mengenai kode angkutan umum dapat mengurangi minat pengguna jalan untuk beralih ke transportasi umum, sehingga dibutuhkan suatu sistem yang dapat memberikan informasi kepada pengguna jalan secara *instant.*

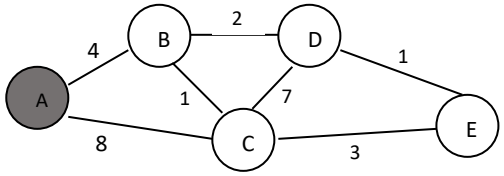
Dalam ilmu komputer terdapat beberapa algoritma yang dapat menentukan jalur terpendek seperti Algoritma *Dijkstra* dan Algoritma *Ant Colony Optimization.* Algoritma *Dijkstra* adalah algoritma yang digunakan untuk menentukan dan memecahkan masalah jarak pada sebuah graf yang berbobot [3]. Bobot pada graf dapat berupa jarak, waktu, biaya ataupun bobot lainnya. Algoritma *Dijkstra* bekerja dengan cara menghitung atau mengunjungi semua bobot yang terdapat pada graf dengan dimulai pada bobot sumber dan diakhiri dengan bobot tujuan[4]. Algoritma *Ant Colony Optimization* adalah suatu Algoritma yang dirancang oleh Urszula Boryczka[5]. Algoritma *Ant Colony Optimization* 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.

Penelitian ini bertujuan untuk mengetahui prinsip kerja serta karakteristik antara algoritma *Dijkstra* dan Algoritma *Ant Colony Optimization* dalam penentuan jalur. Adapun objek penelitian yang digunakan adalah jalur transportasi umum TransJogja. Hasil yang diperoleh dapat diterapkan sebagai media informasi kepada pengguna transportasi umum dan diharapkan dapat membantu memperbaiki sistem informasi yang sudah ada.

II. LANDASAN TEORI

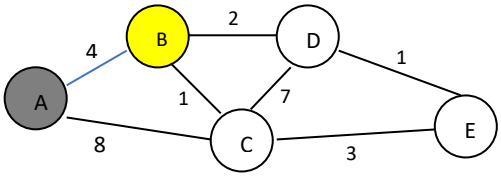
A. Algoritma Dijkstra

Algoritma *Dijkstra* diambil dari nama penemunya Edsger Dijkstra yang merupakan ilmuwan komputer. Algoritma *Dijkstra* digunakan untuk memecahkan permasalahan jarak pada suatu graf[3]. Bobot pada graf bisa berupa jarak, waktu, biaya atau bobot lainnya. Prinsip algoritma ini adalah menggunakan prinsip *greedy* dengan membandingkan setiap bobot minimum yang dilewati kemudian disimpan dalam himpunan[6]. Dalam penerapannya algoritma *Dijkstra* membutuhkan parameter asal dan akhir. Untuk lebih detail dapat dilihat skema algoritma *Dijkstra* seperti gambar 1 :



Gambar. 1 Menentukan Titik Awal

Gambar 1 menunjukkan titik A adalah titik awal dan titik tujuan adalah E. Algoritma Dijkstra akan mencari titik terdekat yang terhubung dengan titik A dan memiliki bobot terkecil.

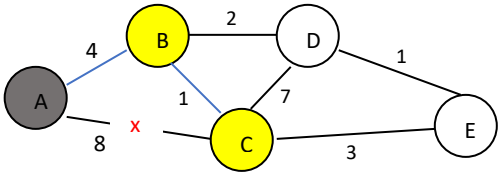


Gambar. 2 Menentukan Bobot Terendah dengan A

Gambar 2 menunjukkan bahwa titik yang memiliki bobot minimum adalah B. Selanjutnya, Algoritma *Dijkstra* akan membandingkan bobot dari titik yang terhubung dengan B dengan cara sebagai berikut:

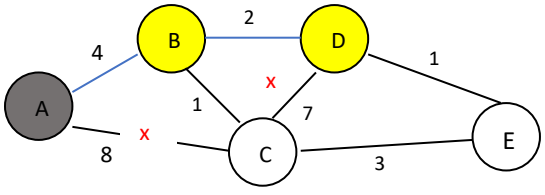
- Titik A-B-D = 6
- Titik A-B-C = 5

Dari perbandingan diatas maka didapatkan titik A-B-C memiliki bobot yang terkecil, sehingga algoritma *Dijkstra* akan melanjutkan perhitungan bobot yang terhubung dengan C. Pada tahap ini titik D tidak menjadi pilihan karena bobot yang dimiliki lebih besar dari bobot A-B-D dan A-B-C seperti gambar 3.



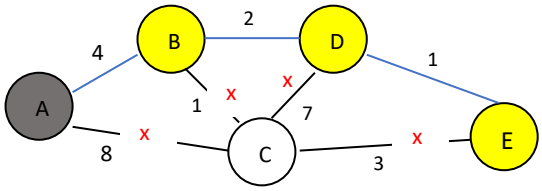
Gambar. 3 Menentukan Bobot Terendah Dengan C

Gambar 3 menunjukkan bahwa titik A-B-C-D = 12, sedangkan A-B-C-E = 8 dan sudah mencapai titik tujuan yaitu titik E, akan tetapi A-B-C-E belum bisa menjadi pilihan karena masih ada jalur lain yang memungkinkan memiliki bobot lebih rendah yaitu A-B-D = 6 dan belum dibandingkan oleh algoritma.



Gambar. 4 Menentukan Bobot Terendah Dengan D

Gambar 4 menunjukkan bahwa titik D terhubung dengan C dan E, akan tetapi titik C tidak menjadi pilihan karena bobot yang dimiliki A-B-D-C melebihi bobot minimum A-B-C-E, sehingga algoritma *Dijkstra* akan langsung menghitung bobot A-B-D-E. Pada tahap ini, didapatkan nilai minimum A-B-D-E = 7 sedangkan A-B-C-E = 8, maka hasil jalur yang memiliki bobot terendah adalah A-B-D-E dapat dilihat pada gambar 5.



Gambar 5. Hasil Akhir Jalur Terpendek

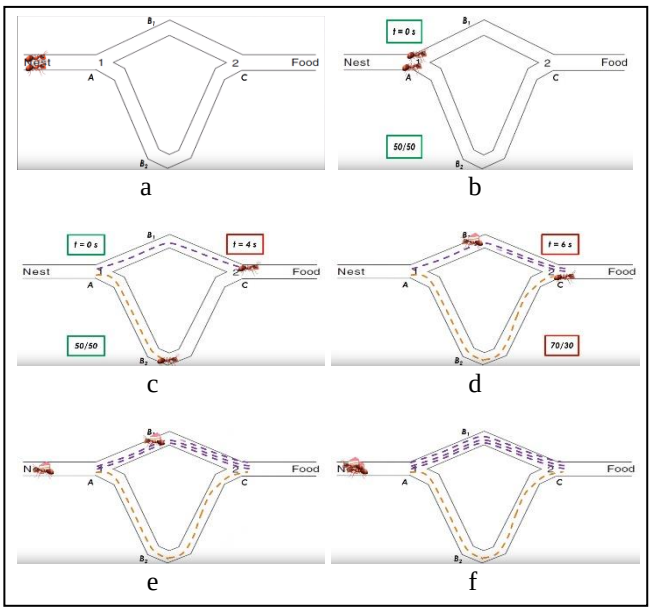
B. Algoritma Ant Colony Optimazition

Algoritma *Ant Colony Optimazition* dirancang Urszula Boryczka pada tahun 2008. Algoritma *Ant Colony Optimization* (ACO) 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[7].

Terdapat beberapa jenis algoritma *Ant Colony Optimazition* yaitu *Ant System (AS), Ant Colony System (ACS), Min-Max Ant System (MMAS), Elitis Ant System (EAS), Rank-Based Ant Systerm (ASRank), Approximate Nondeterministiv Tree Search (ANTS).* Dari beberapa jenis tersebut AS dan ACS merupakan jenis ACO yang paling populer digunakan. Algoritma AS merupakan algoritma versi pertama ACO yang diusulkan pada tahun 1992[8].

Gambar 6 menjelaskan tentang skema kerja lengkap dari Algoritma ACO. Warna ungu dan orange adalah jejak dan feronom yang dihasilkan oleh semut. Gambar 6a menunjukkan jumlah semut yang akan digunakan, gambar 6b menjelaskan jalur yang akan dilewati oleh masing-masing pada tahap ini persentase kedua jalur yang akan dilalui yaitu 50/50 karena belum pernah dilalui maka semut akan berpencar seperti Gambar 6c. Pada gambar 6c, semut pertama memerlukan waktu 4 detik untuk sampai ke sumber makanan dibandingkan semut kedua. Gambar 6d semut pertama akan kembali ke sarang dengan mengikuti jejak feronom yang ditinggalkan. Seperti penjelasan

diaatas, semakin sering dilewati maka semakin kuat feronom yang dihasilkan, sehingga pada Gambar 6e semut kedua akan memilih jalur semut pertama karena jalur tersebut sudah dilewati dua kali (pulang pergi) oleh semut pertama sehingga feronom yang dihasilkan lebih besar dibandingkan jalur yang dilewati sebelumnya oleh semut kedua. Gambar 6f adalah hasil jalur terpendek dari skema kerja Algoritma ACO.



Gambar 6. *Skema Ant Colony Optimization*

III. TINJAUAN PUSTAKA

Kajian yang terkait dengan penelitian ini antara lain :

1) Penelitian berjudul *"Penentuan Rute Terpendek Pengambilan Sampah Di Kota Merauke Menggunakan Algortima Dijkstra"*. Algoritma *Dijkstra* diterapkan untuk pengambilan sampah di Kota Merauke dengan melibatkan beberapa pertimbangan utama meliputi rute kendaraan dan meminimalisasi onkos distribusi, sehingga dapat memperluas wilayah pengambilan sampah dengan armada yang terbatas[9].
2) Penelitian berjudul *"Finding The Shortest Paths Among Cities In Java Island Using Node Combination Based On Dijkstra Algorithm"* menerapkan algoritma *Dijkstra* untuk mendapatkan jalur terpendek antara kota-kota yang ada di pulau Jawa[10].
3) Penelitian berjudul "Deteksi Tepi Citra Daun Mangga Menggunakan Algoritma Ant Colony Optimization" mendeteksi tepi daun mangga dengan menggunakan algoritma *Ant Colony Optimization* dengan tujuan untuk mengetahui perbedaan hasil deteksi tepi konvensional[8].

Dari beberapa kajian peneliti sebelumnya terdapat beberapa kesamaan seperti penggunaan algoritma untuk menentukan jalur terpendek. Pada penelitian ini akan menganalisa karakteristik dari algortima *Dijkstra* dan *Ant Colony Optimization* serta penerapan kedua algoritma tersebut.

IV. HASIL DAN PEMBAHASAN

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 7.

Gambar 7. *Jalur Trans Jogja Trayek 1 dan 2*

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.

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 1. *Pengujian Jarak dan Jalur Dijkstra*

No	Input		Output	
	Titik Awal (Halte)	Titik Tujuan (Halte)	Jarak	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 dan Jalur Ant Colony Optimazition*

Adapun perbandingan waktu eksekusi dari kedua algoritma tersebut pada tiap percobaan 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 seperti pada tabel 4.

No	Algoritma Dijkstra	Algoritma Ant Colony Optimazition
1	0.0076 detik	0.0246 detik
2	0.0045 detik	0.0126 detik
3	0.0068 detik	0.0205 detik

Tabel 3. *Perbandingan Waktu Eksekusi Algoritma*

No	Algoritma Dijkstra (kb)	Algoritma Ant Colony Optimazition (kb)
1	387.14	797.62
2	387.14	712.42
3	387.14	751.81

Tabel 4. *Perbandingan penggunaan Memory*

V. KESIMPULAN

Berdasarkan hasil perbandingan algoritma *Dijkstra* dan *Ant Colony Optimazition* maka dapat diambil kesimpulan 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* tidak memerlukan bobot karena jarak 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] R. Umar and R.K. Ikhsan., "Sistem Informasi Pariwisata Berbasis Lokasi Di Propinsi Daerah Istimewa Yogyakarta Yang Didukung Oleh Fotografi". Simposium Nasional Teknologi Terapan (SNTT), 2015.

[2] Sunardi, A.Yudhana, K. Ahmad Azhar, "Implementasi Algoritma Dijkstra Dalam Penentuan Jalur dan Pemesanan Online Transportasi Umum Berbasis Android," Semantikom., pp. 71–78, 2017.

[3] B. Junanda, D. Kurniadi, and Y. Huda, "Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra pada Sistem Informasi Geografis Pemetaan Stasius Pengisian Bahan Bakar Umum," *J. Vokasional Tek. Elektron. Inform.*, vol. 4, no. 1, pp. 1–8, 2016.

[4] A. Gusmão and S. H. Pramono, "Sistem Informasi Geografis Pariwisata Berbasis Web Dan Pencarian Jalur Terpendek Dengan Algoritma Dijkstra," *J. EECCIS*, vol. 7, no. 2, pp. 125–130, 2013.

[5] K. M. Clustering and A. C. Optimization, "PENERAPAN METODE ANT COLONY OPTIMZATION PADA METODE K-HARMONIC MEANS UNTUK KLASTERISASI DATA I Made Kunta Wicaksana , I Made Widiartha Jurusan Ilmu Komputer , Fakultas MIPA , Universitas Udayana , Bali," vol. 5, no. 1, pp. 55–62, 2012.

[6] 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.*, vol. 2013, no. November, pp. 29–34, 2013.

[7] I. Maryati and H. K. Wibowo, "Optimasi penentuan rute kendaraan pada sistem distribusi barang dengan ant colony optimization," *Semin. Nas. Teknlogi Infromasi Komun. Terap. 2012*, vol. 2012, no. Semantik, pp. 163–168, 2012.

[8] F. Liantoni, "Deteksi Tepi Citra Daun Mangga Menggunakan," pp. 411–418, 2015.

[9] S. Andayani and E. W. Perwitasari, "Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra," *Aeminar Nas. Teknol. Inf. Komun. Terap.*, vol., no., pp. 164–170, 2014.

[10] 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.