

Algoritma Dijkstra untuk Pendistribusian Carica Nida Food Wonosobo

Hamdun Sulaiman^{#1}, Yuri Yuliani^{#2}, Evita Fitri^{*3}, Nuraeni Herlinawati^{#4}, Sri Watmah^{*5}

[#]Universitas Bina Sarana Informatika

Jl. Kamal Raya No. 18, Ringroad Barat, Cengkareng, Jakarta Barat

¹hamdun.hsl@bsi.ac.id

²yuriyuli2404@bsi.ac.id

⁴nuraenih2803@bsi.ac.id

^{*}STMIK Nusa Mandiri

Jl. Damai No. 8, Warung Jati Barat (Margasatwa), Jakarta Selatan

³evitafit0102@bsi.ac.id

⁵sri.watmah@gmail.com

Abstrak

Travelling Salesmen Problem (TSP) pencarian rute terdekat, waktu tercepat, minimalisir biaya distribusi, dan perjalanan yang hanya ditempuh satu kali dari setiap titik dan kembali ke titik awal. Studi kasus yang diteliti dari pendistribusian carica Nida Food yang memiliki banyak konsumen di toko-toko wonosobo. Dalam memecahkan permasalahan ini memakai algoritma dijkstra yang bekerja memakai prinsip greedy, setiap langkahnya dipilih berdasarkan nilai jarak terdekat dan kemudian menghubungkan simpul yang terpilih dengan membandingkan nilai jarak dari setiap node sehingga dapat menemukan jalur yang sangat efisien. Data yang kita dapat untuk pencarian rute pendistribusian ini dari penelitian sebelumnya yang mencari rute tependek dengan menggunakan algoritma berbeda yang menghubungkan 4 titik pendistribusian akan bergerak ke titik berikutnya dan akan berakhir di titik awal. Algoritma dijkstra ini tidak memakan waktu yang cukup lama untuk menemukan rute terpendek untuk mendistribusikan produk carica dan hasil yang didapat sangat akurat. Informasi jalur yang akan dilalui menjadi tampilan dari hasil akhir.

Kata kunci: Travelling Salesmen Problem (TSP), Rute Terpendek, Algoritma Dijkstra, Carica, Nida Food Wonosobo

Dijkstra's Algorithm for Distributing Carica Nida Food Wonosobo

Abstract

Traveling Salesmen Problem (TSP) finding the closest route, the fastest time, minimizing distribution costs, and trips that are only taken once from each point and return to the starting point. A case study examined from the distribution of Carica Nida Food which has many consumers in wonosobo stores. In solving this problem using the dijkstra algorithm that works using the greedy principle, each step is chosen based on the value of the closest distance and then connects the selected node by comparing the distance value of each node so that it can find a very efficient path. The data we get for this distribution route search from previous research that looks for a dependent route by using a different algorithm that connects the 4 distribution points will move to the next point and will end at the starting point. The dijkstra algorithm does not take a long time to find the shortest route to distribute carica products and the results obtained are very accurate. Information on the path to be traversed is a display of the final result.

Keywords: Traveling Salesmen Problem (TSP), Shortest Route, Dijkstra Algorithm, Carica, Nida Food Wonosobo

I. PENDAHULUAN

Travelling Salesman Problem (TSP) merupakan pencarian jarak terdekat, waktu tercepat dengan metode yang dapat meminimalisir biaya distriusi [1]. Perjalanan yang ditempuh cukup satu kali, dari setiap titik lokasi ke lokasi semula [2]. Distributor carica Nida Food yang banyak memiliki konsumen di toko-toko Wonosobo harus menentukan rute minimum untuk dilalui dalam

mengantarkan hasil produksinya agar tepat disatu jalur hingga ketempat asal [3].

Pada penelitian sebelumnya permasalahan ini menggunakan metode Artificial Bee Colony (ABC) yang merupakan suatu metode yang mengambil dari foraging behavior koloni lebah madu. Meski hasil dari metode ini memiliki performa yang unggul, akan tetapi terdapat kelemahan dimana membutuhkan waktu yang cukup lama untuk mendapatkan solusi yang optimal [4].

Algoritma dijkstra bekerja dengan menggunakan prinsip greedy, setiap langkah dipilih berdasarkan bobot minimum dengan menghubungkan simpul yang terpilih terhadap simpul yang belum terpilih [5], membandingkan jarak terkecil dari node awal sampai node akhir sehingga menemukan jalur yang paling efisien [6]. Penyelesaian masalah dengan tidak terdapat sisi negatif dalam hal biaya serta menghasilkan pohon keputusan lintasan terpendek [7], informasi lajur jalan yang harus dilalui, merupakan tampilan dari hasil akhir [8].

II. TINJAUAN PUSTAKA

A. Travelling Salesman Problem (TSP)

Pada tahun 1930 TSP pertamakali dipelajari oleh ilmuwan- ilmuwan matematika di Viena dan Harvard oleh Karl Meger yang menemukan masalah, pertimbangan algoritma brute- force dan meneliti ketidak optimalan dari heuristic dengan tetangga terdekat. Kemudian Hassler Whitney dari Universitas Princeton mempublikasikan nama Traveling Salesman Problem (TSP) tidak lama setelahnya [9].

Permasalahan optimalisasi pencarian terhadap rute awal ke rute-rute berikutnya dengan syarat setiap rute di kunjungi satu kali terkecuali rute awal [10], dan menghasilkan graf yang bernilai [11]. Selain sebagai masalah transfortasi, pengiriman barang yang tepat dapat di tentukan oleh lintasan yang di lewati untuk mengirimkan barang tersebut, sehingga pemecahan masalah secara optimal dari TSP ini, dapat membantu perusahaan dalam mengirimkan barang dengan efisien dari prihal waktu maupun dana [12].

B. Algoritma Dijkstra

Algoritma Dijkstra ditemukan Edsger Dijkstra yang kemudian di berikan nama sesuai dengan penemunya algoritma ini menggunakan prinsip greedy, karena pada setiap jarak dipilih berdasarkan nilai jarak yang terkecil kemudian menghubungkan titik yang sudah terpilih terhadap titik lain yang belum terpilih dengan nilai jarak tak-negatif [13]. Input algoritma ini adalah sebuah graf berarah dan nilai jarak, G dan sebuah source vertex s dalam G. V adalah himpunan semua simpul dalam graph G. Setiap jarak dari graph ini adalah pasangan vertices (u,v) yang oleh melambangkan hubungan dari vertex u ke vertex v. Himpunan semua edge disebut E.

Weights dari edges dihitung dengan fungsi $w: E \rightarrow [0, \infty]$;

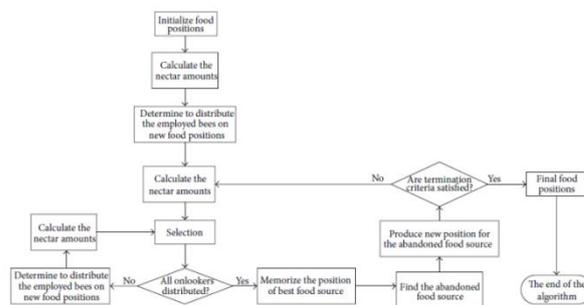
jadi $w(u,v)$ adalah jarak non-negatif dari vertex u ke vertex v.

Cost dari sebuah edge dapat dianggap sebagai jarak antara dua vertex, yaitu jumlah jarak semua edge dalam path tersebut. Untuk sepasang vertex s dan t dalam V, algoritma ini menghitung jarak terpendek dari s ke t [14].

C. Algoritma Artificial Bee Colony

Artificial Bee Colony (ABC) Algorithm adalah salah satu algoritma berdasarkan suatu pendekatan population-based metaheuristic yang diusulkan oleh Karaboga dan Basturk. Cara kerja algoritma ini seperti pada kawan

lebah dalam koloni yang bekerjasama untuk mencari sumber makanan. Pada model algoritma ini membagi lebah menjadi 3 kelompok, yaitu: employed bees, onlookers, dan scouts. Employed bee disini merupakan lebah yang berhubungan dengan sumber makanan tertentu, onlooker bee bertugas memilih sumber makanan setelah menyaksikan tarian lebah dalam sarang, dan scout bee bertugas mencari sumber makanan secara acak. Onlooker dan scout bee merupakan unemployed bee [15]. Flowchart algoritma artificial bee colony dapat dilihat pada Gambar 1.



Gambar 1. Flowchart algoritma artificial Bee Colony

III. HASIL DAN PEMBAHASAN

Hasil dari perbandingan yang dilakukan oleh Arifin dan Laksito dengan mencari jarak terpendek menggunakan Algoritma Artificial Bee Colony (ABC) pada studi kasus optimasi jalur terpendek untuk pendistribusian Carica Nida Food Wonosobo yang proses pencariannya memakan waktu yang cukup lama dikarenakan penyebaran lebah dalam menghasilkan solusi, dengan hasil dapat dilihat pada Tabel I.

TABEL I
JARAK MASING-MASING TOKO

	Nida Food	Bu Likot	Tambi	Sudina
A. Nida Food	0	2,78	5,36	7,83
B. Bu Likito	2,78	0	2,66	7,50
C. Tambi	5,36	2,66	0	7,24
D. Sadina	7,83	7,50	7,24	0

A. Algoritma Artificial Bee Colony

Dengan menggunakan algoritma Artificial Bee Colony menggunakan parameter berikut ini diperoleh hasil perhitungan seperti pada Gambar 2. Parameter yang digunakan: (1) Total lebah= 10, (2) Total lebah aktif= 7, (3) Total lebah nonaktif= 1, (4) Total lebah pencari= 2, (5) Jumlah maksimal perjalanan= 10, (6) Jumlah iterasi= 500.

B. Pseudocode Dijkstra (Gambar 3)

C. Skema Algoritma Dijkstra

Pada graf berikut akan menentukan jalur terpendek dari data berikut:

- Menentukan *node* awal dengan memberikan nilai 0 dan berikan nilai tak hingga (∞) pada *node* yang lainnya (Gambar 4).
- Menentukan nilai jarak paling kecil untuk melangkah ke *node* selanjutnya. Pada Gambar 5

diketahui bahwa berjalan ke *node* B memiliki nilai terkecil, kemudian memberikan nilai atau label permanen pada *node* B yaitu bernilai 2,78.

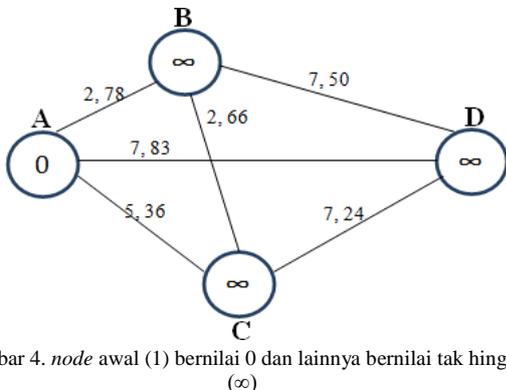
Proses	Jalur	Nilai
Lebah 1	Nida Food - Toko Tambi - Toko Sadina - Toko Bu Lukito - Nida Food	20,88 km
Lebah 2	Nida Food - Toko Sadina - Toko Bu Lukito - Toko Tambi - Nida Food	23,35 km
Lebah 3	Nida Food - Toko Sadina - Toko Bu Lukito - Toko Tambi - Nida Food	23,35 km
Lebah 4	Nida Food - Toko Bu Lukito - Toko Sadina - Toko Tambi - Nida Food	20,88 km
Lebah 5	Nida Food - Toko Tambi - Toko Sadina - Toko Bu Lukito - Nida Food	20,88 km
Lebah 6	Nida Food - Toko Sadina - Toko Tambi - Toko Bu Lukito - Nida Food	20,51 km
Lebah 7	Nida Food - Toko Sadina - Toko Tambi - Toko Bu Lukito - Nida Food	20,51 km
Lebah 8	Nida Food - Toko Tambi - Toko Sadina - Toko Bu Lukito - Nida Food	20,88 km
Lebah 9	Nida Food - Toko Bu Lukito - Toko Sadina - Toko Tambi - Nida Food	20,88 km
Lebah 10	Nida Food - Toko Bu Lukito - Toko Sadina - Nida Food	20,51 km

Gambar 2. Proses pencarian rute dengan algoritma artificial Bee Colony

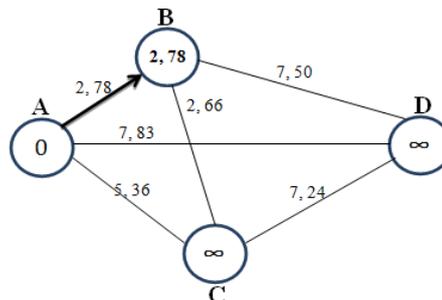
```

1  function Dijkstra(Graph, source):
2  for each vertex v in Graph:
3  dist[v] := infinity;
4  previous[v] := undefined;
5  end for
6
7  dist[source] := 0;
8  Q := the set of all nodes in Graph;
9
10 while Q is not empty:
11 u := vertex in Q with smallest
12 distance in dist[];
13 remove u from Q;
14 if dist[u] = infinity:
15 break;
16 end if
17 for each neighbor v of u:
18 alt := dist[u] + dist_between(u, v);
19 if alt < dist[v]:
20 dist[v] := alt;
21 previous[v] := u;
22 decrease-key v in Q;
23 end if
24 end for
25 end while
26 return dist;
    
```

Gambar 3. Pseudo code algoritma Dijkstra

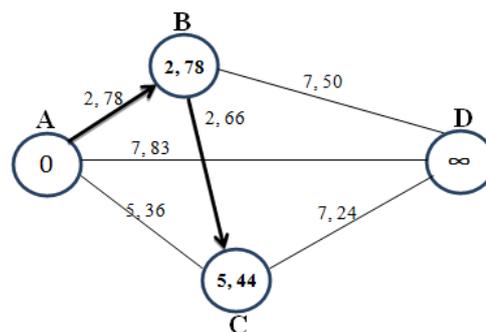


Gambar 4. *node* awal (1) bernilai 0 dan lainnya bernilai tak hingga (∞)



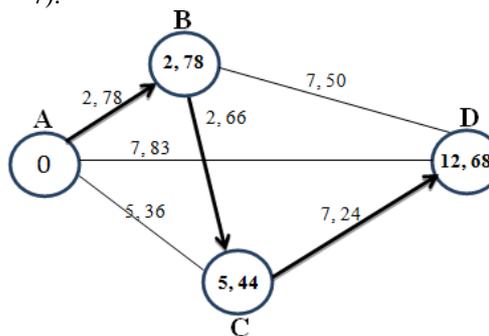
Gambar 5. Menentukan nilai jarak minimum

- Mencari *node* terpendek berikutnya yaitu memilih bobot minimum dengan membandingkan nilai biaya menuju *node* tersebut atau melalui *node* yang telah memiliki nilai permanen (Gambar 6).



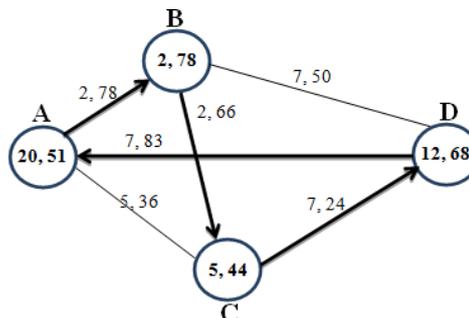
Gambar 6. Menentukan nilai jarak minimum untuk menuju *node* selanjutnya

- Mencari *node* berikutnya yang belum dilalui untuk kembali ke *node* awal (*node* A) (Gambar 7).



Gambar 7. Menentukan *node* selanjutnya untuk kembali ke *node* A

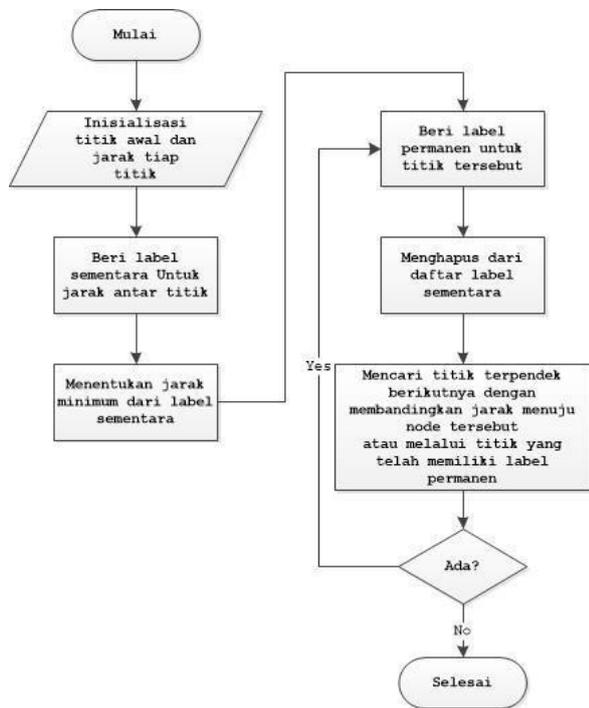
- Semua *node* telah dilalui maka dari *node* yang dikunjungi terakhir kemudian menuju *node* A (Gambar 8).



Gambar 8. Kembali ke *node A*

Sehingga dapat disimpulkan jalur terpendek yang dapat dilalui dari titik awal (*node A*) dan kembali lagi ke titik awal (*node A*) untuk mendistribusikan produk carica adalah $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$ dengan jarak tempuh 20,51.

D. Diagram Alir Algoritma Dijkstra (Gambar 9)



Gambar 9. Diagram Alir Algoritma Dijkstra

IV. KESIMPULAN

Dapat disimpulkan dengan menggunakan pencarian rute menghasilkan jarak tempuh terdekat adalah 20,51, akan tetapi dengan algoritma dijsktra dapat lebih cepat dan akurat dalam menentukan solusi terbukti dengan satukali jalan untuk membandingkan nilai terkecil pada setiap rute yang harus dilalui. dibandingkan dengan menggunakan algoritma artificial bee colony yang membutuhkan waktu yang cukup lama untuk menghasilkan solusi karena ada proses penyebaran lebah dengan total 10 lebah, serta dilakukan perbandingan jarak yang dihasilkan oleh setiap lebah yang menghasilkan 10 rute sesuai lebah yang di sebarakan jadi tidak langsung diketahui rute terdekat.

DAFTAR PUSTAKA

[1] P. Tsp, D. I. Pt, P. Cabang, D. Dermawan, and V. Ramadiansyah, "Dedi Dermawan , Vedri Ramadiansyah," vol. 1, no. 2, pp. 1–8, 2018.

[2] T. Kolo, A. Fanggidae, Y. N. Nabuasa, J. I. Komputer, F. Sains, and U. N. Cendana, "ANALISIS METODE CYCLE CROSSOVER (CX) DAN METODE PARTIAL- MAPPED CROSSOVER (PMX) PADA PENYELESAIAN KASUS TRAVELING SALESMAN PROBLEM (TSP) Selection merupakan suatu proses untuk mendapatkan kromosom-kromosom berkualitas pada generasi berikutnya . Met," vol. 7, no. 1, pp. 61–66, 2019.

[3] J. S. Informasi, I. Algoritma, B. Colony, U. Optimasi, and R. Distribusi, "Arifin, Implementasi Algoritma Bee Colony Untuk Optimasi Rute Distribusi Carica nida food wonosobo 243," vol. 8, pp. 243–253, 2019.

[4] P. Studi, T. Informatika, J. I. Komputer, and U. Udayana, "PENERAPAN CROSSOVER PADA PERILAKU LEBAH SCOUT DALAM ALGORITMA ARTIFICIAL BEE COLONY UNTUK OPTIMASI VEHICLE," pp. 50–57.

[5] A. C. Prasetyo, M. P. Armandi, H. S. Hudnanto, and B. Setiaji, "Perbandingan Algoritma Astar dan Dijkstra Dalam Menentukan Rute Terdekat," *Sisfotenika*, vol. 9, no. 1, p. 36, 2019.

[6] M. Masri, A. P. Kiswanto, and B. S. Kusuma, "IMPLEMENTASI ALGORITMA DIJKSTRA DALAM PERANCANGAN PARIWISATA DANAU TOBA DAN SEKITARNYA," pp. 221–225, 2019.

[7] D. Search, R. Using, D. Algorithm, and M. S. Method, "Perancangan Robot Penelusur Menggunakan Algoritma Dijkstra dan Metode Maze Solver," vol. 18, no. 2, pp. 166–177, 2019.

[8] J. Sauwani, V. N. Putra, and H. Agung, "Implementasi Algoritma Dijkstra Untuk Menentukan Lokasi Dan Jarak Tempuh Terpendek Kampus It Di Jakarta," *J. Inform.*, vol. 6, no. 1, pp. 29–36, 2019.

[9] J. Pedro and M. Ortiz, "No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析 Title," 2018.

[10] N. Juni, P. Tsp, P. Graf, L. Studi, K. Pt, and K. Rimba, "Vol 3, No.1 Juni 2019," vol. 3, no. 1, pp. 88–96, 2019.

[11] R. G. Utomo, D. S. Maylawati, and C. N. Alam, "Implementasi Algoritma Cheapest Insertion Heuristic (CIH) dalam Penyelesaian Travelling Salesman Problem (TSP)," *J. Online Inform.*, vol. 3, no. 1, p. 61, 2018.

[12] "Penyelesaian Travelling Salesman Problem (TSP) Menggunakan Algoritma Hill Climbing dan MATLAB," *Matematika*, vol. 17, no. 1, pp. 13–20, 2018.

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

[14] H. Sulistiani and D. A. Wibowo, "Perbandingan Algoritma A * dan Dijkstra dalam Pencarian Kecamatan dan Kelurahan di Bandar Lampung," *Knsi* 2018, pp. 423–428, 2018.

[15] Arifin, M. D., & Laksio, A. D. (2019). IMPLEMENTASI ALGORITMA BEE COLONY UNTUK OPTIMASI RUTE DISTRIBUSI CARICA NIDA FOOD WONOSOBO. SISTEMASI : Jurnal Sistem Informasi, 2