



Tersedia secara online di <http://ojs.unik-kediri.ac.id/index.php/jurmatis/index>

JURMATIS

Jurnal Manajemen Teknologi dan Teknik Industri Universitas Kediri



Penerapan Algoritma Dijkstra untuk Mencari Rute Terpendek pada Pengiriman Produk Wafer di PT. XYZ

Ikhsan Ahadi*¹, Mimi Nur Habibah², Popy Primaviani Desi Deria³, Muchammad Fauzi⁴

ikhsan.ahadi@widyatama.ac.id*¹, mimi.habibah@widyatama.ac.id², popy.primaviani@widyatama.ac.id³,
muchammad.fauzi@widyatama.ac.id⁴

^{1,2,3,4}Program Studi Teknik Industri, Fakultas Teknik, Universitas Widyatama

Informasi Artikel

Riwayat Artikel :

Received : 19 – Juni – 2021

Revised : 6 – Juli – 2021

Accepted : 9 – Juli – 2021

Kata kunci :

Dijkstra's Algorithm

Distribution Point

Shortest Route

Weight

Untuk melakukan sitasi pada penelitian ini dengan format :

B. Junanda, D. Kurniadi, and Y. Huda, "Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Pada Sistem Informasi Geografis Pemetaan Stasiun Pengisian Bahan Bakar Umum," *Voteteknika (Vocational Tek. Elektron. dan Inform., vol. 4, no. 1, 2018, doi: 10.24036/voteteknika.v4i1.6014*

Abstract

Dijkstra's algorithm is used to find the shortest path based on the smallest weight from one point to another, then Dijkstra's Algorithm calculates all possible smallest weights from each point. Dijkstra's algorithm in this study is used as a tool to calculate all possible smallest weights from the distribution line of PT. XYZ to retailer point. The problems faced by PT. XYZ, the company wants to save shipping time by finding the shortest route that must be passed. The purpose of this study is to determine the shortest route that must be traversed and how far must be taken by PT. XYZ to ship its products to retailers. Based on the analysis of Dijkstra's Algorithm, the results of the shortest route pass through the points 0-3-5-4-6, where point 0 is the starting point or PT. XYZ, point 3 is Jalan Raya Curug – Kosambi, point 5 is Cilangkap, Purwakarta, point 4 is Kalihurip, Cikampek and point 6 is the destination retailer in Dian Anyar, Purwakarta with a total distance of 42.9 km.

Abstrak

Algoritma Dijkstra digunakan untuk menemukan lintasan terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya, maka Algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik. Algoritma Dijkstra dalam penelitian ini digunakan sebagai alat untuk mengkalkulasikan semua kemungkinan bobot terkecil dari jalur distribusi produsen PT. XYZ ke titik *retailer*. Permasalahan yang dihadapi oleh PT. XYZ yaitu perusahaan ingin menghemat waktu pengiriman dengan cara mencari rute terpendek yang harus dilewati. Tujuan dari penelitian ini yaitu untuk mengetahui rute terpendek yang harus dilalui dan berapa jarak yang harus ditempuh oleh PT. XYZ untuk mengirimkan produknya ke *retailer*. Berdasarkan analisis Algoritma Dijkstra didapatkan hasil rute terpendek melewati titik-titik 0-3-5-4-6, dimana titik 0 sebagai titik awal atau PT. XYZ, titik 3 sebagai Jalan Raya Curug – Kosambi, titik 5 sebagai Cilangkap, Purwakarta, titik 4 sebagai Kalihurip, Cikampek dan titik 6 sebagai *retailer* tujuan yang berada di Dian Anyar, Purwakarta dengan total jarak yang harus ditempuh yaitu sebesar 42,9 km.

1. Pendahuluan

Perjalanan untuk pengiriman produk dilakukan dari satu tempat ke tempat lainnya atau dari kota ke kota lain dengan mempertimbangkan berbagai hal, seperti efisiensi, waktu dan biaya yang dikeluarkan sehingga diperlukan pemilihan rute terpendek yang tepat untuk mengirim produk agar tidak memakan waktu dan juga mengeluarkan biaya yang mahal. Pencarian rute yang dapat dilewati untuk sampai ke tujuan yang diinginkan dapat dilihat melalui peta digital atau biasa disebut dengan *Google Maps*. Jalur yang dapat dilewati agar sampai ke tujuan dapat terlihat dengan jelas, mulai dari kota mana saja yang akan dilewati, total jarak yang harus ditempuh serta perbandingan jalan yang akan dilalui untuk sampai ke tujuan.

Pencarian rute terpendek untuk pengiriman produk dapat dihitung melalui penerapan algoritma djikstra didalam permasalahan yang sedang dihadapi ini. Algoritma djikstra merupakan sebuah algoritma yang dipakai dalam memecahkan permasalahan jarak terpendek (*shorter path problem*) untuk sebuah grafik berarah (*direct graph*) [1]. Permasalahan rute terpendek dari sebuah titik ke titik akhir merupakan masalah yang bisa digunakan untuk menguji Algoritma Dijkstra. Hasil dari Algoritma Dijkstra merupakan solusi optimal dan dapat dijadikan pengambilan keputusan dari permasalahan yang sedang dihadapi.

PT. XYZ merupakan distributor produk wafer yang berlokasi di Karawang akan mengirimkan produknya ke *retailer* yang berada di Purwakarta. Permasalahan yang dihadapi oleh PT. XYZ yaitu perusahaan ingin menghemat waktu pengiriman dengan cara mencari rute terpendek yang harus dilewati. Alternatif jalan yang harus dilalui untuk pengiriman produk bermacam-macam, sehingga dipilih titik lokasi yang dapat dilalui untuk sampai ke lokasi tujuan. Algoritma Dijkstra dapat memecahkan permasalahan mengenai pencarian rute terpendek. Tujuan dari penelitian ini yaitu untuk mengetahui rute terpendek yang harus dilalui dan berapa jarak yang harus ditempuh oleh PT. XYZ untuk mengirimkan produknya ke *retailer*.

2. Tinjauan Pustaka

2.1 Teori Graf

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Bobot pada tiap sisi dapat berbeda-beda bergantung pada masalah yang dimodelkan dengan graf. Bobot dapat menyatakan jarak antara dua buah kota, biaya perjalanan antara dua

buah kota, waktu tempuh pesan (*message*) dari sebuah simpul komunikasi ke simpul komunikasi lain (dalam jaringan komputer), ongkos produksi, dan sebagainya [2].

Teori *graph* merupakan teori yang memiliki bagian salah satunya yaitu jalur lintas terpendek, ketika diberikan *graph* berbobot, permasalahan jarak terpendek yaitu bagaimana mendapatkan jalur pada *graph* yang dapat meminimalkan hasil jumlah bobot sisi pembentuk jalur tersebut [3]. Setiap sisi berhubungan dengan satu atau dua simpul. Dua buah simpul dikatakan berhubungan atau bertangga (*adjenct*) jika ada sisi yang menghubungkan keduanya [4]. Misalkan terdapat graf yang sangat kompleks, dimana ada beberapa alternatif untuk melakukan kunjungan-kunjungan (*visiting*) dari satu simpul ke simpul-simpul yang lainnya, tentu dapat segera dicari tahu alternatif yang terbaik untuk melakukannya. Dalam hal ini, alternatif yang cukup baik adalah dengan mencari jarak yang terdekat antar simpul itu [5].

2.2 Jalur Lintasan Terpendek (*Shortest Path*)

Shortest path atau lintasan terpendek pada permasalahan lintasan yaitu sebuah lintasan *graph* yang diminimalisasi nilai bobotnya. Lintasan terpendek terdiri dari beberapa jenis, diantaranya sebagai berikut:

- a. *Shortest path* antara dua buah *node*.
- b. *Shortest path* antara semua pasangan *node*.
- c. *Shortest path* dari *node* tertentu ke semua *node* yang lain.
- d. *Shortest path* antara dua buah *node* yang melalui beberapa *node* tertentu [6].

Shortest path merupakan suatu permasalahan untuk mencari lintasan terpendek yang terdiri dari dua *node* atau lebih pada sebuah graf berbobot yang gabungan bobot sisi graf yang dilalui berjumlah paling minimum. Optimasi pada graf yang berbobot dapat dinyatakan dalam jarak antar kota, waktu pengiriman pesan, biaya dan lain sebagainya [4].

Persoalan yang sering ditemui dalam kehidupan sehari-hari yaitu permasalahan dalam menentukan rute terpendek (*shortest route*). Permasalahan ini biasanya dimodelkan ke dalam suatu graf berbobot dengan nilai pada masing-masing sisi yang mewakili persoalan yang akan dipecahkan. Kata terpendek pada permasalahan ini tidak hanya diartikan sebagai jarak secara fisik, namun hal itu tergantung dari tipe permasalahan yang akan diselesaikan. Kata terpendek tersebut dapat memiliki makna lain yang tingkat aksesibilitas suatu titik di dalam graf dari titik lain [7].

Jalur terpendek (*shortest path*) dipublikasikan pada tahun 1959 yang berada di *Numerische Mathematik* yang di edit oleh *F.L. Bauer*. Pada saat itu algoritma untuk jalur terpendek hampir tidak dianggap. Jalur terpendek dapat didefinisikan sebagai masalah kombinatorial dalam grafik dengan bobot terbatas, atau sebagai masalah optimasi yang terus berkelanjutan dalam *Geometri Euclidian*. Kunci untuk menemukan jalur terpendek adalah dengan mengetahui bagaimana cara menggambarkannya, salah satunya menggunakan struktur pohon [8]. Proses penghitungan rute terpendek adalah proses mencari jarak terpendek atau biaya terkecil suatu rute dari node awal ke node tujuan dalam sebuah jaringan [9].

2.3 Algoritma

Algoritma merupakan kumpulan instruksi atau perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis untuk penyelesaian suatu masalah [10]. Algoritma terdiri dari lima komponen urutan yaitu *finiteness* yang berarti terbatas, *definiteness* yang berarti kepastian, *input* yang berarti masukan, *output* yang berarti keluaran, dan yang terakhir yaitu *effectiveness* yang berarti efektivitas. Komponen yang digunakan dalam merancang algoritma terdiri dari tiga komponen yaitu komponen masukan yang berarti *input*, komponen keluaran yang berarti *output*, komponen proses yang berarti *processing* [6]. Penulisan algoritma terkadang sulit untuk dipahami dan maksud dari algoritma tersebut. Selain itu juga sulit untuk menuliskan algoritmanya, agar mempermudah bisa dilakukan notasi-notasi algoritma. Notasi algoritma merupakan rancangan penyelesaian masalah (algoritma) yang dituliskan dalam notasi (tata cara penulisan). Notasi algoritma yang sering dijumpai ada 3 macam yaitu diagram alir atau *flowchart*, deskriptif dan *pseudo-code* [6]. Algoritma merupakan sebuah metode atau langkah-langkah untuk menyelesaikan suatu permasalahan komputasi yang mentransformasikan sejumlah *input* atau masukan menjadi sejumlah *output* atau keluaran. Algoritma dapat dikatakan benar atau *correct* apabila setiap *input* yang dihasilkan mengeluarkan *output* yang benar [8].

2.4 Algoritma Dijkstra

Algoritma Dijkstra ditemukan oleh *Edsger Dijkstra*, sehingga kata Dijkstra diambil dari nama penemunya. Algoritma dijkstra adalah salah satu algoritma yang sering digunakan untuk memecahkan masalah *search problem* [11]. Prinsip yang digunakan dalam Algoritma Dijkstra yaitu prinsip *greedy*, setiap langkah yang dipilih yaitu memiliki bobot minimum yang menghubungkan sebuah *node* yang sudah terpilih

dengan *node* lain yang belum terpilih [12]. Berbagai permasalahan untuk pencarian rute terpendek dapat diselesaikan dengan menggunakan Algoritma Dijkstra, pencarian Algoritma Dijkstra dapat diselesaikan dengan menggunakan pencarian lintasan terpendek diantara dua buah *node* tertentu atau biasa disebut dengan *a pair shortest path*, pencarian lintasan terpendek diantara semua pasangan *node* atau biasa disebut dengan *all pairs shortest path*, pencarian lintasan terpendek dari suatu *node* tertentu ke semua *node* yang lain atau biasa disebut dengan *single-source shortest path*, serta pencarian lintasan terpendek diantara dua buah *node* yang melalui beberapa *node* tertentu atau biasa disebut dengan *intermediate shortest path* [13].

Penyelesaian algoritma dikstra dapat diselesaikan dengan mencari jalur terpendeknya, Algoritma Dijkstra akan mencari nilai yang paling minimal dari suatu graf berbobot, sehingga jarak terpendek akan diperoleh dari dua atau lebih titik dari suatu graf dan nilai total yang didapat adalah yang bernilai paling kecil [14]. Misalnya, bila *vertices* dari sebuah graf melambangkan kota-kota dan bobot sisi (*edge weights*) melambangkan jarak antara kota-kota tersebut, maka Algoritma Dijkstra dapat digunakan untuk menemukan jarak terpendek antara dua kota [15]. Pada setiap iterasi, jarak titik memberikan jarak terpendek [16]. Manfaat Algoritma Dijkstra untuk mencari lintasan terdekat, maka dapat membantu para pengguna untuk mengetahui jalur terdekat tanpa harus membuang-buang waktu [17].

Langkah untuk menyelesaikan permasalahan rute terpendek dengan menggunakan Algoritma Dijkstra yaitu pertama kali harus menentukan titik mana yang akan dipilih menjadi *node* awal, kemudian diberi nilai jarak pada *node* pertama ke *node* terdekat satu per satu, setelah menemukan titik awal Algoritma Dijkstra akan melakukan pengembangan pencarian dari satu *node* ke *node* lain dan ke *node* selanjutnya secara langkah demi langkah. Prosedur atau tahapan dari Algoritma Dijkstra adalah sebagai berikut:

- a. Memberikan nilai atau bobot jarak pada setiap *node* ke *node* lainnya, kemudian tentukan nilai 0 pada *node* awal dan nilai tak hingga pada *node* lain yang belum terisi.
- b. Isi semua *node* yang belum diberi nilai atau bobot jarak, kemudian tentukan *node* awal sebagai *node* keberangkatan.
- c. Setelah menentukan *node* keberangkatan, pertimbangkan *node* yang berdekatan yang belum diberi nilai dan hitung jarak dari titik awal atau titik keberangkatan. Sebagai contoh, jika pada titik awal atau titik keberangkatan A menuju B memiliki

nilai atau bobot jarak sejauh 6 dan dari *node* B menuju *node* C memiliki nilai atau bobot yang berjarak 2, maka jarak menuju *node* C melewati *node* B menjadi $6+2=8$. Apabila jarak yang dihasilkan memiliki nilai atau bobot yang lebih kecil dari jarak sebelumnya maka data lama dapat dihapus dan mengganti dengan data jarak yang baru.

- d. Apabila *node* terdekat yang belum terlewati belum diberikan tanda, maka *node* tersebut harus ditandai. *Node* yang sudah dilewati tidak akan pernah di cek kembali, jarak yang disimpan pada *node* merupakan jarak terakhir dengan nilai atau bobot yang paling minimal.
- e. Tentukan *node* belum terlewati dengan jarak terkecil dari *node* awal atau titik keberangkatan sebagai *node* keberangkatan selanjutnya dan diulangi sampai di *node* akhir atau tujuan dengan nilai atau bobot yang paling minimal [18].

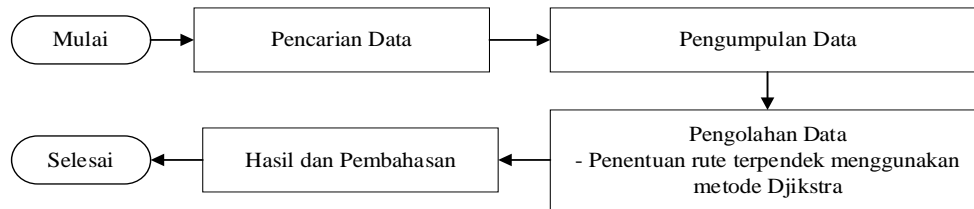
2.5 Google Maps

Google Maps merupakan sebuah peta digital yang dapat diakses secara *online* dan tidak berbayar yang disediakan oleh *google* yang dapat diakses melalui laman <http://maps.google.com>. Informasi yang dapat dilihat yaitu kondisi geografis pada hampir semua wilayah yang berada di muka bumi kecuali pada daerah kutub utara dan kutub selatan [19]. Situs *google maps* ini merupakan layanan interaktif, karena peta digital tersebut dapat digeser sesuai keinginan pengguna, dapat memperbesar atau memperkecil peta, serta dapat mengubah tampilan peta. *Google Maps* juga memberikan informasi peta yang dapat dilihat dengan gambar satelit untuk seluruh dunia, fitur lainnya *google maps* dapat memberikan alternatif perjalanan yang harus dilalui apabila pengguna akan pergi ke suatu tempat.

Google Maps digunakan sebagai parameter keoptimalan karena *Google Maps* merupakan peta digital yang sering digunakan oleh masyarakat, dengan adanya kecerdasan buatan dan algoritma pencarian rute terdekat pada *Google Maps*, kini semakin dimudahkan untuk mencari lokasi dari rute yang terdekat sehingga lebih efisien dalam hal waktu dan tenaga [20]. *Google Maps* dibuat dengan menggunakan kombinasi atau kumpulan dari gambar peta, *database*, serta objek-objek interaktif yang dibuat dengan bahasa pemrograman HTML, Javascript, dan AJAX, serta beberapa bahasa pemrograman lainnya. Komponen tersebut kemudian diintegrasikan ke dalam suatu *database* pada *google server*. Gambar pada *google maps* merupakan gabungan dari gambar-gambar yang berukuran 256 x 256 *pixel* [18].

3. Metode Penelitian

Metodologi penelitian merupakan tahapan atau langkah-langkah yang dilakukan pada saat penelitian. Alur metodologi penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur Metodologi Penelitian
(Sumber : Olah Data, 2021)

Berdasarkan Alur Metodologi Pendahuluan penelitian ini dilakukan dengan beberapa tahapan penjelasan dari tahapan tersebut, yaitu:

3.1 Pencarian Data

Pencarian data adalah suatu proses yang dilakukan untuk mengumpulkan sejumlah informasi yang diperlukan untuk mendukung jalannya penelitian serta mengidentifikasi masalah yang terjadi dari data yang dicari. Studi kasus yang diambil pada penelitian ini yaitu permasalahan rute terpendek yang harus dilalui apabila PT. XYZ yang berada di Walahar, Kabupaten Karawang akan mengirimkan produknya ke *retailer* yang ada di Dian Anyar, Purwakarta.

3.2 Pengumpulan Data

Pengumpulan data yaitu aktivitas mengumpulkan data setelah mencari data apa saja yang dibutuhkan untuk mencapai tujuan penelitian. Data dalam penelitian ini merupakan data sekunder yang diperoleh dari pengumpulan data yang berdasarkan jarak dari PT. XYZ ke *retailer* yang didapat dari *Google Maps*.

3.3 Pengolahan Data

Pengolahan data merupakan prosedur pengolahan data yang sesuai dengan pendekatan yang dilakukan yang akan menghasilkan informasi bagi para pembaca yang sesuai dengan tujuan penelitian. Pendekatan dalam studi kasus ini yaitu dengan menggunakan penerapan Algoritma Dijkstra. Pada teknik analisis data dilakukan dengan urutan sebagai berikut:

Tahap 1. Memberikan nilai atau bobot jarak pada setiap *node* ke *node* lainnya.

Tahap 2. Isi semua *node* yang belum diberi nilai atau bobot jarak kemudian tentukan *node* awal sebagai *node* keberangkatan.

Tahap 3. Pertimbangkan *node* yang berdekatan yang belum diberi nilai dan hitung jarak dari titik awal

Tahap 4. Apabila *node* terdekat yang belum terlewati belum diberikan tanda maka *node* tersebut harus ditandai.

Tahap 5. Tentukan *node* yang belum terlewati dengan jarak terkecil dari *node* awal atau titik keberangkatan sebagai *node* keberangkatan selanjutnya dan diulangi sampai di *node* akhir atau tujuan dengan nilai atau bobot yang paling minimal.

4. Hasil dan Pembahasan

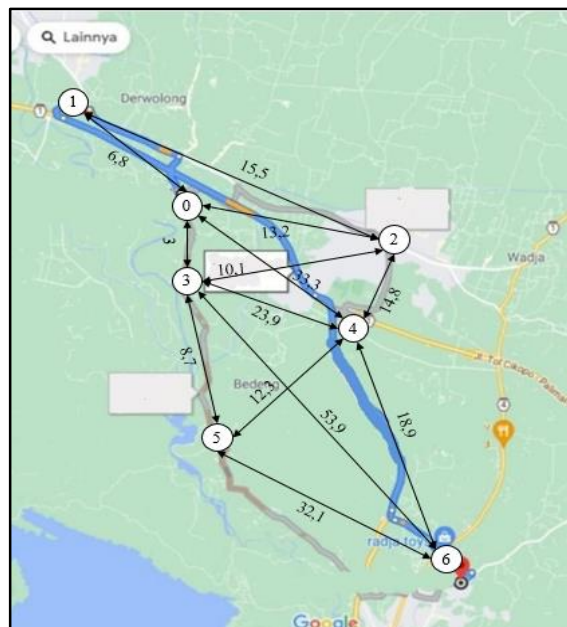
Perusahaan XYZ berlokasi di Karawang akan akan mengirimkan produk wafer ke *retailer* yang berlokasi di Purwakarta. Terdapat beberapa jalan alternatif yang dapat dilalui menuju lokasi *retailer* ditunjukkan pada Tabel 1.

Tabel 1. Alternatif Jalan

No	Alternatif Jalan	Jarak
1	Jalan Kapten Halim	54,4 km
2	Jalan Raya Peruri	67,4 km
3	Jalan Raya Syeh Quro	52,2 km

(Sumber : *Google Maps*, 2021)

Data yang digunakan untuk penelitian ini berupa data sekunder yang diperoleh dari *Google Maps*. Jarak dan *node* yang dilewati untuk sampai ke tujuan akhir dapat dilihat pada Gambar 2.



Gambar 2. Jarak dan *Node* yang dilalui
(Sumber : Olah Data, 2021)

Gambar 2. merupakan nama lokasi setiap *node*, berikut ini merupakan penjelasan dari gambar diatas:

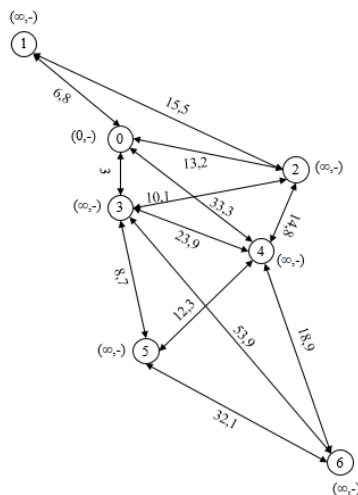
Tabel 2. Data Lokasi *Node*

No.	Nama Lokasi <i>Node</i>	<i>Node</i>
1.	PT. XYZ	0
2.	Jalan Anggadita, Klari, Kabupaten Karawang	1
3.	Cikampek, Dawuan Tengah	2
4.	Jalan Raya Curug – Kosambi	3
5.	Kalihurip, Cikampek	4
6.	Cilangkap, Purwakarta	5
7.	Dian Anyar, Purwakarta	6

(Sumber : Olah Data, 2021)

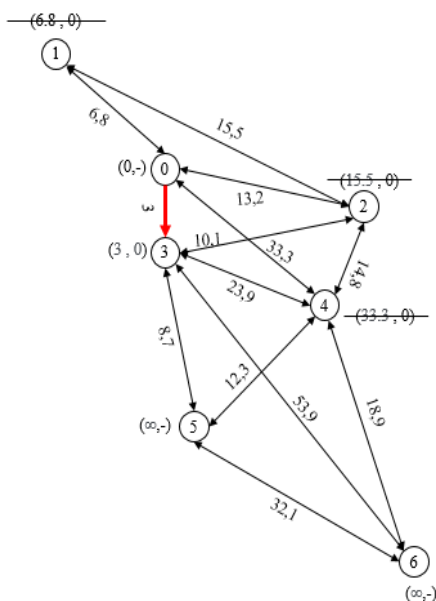
Setelah menentukan *node* dan lokasi yang bisa dilewati maka langkah yang bisa dilakukan untuk menentukan rute terpendek dengan menggunakan Algoritma Dijkstra adalah sebagai berikut:

Iterasi 0



Pada iterasi 0 dilakukan pemberian nilai pada setiap *node* berdasarkan hasil pengumpulan data yang sudah didapatkan. Selain itu, menentukan titik awal sebagai titik keberangkatan yaitu pada *node* 0 dan titik akhir atau tujuan yaitu pada *node* 6.

Iterasi 1

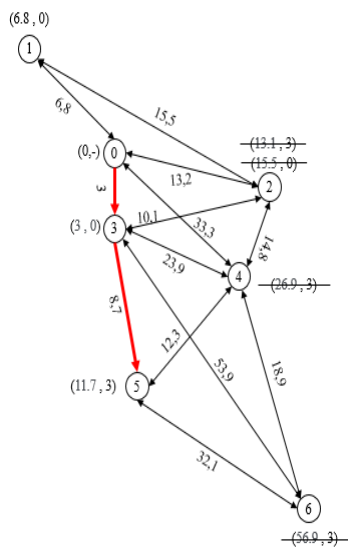


Step 1: Memilih *node* 0 sebagai simpul asal kemudian diberi label [0, -].

Step 2: Dari *node* 0 terdapat empat *node* yang terhubung yaitu *node* 1 dengan label [6.8,0]; *node* 2 dengan label [15.4,0]; *node* 3 dengan label [33.3,0]; *node* 4 dengan label [3,0].

Step 3: jarak terpendek dari *node* 0 yaitu *node* 3 dengan label [3,0], maka *node* 3 merupakan permanen label terbaru.

Iterasi 2



Step 1: Node 3 merupakan node dengan jarak terpendek, sehingga node 3 dipilih sebagai permanen label terbaru.

Step 2: Node yang terhubung dengan node 3 terdiri dari empat node, yaitu node 2, node 4, node 5 dan node 6. Node yang dipilih yaitu node yang memiliki jarak terpendek dengan cara menjumlahkan jarak dari node permanen terbaru.

Node 2: $3 + 10.1 = 13.1 \rightarrow [13.3, 3]$

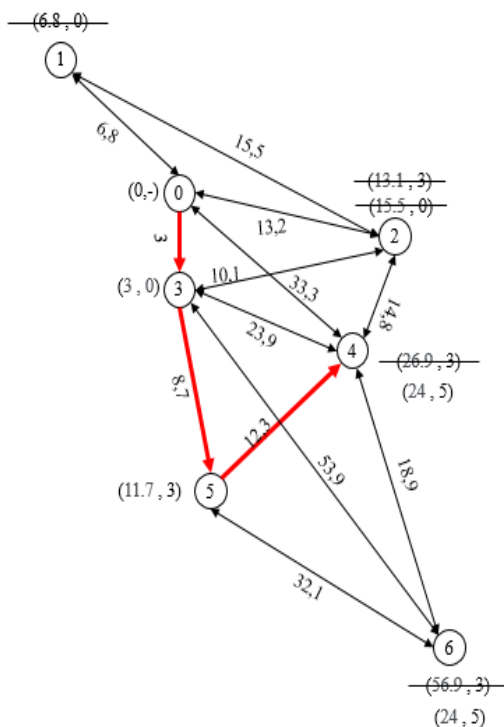
Node 4: $3 + 23.9 = 26.9 \rightarrow [26.9, 3]$

Node 5: $3 + 8.7 = 11.7 \rightarrow [11.7, 3]$

Node 6: $3 + 53.9 = 56.9 \rightarrow [56.9, 3]$

Maka, dipilih node 5 dengan label [11.7, 3]

Iterasi 3



Step 1: Node 5 merupakan node dengan jarak terpendek, sehingga node 5 dipilih sebagai permanen label terbaru.

Step 2: Node yang terhubung dengan node 5 terdiri dari dua node, yaitu node 4 dan node 6. Node yang dipilih yaitu node yang memiliki jarak terpendek dengan cara menjumlahkan jarak dari node permanen terbaru.

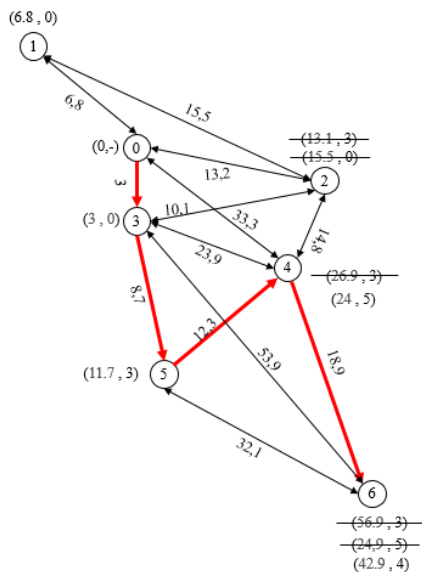
Node 4: $11.7 + 12.3 = 24 \rightarrow [24, 5]$

Node 6: $11.7 + 32.1 = 43.8 \rightarrow [43.8, 5]$

Maka, dipilih node 4 dengan label [24, 5]

Iterasi 4

Step 1: Node 4 merupakan node dengan jarak terpendek, sehingga node 4 dipilih sebagai permanen label terbaru.

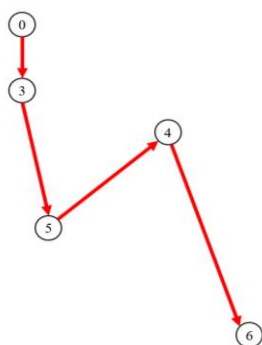


Step 2: *Node* yang terhubung dengan *node* 4 terdiri dari satu *node*, yaitu *node* 6. *Node* yang dipilih yaitu *node* yang memiliki jarak terpendek dengan cara menjumlahkan jarak dari *node* permanen terbaru.

Node 6: $24 + 18.9 = 42.9 \rightarrow [42.9, 4]$

Maka, dipilih *node* 6 dengan label $[42.9, 4]$

Hasil Iterasi



Untuk sampai ke simpul tujuan yaitu *node* 6 dari simpul asal yaitu *node* 0 jarak yang harus ditempuh dengan melewati 0-3, 3-5, 5-4, 4-6 yaitu $3 + 8,7 + 12,3 + 18,9 = 42,9$ km. Jalan yang harus dilalui yaitu dari PT. XYZ \rightarrow Jalan Raya Curug – Kosambi \rightarrow Cilangkap, Purwakarta \rightarrow Kalihurip, Cikampek \rightarrow Dian Anyar, Purwakarta.

5. Kesimpulan dan Saran

Kesimpulan dari hasil penelitian ini yaitu Algoritma Dijkstra berhasil diterapkan untuk mencari rute terpendek dari PT. XYZ yang berada di Karawang ke *retailer* yang ada di Purwakarta. Titik harus yang dilalui oleh PT. XYZ untuk sampai ke tujuan di *retailer* yaitu titik 0-3-5-4-6 atau melalui Walahar \rightarrow Cikampek \rightarrow Kalihurip \rightarrow Jalan Raya Curug – Komsani \rightarrow Dian Anyar, Purwakarta dengan jarak yang harus ditempuh yaitu sebesar 42,9 km. Rute tersebut merupakan hasil optimal yang harus dilalui, sehingga permasalahan dapat terselesaikan.

Daftar Pustaka

[1] E. Indra, M. Turnip, and Y. Laia, “Penerapan Metode Dijkstra Pada Sistem Informasi Geografis,” no. November, 2016.
 [2] H. Fukahori *et al.*, “PENENTUAN RUTE WISATA MINIMUM DI PULAU LOMBOK NUSA TENGGARA BARAT DENGAN PENDEKATAN

- ALGORITMA DIJKSTRA,” vol. 12, no. 12, pp. 103–107, 2018.
- [3] P. M. Sari, F. Fauziah, and A. Gunaryati, “Implementasi Algoritma Dijkstra pada Aplikasi Go-Tahu dengan Pencarian Rute Terpendek ke Pabrik Tahu,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 5, no. 2, p. 103, 2021, doi: 10.35870/jtik.v5i2.210.
- [4] 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.
- [5] E. R. Syahputra, “Analisis Perbandingan Algoritma Prim Dengan Algoritma Dijkstra Dalam Pembentukan Minimum Spanning Tree (Mst),” vol. 01, pp. 50–55, 2016.
- [6] M. K. Harahap and N. Khairina, “Pencarian Jalur Terpendek dengan Algoritma Dijkstra,” *Sinkron*, vol. 2, no. 2, p. 18, 2017, doi: 10.33395/sinkron.v2i2.61.
- [7] P. Algoritma, D. Dan, L. Marlina, A. Suyitno, M. Penerapan, and A. Dijkstra, “Penerapan Algoritma Dijkstra Dan Floyd-Warshall Untuk Menentukan Rute Terpendek Tempat Wisata Di Batang,” *Unnes J. Math.*, vol. 6, no. 1, pp. 36–47, 2017, doi: 10.15294/ujm.v6i1.13544.
- [8] A. G. Wibowo and A. P. Wicaksono, “Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Rumah Sakit di Purbalingga dengan Metode Algoritma Dijkstra (An Application Design for Determining the Shortest Path of Hospital in Purbalingga Using Dijkstra Algorithm Method),” *Juita*, vol. II, no. 1, pp. 21–35, 2012.
- [9] W. E. Y. Retnani, D. Istiadi, and A. Roqib, “Pencarian SPBU Terdekat dan Penentuan Jarak Terpendek Menggunakan Algoritma DIJKSTRA (Studi Kasus di Kabupaten Jember),” *J. Nas. Tek. Elektro*, vol. 4, no. 1, p. 89, 2015, doi: 10.25077/jnte.v4n1.132.2015.
- [10] M. H. Nugroho, Y. Arnandha, and A. Rakhmawati, “ANALISIS PETA JALUR EVAKUASI DAN PENENTUAN TITIK KUMPUL DENGAN METODE ALGORITMA DIJKSTRA,” *Concept Commun.*, vol. null, no. 23, pp. 301–316, 2019, doi: 10.15797/concom.2019..23.009.
- [11] A. D. Hartanto, A. S. Mandala, D. R. P.L., S. Aminudin, and A. Yudirianto, “Implementasi Algoritma Dijkstra Pada Game Pacman,” *CCIT J.*, vol. 12, no. 2, pp. 170–176, 2019, doi: 10.33050/ccit.v12i2.687.
- [12] B. Web, P. Kota, D. O. Pugas, M. Somantri, and K. I. Satoto, “Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra dan Astar (A*) pada SIG Berbasis Web untuk Pemetaan Pariwisata Kota Sawahlunto,” *Transmisi*, vol. 13, no. 1, pp. 27–32, 2011, doi: 10.12777/transmisi.13.1.27-32.
- [13] R. A. D. Novandi, “Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan Terpendek (Single Pair Shortest Path),” *IF2251 Strateg. Algoritm.*, vol. 1, pp. 1–5, 2013.
- [14] M. N. Parapat, D. Kusbianto, and C. Rahmad, “Rancang Bangun Aplikasi Pencarian Rute Terpendek Jasa Kiriman Barang Berbasis Mobile Dengan Metode Algoritma Dijkstra,” *J. Inform. Polinema*, vol. 3, no. 3, p. 15, 2017, doi: 10.33795/jip.v3i3.28.
- [15] B. Junanda, D. Kurniadi, and Y. Huda, “Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Pada Sistem Informasi Geografis Pemetaan Stasiun Pengisian Bahan Bakar Umum,” *Voteteknika (Vocational Tek. Elektron. dan Inform.)*, vol. 4, no. 1, 2018, doi: 10.24036/voteteknika.v4i1.6014.
- [16] E. C. Galih and R. A. Krisdiawan, “Implementasi Algoritma Dijkstra Pada Aplikasi Wisata Kuningan Berbasis Android,” *Nuansa Inform.*, vol. 12, no. 1, 2018, doi: 10.25134/nuansa.v12i1.1344.
- [17] A. Noviriandini and M. Safitri, “Implementasi Algoritma Dijkstra Untuk Menentukan,” *J. Pilar Nusa Mandiri*, vol. 13, no. 2, pp. 181–186, 2017.

- [18] D. Ardana and R. Saputra, “Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang,” *Skripsi Jur. Ilmu Komputer, Fak. Sains Dan Mat. Univ. Diponegoro*, no. Snik, pp. 299–306, 2016.
- [19] J. Setiawan *et al.*, “Perbelanjaan Di Jakarta Menggunakan Algoritma Dijkstra,” vol. 21, no. 3, pp. 156–165, 2019.
- [20] M. N. Latuconsina *et al.*, “Kontrakan Berbasis Algoritma Dijkstra Dan,” vol. 08, no. 1, pp. 15–27, 2021.