

Application of Dijkstra's Algorithm in Searching the Shortest Path of Coal Production Locations

Amelia Yusnita¹, Siti Lailiyah², I Gede Adi Merta³

¹Prodi Sistem Informasi, STMIK Widya Cipta Dharma, Samarinda, Kalimantan Timur

^{2,3}Prodi Teknik Informatika, STMIK Widya Cipta Dharma, Samarinda, Kalimantan Timur

^{1,2,3}Jl. M. Yamin No 25, Samarinda, Kalimantan Timur, Indonesia

Email: amelia@wicida.ac.id¹, lail.59a@gmail.com², gede.adimerta@gmail.com³

Received: 25 November 2020

Revised: 22 Desember 2020

Accepted: 20 Januari 2021

Abstract-To exploit coal the company needs to hold discussions and it takes time to reach the location, each coordinate point traveled has a different distance, the shorter the distance, the faster the time it will take. In this problem the company does not yet have hauling distance data at each coordinate point, the solution to this problem is how to build an application for determining coal production points using the dijkstra algorithm. There are three stages used to collect data, namely, literature study, observation, and interviews. The development of the system used to build this application uses the prototype method whose stages consist of communication, rapid planning, rapid design modeling using flowcharts, prototype formation, and submission of the customer system in the form of an application that analyzes the determination of PIT location distances using algorithms dijkstra, the output of this application is a description of the track map display, the destination location and its node points with detailed routes with a distance of kilometers. At this stage, the application is tested using the blackbox test, by checking whether the buttons can function properly and the expected result from this test is a success.

Keywords; *Dijkstra's Algorithm, Shortest Path*

I. PENDAHULUAN

Divisi Mining bagian produksi merupakan bagian divisi menunjang operasional sehari-hari disalah satu perusahaan yang bergerak dibidang pertambangan batubara. Dalam mengeksplotasi batubara perusahaan membutuhkan waktu untuk mencapai lokasi. Setiap titik koordinat yang ditempuh memiliki jarak yang berbeda disetiap titik-titik koordinatnya, semakin pendek jarak tempuh, maka semakin cepat waktu yang digunakan.

Permasalahan yang dihadapi oleh perusahaan adalah belum adanya data jarak tempuh *hauling* pada setiap titik koordinat, sehingga jika ingin menentukan lokasi eksploitasi perlu forum diskusi untuk menentukannya. Solusi dari permasalahan ini adalah bagaimana membangun aplikasi penentuan titik-titik produksi batubara menggunakan algoritma *dijkstra*.

Tujuan menggunakan algoritma *dijkstra* dalam penentuan titik produksi adalah untuk

mempermudah mendapatkan informasi secara cepat dan akurat mengenai bagaimana pemetaan titik-titik produksi, menambah keuntungan yang akan dikelola oleh bagian produksi, meminimalisir kesalahan pada level staf atau manajer bagian, saat mengambil keputusan berdasarkan informasi yang kurang akurat, dan meminimalisir anggaran yang dikeluarkan dalam proses pengangkutan batubara..

Beberapa penelitian yang lain tentang Algoritma dijkstra yaitu Algoritma *dijkstra* digunakan untuk sistem informasi geografis pemetaan pengisian bahan bakar umum, sistem ini memudahkan wisatawan mencari lokasi SPBU bagi pengguna kendaraan bermotor yang singgah dan melintasi kota Padang. Sistem ini menggunakan *MapInfo* dan *ArcGis* sebagai aplikasi untuk mendigitasi peta. Informasi yang dihasilkan berupa penyebaran, pencarian dan petunjuk rute yang ditempuh antar lokasi SPBU, menyediakan data tata ruang yang melingkupi jalan utama, letak dan informasi umum berupa nama dan alamat SPBU di kota Padang [1]. Penelitian yang kedua yaitu, Algoritma *dijkstra* untuk aplikasi pencarian rute bus trans semarang, pada aplikasi ini penumpang dapat mencari informasi melalui aplikasi digital untuk menentukan rute dan lokasi perpindahan koridor atau *transfer point* menggunakan algoritma *dijkstra*. Output dari aplikasi ini akan menampilkan peta rute yang akan dilalui dari titik awal menuju lokasi tujuan [2].

II. TINJAUAN PUSTAKA

A. Teori Dasar Graf

Graf didefinisikan sebagai pasangan himpuna (V,E) ditulis dengan notasi $G=(V,E)$ yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*Vertices* atau *Node*) dan E adalah himpunan sisi (*Edges* atau *arcs*) yang menghubungkan sepasang simpul [3].

B. Algoritma Dijkstra

Algoritma dijkstra (Minimum dari path dari *graph* berbobot) adalah algoritma untuk pencarian

jalur (*path*) yang termurah dari suatu *vertex* awal ke *vertex* akhir. Algoritma ini berbasis pada teknik *greedy*, dimana digunakan sebuah himpunan *vertex* S yang mula-mula diisi dengan *vertex* awal pada setiap langkah *vertex* v berikutnya yang memiliki bobot terkecil ditambahkan. Sebuah array D biasa digunakan untuk merekam jalur terpendek dari *vertex* awal ke *vertex* akhir [4].

III. METODE PENELITIAN

A. Pengumpulan Data

Pengumpulan data dapat dilakukan dengan cara:

1. Studi pustaka, yaitu dengan cara mendapatkan beberapa buku teks, jurnal dan sebagainya dengan membaca sumber pustaka dapat dijadikan untuk menganalisis permasalahan [5].
2. Observasi, yaitu melakukan pengamatan langsung untuk mendapatkan data yang berhubungan dengan titik titik produksi batubara. Data-data tersebut dikumpulkan dengan pengamatan langsung ke tempat objek pada bagian produksi
3. Wawancara dilakukan tentang bagaimana proses dan mekanisme dari kegiatan eksploitasi untuk menentukan jalur terpendek dari titik-titik produksi

B. Metode Pengembangan Sistem

Model prototype merupakan salah satu model pengembangan perangkat lunak yang digunakan dalam penelitian ini. tahapan dalam model ini adalah sebagai berikut [6]:

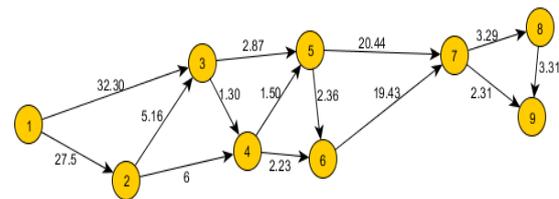
1. Komunikasi, pada tahap ini meliputi penepati identifikasi masalah dan kebutuhan sistem melalui komunikasi yang intensif dengan pengguna
2. Perencanaan secara cepat, pada tahap ini berfokus pada representasi semua aspek perangkat lunak yang terlihat oleh pengguna akhir seperti rancangan antarmuka pengguna dalam bentuk desain tampilan
3. Pemodelan perancangan secara cepat, pada tahap ini bertujuan untuk memberikan gambaran umum kepada pemakai tentang sistem yang digunakan dengan menggunakan alat bantu perancangan sistem yaitu *flowchart*
4. Pembentukan prototype, pada tahap ini, perancangan sistem yang telah dibuat direalisasikan dan aplikasikan dengan cara membuat prototype aplikasi penentuan titik-titik produksi batubara agar lebih mudah dipahami
5. Penyerahan sistem ke pelanggan, pengiriman dan umpan balik, pada tahap ini dilakukan evaluasi terhadap prototype yang telah dibagun dengan pengguna

IV. HASIL DAN PEMBAHASAN

A. Analisis Penentuan Jarak Lokasi

Lingkungan yang akan digunakan adalah sebuah graf berbobot dan berarah. Dimana graf digunakan untuk merepresentasikan lokasi yang akan dituju. Pemanfaatan teori graf dalam penelitian ini dilakukan sebagai berikut:

1. *Node* dalam suatu graf digunakan untuk mengetahui nama lokasi.
2. Busur dari suatu graf digunakan untuk melambangkan jalan
3. Bobot atau jarak dari busur graf merepresentasikan jarak antara suatu simpangan ke simpangan lainnya,
4. Arah graf dipresentasikan untuk arah yang melewati jalan dari *node* awal menuju ke *node* tujuan



Gambar 1. Arah graf

Teknik pembobotan yang dilakukan untuk nilai bobot adalah: $G=(V,E)$ merupakan graf dari V himpunan tak kosong dari *vertex* (titik atau node) dan E himpunan *edge* (Busur) yang merupakan pasangan tak terurut dari *vertex*. Titik-titik $(V,E) = (V_1, V_2, \dots, V_n)$ dan *path* terpendek yang dicari adalah dari V_1 ke V_N

TABLE I. NODE DAN LOKASI

No	Kode Node	Nama Lokasi
1	P01	Pelabuhan
2	P02	PIT J-21
3	P03	PIT K-01 A
4	P04	PIT K-01 B
5	P05	PIT K-01 C
6	P06	PIT M-34
7	P07	PIT E-01 A
8	P08	PIT E-01 B
9	P09	PIT E-01 C

TABLE II. DATA JALAN

No	Node Awal	Node Tujuan	Jarak Nyata	Bobot (Skala 1 : 1 Km)
1	P01	P02	27.5 Km	27.5
2	P01	P03	32.30 Km	32.30
3	P02	P03	5.16 Km	5.16
4	P02	P04	6 Km	6
5	P03	P04	1.30 Km	1.30
6	P03	P05	2.87 Km	2.87
7	P04	P05	1.50 Km	1.50

8	P04	P06	2.23 Km	2.23
9	P05	P06	2.35 Km	2.35
10	P05	P07	20.44 Km	20.44
11	P06	P07	19.43 KM	19.43
12	P07	P08	3.29 Km	3.29
13	P07	P09	2.31 Km	2.31
14	P08	P09	3.31 Km	3.31

Keterangan:

1. *Node* awal : Nama lokasi awal yang ingin dikunjungi
2. *Node* tujuan : *Node* yang menghubungkan ke node awal sebagai tujuan tempat yang ingin dikunjungi.
3. Jarak nyata : Jarak jalan dalam satuan kilometer.
4. Bobot : Nilai dari bobot dari setiap arah node dalam skala 1:1 Km

B. Analisis Pencarian Jarak Lokasi PIT dengan Algoritma Dijkstra

Pada algoritma *dijkstra*, semua elemen matrik M diisi dengan jarak antara simpul awal dengan simpul lainnya, jika sisi yang menghubungkan kedua simpul tersebut, apabila tidak ada, maka elemen matrik diisi dengan simbol ∞ (*infinity*).

Selanjutnya, dijalankan proses interaktif yang akan memeriksa setiap simpul kecuali simpul awal. Dalam proses ini, dicari terlebih dahulu simpul yang memiliki jarak terpendek dengan simpul yang sebelumnya (untuk pertama kali adalah simpul awal), dan nilai S dari simpul tersebut diisi dengan nilai 1 (satu). Simpul ini untuk selanjutnya disebut simpul antara. Dari simpul antara tersebut, jarak antara simpul awal dengan simpul yang lain di periksa, jika jarak antara simpul awal dengan sebuah simpul lebih besar dari jarak simpul awal dengan simpul antara ditambah jarak simpul antara dengan simpul tujuan tersebut, maka jarak antara simpul awal diisi dengan simpul tujuan. Proses ini diulangi sebanyak $n - 1$ kali, dengan n adalah jumlah simpul dari graf. Maka dari itu dapat di ambil kesimpulan yang dilakukan algoritma *dijkstra*:

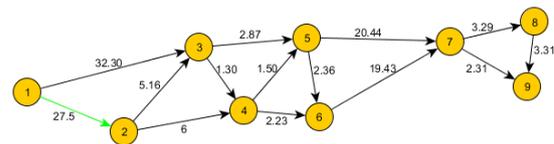
1. Dapat menentukan lintasan terpendek antara dua buah simpul tertentu.
2. Dapat menentukan lintasan terpendek antara semua pasangan simpul.
3. Dapat menentukan lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
4. Dapat menentukan lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu.

Dari hasil yang ada dapat menghasilkan nilai bobot yang minimum dari suatu perhitungan simpul antara dengan simpul awal dalam menentukan jarak sebenarnya.

C. Proses Penelusuran Graf Algoritma Dijkstra

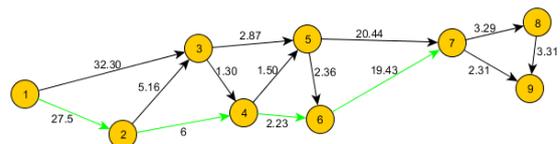
Dalam menentukan graf ini terdapat 9 lokasi yang dikunjungi yang menghubungkan *node* P01 ke tujuan P08 serta jarak dari setiap lokasi yang penelusuran. Berikut lintasan penelusuran yang dilakukan :

1. Mula-mula proses berawal dari *node* P01 sebagai *node* keberangkatan. Terdapat 2 jalur yaitu jalur P01 menuju ke P02 dengan bobot 27.5, dan P01 menuju P03 dengan bobot 32.30, maka jalur P01 menuju ke P02 yang terpilih pertama kali dengan bobot 27.5. Berikut penelusuran graf P01 ke P02



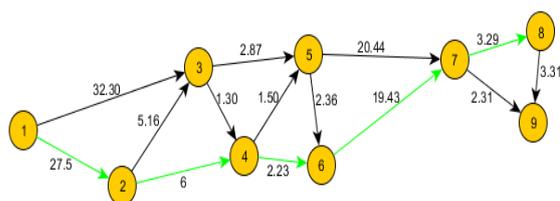
Gambar 2. Penelusuran graf P01 ke P02

2. Dari P01 menuju ke P07 terdapat 4 jalur yang memungkinkan yaitu P01 menuju ke P02 dengan bobot 27.5, P02 menuju ke P04 dengan bobot 6, P04 menuju ke P06 dengan bobot 2.23, P06 menuju ke P07 dengan bobot 19.43 (P01 → P02 → P04 → P06 → P07) $27.5 + 6 + 2.23 + 19.43 = 55.16$, maka lintasan P01 → P02 → P04 → P06 → P07 terpilih untuk lintasan yang dilalui dengan bobot 55.16.



Gambar 3. Penelusuran graf P01 ke P07

3. Dari P01 menuju ke P08 terdapat 5 jalur yang memungkinkan yaitu P01 menuju ke P02 dengan bobot 27.5, P02 menuju ke P04 dengan bobot 6, P04 menuju ke P06 dengan bobot 2.23, P06 menuju ke P07 dengan bobot 19.43, P07 menuju ke P08 dengan bobot 3.29 (P01 → P02 → P04 → P06 → P07 → P08) $27.5 + 6 + 2.23 + 19.43 + 3.29 = 58.45$, maka lintasan P01 → P02 → P04 → P06 → P07 → P08 terpilih untuk lintasan yang dilalui dengan bobot 58.45. Sehingga proses penelusuran ditemukan dari *node* P01 menuju ke *node* P08.



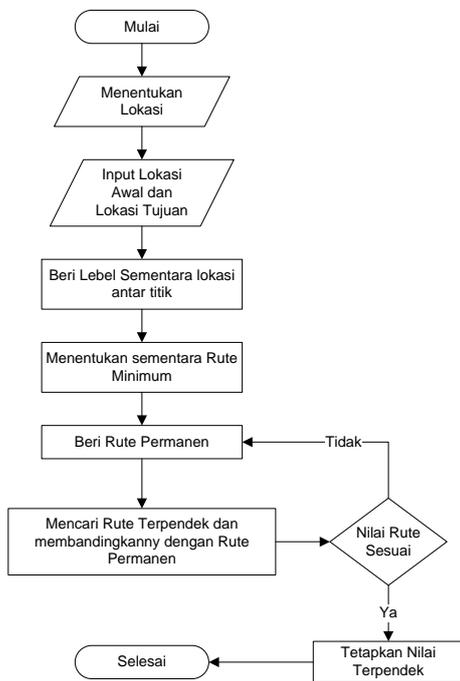
Gambar 4. Penelusuran graf P01 ke P08

TABLE III. PENELUSURAN GRAF

Node Awal	Node Tujuan	Jalur Rute	Bobot (Skala 1:1 Km)	Jumlah Jarak (Km)
P01	P02	P01 → P02	27.5	27.5 Km
P01	P03	P01 → P03	32.3	32.30 Km
P01	P04	P01 → P02 → P04	27.5 + 6	32.66 Km
P01	P05	P01 → P02 → P04 → P05	27.5 + 6 + 1.50	35 Km
P01	P06	P01 → P02 → P04 → P06	27.5 + 6 + 2.23	35.73 Km
P01	P07	P01 → P02 → P04 → P06 → P07	27.5 + 6 + 2.23 + 19.43	55.16 Km
P01	P08	P01 → P02 → P04 → P06 → P07 → P08	27.5 + 6 + 2.23 + 19.43 + 3.29	58.45 km

D. Flowchat Penelusuran Graf Algoritma Dijkstra

Dijelaskan untuk algoritma dimulai dengan menentukan lokasi, dan menginput lokasi awal dan lokasi tujuan, dari hasil inputan akan didapat titik lokasi sementara dan nilai rute minimum, tentukan nilai permanennya yang akan dibandingkan dengan rute terpendek, jika nilai sesuai tetapkan nilai terpendeknya dan jika tidak proses akan kembali ke penentuan nilai permanen



Gambar 5. Flowchat Penelusuran Graf Algoritma Dijkstra

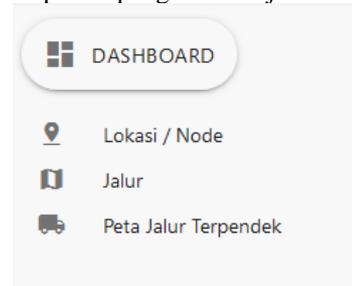
E. Implementasi

Adapun tahapan dalam implementasi Algoritma Dijkstra ini adalah :

1. Halaman Utama

Tampilan halaman utama terdiri dari Lokasi/Node, Jalur, dan Peta jalur terpendek.

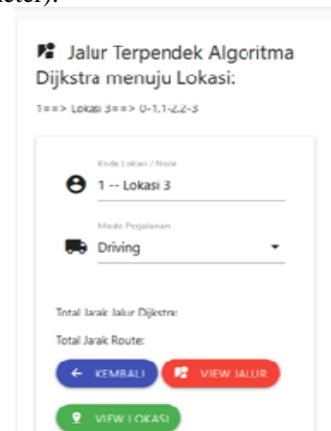
Lokasi/Node menyimpan titik-titik koordinat lokasi dari hasil survey dan titik hauling batubara. Jalur berdasarkan hubungan antar titik-titik koordinat digunakan dalam metode algoritma dijkstra. Peta jalur terpendek, pada menu ini akan menampilkan hasil form input map algoritma dijkstra.



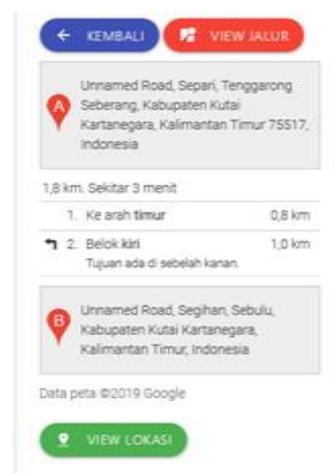
Gambar 6. Halaman menu utama

2. Keterangan Jalur terpendek

Pemakai memilih jalur 1, dengan tujuan lokasi 3, sedangkan titik-titik jalur/simpulnya adalah 0-1, 1-2, 2-3 dengan tidak alternatif jalur. Akan terlihat rincian jalur dengan jarak hitungan adalah km(kilometer).



Gambar 7. Keterangan jalur terpendek



Gambar 8. Keterangan jarak hitungan dalam Km

F. Pengujian

Metode pengujian yang digunakan adalah metode *Blackbox*. Metode ini mempunyai mekanisme untuk mengecek apakah fungsi-fungsi tombol dapat berjalan sebagaimana mestinya. Menu keterangan jalur akan diuji secara detail dan hasil akhir yang diharapkan adalah keberhasilan.

Menggunakan Metode Prototype,” *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 8, no. 2, p. 149, 2019.

TABLE IV. PENGUJIAN

Uraian	Hasil Yang Diharapkan	Hasil Pengujian
Pilih Lokasi	Dapat memilih lokasi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil
Tombol Kembali Tombol View Jalur	Dapat diproses Dapat menampilkan jalur peta titik produksi dengan algoritma	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak berhasil

V. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan, maka dapat disimpulkan bahwa aplikasi menentukan titik-titik produksi batubara menggunakan algoritma dijkstra ini dibuat dengan tujuan untuk mendapatkan informasi secara cepat dan akurat dalam mengambil keputusan, sehingga dapat meminimalisir anggaran yang dikeluarkan dalam proses pengangkutan batubara.

References

- [1] D. Ardana and R. Sa, “Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang,” *Semin. Nas. Ilmu Komput.*, vol. 2, no. Snik, pp. 299–306, 2016.
- [2] U. Efektivitas, S. Alkaloid, P. Tanaman, A. Gram, and G. Positif, “Universitas Negeri Semarang Semarang 2017,” no. 0024078603, pp. 1–52, 2017.
- [3] Fitria and A. Triansyah, “Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan,” *J. Sist. Inf.*, vol. 5, no. 2, pp. 611–621, 2013.
- [4] 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.
- [5] Suarga, *Algoritma dan Pemrograman Yogyakarta*. CV, Andi Offset, 2012.
- [6] A. Syarifudin, “Perancangan Sistem Informasi Pengajuan dan Pelaporan Pembayaran Tunjangan Kinerja Kementerian Keuangan