

Penentuan Jalur Terdekat Angkutan Kelapa Sawit Berbasis *Algoritma Dijkstra*

Taufiq^{1*}, Ir. Yulia Yudihartanti²

^{1,2}Prodi Teknik Informatika, STMIK Babjarbaru

Jl. Jend. A. Yani Km 33,5 Loktabat Banjarbaru Telp (0511) 4782881

*Email Corresponding Author: pa_tauw@yahoo.com

Abstrak

Banyaknya jalur alternatif yang dapat dilalui oleh armada pengangkutan buah Kelapa Sawit dari suatu Blok Kebun Produksi ke Pabrik Pengolahan pada Perusahaan Perkebunan Kelapa Sawit PT. Pucuk Jaya, menyebabkan tidakefisiennya waktu tempuh pengangkutan jika keliru dalam menentukan Jalur terpendek. Artikel ini menyajikan model aplikasi penentuan jalur angkut terdekat bagi armada angkutan untuk mengangkut buah Kelapa Sawit dari blok Kebun produksi ke Pabrik Pengolahan. Metode penalaran yang digunakan dalam aplikasi adalah algoritma Dijkstra. Sepuluh Blok Kebun dengan berbagai jalur alternatif yang tersedia menuju ke Pabrik Pengolahan diujicoba dalam pengujian kinerja algoritma. Hasil pengujian kinerja algoritma menunjukkan 30% data jalur yang diuji mampu dinalar oleh algoritma sesuai dengan jalur rekomendasi terbaik/terpendek yang disarankan oleh manajemen pabrik, sedangkan 70% data jalur lainnya mampu dinalar oleh algoritma secara lebih baik dari jalur rekomendasi terbaik yang disarankan oleh manajemen pabrik.

Kata Kunci: *Penentuan Jalur Terdekat, Armada angkutan Kelapa Sawit, Algoritma Dijkstra*

Abstract

The number of alternative routes that can be traversed by the oil palm fruit transportation fleet from a Production Plantation Block to a Processing Plant at the Oil Palm Plantation Company PT. Pucuk Jaya, causing inefficiency in transportation travel time if you make a mistake in determining the closest route. This article presents an application model for determining the closest transportation route for transportation fleets to transport oil palm fruit from the production plantation block to the processing factory. The reasoning method used in the application is Dijkstra's algorithm. Ten Farm Blocks with various alternative paths available to the Processing Plant were tested in algorithm performance testing. The results of the algorithm performance test show that 30% of the path data tested are able to be reasoned by the algorithm according to the best recommendation path suggested by the factory management, while 70% of the other path data are able to be reasoned by the algorithm better than the best recommendation path suggested by the factory management.

Keywords: *Determination of the Nearest Path, Oil Palm Transportation Fleet, Dijkstra's Algorithm*

1. Pendahuluan

Dalam kehidupan sehari-hari sering kali seseorang melakukan perjalanan dari suatu tempat ke tempat lain. Dan sangat lazim sekali apabila memilih jarak terdekat dari kedua tempat tersebut untuk dilalui, karena dapat menghemat waktu, tenaga serta bahan bakar tentunya. Untuk suatu perusahaan tertentu jarak terdekat dan waktu tempuh sangat diperhitungkan sekali untuk menghindari kerugian, seperti pengiriman buah sawit. Penentu jarak terpendek dan menghitung waktu tempuh dan jarak tempuh sangat dibutuhkan sekali agar buah sawit yang dibawa mencapai tujuan lebih cepat [1].

Kesulitan menentukan jarak dikarenakan terdapat banyak sekali jalur yang ada pada perusahaan PT. PUCUK JAYA karena dari beberapa blok A-Z ke pabrik tidak hanya memiliki satu jalur saja, banyak sekali jalur yang dapat dilalui sehingga diperlukan suatu sistem yang dapat membantu menentukan jalur alternatif terbaik dari tempat asal ke tempat tujuan secara lebih efisien.

Ada banyak algoritma yang dapat digunakan untuk menemukan jalur terbaik pada sebuah graf diantaranya Algoritma Dijkstra, Algoritma Dijkstra memiliki kelebihan menentukan

jalur tercepat dan mengetahui jarak atau lintasan terpendek serta penampilan rute lebih mudah dibaca dan dipahami dan lintasan pun dapat diberikan warna sehingga penampilannya lebih menarik [2].

Artikel ini menyajikan penggunaan Algoritma *Dijkstra* untuk menentukan jalur alternatif terbaik pada banyaknya jalur alternatif yang dapat dilalui oleh armada pengangkutan buah Kelapa Sawit dari suatu Blok Kebun Produksi ke Pabrik Pengolahan.

2. Tinjauan Pustaka

Penelitian Triasnyah yang berjudul 'Implementasi Algoritma Dijkstra Dalam Aplikasi Menentukan Lintasan Terpendek Jalan Darat Antar Kota di Sumatra Bagian Selatan' Dengan menerapkan Algoritma Dijkstra untuk mencari lintasan terdekat dapat membantu para pengguna jalan, traveling salesman, perusahaan yang bergerak dibidang pariwisata dan angkutan antar provinsi, instansi pemerintah dan lain sebagainya terutama bagi yang membutuhkan informasi lintasan terdekat. Untuk menuju ke suatu kota tujuan dapat ditempuh melalui beberapa lintasan. Dalam hal ini, kita akan menentukan kota-kota atau jalan manakah yang harus dilalui sehingga kita dapat mencari tempat tujuan dengan jarak terpendek. Dengan demikian lintasan terpendek dapat diartikan sebagai bobot minimal dari suatu lintasan, yaitu jumlah bobot dari seluruh busur yang membentuk lintasan. Dalam menentukan lintasan terpendek dapat diperoleh dengan beberapa algoritma matematika, antara lain algoritma Dijkstra, algoritma Floyd-Warshall dan algoritma Bellman-Ford. Algoritma yang akan di pakai dalam sistem ini adalah algoritma Dijkstra. Algoritma ini bertujuan untuk menemukan lintasan terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan kota, garis menggambarkan jalan dan bobot menggambarkan jarak, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik. Dengan kata lain algoritma ini menghitung lintasan berdasar jarak terpendek yang di tempuh di tiap-tiap kota. [3]

Penelitian untuk penentuan lokasi terpendek pada Universitas Sam Ratulangi Manado adalah salah satu perguruan tinggi di Sulawesi Utara yang terdiri atas 11 fakultas dan satu gedung rektorat. Setiap fakultas dan rektorat terhubung dengan fasilitas jalan raya. Secara matematis kondisi seperti ini dapat direpresentasikan sebagai sebuah graf yang bisa diterapkan untuk mencari lintasan terpendek. Pada penelitian ini akan dicari lintasan terpendek dari FMIPA ke rektorat dan fakultas lainnya. Dengan menggunakan algoritma Dijkstra, lintasan terpendek dari FMIPA diperoleh dengan memilih minimum lokal atau akses dengan jarak terdekat dari setiap lokasi yang kemudian digabungkan menjadi sebuah kumpulan lintasan dari satu lokasi ke lokasi lainnya dengan jarak terpendek. [4]

Penelitian lainnya dilakukan oleh Arif et al, tentang wisata air adalah salah satu jenis wisata unggulan yang ada di Klaten, umumnya wisata air di Klaten adalah objek wisata untuk pemandian umum. Wisata ini kebanyakan berada diwilayah yang mempunyai sumber mata air sendiri, misalnya wilayah Cokro, Polanharjo dan Janti. Untuk pengunjung yang berdatangan terdiri dari berbagai daerah bahkan ada yang dari mancanegara. Dalam pengambilan jalur untuk memudahkan bagi wisatawan lokal maupun mancanegara, ini sangat diperlukan untuk mencari jalur tercepat menuju salah satu titik wisataair, hal ini dapat memudahkan wisatawan dalam pencarian jalur dan hal ini juga dapat membuat pertambahan wisatawan yang akan datang ke wisata air yang ada di Klaten. Oleh karena itu, dalam makalah ini akan dibahas bagaimana penerapan algoritma Dijkstra dalam mencari rute terpendek pada pencarian objek wisata di Kabupaten Klaten dengan perhitungan manual dan dengan bantuan program Tora.[5]

Konsep sistem yang diajukan pada artikel ini adalah dengan menerapkan algoritma Dijkstra untuk menemukan lintasan terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Dalam hal ini menghitung jarak setiap beberapa blok A-Z ke pabrik tidak hanya memiliki satu jalur saja, banyak sekali jalur yang dapat dilalui menentukan jalur terdekat. Sehingga nantinya setiap unit driver akan mengetahui rute terpendek jika berapa pada blok manapun, tidak hanya mengacu pada satu tidak sentral yaitu pabrik. Suatu blok memiliki banyak jalur alternative dari blok semula ke blok yang dikehendaki. Maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik. Dengan kata lain algoritma ini menghitung lintasan berdasar jarak terpendek yang di tempuh di tiap-tiap titik rute.[6]

3. Metodologi

3.1 Analisa Kebutuhan

Analisa kebutuhan dilakukan untuk mengetahui semua permasalahan serta kebutuhan yang diperlukan dalam pengembangan aplikasi penentuan rute terdekat menggunakan metode algoritma Dijkstra. Penelitian tentang penentuan rute terdekat adalah untuk mengetahui keakuratan metode algoritma Dijkstra dalam menentukan dan mencari rute terdekat dan menghasilkan pengetahuan tentang pola rute/alternatif pilihan node lainnya. Pada penelitian ini, kinerja algoritma diuji menggunakan parameter jarak dan kondisi jalan pada setiap rute yang diuji.

3.2. Rancangan Sistem

Diagram konteks adalah diagram yang menyajikan aliran data dalam sistem yang akan dibuat dan hubungannya dengan bagian luar. Dengan diagram ini akan mempermudah pemahaman terhadap hasil analisa, sehingga apabila terjadi kesalahan dapat diketahui sedini mungkin. Pada diagram konteks digambarkan proses umum yang terjadi di dalam sistem. [7]

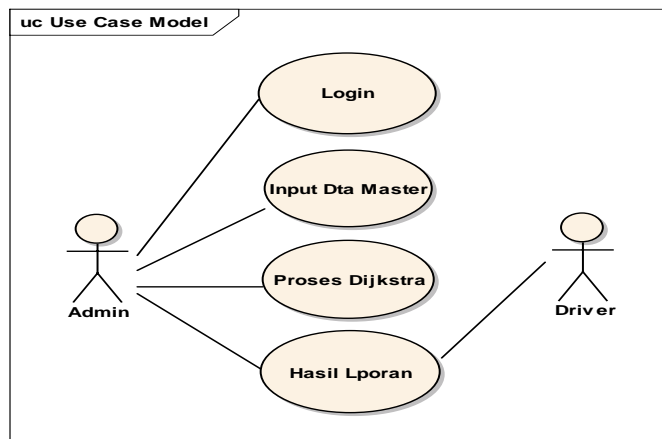
Berikut adalah diagram Konteks yang digunakan dalam sistem ini :



Gambar 1. Diagram Konteks

3.3. Use case

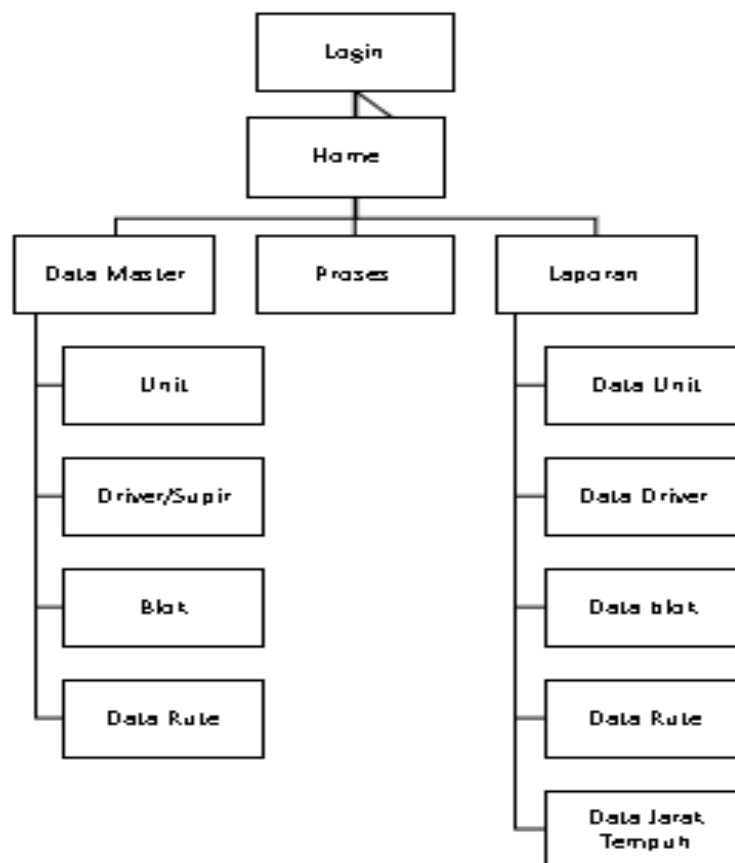
Use case digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Berikut diagram use case diagram pada aplikasi ini.



Gambar 2. Use Case Diagram

3.4. Struktur Aplikasi

Model arsitektur sistem ini terdapat dapat digambarkan dalam diagram Gambar 4 berikut ini:

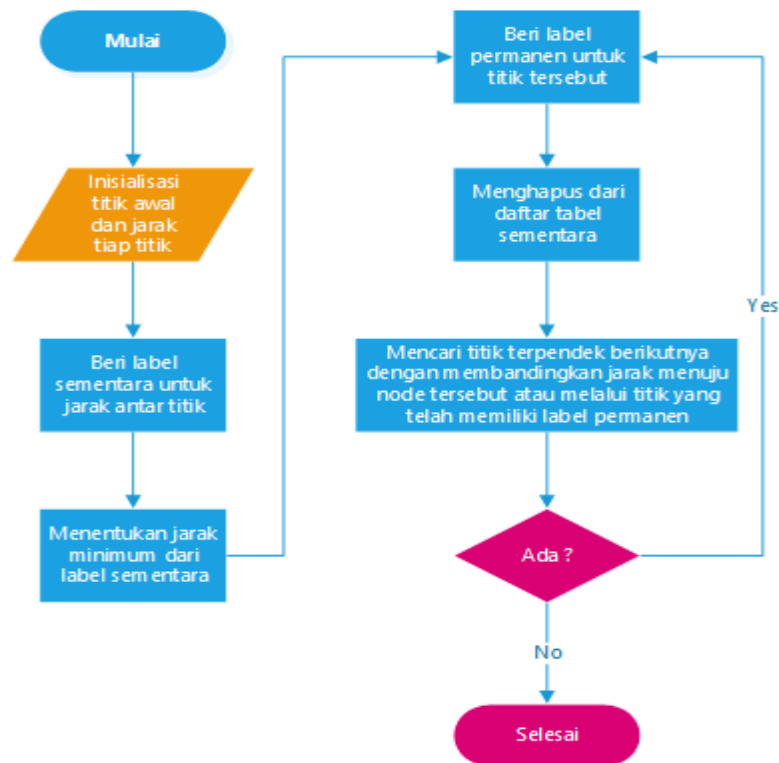


Gambar 3. Arsitektur Sistem Aplikasi

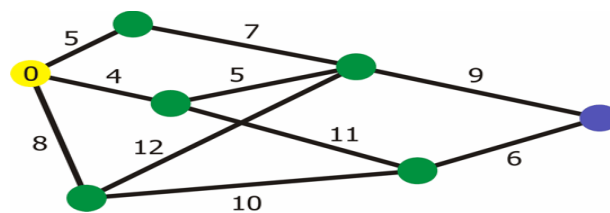
Diagram arsitektur menunjukkan fungsional *top-down* suatu sistem. Melalui diagram ini, dapat diketahui bagian-bagian sistem secara keseluruhan baik root *process* yang berhubungan dengan sistem, subsistem dan sistem serta rincian dari masing-masing.

3.4 Algoritma Dijkstra

Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infinity (Tak Hingga). Pada Algoritma Dijkstra, *node* digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute listasan terpendek. Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalnya titik menggambarkan gedung dan garis menggambarkan jalan, maka Algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik. *Flowchart* Algoritma Dijkstra disajikan pada Gambar 4 [8].



Gambar 4. Flowchart Algoritma Dijkstra



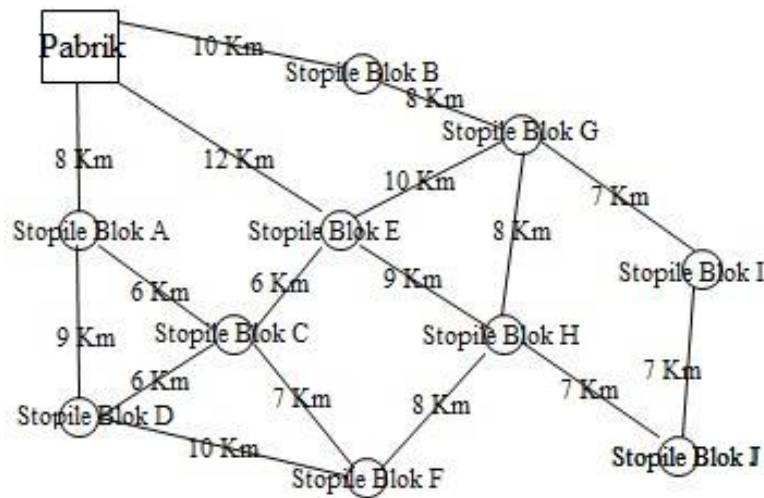
Gambar 5. Algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadikan node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu persatu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap inilah urutan logika dari algoritma Dijkstra:

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node "Belum Terjamah" dan set node awal sebagai "Node keberangkatan"
3. Dari no keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai "Node terjamah". Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set "Node belum terjamah" dengan jarak terkecil (dari node keberangkatan) sebagai "Node Keberangkatan" selanjutnya dan lanjutkan dengan kembali ke step 3 [9][10].

3.5 Teknik Analisa Data

Peta jalur angkutan kelapa sawit yang terdapat pada Area Perkebunan PT. Pucuk jaya



Gambar 6. Gambar Peta Area Perkebunan PT. Pucuk jaya

Selanjutnya setelah mengetahui jalur yang ada pada obyek penelitian maka akan dilakukan perhitungan dengan menggunakan algoritma Dijkstra, Pertama-tama tentukan titik mana yang akan menjadikan node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu persatu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya

Tabel 1 Hasil Perhitungan Algoritma Dijkstra

No	TITIK AWAL	Jalur AWAL	JARAK TEMPUH	JALUR TERDEKAT	JARAK TEMPUH
1	Stopile A	S/A, Pabrik (8 Km)	8 Km		
2	Stopile B	S/B, Pabrik (10 Km)	10 Km		
3	Stopile C	S/C, S/E (6 Km)	18 Km	S/C, S/A (6 Km)	14 Km
		S/E, Pabrik (12 Km)		S/A, Pabrik (8 Km)	
4	Stopile D	S/D, S/C (6 Km)	20 Km	S/D, S/A (9 Km)	17 Km
		S/C, S/A (6 Km)		S/A, Pabrik (8 Km)	
		S/A, Pabrik (8 Km)			
5	Stopile E	S/E, Pabrik (12 Km)	12 Km		
6	Stopile F	S/F, S/C (7 Km)	25 Km	S/F, S/C (7 Km)	21 Km
		S/C, S/E (6 Km)		S/C, S/A (6 Km)	
		S/E, Pabrik (12 Km)		S/A, Pabrik (8 Km)	
7	Stopile G	S/G, S/E (10 Km)	12 Km	S/G, S/B (8 Km)	18 Km
		S/E, Pabrik (12 Km)		S/B, Pabrik (10 Km)	
8	Stopile H	S/H, S/G (8 Km)	26 Km	S/H, S/E (9 Km)	21 Km
		S/G, S/B (8 Km)		S/E, Pabrik (12 Km)	
		S/B, Pabrik (10 Km)			

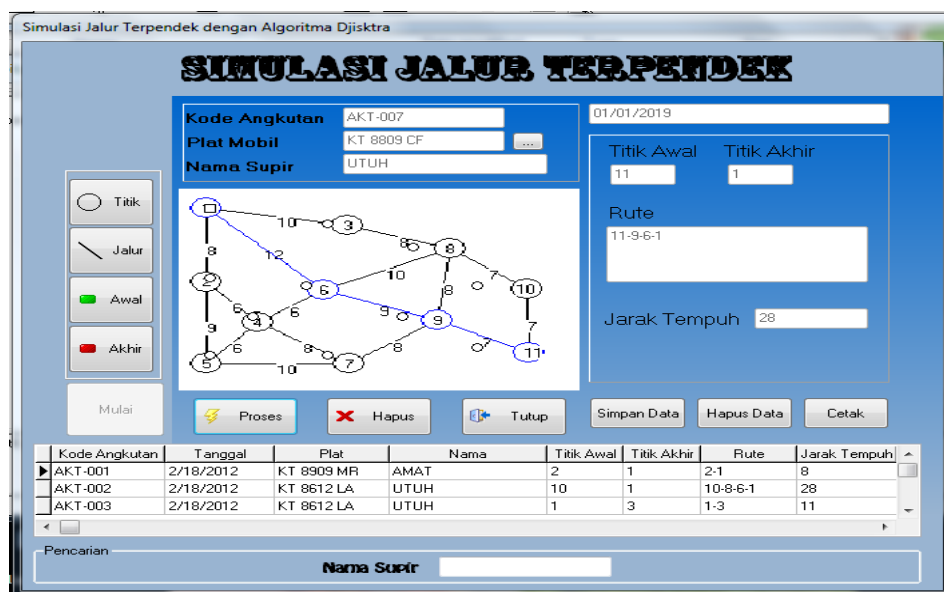
No	TITIK AWAL	Jalur AWAL	JARAK TEMPUH	JALUR TERDEKAT	JARAK TEMPUH
9	Stopile I	S/I, S/G (7 Km)	29 Km	S/I, S/G (7 Km)	25 Km
		S/G, S/E (10 Km)		S/G, S/B (8 Km)	
		S/E, Pabrik (12 Km)		S/B, Pabrik (10 Km)	
10	Stopile J	S/J, S/I (7 Km)	32 Km	S/J, S/H (7 Km)	28 KM
		S/I, S/G (7 Km)		S/H, S/E (9 Km)	
		S/G, S/B (8 Km)		S/E, Pabrik (12 Km)	
		S/B, Pabrik (10 Km)			

Penjelasan pada Tabel 1 perhitungan Algoritma Dijkstra sebagai berikut:
 Dari tabel perhitungan Algoritma Dijkstra dilihat dari stopile D yang mana jalur awal dari D ke stopile C berjarak 6 Km, dari stopile C ke stopile A berjarak 6 Km, dari Stopile A Ke pabrik 8 Km sehingga jalur awal stopile D memiliki total jarak tempuh $6+6+8 = 20$ Km. Setelah menggunakan dari sistem aplikasi yang dibuat sehingga jalur terdekat stopile D nya adalah dari stopile D ke stopile a berjarak 9 km, dari stopile Akepabrik berjarak 8 Km sehingga jalur terdekat stopile D memiliki total jarak tempuh $9+8= 17$ km.

4. Hasil dan Pembahasan

4.1 Antarmuka Aplikasi

Form Gambar 7 adalah form transaksi utama, yaitu form yang digunakan untuk mensimulasikan jalur alternatif terpendek yang direkomendasikan oleh algoritma untuk dilalui oleh armada pengangkutan.



Gambar 7. Form Proses Algoritma Dijkstra

4.2. Validasi Kinerja Algoritma

Mekanisme yang digunakan untuk memvalidasi kinerja Algoritma adalah dengan membandingkan keputusan jalur terbaik yang telah dikaji oleh pihak manajemen perusahaan dengan jalur alternatif yang direkomendasikan oleh Algoritma. Terdapat dua kelompok data yang diujikan pada uji kinerja algoritma ini, yaitu Kelompok data yang dinyatakan Akurat dan kelompok data yang dinyatakan Tidak Akurat. Jika Algoritma dapat menganalisis dan dapat merekomendasikan jalur terbaik sama seperti Jalur Terbaik yang telah dianalisis oleh pihak

manajemen perusahaan, atau mampu merekomendasikan Jalur Terbaik dari Jalur yang dinyatakan Tidak akurat, maka kinerja algoritma dinyatakan akurat.

Berdasarkan Gambar 6 dan Tabel 1, Keputusan jalur terbaik (Akurat) dan Jalur Tidak Akurat yang telah dikaji oleh pihak manajemen perusahaan disajikan pada Tabel 2, yang selanjutnya akan dibandingkan dengan Jalur yang direkomendasikan oleh Algoritma (Tabel 3).

Tabel 2. Data jalur terbaik (Akurat) dan Jalur Tidak Akurat yang telah dikaji oleh pihak manajemen perusahaan

No	Nama Stopile	Tujuan	Pretest	
			Jarak Tempuh	Waktu Tempuh
1	Stopile A	Pabrik	8 KM	24 Menit
2	Stopile B	Pabrik	10 KM	30 Menit
3	Stopile C	Pabrik	18 KM	54 Menit
4	Stopile D	Pabrik	20 KM	60 Menit
5	Stopile E	Pabrik	12 KM	36 Menit
6	Stopile F	Pabrik	25 KM	75 Menit
7	Stopile G	Pabrik	22 KM	66 Menit
8	Stopile H	Pabrik	26 KM	63 Menit
9	Stopile I	Pabrik	29 KM	87 Menit
10	Stopile J	Pabrik	32 KM	96 Menit

Pada Tabel 2, Jalur Nomor 1, Nomor 2, dan nomor 5 adalah Jalur yang dianggap terbaik (Akurat), sedangkan Jalur lainnya dianggap sebagai Jalur Tidak Akurat. Ke-10 data tersebut selanjutnya diujikan pada Algoritma untuk mengetahui Tingkat Akurasi Kinerja tersebut. Hasil Rekomendasi yang diberikan oleh Algoritma seperti disajikan pada Tabel 3.

Tabel 3. Data jalur terbaik yang direkomendasikan oleh Algoritma

No	Nama Stopile	Tujuan	Pretest		Posttest		Ket
			Jarak Tempuh	Waktu Tempuh	Jarak Tempuh	Waktu Tempuh	
1	Stopile A	Pabrik	8 KM	24 Menit	8 Km	24 Menit	Sama
2	Stopile B	Pabrik	10 KM	30 Menit	10 Km	30 Menit	Sama
3	Stopile C	Pabrik	18 KM	54 Menit	14 Km	42 Menit	Lebih Cepat
4	Stopile D	Pabrik	20 KM	60 Menit	17 Km	51 Menit	Lebih Cepat
5	Stopile E	Pabrik	12 KM	36 Menit	12 Km	36 Menit	Sama
6	Stopile F	Pabrik	25 KM	75 Menit	21 Km	63 Menit	Lebih Cepat
7	Stopile G	Pabrik	22 KM	66 Menit	18 Km	54 Menit	Lebih Cepat
8	Stopile H	Pabrik	26 KM	63 Menit	21 Km	63 Menit	Lebih Cepat
9	Stopile I	Pabrik	29 KM	87 Menit	25 Km	75 Menit	Lebih Cepat
10	Stopile J	Pabrik	32 KM	96 Menit	28 Km	84 Menit	Lebih Cepat

Pada Tabel 3 terlihat bahwa dari 10 Blok Kebun dengan berbagai jalur alternatif yang tersedia menuju ke Pabrik Pengolahan yang diujicoba dalam pengujian kinerja algoritma *Dijkstra*, hasil pengujian kinerja algoritma menunjukkan 30% data jalur (data Nomor 1, nomor 2, dan Nomor 5) yang diuji mampu dinalar oleh algoritma sesuai dengan jalur rekomendasi terbaik (akurat) yang disarankan oleh manajemen pabrik (ket. "Sama"), sedangkan 70% data jalur lainnya (Jalur Tidak Akurat) juga mampu dinalar oleh algoritma dengan merekomendasikan Jalur terbaik (Ket. "Lebih Cepat").

5. Kesimpulan

Berdasarkan 10 kombinasi data “akurat” dan data “tidak Akurat” yang diujikan pada Algoritma *Dijkstra* dalam merekomendasikan jalur alternatif terbaik/terpendek (studi kasus pada jalur pengangkutan kebun sawit PT. Pucuk Jaya) menunjukkan 30% data jalur yang diuji mampu dinalar oleh algoritma sesuai dengan jalur rekomendasi terbaik (Akurat) yang disarankan oleh manajemen pabrik, dan 70% data jalur lainnya yang diuji (jalur tidak akurat) mampu dinalar oleh algoritma secara lebih baik dengan memberikan jalur alternatif yang lebih akurat (waktu tempuh lebih cepat). Hal ini menunjukkan bahwa akurasi kinerja Algoritma *Dijkstra* dalam merekomendasikan jalur alternatif terbaik pada kasus yang diuji dapat mencapai 100% untuk 10 sampel data yang diuji.

Daftar Referensi

- [1] Wibowo G., Abdul, dan Agung P.W. Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Rumah Sakit di Purbalingga dengan Metod eAlgoritma Dijkstra. *JUITA*. 2012; 2(1): 21-35
- [2] Noto, M., & Sato, H. "A method for the shortest path search by extended Dijkstra algorithm". *ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'* 2002; 3: 2316-2320.
- [3] Triansyah, Apri. Implementasi algoritma Dijkstra dalam aplikasi untuk menentukan lintasan terpendek jalan darat antar kota di Sumatera Bagian Selatan. *JSI: Jurnal Sistem Informasi (E-Journal)*. 2013; 5(2): 611-621
- [4] Salaki, Deiby T. Penentuan Lintasan Terpendek Dari Fmipa Ke Rektorat Dan Fakultas Lain Di Unsrat Manado Menggunakan Algoritma Dijkstra. *Jurnal Ilmiah Sains*. 2011; 11(1): 73-76.
- [5] Ardyan S, Suyitno A, Mulyono. Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten Klaten. *UNNES J Math*. 2017; 6(2): 108-116.
- [6] Chen, Y. Z., Shen, S. F., Chen, T., & Yang, R. Path optimization study for vehicles evacuation based on Dijkstra algorithm. *Procedia Engineering*. 2014; 71: 159-165.
- [7] Muslihudin, Muhamad. Analisis Dan Perancangan Sistem Informasi Menggunakan Model Terstruktur Dan UML. Penerbit Andi, 2016.
- [8] Zaki, Abdul. "Algoritma Dijkstra: Teori Dan Aplikasinya." *Jurnal Matematika UNAND*. 2017; 6(4): 1-8.
- [9] Prajapati, G. L., Singhal, P., Ranjan, A., & Chourasia, N. "An Efficient Scheme for the Single Source Shortest Path Problem based on Dijkstra and SPFA Methodologies". *International Journal of Computer Applications*. 2017; 163(8): 46–52.
- [10] Harahap, M. K., & Khairina, N. Pencarian Jalur Terpendek dengan Algoritma Dijkstra. *Sinkron: jurnal dan penelitian teknik informatika*, 2017; 2(2): 18-23.