

Penerapan Metode Dijkstra Pada Jalur Distribusi LPG Untuk Penentuan Jarak Terpendek

Novi Hendri Adi¹, Muhammad Giatman², Wakhinuddin Simatupang², Afrina¹, Ronal Watrianthos³

¹ Teknik Informatika, Universitas Ibnu Sina, Batam, Indonesia

² Pendidikan Teknologi dan Kejuruan, Universitas Negeri Padang, Padang, Indonesia

³ Teknik Informatika, Universitas Al Washliyah, Rantauprapat, Indonesia

Email: ¹novi.hendriadi@gmail.com, ²giatman@ft.unp.ac.id, ³wakhinuddins@gmail.com, ^{5,*}ronal.watrianthos@gmail.com

Email Penulis Korespondensi: ronal.watrianthos@gmail.com

Submitted: 18/12/2021; Accepted: 26/12/2021; Published: 31/12/2021

Abstrak—Penentuan sesuatu jalur terpendek adalah hal yang krusial serta diperlukan sehubungan menggunakan optimasi waktu yang dipergunakan dan beberapa penghematan dibidang lainnya. Penelitian ini bertujuan merancang software penentuan jarak terpendek pada jalur distribusi gas LPG berbasis web dengan menerapkan prosedur pemecahan Dijkstra pada PT. Amartha Anugrah Mandiri. Penelitian ini menggunakan metode pengembangan SDLC (Software Development Life Cycle) dengan menggunakan pemodelan *waterfall*, untuk menentukan rute terdekat pada saat distribusi gas LPG memakai prosedur pemecahan Dijkstra yaitu dengan menentukan lokasi mana yang dijadikan node awal, selanjutnya set nilai jarak di node awal ke node tetangga yang terdekat satu per satu. Hasil dari penelitian ini memudahkan sopir dalam mencari jalur terdekat yang bisa dilewati menuju lokasi pangkalan. Prosedur pemecahan Dijkstra pada software penentuan jarak terpendek di jalur distribusi gas LPG PT. Amartha Anugrah Mandiri bisa membentuk jarak terpendek yang dilalui untuk menuju ke pangkalan.

Kata Kunci: Penentuan Jarak; Algoritma Dijkstra; SDLC; Waterfall

Abstract—Determining a shortest path is a crucial and necessary thing in connection with optimizing the time used and some savings in other fields. This research aims to design a software for determining the shortest distance on a web-based LPG gas distribution line by applying the Dijkstra solving procedure at PT. Amartha Anugrah Mandiri. This study uses the SDLC (software development life cycle) development method using waterfall modeling, to determine the closest route during the distribution of LPG gas using the Dijkstra solution procedure, namely by determining which location is used as the initial node, then set the distance value at the initial node to neighboring nodes. the closest one by one. The results of this study make it easier for drivers to find the closest route that can be passed to the base location. Dijkstra's solving procedure in the software for determining the shortest distance in the PT. Amartha Anugrah Mandiri can form the shortest distance traveled to get to the base.

Keywords: Distance Determination; Dijkstra Algorithm; SDLC; Waterfall

1. PENDAHULUAN

Semakin tinggi teknologi informasi pada saat ini bisa digunakan untuk menaikkan efisiensi serta keefektifan suatu perusahaan. salah satu contohnya yaitu pemanfaatan sistem untuk distribusi *Liquified Petroleum Gas (LPG)* 3 Kilo Gram. Penggunaan sistem yang terkomputerisasi akan sangat membantu dalam menghemat waktu, tidak banyak menggunakan tenaga serta membentuk ketepatan dalam penyajian sebuah data. Kemajuan teknologi saat ini bisa membantu segala pekerjaan manusia. Mulai dari pekerjaan yang sifatnya praktis hingga hal yang sangat susah[1].

PT. Amartha Anugrah Mandiri yang berlokasi di jalan L Sudirman, Komplek Ruko Mega Legenda Blok A2 No. 12, Baloi Permai, Batam Kota, Kota Batam, Kepulauan Riau. Merupakan agen gas LPG di kota Batam yang beranjak pada bidang pendistribusian gas LPG berasal perusahaan Pertamina ke sub agen yang menjual gas LPG secara eksklusif ke masyarakat umum. Semakin tingginya taraf persaingan pada dunia industri, menuntut perusahaan untuk bisa membuat taktik-taktik distribusi yang lebih baik. Salah satu taktik yang bisa dipergunakan adalah pemanfaatan sistem informasi pada perencanaan serta penentuan jarak terpendek di jalur distribusi secara tepat.

Penentuan sesuatu jalur terpendek adalah hal yang krusial serta diperlukan sehubungan menggunakan optimasi waktu yang dipergunakan dan beberapa penghematan dibidang lainnya. Penentuan jalur terpendek memiliki alasan untuk menghemat bahan bakar, waktu dan tenaga yang diberikan[2]. Pencarian jarak terpendek pada jalur perjalanan merupakan paling banyak dibicarakan. Dengan mencari jalan terdekat pada jalur perjalanan ini telah dipakai di berbagai macam bidang untuk mempermudah kinerja suatu aplikasi, baik dalam menentukan jarak perjalanan yang dilewati agar mempersingkat waktu proses pengiriman barang[3]. Adapun dalam menentukan jalur terpendek pada distribusi dapat dihitung dengan salah satu algoritma matematika, antara lain algoritma Dijkstra.

PT. Amartha Anugrah Mandiri belum memiliki aplikasi dalam menentukan jarak terpendek pada jalur distribusi gas LPG, saat ini masih berdasarkan pengalaman sopir dengan beranggapan jalur mana yang lebih dekat Misalnya, pada hari Senin tanggal 28 bulan Januari 2019 agen harus mengirimkan gas LPG 3 kg ke pangkalan yang berlokasi di Botania Garden. Jalur yang biasa dilalui oleh sopir menuju lokasi pangkalan tersebut sekitar 8,9 Km. Hal ini membuktikan bahwa dalam pengiriman tabung gas LPG ke pangkalan jarak yang ditempuh lumayan cukup jauh, sehingga perusahaan terkadang mengalami keterlambatan dan banyak mengeluarkan biaya.

Maka dari itu dalam pencarian jarak terpendek pada jalur distribusi ini, dapat dihitung dengan salah satu algoritma matematika, algoritma matematika yang biasa digunakan salah satunya yaitu algoritma Dijkstra. Algoritma Dijkstra adalah algoritma yang efektif dalam menentukan jalan terdekat dari satu tempat ke tempat lain. Algoritma

Dijkstra mempunyai iterasi untuk menentukan titik yang jaraknya dari titik pertama adalah jarak yang paling pendek[4].

Salah satu referensi penelitian sebelumnya[5] dengan penelitian penerapan algoritma Dijkstra untuk menentukan rute terbaik pada Mobile E-Parking berbasis sistem Informasi Geografis. Penelitian ini membahas tentang objek yang diteliti serta teknologi yang dipakai, untuk mendapatkan informasi yang lebih cepat, peneliti memberikan informasi yang mudah dapat diakses yaitu dengan aplikasi mobile berbasis android dimana dapat mempermudah user mencari informasi lokasi parkir dan menemukan jalur terbaik untuk mencapai lokasi parkir dari titik user mobile berada.

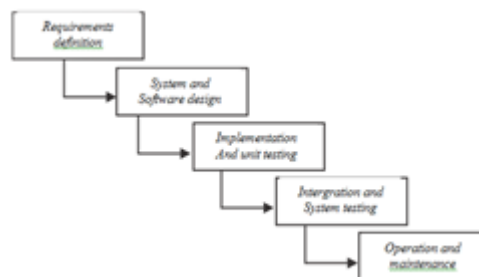
Referensi penelitian berikutnya diusulkan oleh[6], yang berjudul Pencarian SPBU Terdekat dan Penentuan Jarak Terpendek Menggunakan Algoritma Dijkstra (Studi Kasus di Kabupaten Jember). Penelitian ini membahas yaitu menampilkan pencarian SPBU terdekat dan mencari jalan terpendek di Kabupaten Jember menggunakan metode algoritma Dijkstra. Diharapkan penelitian ini mampu memberikan informasi letak SPBU dan memberikan informasi jalur terdekat sehingga dapat menghasilkan efisiensi penggunaan bahan bakar kendaraan.

Penelitian berikutnya berjudul Penentuan Alternatif Jarak Terpendek Menggunakan Algoritma Dijkstra dan Estimasi Waktu Perjalanan, penelitian ini membahas pencarian jalan terdekat pada suatu graf menggunakan algoritma dijkstra. Hasil yang diperoleh yaitu sistem yang dapat mencari jalan terdekat dan alternatif jalan terdekat ketika ada penutupan jalan pada jalan terdekat utama dan juga dapat mengetahui berapa lama waktu tempuh.[7]

Berdasarkan hal tersebut Setelah penulis mencari referensi yang akan diteliti, penulis menemukan permasalahan yang ada di PT. Amartha Anugrah Mandiri. Oleh karenanya penulis dalam penulisan ini tertarik mengambil judul : “Aplikasi Penentuan Jarak Terpendek Pada Jalur Distribusi LPG PT. Amartha Anugrah Mandiri”. Dengan menerapkan Algoritma Dijkstra, ini dapat memudahkan sopir PT. Amartha Anugrah Mandiri dalam mencari jalur terdekat yang dapat dilalui menuju lokasi pangkalan PT. Amartha Anugrah Mandiri. sehingga dapat menghemat waktu dan meminimalisir biaya perjalanan.

2. METODOLOGI PENELITIAN

Pada penelitian ini peneliti menggunakan metode SDLC dengan pemodelan waterfall model, dikarenakan waterfall model memiliki tahapan-tahapan yang tersusun secara linier, identifikasi dan dokumentasi yang lengkap, sehingga cara kerja waterfall model ini mudah dimengerti [8]. Terdapat tahapan-tahapan dalam Waterfall Model sebagai berikut[9]:



Gambar 1. Tahapan Waterfall Model

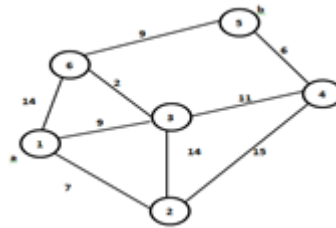
- Requirements analysis and definition**
Tahapan ini bertujuan untuk pengumpulan data seperti masalah sedang terjadi dan tujuan yang ingin dicapai dengan melakukan konsultasi terlebih dahulu dengan pengguna.
- System and software design**
Perancangan aplikasi ini meliputi penggambaran abstraksi sistem dasar aplikasi, identifikasi dan hubungannya.
- Implementation and unit testing**
Tahapan ini bertujuan melakukan perancangan aplikasi selanjutnya diproses sebagai kesatuan program. Adapun Pengujian sistem dilakukan untuk pemeriksaan apakah sistem sudah memenuhi spesifikasinya.
- Integration and system testing**
Bagian-bagian tersendiri sistem atau sistem di satukan dan dicoba sebagai kesatuan sistem yang komplit. Jika pengujian sudah selesai, sistem dapat diberikan ke pengguna.
- Operation and maintenance**
Langkah atau tahapan operasi dan pemeliharaan merupakan tahapan yang paling lama. Sistem di install dan digunakan secara langsung. Maintenance atau perbaikan sistem bertujuan untuk memperbaiki kerusakan yang belum ditemukan pada tahapan proses sebelumnya, memperbaiki tampilan sistem, dan meningkatkan kinerja pada sistem sebagai keperluan baru.

Algoritma Dijkstra sesuai dengan maknanya yang berarti kan serakah tetapi tidak dalam konteks negatif. untuk mencari solusi dari permasalahan, algoritma Dijkstra menggunakan Greedy, yaitu melakukan pencarian solusi yang optimal terhadap titik yang dilalui, tujuannya mendapatkan solusi untuk langkah berikutnya merujuk pada solusi terbaik. Prinsip kerja algoritma Dijkstra dengan menerapkan prinsip antrian, namun prinsip antrian yang dijalankan

pada algoritma Dijkstra yaitu prinsip antrian berdasarkan prioritas. Maka dari itu titik yang memiliki prioritas tertinggi yang akan dicari. Untuk memilih titik yang di jadikan prioritas, algoritma Dijkstra akan melakukan perbandingan dari setiap nilai bobot titik yang ada pada satu tingkat. algoritma ini akan menyimpan nilai bobot dari setiap titik dan kemudian dibandingkan bersama nilai yang akan dijumpai dari jalur yang baru dijumpai. Begitu seterusnya sampai titik yang dicari ditemukan[10].

Algoritma Dijkstra merupakan metode yang dapat menyelesaikan masalah dalam mencari jalur terdekat dari suatu graf di setiap titik yang memiliki bernilai positif. pada pencarian jalur terdekat dari suatu graf berbobot, algoritma Dijkstra akan melakukan pencarian nilai jarak yang paling kecil, jarak terdekat dapat dihasilkan dari 2 atau lebih titik awal dari suatu nilai total dan graf yang dihasilkan adalah nilai yang paling kecil [7][11]. Dalam metode algoritma Dijkstra, titik atau tanda lokasi tempat digunakan karena algoritma Dijkstra menggunakan graf langsung untuk membuat jarak terdekat. Algoritma Dijkstra bekerja untuk menentukan jarak terdekat yang didapat dari hasil paling minimal dari satu titik ke titik berikutnya. Dengan asumsi bahwa titik-titik mewakili lokasi dan garis mewakili jalur, solusi masalah Dijkstra menghitung semua nilai terkecil yang mungkin untuk setiap titik[4].

Berikut contoh tampilan graf dengan menggunakan algoritma Dijkstra serta nilai bobot dari titik ke titik lainnya dalam menentukan jalur:



Gambar 2. Tampilan Graf Menggunakan Algoritma Dijkstra

Dalam mencari jalur terdekat Algoritma Dijkstra akan melakukan langkah demi langkah dalam mencari titik satu ke titik berikutnya yang memiliki tahapan sebagai berikut:

- Berikan nilai jarak pada titik mula ke titik berikutnya, selanjutnya tentukan nilai 0 pada titik mula dan nilai tak hingga ke titik lain.
- Isi semua pada titik belum ditemukan dan set titik awal menjadi titik kepergian.
- Pada titik kepergian, bandingkan titik selanjutnya yang belum dijumpai dan hitunglah jaraknya dari titik kepergian. misalkan, jika titik kepergian C ke D mempunyai nilai jarak 10 dan dari titik D ke titik E memiliki jarak 3, maka dari itu nilai jarak ke E melewati D menjadi $10+3=13$. Apabila nilai jarak ini lebih kecil dari nilai jarak sebelumnya (sudah tersimpan sebelumnya) hapus data lama sebelumnya, simpan data jarak dengan jarak yang terbaru.
- Setelah selesai menentukan setiap jarak terhadap titik selanjutnya, pilih titik yang sudah ditemukan sebagai titik ditemukan. titik yang sudah ditemukan tidak akan pernah diperiksa kembali, jarak yang sudah tersimpan artinya jarak terakhir serta yang memiliki bobot terkecil.
- Isi nilai titik yang belum ditemukan dengan jarak minimal (berasal titik kepergian) menjadi titik kepergian selanjutnya dan lanjutkan dengan kembali kelangkah tiga.

Dari pengertian diatas dapat disimpulkan algoritma Dijkstra merupakan sebuah algoritma yang tamak atau rakus (greedy algorithm) yang biasa digunakan untuk menyelesaikan permasalahan dalam pencarian jarak terpendek dari sebuah graf berarah dengan bobot-bobot garis yang bernilai positif.

3. HASIL DAN PEMBAHASAN

3.1 Requirements Analysis And Definition

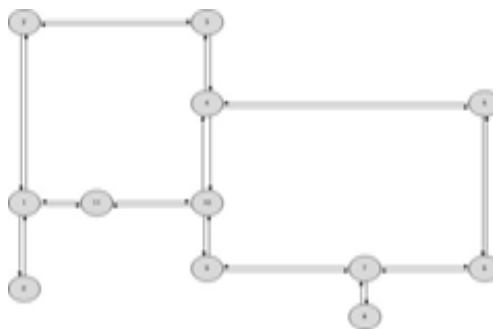
- Analisa Sistem**
Analisa sistem bertujuan untuk menentukan permasalahan yang sering muncul dalam pembuatan sistem, hal ini dikerjakan agar saat proses pembuatan sistem tidak terjadi kesalahan-kesalahan sehingga sistem dapat dijalankan dengan baik dan selesai tepat pada waktunya. Dalam analisa sistem ini penulis melakukan pencarian data atas sistem yang sedang berjalan.
- Analisa Sistem Informasi**
Data yang sudah terkumpul selanjutnya diproses untuk digunakan pada sistem baru yang sudah diusulkan, dan dirancang dengan menerapkan tahapan aliran sistem yang sedang berjalan saat ini dan selanjutnya memberikan solusi pada sistem yang baru.
- Analisa Metode Algoritma Dijkstra**
Algoritma Dijkstra memiliki tujuan untuk melakukan pencarian jalur terdekat berdasarkan nilai jarak terkecil dari titik satu ke titik lainnya.

Penulis menggunakan algoritma Dijkstra untuk memilih rute terpendek dari suatu pangkalan ke pangkalan gas LPG PT. Amaritha Anugrah Mandiri lainnya. Berikut data lokasi pengiriman gas LPG ke pangkalan PT. Amaritha Anugrah Mandiri pada tanggal 25 s/d 28 Februari 2019.

Tabel 1. Data Lokasi Pangkalan Gas LPG PT. Amaritha Anugrah Mandiri

No.	Nama Pangkalan	Alamat	No.	Nama Pangkalan	Alamat
1	PT. AAM	Mega Legenda	16	Supardi saidi	Meditermia blk HH5
2	Muntoib	Komplek Suka jadi	17	Sahrudin	Mediterania KK5
3	M.Ishak	Komplek ditpam OB	18	Iamudin	Puri legenda
4	Sama'ati	Komplek ditpam OB	19	Lambok tua sari	Puri legenda
5	Andi Sarbiah	Komplek Pribumi	20	Hari kasman	Legenda Malaka
6	Nasirun	Komplek Grown Hill	21	Gono purwanto	Bida Kharisma
7	Asnidar mayanti	Kampung Nanas	22	Andi Saidah	Baloi kebun
8	Man Gani	Komplek Nanas	23	Putra cunda	Orchidpark
9	Wheni	Legenda Bali	24	sudarta	Baloi Kebun
10	Syaril	Legenda Malaka	25	Said rusli	Baloi Kebun
11	Ikhwan	Legenda Malaka	26	Fatonah	Baloi Kebun
12	Kamaliyah	Legenda Malaka	27	Moi Kim	Orchid Park
13	Taripin	Legenda Malaka	28	Rasidi	Ruli Kampung Belian
14	Izah Pasari	Legenda Malaka	29	Sulsilawati	Bida Garden
15	Berlian	Legenda Malaka	30	Sugeng	Mitra Raya

Sebagai contoh penerapan Algoritma Dijkstra penulis mengambil contoh pengiriman gas LPG dari lokasi awal pangkalan Lamudin yang beralamat Mega Legenda ke lokasi tujuan pangkalan Izah Pasari yang beralamat di legenda malaka. Dari data lokasi pengiriman gas LPG dari pangkalan Lamudin ke pangkalan Izah Pasari yang di dapat, dapat di gambarkan graf yang menghubungkan titik satu dengan titik yang lain beserta arahnya. Berikut gambarannya:



Gambar 3. Graf Dari Titik Satu Ke Titik Yang Lain

Berikut daftar tabel titik yang ada di graf di atas:

Tabel 2. Daftar titik pada graf

Titik	Lokasi
0	Jl. Lancang Kuning
1	Simpang Warung khas Probolinggo
2	Jl. Pemuda
3	Pertigaan Jl. Pemuda
4	Simpang Jl. Pemuda
5	Simpang Jl. Raja Husin
6	Perempatan Jl. Raja Husin
7	Pertigaan Pintu gerbang Sentosa
8	Pt. Amaritha Anugrah Mandiri
9	Simpang masjid nurul salam
10	Pertigaan Jl. Villa Hang Lekir
11	Perempatan Soto Sokaraja

Tahapan algoritma Dijkstra untuk mencari jalur terpendek yaitu tahap pertama pilih simpul atau titik mana yang akan menjadi simpul pertama, selanjutnya tentukan nilai jarak pada titik pertama ke titik paling dekat satu per satu, dan ke titik selanjutnya langkah demi langkah.

- Tentukan nilai jarak untuk setiap titik ke titik lainnya, kemudian berikan nilai 0 dititik mula dan kosongkan nilai simpul lain. Berikut ini daftar bobot dari titik yang satu ke titik lain yang ada di dalam graf yang telah digambarkan sebelumnya.

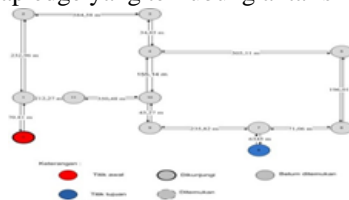
Tabel 3. Daftar Bobot Dari Titik Awal Ke Titik Tujuan

No.	Titik Awal	Titik Tujuan	Bobot (m)	No.	Titik Awal	Titik Tujuan	Bobot (m)
1	0	1	70,81 m	14	7	6	71,06 m
2	1	0	70,81 m	15	7	8	6345 m
3	1	2	232,96 m	16	8	7	6345 m
4	2	1	232,96 m	17	7	9	235,82 m
5	2	3	584,58 m	18	9	7	235,82 m
6	3	2	584,58 m	19	9	10	43,27 m
7	3	4	34,85 m	20	10	9	43,27 m
8	4	3	34,85 m	21	10	4	155,14 m
9	4	5	305,11 m	22	4	10	155,14 m
10	5	4	305,11 m	23	10	11	350,48 m
11	5	6	196,01 m	24	11	10	350,48 m
12	6	5	196,01 m	25	11	1	212,27 m
13	6	7	71,06 m	26	1	11	212,27 m

- b. Set seluruh titik yang belum dijumpai serta set titik awal sebagai titik telah dijumpai.
- c. Dari titik yang telah dijumpai, pertimbangkan simpul tetangga yang belum dijumpai serta hitung jarak asal titik kepergian. Jika jarak yang dijumpai lebih kecil berasal jeda sebelumnya (yang telah tersimpan sebelumnya) hapus data usang, simpan ulang data jarak menggunakan jarak yang baru.
- d. Setelah sudah terselesaikan memilih setiap jarak terhadap titik sebelah, tandai titik yang sudah dijumpai menjadi “titik dijumpai”. titik dijumpai tidak akan pernah di cek kembali, jarak yang disimpan merupakan jarak terakhir dan yang paling kecil nilai jaraknya.
- e. Tentukan titik belum dijumpai dengan jarak terkecil menjadi “Titik Dikunjungi” dan dilanjutkan kembali ke step tiga.

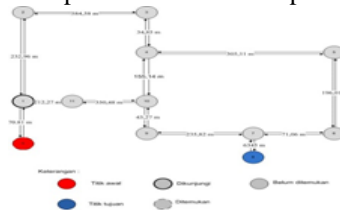
Berikut menjelaskan langkah pencarian jalur terpendek dimulai dari simpul awal (pangkalan izah pasari) sampai simpul tujuan (pangkalan Lamudin) dengan nilai jarak terkecil.

- a. Simpul awal 0, Simpul tujuan 8. Setiap edge yang terhubung antar simpul telah diberi nilai



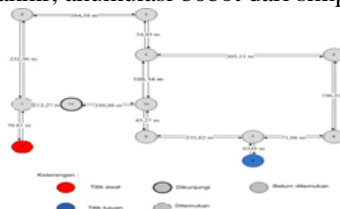
Gambar 4. Dijkstra - Langkah 1

- b. Dijkstra melakukan perhitungan pada simpul tetangga yang terhubung langsung dengan simpul keberangkatan (simpul 0), dan hasil yang didapat adalah simpul 1. Bobot dari simpul 0 ke simpul 1 yaitu 70,81 m.



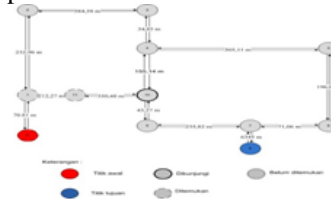
Gambar 5. Dijkstra - Langkah 2

- c. Simpul 1 dipilih menjadi simpul dikunjungi dan dipilih sebagai simpul yang telah dijumpai. Dijkstra melakukan perhitungan kembali terhadap simpul-simpul tetangga yang terkait langsung dengan simpul yang telah dijumpai. Dan perhitungan dijkstra menunjukkan bahwa simpul 2 dan simpul 11 yang menjadi simpul dikunjungi selanjutnya. Dari ke simpul tersebut simpul 11 sebagai simpul dikunjungi selanjutnya, sebab bobotnya yang paling kecil berasal akibat kalkulasi terakhir, akumulasi bobot dari simpul awal $0 \Rightarrow 1 \Rightarrow 11 = 283,08$ m.



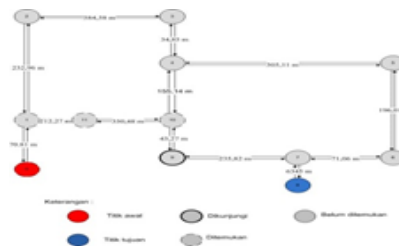
Gambar 6. Dijkstra - Langkah 3

- d. Perhitungan simpul 11 ditandai menjadi simpul yang telah dikunjungi. simpul 10 yang menjadi simpul dikunjungi selanjutnya. Dengan akumulasi dari simpul awal $0 \Rightarrow 1 \Rightarrow 11 \Rightarrow 10 = 633,56$ m.



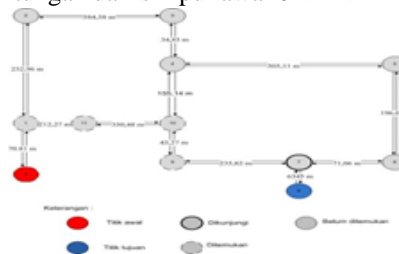
Gambar 7. Dijkstra - Langkah 4

- e. Simpul 10 atur sebagai simpul yang dikunjungi dan ditandai sebagai simpul yang sudah dikunjungi. algoritma Dijkstra melakukan perhitungan kembali terhadap simpul-simpul tetangga yang terkait eksklusif dengan simpul yang telah dijumpai. perhitungan dijkstra melihat bahwa node 4 dan node 9 yang menjadi simpul keberangkatan selanjutnya. Dari ke simpul tersebut simpul 9 menjadi simpul dikunjungi selanjutnya, karena bobotnya yang paling kecil dari hasil perhitungan terakhir, total bobot dari simpul awal $0 \Rightarrow 1 \Rightarrow 11 \Rightarrow 10 \Rightarrow 9 = 676,83$ m.



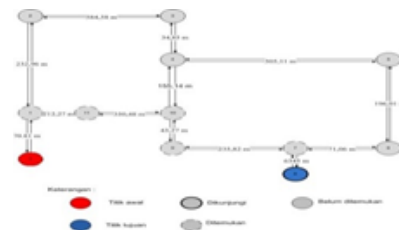
Gambar 8. Dijkstra - Langkah 5

- f. Selanjutnya perhitungan dengan simpul 9 menjadi simpul yang telah dijumpai dan simpul 7 yang menjadi simpul dikunjungi selanjutnya. Dengan perhitungan dari simpul awal $0 \Rightarrow 1 \Rightarrow 11 \Rightarrow 10 \Rightarrow 9 \Rightarrow 7 = 912,65$ m.



Gambar 9. Dijkstra - Langkah 6

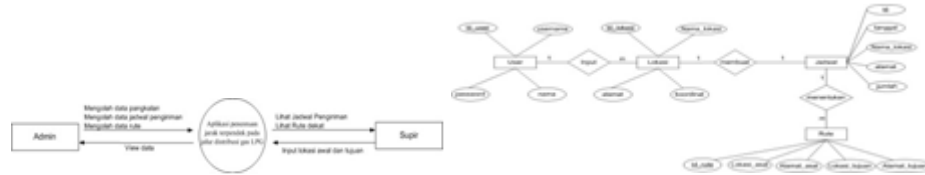
- g. Simpul 7 menjadi simpul dijumpai, dijkstra melakukan perhitungan kembali, dan menemukan bahwa simpul 8 merupakan simpul tujuan telah tercapai lewat simpul 7. Jalur terpendeknya adalah $0 \Rightarrow 1 \Rightarrow 11 \Rightarrow 10 \Rightarrow 9 \Rightarrow 7 \Rightarrow 8$, dan perhitungan bobotnya yaitu 918,995 m. Bila simpul tujuan telah tercapai maka perhitungan dijkstra dinyatakan selesai.



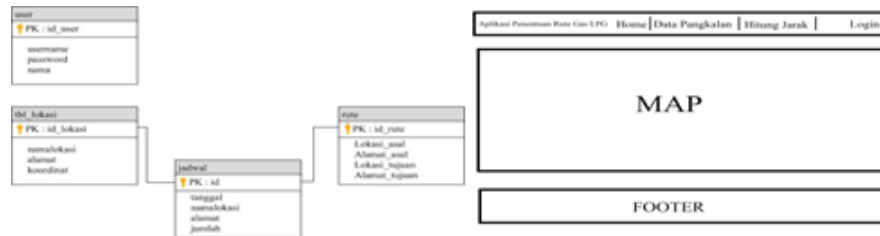
Gambar 10. Dijkstra - Langkah 7

3.2 Design System

Pada perancangan aplikasi penentuan rute pada distribusi gas LPG PT. Amarta Anugrah Mandiri, perancangan sistem memenuhi kebutuhan pemakaian sistem, memberikan gambaran yang jelas dan perancangan sistem harus efektif dan efisien untuk mendukung pengolahan sistem penentuan rute. Untuk mencapai harapan tersebut, rancangan sistem ini akan digambarkan menggunakan Data Flow Diagram merupakan gambaran rancangan untuk mencapai tujuan yang terdiri dari perancangan output, perancangan input, perancangan database, entity relationship diagram.



Gambar 11. Diagram Konteks dan Entity Relationship Diagram (ERD)



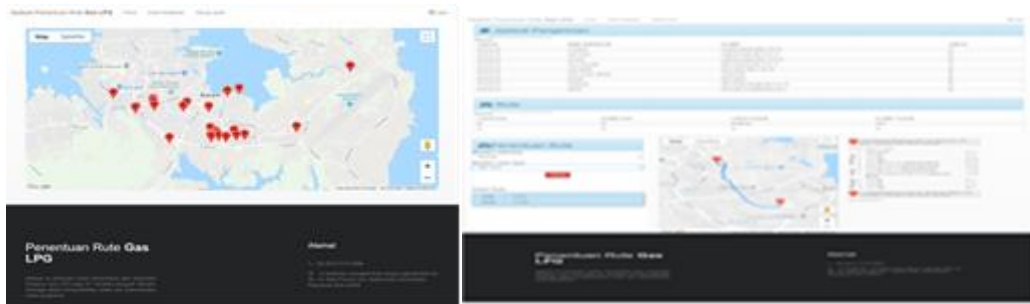
Gambar 12. Relasi Tabel dan Rancangan Halaman Home

3.3 Rancangan Interface

Sistem informasi yang dipergunakan pada perangkat lunak ini artinya teknologi sistem informasi berbasis website, yang menghasilkan sebuah software yang bisa berdiri sendiri serta dapat dijalankan dalam lingkungan internet. Sehingga dimanapun pengguna berada akan bisa memakai sistem informasi ini dengan mengakses situs tersebut dengan cepat serta mudah. Berikut implementasi dari aplikasi penentuan jarak terpendek pada jalur distribusi gas LPG PT. Amartha Anugrah Mandiri:

a. Halaman Tampilan Home

Tampilan halaman home adalah tampilan pertama dari aplikasi penentuan Jalur gas LPG, dimana terdapat map yang menunjukkan lokasi-lokasi pangkalan yang telah diinput titik koordinatnya ke dalam database.



Gambar 13. Halaman Home dan Halaman Hitung Jarak

b. Halaman Hitung Jarak

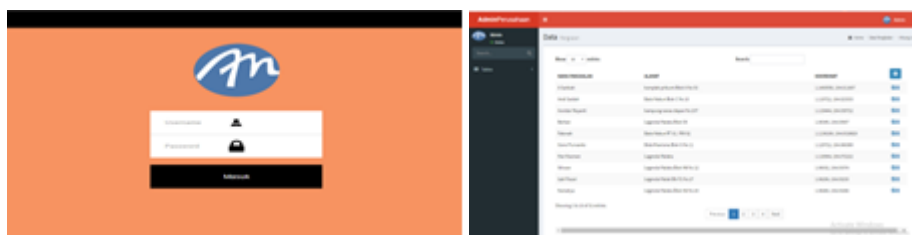
Halaman hitung jarak merupakan tampilan yang memuat halaman pencarian rute terdekat dari pangkalan satu ke pangkalan lainnya pada PT. Amartha Anugrah Mandiri yang telah diinput oleh admin kedalam database.

c. Halaman Login

Dimana Halaman ini berisikan perintah untuk memasukkan username dan password yang sesuai. Apabila login berhasil maka akan masuk ke dalam tampilan admin.

d. Halaman Data Pangkalan

Halaman Admin Data Pangkalan merupakan tampilan yang memuat halaman data pangkalan setelah admin melakukan login ke dalam sistem. Di halaman ini juga admin bisa menginput data, mengubah data dan menghapus data pangkalan.



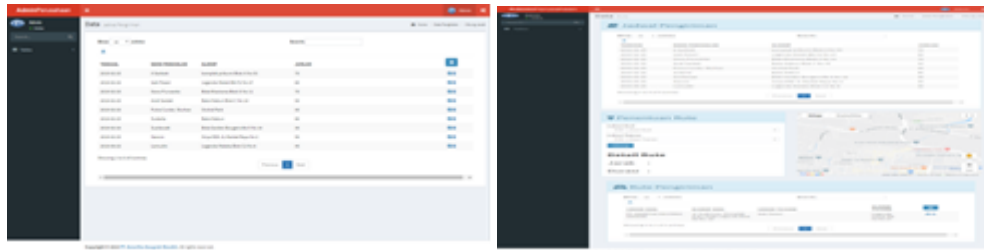
Gambar 14. Halaman Login dan Halaman Data Pangkalan

e. Halaman Jadwal Pengiriman

Halaman Admin Jadwal Pengiriman merupakan tampilan yang memuat halaman data jadwal pengiriman setelah admin melakukan login ke dalam sistem. Di halaman ini juga admin bisa menginput data, mengubah data dan menghapus data jadwal pengiriman.

f. Halaman Admin Data Rute

Halaman Admin Data Rute merupakan tampilan yang memuat halaman data rute setelah admin melakukan login ke dalam sistem. Di halaman ini juga admin bisa menginput data, mengubah data dan menghapus data rute.



Gambar 15. Halaman Data Jadwal Pengiriman dan Halaman Data Rute

3.4 Pengujian Sistem

Pengujian Aplikasi Penentuan Jarak Terpendek Pada Jalur Distribusi Gas LPG Pada PT. Amartha Anugrah Mandiri dapat ditampilkan sebagai berikut:

Tabel 4. Pengujian Sistem

No.	Input	Proses	Output	Hasil
1	Halaman Home	http://localhost/jalur2/index.php	Tampil halaman home	ok
2	Halaman Data Pangkalan	http://localhost/jalur2/datapangkalan.php	Tampil halaman data pangkalan	ok
3	Halaman Hitung Jarak	http://localhost/jalur2/hitungjarak.php	Tampil halaman hitung jarak	ok
4	Halaman Login	http://localhost/jalur2/login.php	Tampil halaman login	ok
5	Halaman admin data pangkalan	http://localhost/jalur2/admin/application/pangkalan/index.php	Tampil halaman admin data pangkalan	ok
6	Halaman input data pangkalan	http://localhost/jalur2/admin/application/pangkalan/index.php	Tampil halaman admin input data pangkalan	ok
7	Halaman admin jadwal pengiriman	http://localhost/jalur2/admin/application/pangkalan/jadwal.php	Tampil halaman admin data pangkalan	ok
8	Halaman input jadwal pengiriman	http://localhost/jalur2/admin/application/pangkalan/jadwal.php	Tampil halaman admin input data pangkalan	ok
9	Halaman admin rute	http://localhost/jalur2/admin/application/pangkalan/rute.php	Tampil halaman admin data rute	ok
10	Halaman input data rute	http://localhost/jalur2/admin/application/pangkalan/rute.php	Tampil halaman admin input data rute	ok

3.5 Operasional dan Perawatan

Operasional aplikasi ini untuk digunakan pada PT. Amartha Anugrah Mandiri yang dapat menentukan jarak terpendek antar titik. Perawatan aplikasi ini melakukan pemeliharaan sistem secara berkala dan melakukan update titik lokasi terbaru pada gas LPG kota Batam, sehingga menghasilkan sistem maksimal dan baik.

4. KESIMPULAN

Selesainya penelitian dan pembuatan sistem informasi ini, maka penulis mengambil kesimpulan yaitu Dengan menerapkan Algoritma Dijkstra pada Aplikasi penentuan rute terpendek pada jalur distribusi gas LPG PT. Amartha Anugrah Mandiri dapat menghasilkan rute terpendek yang dilalui untuk menuju ke pangkalan satu ke pangkalan PT. Amartha Anugrah Mandiri lainnya. Aplikasi penentuan rute terpendek pada jalur distribusi gas LPG PT. Amartha Anugrah Mandiri ini dibuat menggunakan bahasa pemrograman PHP, MySQL sebagai basis datanya dan Google Map untuk menampilkan peta.

REFERENCES

[1] Z. Ariansyah, D. D. S. Fatimah, and E. Retnadi, “Perancangan Perangkat Lunak Pendistribusian Liquefied Petroleum Gas (LPG) Di PT. Denas Shantika Perkasa,” *J. Algoritma.*, vol. 11, no. 2, pp. 282–290, 2015, doi: 10.33364/algoritma/v.11-2.282.

[2] E. S. Arga, G. G. Firmansyah, K. Imam, and M. Fauzi, “Penerapan algoritma djikstra pada pencarian jalur terpendek,” vol. 1, no. 2, pp. 134–142, 2021.

[3] 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.
- [4] 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.
- [5] C. Prianto and M. Kusnadi, “Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terbaik Pada Mobile E-Parking Berbasis Sistem Informasi Geografis,” *J. Inform. J. Pengemb. IT*, vol. 3, no. 3, pp. 329–335, 2018, doi: 10.30591/jpit.v3i3.941.
- [6] 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.
- [7] S. Hamdi and Prihandoko, “Analisis Algoritma Dijkstra dan Algoritma Bellman-Ford Sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran (Studi Kasus: Kecamatan Praya Kota),” *J. Ilm. Ilmu-Ilmu Tek.*, vol. 8, no. 1, pp. 26–32, 2018.
- [8] W. Bangkit and T. A. Setiawan, “Rancang Bangun Aplikasi Scanner Car Location Berbasis Teknologi QR-Code menggunakan Algoritma Dijkstra,” pp. 1–5, 2019.
- [9] G. W. Sasmito, “Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal,” *J. Inform. Pengemb. IT*, vol. 2, no. 1, pp. 6–12, 2017.
- [10] Y. Primadasa, “PENCARIAN RUTE TERPENDEK DIJKSTRA PADA SIG BERBASIS WEB UNTUK DISTRIBUSI MINUMAN (STUDI KA ...),” 2016.
- [11] 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.