

Pembuatan Aplikasi Penentuan Jalur Terpendek Pendistribusian Solar dengan Metode Algoritma Dijkstra Berbasis Web Pada CV. Mega Lestari

Tri Oktaviyani¹⁾, Jusak²⁾, Agus Dwi Churniawan³⁾

Program Studi/Jurusan Sistem Informasi
Fakultas Teknologi dan Informatika

Institut Bisnis dan Informatika Stikom Surabaya
Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email : 1)Vivim@gmail.com, 2)Jusak@stikom.edu, 3)Agusdwi@stikom.edu

Abstract: CV. Mega Lestari is a company engaged as service provider to transport diesel fuel manifold to the needs of companies, which is located at Jl. Tropodo Asri Blok F 31 Sidoarjo. During this CV. Mega Lestari still does not have the technology that can help in running the business of the company, especially in terms of the distribution of goods. The process resulted in delays in the distribution of diesel fuel is often reduced profits. Difficult to estimate the distance and time of the place of destination becomes an obstacle for the company, coupled with drivers who are not familiar with the road.

Based on the above issues then made a determination of the application of the shortest paths distribution of diesel with a web-based method dijkstra algorithm on a CV. Mega Lestari. With this application is expected to help the company runs the business.

Applications that have been built in the preparation of this thesis is the application of determining the shortest path to the method of distribution of diesel fuel with dijkstra algorithm web-based on CV. Mega Lestari. This application can generate both shortest path different destinations and determine estimated distance and time to the distribution of diesel fuel destination.

Keywords: Shortest Route, Shortest Path Problem, Dijkstra's Algorithm

Perkembangan teknologi saat ini sangat pesat dan memberikan banyak manfaat di segala bidang termasuk bidang pelayanan jasa. Perusahaan jasa yang bergerak dalam persewaan kendaraan untuk mengangkut barang-barang merupakan bagian penting bagi sebuah perusahaan produksi dalam hal mendistribusikan produk mereka ke berbagai tempat yang tersebar.

Pada pendistribusian atau pengiriman barang, kendaraan tidak melayani satu konsumen saja. Namun melayani beberapa konsumen sekaligus dalam melakukan sebuah pendistribusian atau pengiriman barang. Wilayah-wilayah konsumen yang berbeda tempat menyebabkan suatu kendaraan pendistribusi atau pengiriman barang harus menentukan jalur perjalanan yang akan dilaluinya sebelum melakukan pendistribusian barang. Serta terdapat banyak jalur yang ada pada tiap daerah karena dalam kenyataannya dari daerah A ke daerah Z tidak hanya memiliki satu jalur saja, banyak sekali jalur yang dapat dilalui sebagai rute jalan. CV. Mega Lestari merupakan perusahaan yang bergerak sebagai penyedia jasa untuk mengangkut bahan bakar minyak berjenis solar untuk kebutuhan perusahaan industri. Saat ini CV. Mega Lestari masih belum memiliki teknologi yang dapat membantu dalam menjalankan bisnis perusahaan, terutama dalam hal pendistribusian barang. Terjadinya proses keterlambatan dalam pendistribusian solar mengakibatkan sering berkurangnya keuntungan yang didapat.

Sulitnya memperkirakan jarak dan waktu ketempat tujuan menjadi kendala bagi perusahaan, ditambah dengan supir yang tidak hafal dengan rute jalan. Padahal ketepatan dalam waktu pendistribusian barang sangatlah penting demi menjaga kelangsungan

bisnis antar sesama perusahaan agar terhindar dari kerugian yang bisa terjadi dimasa yang akan datang. Ada beberapa faktor yang berpengaruh terhadap penentuan jarak terpendek diantaranya biaya transportasi dan jarak antar tujuan.

Berdasarkan masalah yang terjadi, perusahaan berharap bisa meningkatkan kualitas dalam hal pelayanan jasa sesuai dengan tujuan perusahaan. Solusi yang diusulkan untuk mengatasi masalah yang dialami dengan membuat aplikasi yang secara langsung dapat membantu perusahaan dalam penentuan jalur terpendek untuk mendistribusikan solar serta dapat memperkirakan jarak dan waktu tempuh perjalanan ketempat tujuan. Dengan adanya aplikasi ini diharapkan dapat mempercepat proses pendistribusian barang.

Algoritma

Menurut Thomas H. Cormen (2009), algoritma merupakan prosedur komputasi yang mengambil beberapa atau kumpulan nilai sebagai input kemudian di proses sebagai output sehingga algoritma merupakan urutan langkah komputasi yang mengubah input menjadi output.

Algoritma Dijkstra

Nama algoritma Dijkstra diturunkan sesuai dengan nama penemunya, yaitu Edsger Wybe Dijkstra. Prinsip dasar dari algoritma ini adalah mencari lintasan terpendek dengan memanfaatkan prinsip *Greedy*, yang menyatakan bahwa pada setiap langkah algoritma akan memilih jalur yang memiliki bobot paling kecil dan selanjutnya memasukkannya nilai tersebut ke dalam sebuah himpunan solusi.

Input algoritma Dijkstra ialah sebuah graph berarah yang berbobot (*weighted directed graph*) G dan sebuah sumber vertex *s* dalam G dan V adalah himpunan semua vertices dalam graph G (Rosen,1999).

Properti algoritma Dijkstra:

1. Matriks ketetanggaan $M[mij]$

$mij =$ bobot sisi (*i*, *j*)

$mii = 0$

$mij = \infty$, jika tidak ada sisi dari simpul *i* ke simpul *j*

2. Larik $S = [si]$ yang dalam hal ini,

$si = 1$, jika simpul *i* termasuk ke dalam lintasan terpendek

$si = 0$, jika simpul *i* tidak termasuk ke dalam lintasan terpendek.

3. Larik/tabel $D = [di]$ yang dalam hal ini,

di merupakan panjang lintasan dari simpul awal *s* ke simpul *i*.

Algoritma Lintasan Terpendek Dijkstra (Mencari lintasan terpendek dari simpul *a* ke semua simpul)

Langkah 0 (inisialisasi): - inialisasi $si = 0$ dan $di =$ mai untuk $i = 1, 2, \dots, n$

Langkah 1:

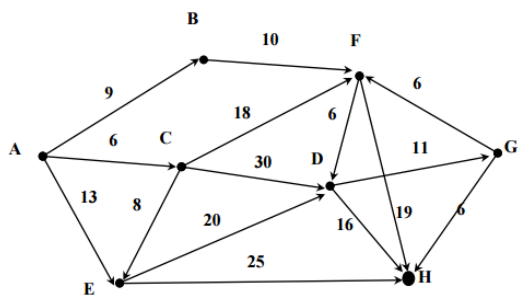
- isi *sa* dengan 1 (karena simpul *a* adalah simpul asal lintasan terpendek, jadi sudah pasti terpilih)

- isi *da* dengan ∞ (tidak ada lintasan terpendek dari simpul *a* ke *a*)

Langkah 2, 3, ..., *n*:

- cari *j* sedemikian sehingga $sj = 0$ dan $dj = \min\{d1, d2, \dots, dn\}$

- isi *sj* dengan 1 perbarui *di*, untuk $i = 1, 2, 3, \dots, n$ dengan: di (baru) = $\min\{di$ (lama), $dj + mji\}$. Contoh Misal diberikan graph berbobot seperti dapat dilihat pada Gambar 1.



Gambar 1 Contoh Graph Berbobot

Misal diberikan grafik berbobot dan berarah seperti pada gambar di atas. Akan dicari lintasan terpendek dari simpul A ke seluruh simpul lain. Maka untuk menyelesaikan permasalahan tersebut dapat menggunakan tabel seperti pada tabel 1.

Jadi, lintasan terpendek dari:

A ke C adalah A, C dengan panjang = 6

A ke B adalah A, B dengan jarak = 9

A ke E adalah A, B, E dengan jarak = 13

A ke F adalah A, B, F dengan jarak = 19

A ke D adalah A, B, F, D dengan jarak = 25

A ke G adalah A, B, F, D, G dengan jarak = 36

A ke H adalah A, B, E, H dengan jarak = 38.

Tabel 1 Penyelesaian Menggunakan Algoritma Dijkstra

Iterasi ke	Simpul yang dipilih	Lintasan	S								D							
			A	B	C	D	E	F	G	H	A	B	C	D	E	F	G	H
inisialisasi	-	-	0	0	0	0	0	0	0	∞	∞	∞	∞	∞	∞	∞	∞	
1	A	A	1	0	0	0	0	0	0	∞	9	6	∞	13	∞	∞	∞	
2	C	AC	1	0	1	0	0	0	0	∞	9	6	36	13	∞	∞	∞	
3	B	AB	1	1	1	0	0	0	0	∞	9	6	36	13	19	∞	∞	
4	E	A,B,E	1	1	1	0	1	0	0	∞	9	6	33	13	19	∞	38	
5	F	A,B,F	1	1	1	0	1	1	0	∞	9	6	25	13	19	∞	38	
6	D	A,B,F,D	1	1	1	1	1	1	0	∞	9	6	25	13	19	36	38	
7	G	A,B,F,D,G	1	1	1	1	1	1	1	0	∞	9	6	25	13	19	36	38
8	H	A,B,E,H	1	1	1	1	1	1	1	1	∞	9	6	25	13	19	36	38

Algoritma Rute Terpendek

Saat kita meringkas algoritma rute terpendek, anggaplah suatu jaringan yang terdiri dari *N* node. Prosedur dibawah ini dapat digunakan untuk mencari jalur terpendek dari node 1 ke setiap node lainnya dalam jaringan:

Langkah 1 : Berikan label permanen [0,S] pada node 1; Angka 0 menunjukkan bahwa jarak dari node 1 ke node itu sendiri adalah nol dan S menunjukkan bahwa node 1 adalah node awal.

Langkah 2 : Hitung label sementara untuk node-node yang dapat dicapai langsung dari node 1. Angka pertama dalam setiap label adalah jarak langsung dari node 1 ke node yang sedang dicari; kita sebut bagian label ini sebagai nilai jarak. Angka kedua dalam setiap label, yang kita sebut nilai node sebelumnya, menunjukkan rute node sebelumnya dari node 1 ke node yang sedang dicari; jadi, dalam langkah ini nilai node sebelumnya adalah 1 karena kita hanya memperhatikan node yang dapat dicapai langsung dari node 1.

Langkah 3: Identifikasi node berlabel sementara dengan nilai jarak terkecil, dan nyatakan node itu berlabel permanen. Bila semua node berlabel permanen, teruskan ke langkah 5.

Langkah 4: Perhatikan node-node tersisa yang tidak berlabel permanen dan dapat dicapai langsung dari node berlabel permanen baru yang diidentifikasi dalam langkah 3. Hitung label sementara baru untuk node-node ini sebagai berikut:

a. Jika node yang sedang dicari memiliki label sementara, tambahkan nilai jarak node berlabel permanen baru ke jarak langsung dari node berlabel permanen baru ke node yang sedang dicari. Jika jumlah ini kurang dari nilai jarak node yang sedang dicari, tentukan nilai jarak untuk node ini sama dengan jumlahnya; sebagai tambahan, tentukan nilai node sebelumnya sama dengan node berlabel permanen baru yang memberikan jarak lebih kecil. Lanjutkan ke langkah 3.

b. Jika node yang sedang dicari belum diberi label, buatlah sebuah label sementara dengan menambah nilai jarak langsung dari node berlabel permanen baru ke node yang sedang dicari. Nilai node sebelumnya ditetapkan sama dengan node berlabel permanen yang baru tersebut. Lanjutkan ke 3.

Langkah 5 : Label permanen mengidentifikasi jarak terpendek dari node 1 ke setiap node sebelumnya pada rute yang terpendek. Rute terpendek ke suatu node yang ditentukan dapat ditemukan dengan memulai dari node yang ditentukan tersebut dan bergerak ke node

sebelumnya. Dengan melanjutkan gerakan kebelakang ini melewati jaringan akan memberikan rute terpendek dari node 1 ke node yang dicari.

System Development Life Cycle

Menurut Tegarden et al (2013) System Development Life Cycle (SDLC) adalah proses untuk memahami bagaimana sebuah sistem informasi dapat mendukung kebutuhan bisnis dengan merancang suatu sistem, membangun sistem tersebut dan menyampaikan kepada pengguna..

SDLC memiliki empat fase dasar yaitu planning, analysis, design dan implementation. Setiap fase itu sendiri terdiri dari serangkaian langkah yang menggunakan cara tertentu dalam menghasilkan goal yang dicapai. Pada poin berikut akan dijelaskan secara singkat dari keempat fase tersebut.

a. Planning

Fase planning merupakan proses dasar dalam memahami mengapa sistem informasi harus dibangun dan menentukan bagaimana tim proyek akan membangun *project* tersebut.

b. Analysis

Fase analysis adalah jawaban dari pertanyaan siapa yang akan menggunakan sistem, apa yang akan dilakukan oleh sistem, dan dimana serta kapan sistem tersebut akan digunakan. Pada fase ini pula tim proyek menginvestigasi sistem yang sudah ada sebelumnya, mengidentifikasi peluang untuk perbaikan dan mengembangkan konsep yang baru untuk sistem yang akan dibuat.

c. Design

Fase design yaitu menentukan bagaimana sistem akan beroperasi, dalam hal ini antara lain perangkat keras, perangkat lunak, infrastruktur jaringan (user interface), forms dan laporan (database dan file yang dibutuhkan aplikasi).

d. Implementation

Fase implementation adalah fase final pada SDLC yaitu pada saat sistem telah selesai dibuat. Implementasi pada fase ini biasanya paling banyak mengambil perhatian karena dalam keseluruhan sistem, tahap implementasi adalah tahap yang paling banyak memakan waktu serta biaya karena mencoba keseluruhan sistem.

Waterfall

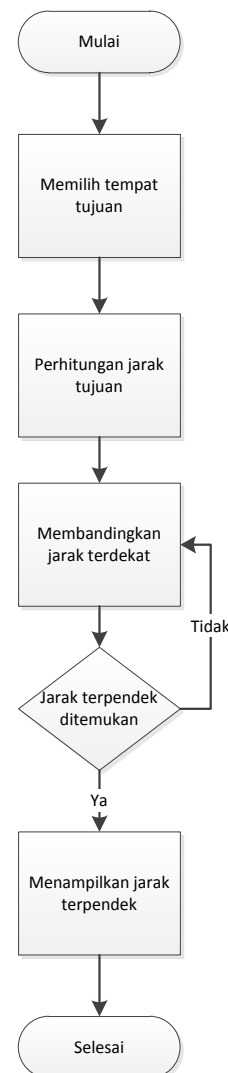
Struktur asli dari metode desain terstruktur adalah metode pengembangan waterfall. Dengan metode pengembangan waterfall, analis dan pengguna memproses melewati tahap demi tahap. Untuk lebih jelasnya dapat dilihat pada Gambar 2.



Gambar 2 Waterfall

Flowchart Pencarian Jalur Terpendek

Aplikasi penentuan jalur terpendek ini diharapkan dapat membantu mempermudah admin dalam memperkirakan jarak tempuh dari pendistribusian solar ke tempat tujuan. Secara umum gambaran dari flowchart pencarian jalur terpendek dapat dilihat pada Gambar 3.



Gambar 3 Flowchart Pencarian Jalur Terpendek

Google Maps

Google Maps Distance Matrix API adalah layanan yang menyediakan waktu dan jarak perjalanan untuk matriks tempat asal dan tujuan. Informasi yang dikembalikan berdasarkan rute yang disarankan antara titik awal dan titik akhir, sesuai perhitungan *Google Maps API*, dan terdiri dari baris-baris berisi nilai *duration* dan *distance* untuk setiap pasang. Layanan ini tidak mengembalikan informasi rute detail. Informasi jalur bisa didapat dengan meneruskan satu tempat asal dan tujuan yang diinginkan ke *Google Maps Directions API*.

Untuk penghitungan jarak, bisa memilih mode transportasi yang akan digunakan. Secara default, jarak dihitung untuk mode mengemudi. Mode perjalanan berikut ini telah didukung:

- Driving (default) menunjukkan penghitungan jarak menggunakan jaringan jalan raya.
- Walking meminta penghitungan jarak untuk berjalan melalui jalur pejalan kaki & trotoar.
- Bicycling meminta penghitungan jarak untuk bersepeda melewati jalur sepeda & jalan yang diinginkan.
- Transit meminta perhitungan jarak melalui jalur angkutan umum. Nilai ini dapat dipilih jika permintaan menyertakan API key atau ID klien Google Maps API for Work. Jika menggunakan mode ke transit bisa menetapkan *departure_time* atau *arrival_time*. Jika waktu yang ditetapkan tidak ada, maka default *departure_time* adalah waktu saat ini. Bisa juga menyertakan *transit_mode* dan atau *transit_routing_preference*.

Jarak dapat dihitung dengan mengikuti batasan tertentu. Batasan ditunjukkan melalui penggunaan parameter *avoid* dan parameter yang menunjukkan batasan untuk *avoid*. Batasan ini meliputi:

- avoid=tolls*
- avoid=highways*
- avoid=ferries*
- avoid=indoor*

Hasil *Distance Matrix* berisi teks dalam bidang *distance* untuk menunjukkan jarak rute yang dihitung. Satuan yang akan digunakan bisa ditetapkan:

- units=metric* (default) mengembalikan jarak dalam kilometer dan meter.
- units=imperial* mengembalikan jarak dalam mil dan kaki.

Unified Modelling Language

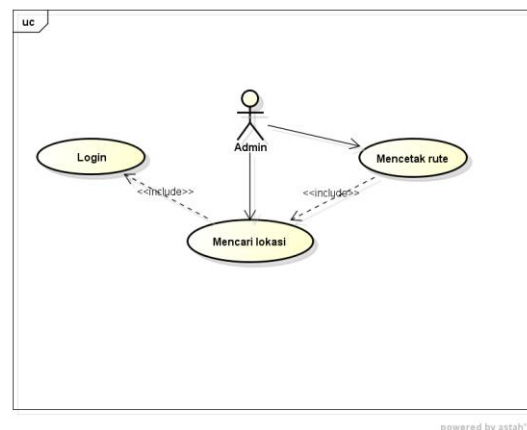
Menurut Nugroho (2005), pemodelan visual merupakan proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Notasi-notasi baku sangat penting demi suatu alasan komunikasi. Dengan notasi-notasi pemodelan yang bersifat baku komunikasi yang baik akan terjalin dengan mudah antar tim pengembang sistem dan antara tim pengembang dengan para *user* atau pengguna. Untuk membuat pemodelan sistem atau *software*, dalam buku ini notasi-notasi Unified Modeling Language (UML) yang akan digambarkan secara elektronik (dengan bantuan komputer) lewat

sarana perangkat lunak. Dengan pemodelan menggunakan UML ini, pengembang dapat melakukan:

- Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
- Penelaahan bagaimana objek-objek pada sistem saling mengirim pesan (message) dan saling bekerja sama satu sama lain.
- Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.
- Dokumentasi sistem atau *software* untuk keperluan-keperluan tertentu di masa yang akan datang.

Diagram Use Case Aplikasi

Use case adalah bagian tingkat tinggi dan fungsional sistem. Dengan kata lain, use case menggambarkan bagaimana seseorang menggunakan sistem (Sholih, 2006). Berikut ini merupakan use case yang telah didapat berdasarkan kebutuhan sistem yang dapat dilihat pada Gambar 4.

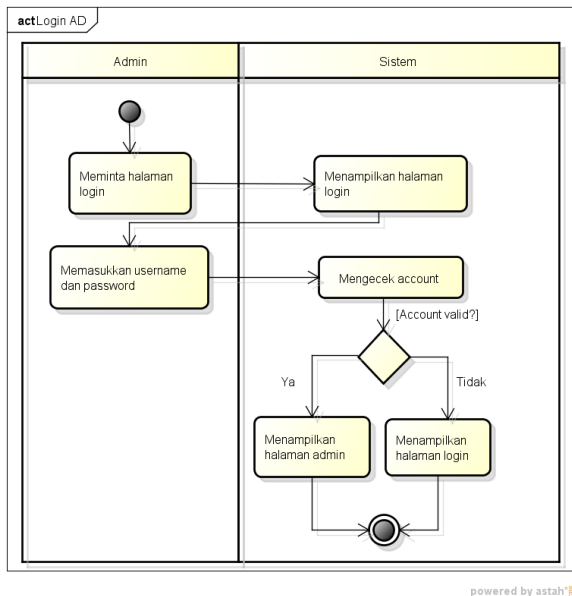


Gambar 4 Diagram Use Case Aplikasi

Pada Gambar 4 terdapat aktor yaitu admin sebagai pengguna aplikasi, yang dapat melakukan tiga aksi dalam sistem yaitu: login, mencari lokasi, dan mencetak rute. Untuk mencari lokasi, admin harus memiliki akun terlebih dahulu dan melakukan login. Login berfungsi untuk keamanan. Apabila admin tidak memiliki akun, maka admin tidak berhak dan tidak dapat mengoperasikan aplikasi. Setelah login, admin dapat mengoperasikan aplikasi dan juga dapat mencetak rute yang terdapat pada aplikasi. Untuk penjelasan yang lebih detail akan dijelaskan pada diagram-diagram di bawah.

Diagram Aktifitas Login

Desain diagram aktifitas *login* beserta penjelasannya dapat dilihat pada Gambar 5.

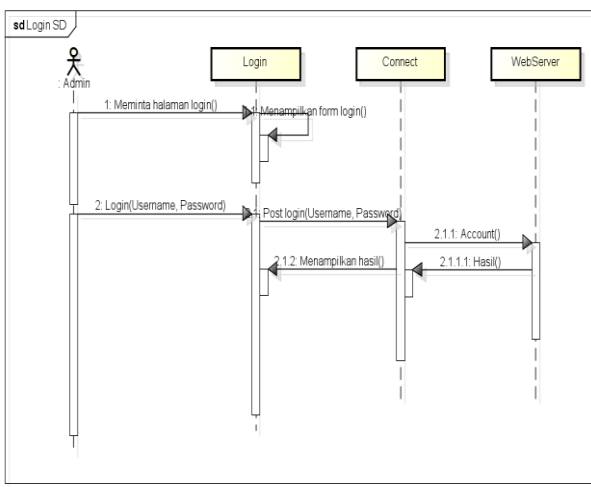


Gambar 5 Diagram Aktifitas Login

Diagram aktifitas login dilakukan oleh admin. Proses dimulai ketika admin meminta atau mengakses halaman login dengan cara memasukkan alamat halaman login. Sistem akan menampilkan form login dan admin memasukkan username dan password. Kemudian sistem akan mengecek apakah username dan password tersebut sama dengan yang ada di dalam database akun. Apabila data sesuai dengan database akun maka sistem akan memberikan notifikasi bahwa login sukses. Setelah login berhasil maka sistem akan menampilkan halaman utama.

Diagram Sequence Login

Desain diagram *sequence login* beserta penjelasannya dapat dilihat pada Gambar 6.



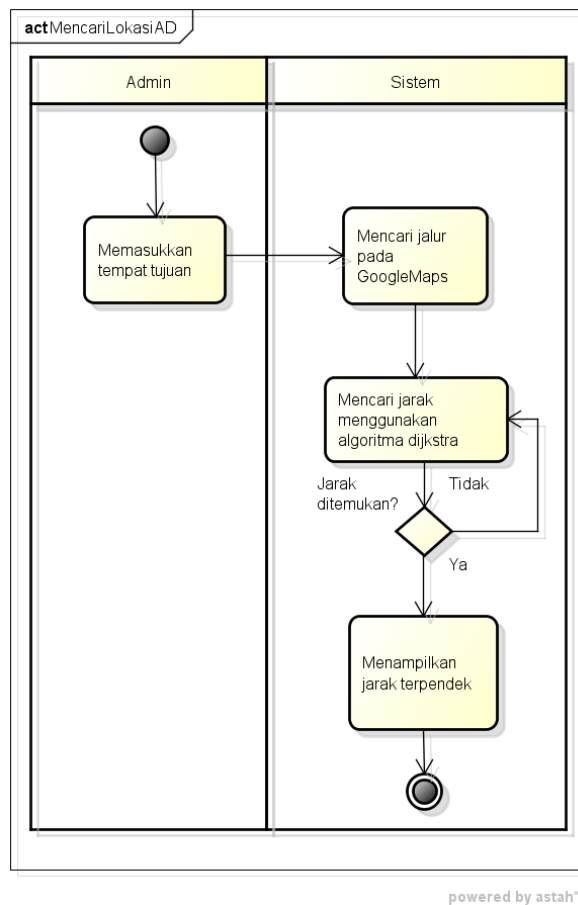
Gambar 6 Diagram Sequence Login

Diagram *sequence login* dilakukan oleh admin. Admin meminta atau mengakses halaman *login*. Sistem akan menampilkan halaman tersebut. Kemudian admin memasukkan *login (username dan password)* dan

sistem akan mengecek data tersebut. Apabila data sesuai dengan *database* akun maka sistem akan memberikan notifikasi bahwa *login* sukses. Setelah *login* berhasil maka sistem akan menampilkan halaman utama.

Diagram Aktifitas Mencari Lokasi

Desain diagram aktifitas mencari lokasi beserta penjelasannya dapat dilihat pada Gambar 7.

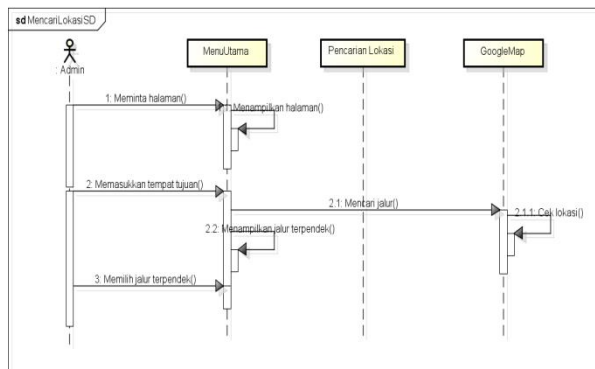


Gambar 7 Diagram Aktifitas Mencari Lokasi

Diagram aktifitas mencari lokasi dilakukan oleh admin. Admin meminta halaman ketika proses login selesai dan sistem akan menampilkan halaman tersebut. Kemudian admin memilih lokasi tujuan distribusi. Sistem akan menentukan jalur sesuai dengan lokasi tujuan dan mencari jalur terpendek ke lokasi tujuan. Setelah itu sistem akan menampilkan jarak terpendek dan lokasi tujuan.

Diagram Sequence Mencari Lokasi

Desain diagram *sequence* mencari lokasi beserta penjelasannya dapat dilihat pada Gambar 8.

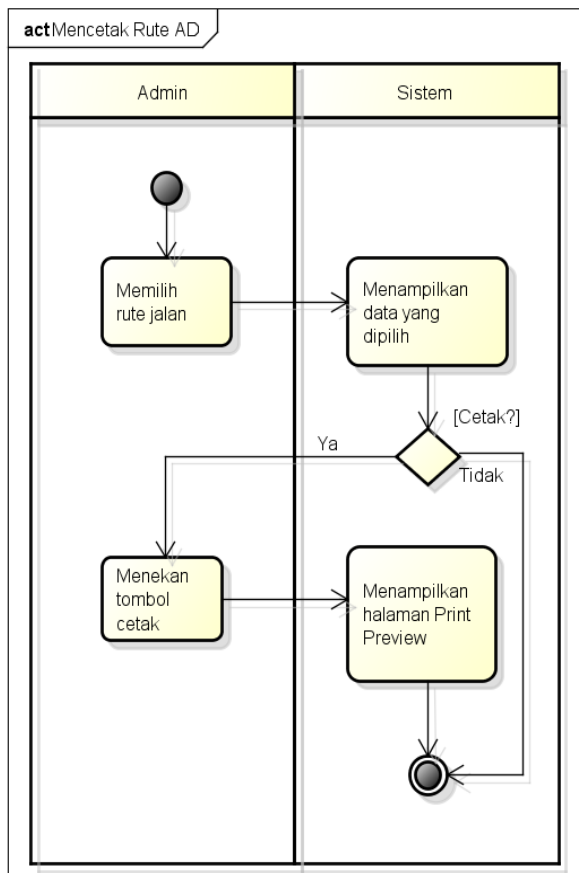


Gambar 8 Sequence Mencari Lokasi

Diagram *sequence* mencari lokasi dilakukan oleh admin. Admin meminta halaman ketika proses login selesai dan sistem akan menampilkan halaman tersebut. Kemudian admin memilih lokasi tujuan distribusi. Sistem akan menentukan jalur sesuai dengan lokasi tujuan dan mencari jalur terpendek ke lokasi tujuan. Setelah itu sistem akan menampilkan jarak terpendek dan lokasi tujuan.

Diagram Aktifitas Mencetak Rute

Desain diagram aktifitas mencetak rute beserta penjelasannya dapat dilihat pada Gambar 9.

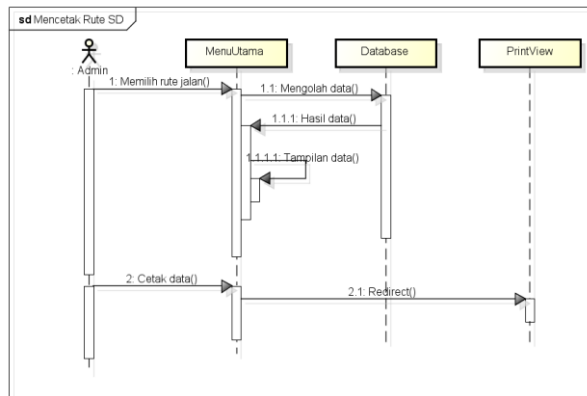


Gambar 9 Diagram Aktifitas Mencetak Rute

Diagram aktifitas mencetak rute yang dilakukan oleh admin. Admin memilih terlebih dahulu rute jalur yang ingin dicetak pada layar dan sistem akan menampilkan informasi yang dipilih tersebut. Kemudian admin dapat menekan tombol cetak lalu sistem akan menampilkan halaman *print preview* yang dapat mencetak informasi tersebut.

Diagram Sequence Mencetak Rute

Desain diagram *sequence* mencetak rute beserta penjelasannya dapat dilihat pada Gambar 10.

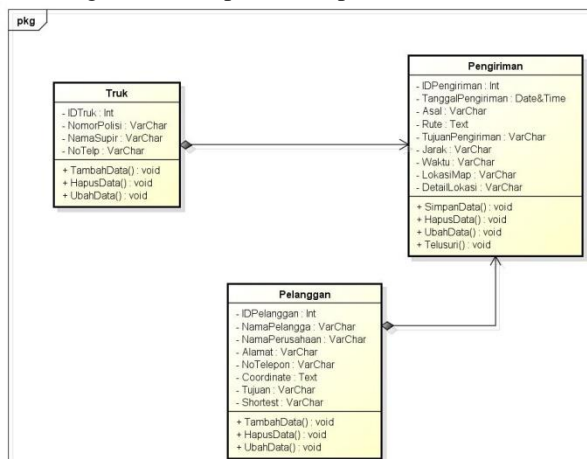


Gambar 10 Diagram Sequence Mencetak Rute

Diagram *sequence* mencetak rute dilakukan oleh admin. Admin memilih terlebih dahulu rute jalur yang ingin dicetak pada layar dan sistem akan menampilkan informasi yang dipilih tersebut. Kemudian admin dapat menekan tombol cetak lalu sistem akan menampilkan halaman *print preview* yang dapat mencetak informasi tersebut.

Class Diagram

Class diagram digunakan untuk menunjukkan interaksi antar kelas dalam sistem (Sholih, 2006). *Class diagram* memberikan gambaran sistem secara statis dan relasi antar mereka. *Class diagram* dari Aplikasi Penentuan Jalur Terpendek Pendistribusian Solar Dengan Metode Algoritma Dijkstra Berbasis Web Pada CV. Mega Lestari dapat dilihat pada Gambar 11.



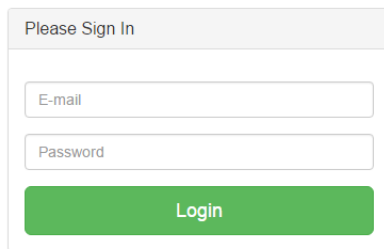
Gambar 11 Class Diagram

HASIL DAN PEMBAHASAN

Aplikasi Penentuan Jalur Terpendek Pendistribusian Solar Dengan Metode Algoritma Dijkstra Berbasis Web Pada CV. Mega Lestari ini menggunakan *browser* atau peramban.

Halaman Master Login

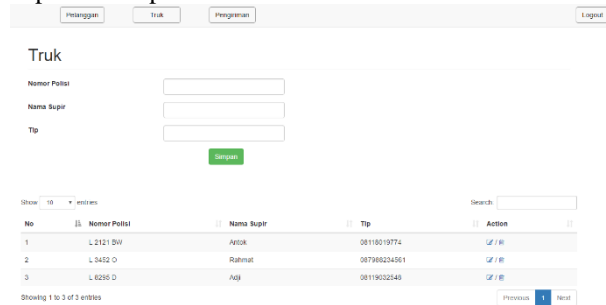
Halaman login digunakan oleh admin untuk masuk ke dalam sistem dengan memasukkan username dan password. Pada halaman ini juga akan memberikan informasi apabila data admin yang dimasukan tidak benar. Halaman master *login* dapat dilihat pada Gambar 13.



Gambar 12 Halaman Master Login

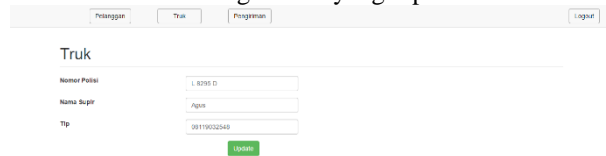
Halaman Master Truk

Halaman master truk berisi tentang data truk, terdiri dari nomor polisi, nama supir, nomor telepon supir dan aksi. Didalam ini juga terdapat menu untuk membuat data baru, fitur ubah untuk mengubah data dan hapus untuk menghapus data. Halaman master truk dapat dilihat pada Gambar 14 dan Gambar 15.



Gambar 13 Halaman Master Truk

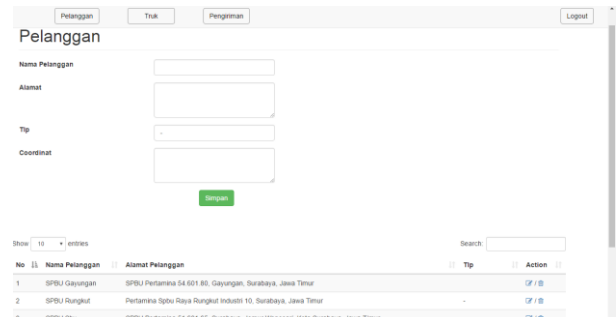
Tombol “Ubah” dipilih maka akan menampilkan form ubah sesuai dengan data yang dipilih.



Gambar 14 Form Ubah Master Truk

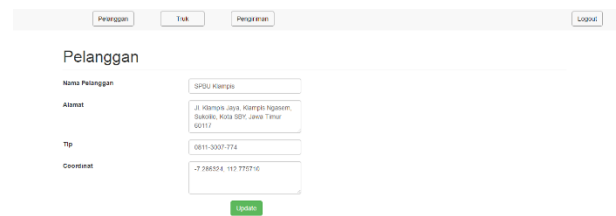
Halaman Master Pelanggan

Halaman master pelanggan digunakan untuk menambah, merubah dan menghapus data pelanggan. Untuk lebih lengkapnya bisa dilihat pada Gambar 16 dan Gambar 17.



Gambar 15 Halaman Master Pelanggan

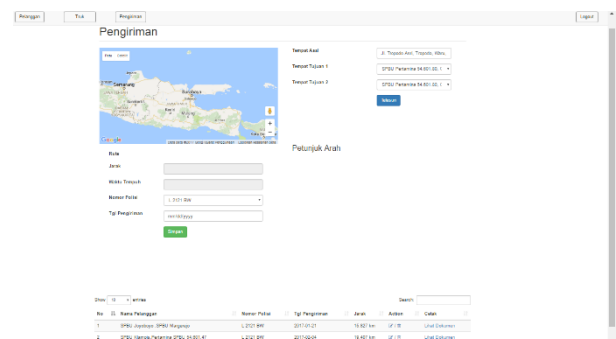
Tombol “Ubah” dipilih maka akan menampilkan form ubah sesuai dengan data yang dipilih.



Gambar 16 Form Ubah Pelanggan

Halaman Pengiriman

Halaman pengiriman digunakan untuk pengisian data pengiriman yang berisi tempat tujuan pengiriman, rute, jarak, waktu tempuh, nomor polisi dan tanggal pengiriman. Pada halaman ini juga terdapat fitur ubah untuk mengubah maupun menghapus data pengiriman. Untuk lebih lengkapnya bisa dilihat pada Gambar 18 dan Gambar 19.

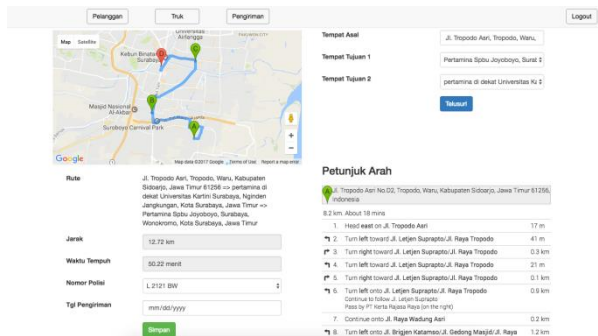


Gambar 17 Halaman Pengiriman

Tombol “Lihat Dokumen” dipilih maka akan menampilkan preview report pengiriman dari data yang dipilih untuk dicetak.

HASIL UJI COBA

Pengujian pencarian jalur terpendek pada program yakni dari titik awal CV. Mega Lestari ke dua tempat tujuan pendistribusian solar. Hasil perhitungan dari total jarak antara titik awal ke dua tempat tujuan pendistribusian solar dimana jalur dari titik awal CV. Mega Lestari menuju titik tujuan pertama ke SPBU UKS dan tujuan akhir ke SPBU Joyoboyo adalah 12,72 km. Untuk lebih lengkapnya dapat dilihat pada Gambar 18.



Gambar 18 Hasil Pencarian Program

KESIMPULAN

Berdasarkan hasil evaluasi dari Pembuatan Aplikasi Penentuan Jalur Terpendek Pendistribusian Solar dengan Metode Algoritma Dijkstra Berbasis Web Pada CV. Mega Lestari. Aplikasi ini dapat memenuhi tujuan yang diharapkan yaitu dapat menentukan jalur

terpendek antara dua tujuan dan memperkirakan jarak dan juga waktu dari tempat asal ke tempat tujuan pendistribusian solar.

RUJUKAN

Arifianto, Sofyan. 2012. *Sistem Aplikasi Penentuan Rute Terpendek Pada Jaringan Multi Moda Transportasi Umum Menggunakan Algoritma Dijkstra*. Semarang: Universitas Diponegoro.

Google. 2017. *Jarak & Durasi Untuk Berbagai Tujuan Dan Moda Transportasi*. (Online). (<https://developers.google.com/maps/documentation/distance-matrix/intro?hl=id>, diakses 31 Januari 2017).

Munir, Renaldi. 2011. *Algoritma & Pemrograman dalam Bahasa Pascal dan C*. Bandung: Informatika Bandung.

Nugroho, Adi. 2010. *Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP*. Yogyakarta: Andi.

Sholih. 2010. *Analisis dan Perancangan Berorientasi Obyek*. Bandung: Muara Indah.

Tegarden, David., Dennis, Alan., Haley Wixom, Barbara. 2013. *System Analysis and Design with UML 4th Edition*. Singapore: John Wiley & Sons.

Report Pengiriman

Kamis, 16 Maret 2017

Nama Pemesan : Pertamina 54 601 115,SPBU Sby
 Alamat : Jl. Raya Nginden, Baratajaya, Gubeng, Kota SBY, Jawa Timur 60284,SPBU Pertamina 51.601.65, Surabaya, Jemur Wonosari, Kota Surabaya, Jawa Timur
 Tip : (031) 5021274,-

No	Rute	Jarak	Waktu Tempuh	Tgl Pengiriman
1	Jl. Tropodo Asri, Tropodo, Waru, Kabupaten Sidoarjo, Jawa Timur 61256 => SPBU Pertamina 51.601.65, Surabaya, Jemur Wonosari, Kota Surabaya, Jawa Timur => Jl. Raya Nginden, Baratajaya, Gubeng, Kota SBY, Jawa Timur 60284	15.03 km	36.10 menit	2017-02-17
<p>A Jl. Tropodo Asri No.D2, Tropodo, Waru, Kabupaten Sidoarjo, Jawa Timur 61256, Indonesia</p> <p>8,2 km. Sekitar 18 menit</p> <p>1. Ke arah timur di Jl. Tropodo Asri 17 m</p> <p>2. Belok kiri menuju Jl. Letjen Suprpto/Jl. Raya Tropodo 41 m</p> <p>3. Belok kanan menuju Jl. Letjen Suprpto/Jl. Raya Tropodo 0,3 km</p> <p>4. Belok kiri menuju Jl. Letjen Suprpto/Jl. Raya Tropodo 21 m</p> <p>5. Belok kanan menuju Jl. Letjen Suprpto/Jl. Raya Tropodo 0,1 km</p> <p>Belok Kiri ke Jl. Letjen Suprpto/Jl. Raya Tropodo</p> <p>6. Lanjutkan untuk mengikuti Jl. Letjen Suprpto Lewati PT Kerta Rajasa Raya (di kanan) 0,9 km</p> <p>7. Terus ke Jl. Raya Wadung Asri 0,2 km</p> <p>Belok kiri ke Jl. Brigjen Katamsa/Jl. Gedung Masjid/Jl. Raya Berbek</p> <p>8. Lanjutkan untuk mengikuti Jl. Brigjen Katamsa Lewati PG Darul Ulum (di kanan) 1,2 km</p> <p>9. Belok kanan ke Jl. Berbek Industri VII Lewati PT.Kanefusa Indonesia (di kanan 450 m lagi) 0,7 km</p> <p>10. Jl. Berbek Industri VII belok kiri dan menjadi Jl. Berbek Industri II Lewati PT Rexplast (di kiri) 0,4 km</p> <p>11. Ambil jalan ke Mojokerto/Tj Perak/Lewat Jalan tol 0,6 km</p> <p>12. Bergabung ke Jl. Tol Waru - Juanda Jalan tol 1,7 km</p> <p>13. Ambil jalan keluar menuju Jl. Wisata Menanggal 0,1 km</p> <p>14. Terus ke Jl. Wisata Menanggal 0,6 km</p> <p>15. Belok kanan ke Jl. Kertomenanggal Lewati Korem 084/Bj (di kanan) 0,4 km</p> <p>16. Belok kiri di Jl. Menanggal Timur 25 m</p> <p>17. Belok kiri ke Jl. Ahmad Yani/Jl. Raya Malang - Surabaya Lewati BNI Graha Pangeran (di kanan) 0,3 km</p> <p>18. Belok sedikit ke kanan menuju Jl. Ahmad Yani 0,2 km</p> <p>19. Terus lurus ke Jl. Ahmad Yani/Jl. Raya Malang - Surabaya Lewati BNI Graha Pangeran (di kiri) 0,5 km</p> <p>Belok kiri ke Jl. Menanggal VI</p> <p>20. Tujuan ada di sebelah kanan. 93 m</p> <p>B Jl. Menanggal VI No.69, Menanggal, Gayungan, Kota SBY, Jawa Timur 60234, Indonesia</p> <p>3,6 km. Sekitar 9 menit</p> <p>1. Ke arah timur di Jl. Menanggal VI menuju Jl. Ahmad Yani/Jl. Raya Malang - Surabaya 92 m</p> <p>2. Belok kiri ke Jl. Ahmad Yani/Jl. Raya Malang - Surabaya Lewati Carrefour (di kiri) 1,5 km</p> <p>3. Belok kanan ke Jl. Ahmad Yani 0,1 km</p> <p>4. Belok kiri ke Jl. Raya Jemur Sari (rambu Prapen/Pt. Sier) Belok kiri untuk tetap di Jl. Raya Jemur Sari Lewati Dante Coffee Shop (di kiri 1,5 km lagi) 0,3 km</p> <p>5. Tujuan ada di sebelah kiri. 1,6 km</p> <p>C Jl. Raya Jemur Sari No.111, Jemur Wonosari, Wonocolo, Kota SBY, Jawa Timur 60237, Indonesia</p> <p>3,2 km. Sekitar 8 menit</p> <p>Ke arah timur laut di Jl. Raya Jemur Sari menuju Kendang Sari</p> <p>1. Lanjutkan untuk mengikuti Jl. Raya Jemur Sari Lewati CIMB Niaga (di kiri) 1,4 km</p> <p>2. Di Autohaus, terus ke Jl. Raya Prapen Lewati Dapur Cokelat Jemursari (di kiri) 0,8 km</p> <p>3. Di Auto Garage, terus ke Jl. Raya Nginden Lewati Duitara Abadi Raya (di kiri) 0,8 km</p> <p>4. Belok kiri ke Jl. Bratang Jaya 0,1 km</p> <p>5. Belok kanan sebelum pompa bensin/SPBU Tujuan ada di sebelah kiri. 29 m</p> <p>D Jl. Bratang Jaya No.80-82, Baratajaya, Gubeng, Kota SBY, Jawa Timur 60284, Indonesia</p> <p>Data peta ©2017 Google</p>				

Gambar 19 Preview Pengiriman