

PERBANDINGAN ALGORITMA A* DENGAN ALGORITMA DIJKSTRA UNTUK PENCARIAN JARAK DAN RUTE TERPENDEK BERBASIS WEB

Nursetia Wati¹, Sri Hastuti Permatasari²

e-mail: ¹ nursetiawati@umgo.ac.id, ² iyaiyako@gmail.com

^{1,2} Prodi S1 Sistem Informasi Universitas Muhammadiyah Gorontalo,

Penggunaan Gas Elpiji di Kota Gorontalo mulai meningkat sejak awal tahun 2012. Hal ini dilihat dengan semakin banyaknya penyebaran pangkalan-pangkalan elpiji di Kota Gorontalo. Namun, tidak semua masyarakat mengetahui lokasi serta rute untuk menuju ke pangkalan-pangkalan tersebut. Oleh karena itu, penulis bermaksud merancang sebuah aplikasi web yang dapat menampilkan lokasi serta rute terpendek untuk sampai ke pangkalan yang akan dituju. Ada beberapa algoritma yang digunakan dalam pencarian rute terpendek, namun algoritma yang digunakan pada penelitian ini yaitu hanyalah algoritma A dan algoritma Dijkstra. Tujuan dari penelitian ini yaitu membandingkan algoritma A* dan algoritma Dijkstra untuk melihat manakah yang lebih cepat dalam pencarian rute terpendek dari posisi awal user ke lokasi tujuan. Metode penelitian yang digunakan pada penelitian ini yaitu Metode Web Development Life Cycle (WDLC). Dimana, penulis membandingkan algoritma A* dan algoritma Dijkstra dari segi pencariannya yang sistem/aplikasinya akan dijalankan pada Web. Hasil penelitian menunjukkan perbedaan dari algoritma A* dan algoritma Dijkstra hanya ada pada perhitungan untuk mencari rute terpendek lokasi tujuan. Dimana, pada algoritma A* dapat dilakukan dengan cepat sebab tidak dilakukan pada semua node yang terhubung, hanya melihat setiap node yang bernilai kecil untuk dilewati, dan mengikuti node-node bernilai kecil tersebut hingga sampai ke lokasi tujuan. Sedangkan pada algoritma Dijkstra, pencariannya dilakukan pada semua node yang terhubung dengan lokasi awal user. Sehingga, perhitungan untuk mencari rute terpendek menjadi lambat untuk sampai ke lokasi tujuan.*

Kata Kunci: Elpiji, Algoritma A*, Algoritma Dijkstra, Web Development Life Cycle (WDLC)

I. PENDAHULUAN

LPG (*Liquified Petroleum Gas*) merupakan salah satu bahan bakar bersubsidi yang di programkan oleh pemerintah sejak tahun 2007 sebagai pengganti bahan bakar minyak tanah yang semakin langka. Sejak awal tahun 2012, penggunaan bahan bakar ini di Kota Gorontalo mulai meningkat.

Dengan meningkatnya pemenuhan kebutuhan LPG oleh masyarakat Kota Gorontalo, maka agen dan pangkalan LPG tersebut sudah tersebar di beberapa kecamatan dan kelurahan yang ada di Kota Gorontalo. Namun, tidak menjamin banyak masyarakat yang mengetahui lokasi-lokasi serta jarak dan rute terpendek dari posisi awal pengguna menuju agen dan pangkalan yang akan dituju.

Berdasarkan uraian diatas, maka penulis bertujuan ingin membandingkan Algoritma A* dengan Algoritma Dijkstra untuk melihat manakah algoritma yang paling cepat dan akurat dalam proses pencariannya dari posisi awal *user* menuju ke lokasi pangkalan tujuan.

Algoritma A*

Algoritma A* adalah Algoritma Komputer yang digunakan secara luas dalam graph traversal dan penemuan jalur serta proses perencanaan jalur yang bias dilewati secara efisien disekitar titik titik yang disebut *node* [1]. Algoritma A* adalah algoritma *best first search* yang paling banyak dikenal. Algoritma ini memeriksa *node* dengan menggabungkan $g(n)$ yaitu *cost* yang dibutuhkan untuk mencapai sebuah *node*, dan $h(n)$ yaitu *cost* yang didapat dari *node* ke tujuan. Sehingga didapatkan rumus Algoritma A* ini yaitu sebagai berikut [2]:

$$F(n) = g(n) + h(n)$$

Keterangan :

$g(n)$ = biaya (*cost*) yang dibutuhkan oleh sebuah jalur (*path*) untuk mencapai *node* n dari *node* awal

$h(n)$ = estimasi biaya (*cost*) sebuah jalur (*path*)

$f(n)$ = estimasi total biaya (*cost*) sebuah jalur (*path*) dari *node* awal ke *node* tujuan (*goal*) melalui *node* n.

Algoritma Dijkstra

Algoritma *Dijkstra* ditemukan oleh Edger Wybe Dijkstra. Algoritma *Dijkstra* adalah algoritma untuk menemukan jarak terpendek dari suatu *vertex* ke *vertex* yang lainnya pada suatu *graph* yang berbobot, dimana jarak antar *vertex* adalah bobot dari tiap *edge* pada *graph* tersebut [3]. Selain itu algoritma *Dijkstra* dapat menyelesaikan beberapa kasus jalur terpendek, antara lain: pencarian jalur terpendek antara dua buah simpul (*a pair shortest path*) [4], pencarian jalur terpendek antara semua pasangan simpul (*all pairs shortest path*), pencarian jalur terpendek dari simpul tertentu ke semua simpul yang lain (*single-*

source shortest path) dan pencarian jalur terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermediate shortest path*) Algoritma *dijkstra* mencari jarak terpendek dari *vertex* asal ke *vertex* terdekatnya, kemudian ke *vertex* yang kedua, dan seterusnya [3].

Spherical Law of Cosines

Rumus *Spherical law of cosines* pada penelitian ini digunakan untuk mengkalkulasi jarak dari posisi pengguna ke agen dan pangkalan yang dituju. Dimana, titik koordinat (Latitude dan Longitude) yang diperoleh dengan menggunakan GPS, di konversi ke dalam bentuk jarak (km) dengan menggunakan rumus *Spherical law of cosines*. Adapun rumus *Spherical law of cosines* yang digunakan yaitu sebagai berikut:

$$d = \text{acos}(\sin(\text{lat}_1) \cdot \sin(\text{lat}_2) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \cos(\text{long}_2 - \text{long}_1)) \cdot R$$

Keterangan:

- R = Radius bumi (rata-rata radius = 6,371 kilometer)
- d = jarak (km)

II. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini yaitu metode *Web Development Life Cycle (WDLC)* dengan beberapa tahapan yang digunakan sebagai berikut [5]:

1. Tahapan Perancangan

Tuntut menghasilkan *website* yang baik diperlukan perencanaan yang matang yaitu dengan melakukan studi kelayakan diantaranya pengumpulan data, dan teknik analisis data.

a. Metode Pengumpulan data

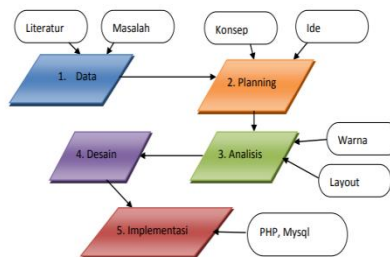


Diagram Metodologi Perancangan

1. Pengumpulan data

Pengumpulan data ini dilakukan untuk mengidentifikasi permasalahan yang ada.

2. *Planning*

Tahapan ini dilakukan untuk menentukan ide dan konsep sebuah *website* yang akan dibuat.

3. Analisis

Tahapan ini dilakukan untuk menganalisis perbandingan algoritma A* dan algoritma Dijkstra untuk menyelesaikan proses pencarian agen dan pangkalan elpiji yang ada dengan menggunakan cara yang manual.

Dalam menyelesaikan proses pencarian agen dan pangkalan elpiji dengan menggunakan algoritma A*, terlebih dahulu menentukan lokasi awal dan lokasi tujuan. Setelah itu, menghitung *cost* untuk mencapai sebuah node n dari node awal yang dalam hal ini disimbolkan dengan g(n). Setelah nilai g(n) diperoleh, maka dilihat *cost* yang terkecil diantara *cost-cost* yang ada. Jika *cost* yang terkecil sudah diperoleh, kemudian menghitung nilai perkiraan/heuristik dari node n yang dalam hal ini adalah *cost* terkecil yang sudah diperoleh ke node tujuan yang disimbolkan dengan h(n). Nilai h(n) diperoleh dari perhitungan antara dua koordinat yaitu lokasi node n dan lokasi tujuan. Setelah nilai h(n) didapatkan, maka yang terakhir yaitu menghitung jumlah/total dari sebuah jalur dari lokasi awal ke lokasi tujuan, yaitu dengan cara menjumlahkan nilai g(n) dan nilai h(n) sehingga menghasilkan nilai f(n). Nilai f(n) itu merupakan jarak serta rute terpendek antara dua lokasi dengan menggunakan penyelesaian algoritma A*.

Sedangkan dengan menggunakan Algoritma Dijkstra, terlebih dahulu menentukan vertex awal dan vertex tujuan. Kemudian, status dari vertex yang belum terpilih diinisialisasikan dengan '0' dan yang sudah terpilih diinisialisasikan dengan '1', dimulai dari vertex awal. Selanjutnya, menentukan bobot dari masing-masing vertex yang terhubung langsung dengan vertex awal, serta menentukan vertex sumbernya. Setelah memperoleh vertex yang memiliki bobot paling kecil dengan status '0', maka status vertex tersebut diubah menjadi '1'. Untuk mendapatkan hasil jarak terpendek dari vertex awal ke vertex tujuan, dilakukan dengan cara seperti diatas dengan melihat vertex yang memiliki

bobot terkecil dengan status '0'. Cara diatas dilakukan sampai semua status dari masing-masing vertex menjadi '1' sehingga mendapatkan jarak terpendek dari vertex awal ke vertex tujuan serta dari vertex awal ke masing-masing vertex.

4. Desain
Tahapan ini dilakukan untuk mendesain *website*.
5. Implementasi
Rancangan sistem/aplikasi yang telah dilakukan sebelumnya, kemudian diterapkan pada tahap ini untuk proses pembuatannya dengan *PHP* yang digunakan sebagai bahasa pemrograman serta penerapan Algoritma A* dan Algoritma Dijkstra pada aplikasi ini.
6. Pengujian
Pada tahap ini dilakukan uji coba terhadap aplikasi yang telah selesai dibuat untuk dilihat kesesuaiannya pada saat dijalankan.
7. Menyusun Laporan dan Menarik Kesimpulan.

III. HASIL DAN PEMBAHASAN

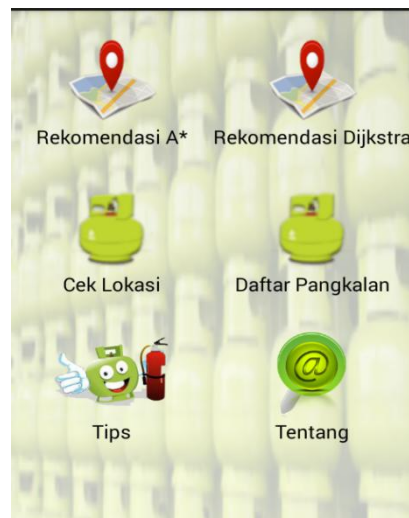
Perancangan sistem atau aplikasi perbandingan Algoritma A* dengan Algoritma Dijkstra, hanya melibatkan satu aktor saja, yaitu *user*.

Pada sistem/aplikasi yang dibuat terdapat 6 (enam) menu yang dapat diakses oleh pengguna, seperti menu Rekomendasi A*, Rekomendasi Dijkstra, Cek Lokasi, Daftar Pangkalan, Tips, dan Tentang. Fungsi menu Rekomendasi A* dan Rekomendasi Dijkstra digunakan untuk melihat rute terpendek lokasi pangkalan yang terdekat dari posisi *user* dengan algoritma A* dan algoritma Dijkstra. Fungsi menu Cek Lokasi yaitu untuk melihat sebaran pangkalan-pangkalan yang terdaftar dan posisi *user*. Menu Daftar Pangkalan digunakan untuk melihat daftar pangkalan-pangkalan yang ada, juga digunakan untuk melihat rute terpendek untuk menuju pangkalan tersebut dengan memilih algoritma A* ataupun algoritma Dijkstra. Sedangkan pada menu Tips berisi beberapa tips seputar elpiji. Untuk menu Tentang merupakan menu tambahan yang berisi informasi menu-menu pada aplikasi.

Berikut ini beberapa *screenshots* implementasi system.

a) Tampilan Halaman Utama

Pada tampilan Halaman Utama ini terdapat 6 (enam) menu yang dapat diakses oleh pengguna, seperti menu Rekomendasi A*, Rekomendasi Dijkstra, Cek Lokasi, Daftar Pangkalan, Tips, dan Tentang, seperti terlihat pada gambar 2 dibawah ini.



Gambar 2. Tampilan Menu Utama

b) Tampilan Rekomendasi A*

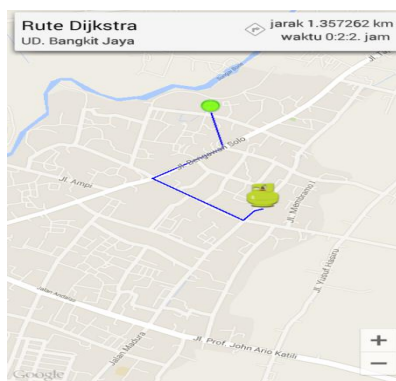
Pada tampilan Rekomendasi A* terlihat rute terpendek algoritma A* dari posisi *user* ke lokasi tujuan, seperti terlihat pada gambar 3 dibawah ini:.



Gambar 3. Tampilan Rekomendasi A*

c) Tampilan Rekomendasi Dijkstra

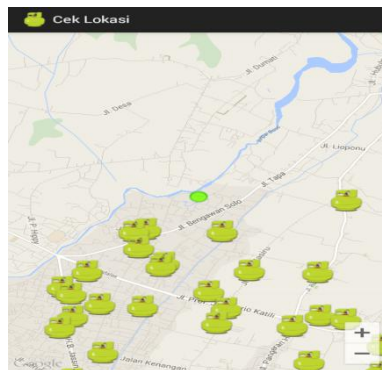
Pada tampilan Rekomendasi A* terlihat rute terpendek algoritma A* dari posisi *user* ke lokasi tujuan, seperti terlihat pada gambar 4 dibawah ini:



Gambar 4. Tampilan Rekomendasi Dijkstra

d) Tampilan Cek Lokasi

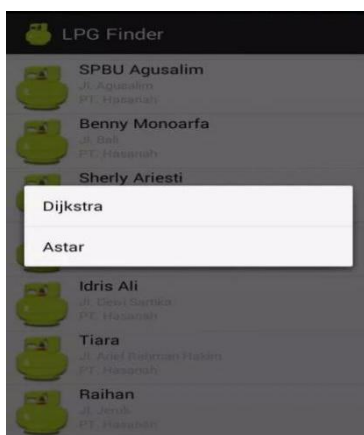
Pada tampilan Menu Cek Lokasi ini terlihat sebaran pangkalan-pangkalan yang terdaftar serta posisi *user*, seperti terlihat pada gambar 5 dibawah ini.



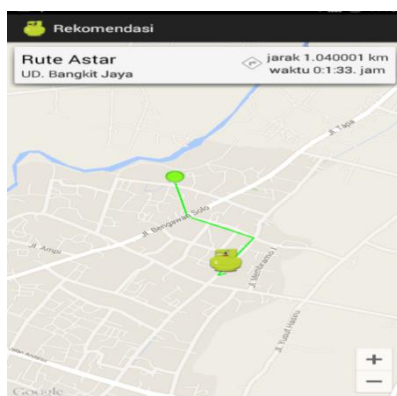
Gambar 5. Tampilan Menu Cek Lokasi

e) Tampilan Menu Daftar Pangkalan

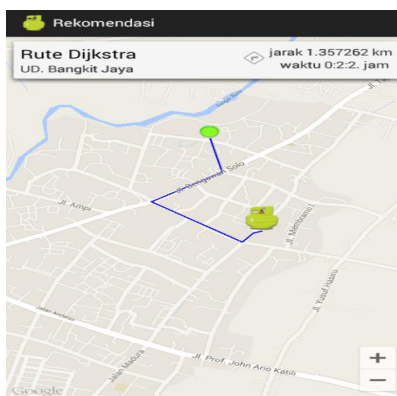
Pada tampilan Menu Daftar Pangkalan ini terdapat daftar-daftar pangkalan LPG di Kota Gorontalo pada beberapa Kecamatan, seperti terlihat pada gambar 6. Pada menu ini juga dapat menampilkan rute terdekat dari lokasi *user* ke lokasi agen/pangkalan yang dituju dengan mengklik salah satu agen/pangkalan yang akan dituju, kemudian memilih algoritma A* atau Dijkstra. Tampilannya dapat dilihat pada gambar 7 dan gambar 8 dibawah ini.



Gambar 6. Tampilan Menu Daftar Pangkalan



Gambar 7. Tampilan Rute Algoritma A*



Gambar 8. Tampilan Rute Algoritma Dijkstra

Berdasarkan kedua gambar tampilan rute algoritma A* dan algoritma Dijkstra diatas, dapat dilihat bahwa kedua algoritma itu menghasilkan dua rute yang berbeda dari posisi *user* dan lokasi tujuan yang sama, yaitu dari posisi *user* Jalan Poowo dan lokasi tujuan Pangkalan UD. Bangkit Jaya. Dimana, terlihat dengan menggunakan algoritma A* jarak tempuhnya 1,040001 km dalam waktu 1 menit 33 detik, sedangkan dengan menggunakan algoritma Dijkstra jarak tempuhnya sama yaitu 1,357262 km dalam waktu 2 menit 20 detik.

Penyelesaian dengan Menggunakan Algoritma A*

Algoritma A* dari titik awal node 23 dan titik tujuan node 580 yaitu memiliki total jarak 1,040001 km. Rute terpendek ini diperoleh dari total jarak dari node 23-24-642-580. Hasil evaluasi perhitungan rute terpendek dengan menggunakan algoritma A*, yaitu sebagai berikut:

Langkah 1

Node awal adalah node 23. Kemudian dilihat node-node yang terhubung dengan node 23, yaitu node 24 dan 638. Fungsi evaluasinya yaitu:

$$f(24) = g(24) + h(24) = 0.370336160 + 0.317555847 = 0.687892007$$

$$f(638) = g(638) + h(638) = 0.332238075 + 0.536718753 = 0.868956829$$

Dari hasil perhitungan langkah 1 diatas, node 24 terpilih sebagai *best node*. Sampai langkah pertama ini menghasilkan rute dari node 23-24.

Penyelesaian dengan Menggunakan Algoritma Dijkstra

Algoritma Dijkstra dari titik awal node 23 dan titik tujuan node 580 memiliki total jarak yang berbeda, yaitu 1.357262 km. Rute terpendek ini diperoleh dari total jarak dari node 23-638-22-12-642-580 yang merupakan total jalur terpendek dari beberapa jalur node yang lain. Total jarak jalur terpendek tersebut jika dihitung secara manual, maka hasilnya akan terlihat sebagai berikut:

1. Jarak dari *node* 23 (koordinat 0.583360, 123.060390) ke *node* 638 (koordinat 0.581641, 123.057946), maka jarak antar kedua *node* tersebut adalah sebagai berikut:

$$d = \text{acos}(\sin(0.010181553) \cdot \sin(0.010151551) + \cos(0.010181553) \cdot \cos(0.010151551) \cdot \cos(2.147766328 - 2.147808984)) \cdot 6371$$

$$d = 0.332238075 \text{ km.}$$

Untuk langkah selanjutnya perhitungannya sama seperti diatas. Kemudian ditotalkan seluruh jarak yang diperoleh tersebut.

Dari perhitungan dua algoritma ini, dapat dilihat bahwa dari segi waktu atau kecepatan pencarian, algoritma A* dikatakan lebih cepat dibandingkan dengan algoritma Dijkstra. Hal ini dikarenakan pencarian dengan menggunakan algoritma A* tidak membutuhkan waktu dan memori yang banyak dalam menampilkan rute terpendek, karena tidak semua node untuk mencapai lokasi tujuan dilakukan pengecekan. Lain halnya dengan algoritma Dijkstra, membutuhkan waktu dan memori yang cukup banyak dalam pencariannya karena semua node yang terhubung untuk mencapai lokasi tujuan dilakukan pengecekan. Namun, karena pada algoritma Dijkstra semua node yang terhubung dilakukan pengecekan, maka dapat dikatakan algoritma Dijkstra ini lebih efektif dalam hal proses pencarian rute terpendek dibandingkan dengan Algoritma A*.

IV. KESIMPULAN

Berdasarkan hasil penelitian perbandingan algoritma A* dengan algoritma Dijkstra untuk mengetahui jarak dan rute terpendek agen serta pangkalan Elpiji di Kota Gorontalo, maka dapat disimpulkan bahwa:

- 1) Aplikasi yang dibuat mampu menampilkan rute terpendek dari lokasi *user* ke lokasi agen/pangkalan yang ada berdasarkan perhitungan dengan algoritma A* dan algoritma Dijkstra.
- 2) Perbedaan dari kedua algoritma ini hanyalah pada pencariannya. Dimana, pencarian dengan menggunakan algoritma A* lebih cepat dibandingkan dengan algoritma Dijkstra. Namun, Algoritma Dijkstra dapat dikatakan lebih efektif dalam pencariannya dibandingkan dengan Algoritma A*, hal ini dikarenakan pencarian dengan algoritma Dijkstra memeriksa semua node yang memungkinkan untuk sampai ke tujuan.

REFERENSI

- [1] Reddy, H., 2013. Path Finding-Dijkstra's and A* Algorithm's. [Online] Available at: <http://cs.indstate.edu/hgopireddy/newalg.html> [Accessed 25 September 2016].
- [2] Russel & Norvig, 2003. *Artificial Intelligence: A Modern Approach (1995) hal 63*
- [3] Muzakkir R. 2013. *Sistem Informasi Geografis Berbasis Web Untuk Menentukan Rute Terpendek Dengan Algoritma Dijkstra Di Kota Gorontalo (Studi kasus tempat perbelanjaan seperti Mall, Supermarket dan Apotik)*. Skripsi. Gorontalo : Universitas Negeri Gorontalo.
- [4] Ardiani, Farida. 2011. Penentuan jarak terpendek dan waktu tempuh menggunakan algoritma Dijkstra dengan pemrograman berbasis objek. Skripsi Universitas Islam Negeri Sunan Kalijaga
- [5] Kamatchi, R. et. 2013. Software Engineering: Web Development Life Cycle. Internasional journal of Engineering Reserch & Technology.