



Aplikasi Sistem Pencarian Halte BRT Terdekat Kota Semarang Menggunakan Metode A* Berbasis Android

Abimanyu Cahya Pramudhita^a, Muljono^b

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro,

^aabi.pramudhita@gmail.com, ^bmuljono@dsn.dinus.ac.id

Abstract

BRT (Bus Rapid Transit) is one of public transportation that can be used in Semarang City. This bus is operated to break down the congestion in Semarang city which is increasing. This paper aims to make the application of the bus stop search system closest and is expected to facilitate the people of Semarang city in the use of BRT (Bus Rapid Transit). A Algorithm, This algorithm is a Best First Search algorithm that combines Uniform Cost Search and Greedy Best-First Search. Where the Price considered $f(n)$ is derived from the actual price $g(n)$ plus the approximate price $h(n)$. The created application provides the nearest stop information by using the A* method. From the calculations performed using the A* method and conducted 20 experiments at different locations, the A* method has 100% accuracy in determining the nearest up and down stops, if in an accurate Global Positioning System.*

Keywords: Information System, A, BRT System, Bus Rapid Transit, Android.*

Abstrak

BRT (Bus Rapid Transit) adalah salah satu transportasi umum yang dapat digunakan di Kota Semarang. Bus ini dioperasikan guna mengurangi kemacetan di Kota Semarang yang semakin meningkat. Makalah ini memiliki tujuan untuk membuat aplikasi sistem pencarian halte bus terdekat dan diharapkan dapat mempermudah masyarakat kota Semarang dalam penggunaan BRT (Bus Rapid Transit). Algoritma A*, Algoritma ini merupakan algoritma Best First Search yang menggabungkan Uniform Cost Search dan Greedy Best-First Search. Dimana Harga yang dipertimbangkan $f(n)$ didapat dari harga sesungguhnya $g(n)$ ditambah dengan harga perkiraan $h(n)$. Aplikasi yang dibuat memberikan informasi halte terdekat dengan menggunakan metode A*. Dari perhitungan yang dilakukan menggunakan metode A* dan dilakukan 20 percobaan pada lokasi yang berbeda, metode A* memiliki akurasi 100% dalam menentukan halte naik dan turun terdekat, apabila dalam keadaan Global Positioning System yang akurat.

Kata Kunci: Sistem Informasi, A, Sistem BRT, Bus Rapid Transit, Android.*

© 2018 Jurnal RESTI

1. Pendahuluan

Pencarian (searching) adalah salah satu topik utama dalam kecerdasan buatan (artificial intelligence). Terdapat banyak algoritma pencarian dapat diimplementasikan untuk memecahkan masalah di kehidupan sehari-hari. Salah satunya penerapan algoritma pencarian adalah untuk pencarian halte BRT (Bus Rapid Transit) terdekat.

BRT (*Bus Rapid Transit*) adalah salah satu angkutan umum atau transportasi umum yang dapat digunakan di Kota Semarang. Bus ini dioperasikan guna mengurangi kemacetan di Kota Semarang yang semakin meningkat. Hal yang membedakan BRT (*Bus Rapid Transit*) dengan bus kota lainnya adalah pintu otomatis yang terletak lebih tinggi, tidak boleh mengangkut atau menurunkan penumpang selain melalui halte BRT.

Sehingga penumpang hanya dapat naik di halte khusus yang telah disediakan. Terdapat banyak sekali halte dan rute untuk BRT (*Bus Rapid Transit*) di Kota Semarang. Bus ini ditujukan agar perjalanan penumpang lebih efisien dan memiliki halte pemberhentian yang tertata. Namun

tidak banyak sistem yang menunjukkan halte terdekat untuk naik dan turun penumpang dengan perhitungan yang tepat atau efisien. Beberapa sistem biasanya hanya menyediakan halte terdekat dengan menggunakan perhitungan yang dilakukan menggunakan API dari *Google Maps* tanpa mengerti algoritma apa yang ada di dalamnya [1].

Dari aplikasi BRT yang telah dibuat sebelumnya, ada banyak fitur kelebihanannya, beberapa fitur tersebut yaitu dapat mengetahui posisi bus terdekat. Dapat

memberikan informasi CCTV di beberapa tempat di Semarang, dan informasi mengenai beberapa titik shelter yang ada. Namun juga terdapat beberapa kekurangan dari aplikasi BRT, aplikasi ini hanya menunjukkan rute naik dan turun untuk shelter atau halte BRT tetapi tidak dapat menunjukkan halte tempat naik dan turun terdekat dari keinginan pengguna.

Algoritma A*, Algoritma ini merupakan algoritma Best First Search yang menggabungkan Uniform Cost Search dan Greedy Best-First Search. Dimana Harga yang dipertimbangkan $f(n)$ didapat dari harga sesungguhnya $g(n)$ ditambah dengan harga perkiraan $h(n)$ [2].

Dari paparan di atas tentang *Bus Rapid Transit* dan algoritma A* dapat dilakukan penelitian tentang pembuatan sistem informasi halte BRT untuk mencari halte bus terdekat saat pengguna ingin naik bus dan halte bus terdekat untuk penurunan dengan menggunakan metode A*. Agar dapat diperoleh halte terdekat sebagai informasi yang dihasilkan khususnya bagi masyarakat awam atau yang berasal dari luar kota.

Berdasarkan pada latar belakang masalah di atas, maka perumusan masalah yang akan diselesaikan dalam penelitian ini yaitu sebagai berikut :

1. Calon penumpang kesulitan menentukan posisi halte bus naik dan turun terdekat pada tujuan tertentu.
2. Bagaimana menginformasikan kepada calon penumpang untuk lokasi halte terdekat menggunakan metode A* (A – star).
3. Apakah A* akurat digunakan untuk mencari lokasi terdekat.

Dari perumusan masalah, maka perlu adanya suatu batasan masalah. Beberapa batasan masalah tersebut diantaranya, yaitu :

1. Pembuatan sistem dengan bahasa pemrograman PHP, Java Android dan *database* MySQL.
2. Data rute dan halte yang digunakan hanya di ruang lingkup BRT kota Semarang.
3. Penelitian dilakukan di dinas perhubungan Kota Semarang untuk pengambilan data.

Tujuan dari penelitian ini, yaitu sebagai berikut:

1. Dengan membuat aplikasi sistem pencarian halte bus tersebut, diharapkan dapat mempermudah masyarakat kota Semarang dalam penggunaan BRT.
2. Mengimplementasikan metode A* dalam pencarian halte BRT terdekat pada tujuan tertentu.
3. Membuktikan apakah algoritma A* akurat digunakan untuk sistem pencarian halte naik dan turun terdekat.

2. Tinjauan Pustaka

Dalam penelitian hal yang termasuk penting adalah penelitian terkait atau penelitian yang sudah dilakukan sebelumnya untuk dijadikan acuan dalam penelitian terkini. Dan berikut adalah beberapa penelitian terkait yang serupa dengan penelitian ini.

Penelitian dengan judul “Penerapan Algoritma A* Pada Permasalahan Optimalisasi Pencarian Solusi Dynamic Water Jug” untuk mencari solusi permasalahan optimalisasi pencarian solusi dynamic water jug. Optimalisasi adalah sebuah proses memodifikasi sistem untuk membuat beberapa aspek agar bekerja lebih efisien atau menggunakan *resource* (sumber) lebih sedikit. Sebuah program komputer di optimalisasi sehingga bisa menjalankan tugasnya lebih dengan cepat, atau mampu untuk beroperasi dalam pengurangan sejumlah memory storage [2].

Algoritma A* pada penelitian [3] digunakan untuk mencari lokasi gedung dan ruangan di Universitas Islam Negeri Sultan Syarif Kasim Riau. Sistem yang dibangun menggunakan platform Android. Peta digital UIN Suska Riau dibangun dengan bahasa pemrograman Action Script 3 dan menampilkan informasi rute sehingga memberikan kemudahan bagi pengguna menemukan lokasi gedung dan ruangan UIN Suska Riau.

Penelitian [4] juga menggunakan algoritma A* untuk pencarian mesin ATM terdekat di Palembang. ATM mulai banyak ditemui di berbagai tempat umum dan strategis seperti pusat perbelanjaan, minimarket dan tempat khusus yang telah disediakan oleh pihak bank. Sistem ini dapat digunakan masyarakat umum untuk mencari lokasi ATM di daerah sekitarnya maupun di daerah yang baru di datangi sehingga pengguna tidak perlu bertanya pada warga sekitar dimana lokasi ATM terdekat. Sedangkan penelitian [5] menerapkan algoritma A* dalam optimasi penentuan halte dan rute transmisi untuk setiap kasus sehingga dapat berjalan sesuai dengan yang diharapkan dan keadaan nyatanya, dalam arti hasilnya sesuai dengan perhitungan manual berdasarkan parameter jarak dan jumlah armada sehingga menghasilkan halte dan rute transmisi yang optimal bagi pengguna sesuai dengan koordinat awal pengguna dan koordinat tujuan.

Pada beberapa penelitian lainnya, algoritma A* diimplementasikan untuk aplikasi game, pada [6] dijelaskan bahwa penggunaan algoritma A* berhasil diterapkan sebagai pembangkit perilaku pencarian pada NPC pada game 3D pembelajaran kosakata Bahasa Arab. Pada [7] algoritma A* diterapkan pada Game Edukasi The Maze Island berbasis Android dan didapatkan bahwa algoritma A* lebih baik daripada algoritma lainnya karena algoritma ini memiliki nilai heuristik sebagai nilai pembandingan sehingga dapat mencari solusi yang terbaik dari permasalahan yang

ada jika solusi itu memang ada. Sedangkan penelitian [8] menerapkan algoritma A* sebagai pembangkit perilaku pencarian pada NPC untuk game 3D Wayang Adventure Penelitian ini difokuskan pada platform desktop [8].

Penelitian “Penerapan Algoritma A* (Star) Untuk Mencari Rute Tercepat Dengan Hambatan” mempelajari cara kerja algoritma A* dalam mencari jarak tercepat, yang disimulasikan dengan kondisi seorang mencari rute dalam keadaan jalanan macet dan memiliki menyimpulkan bahwa algoritma A* dapat diterapkan sebagai algoritma untuk menentukan rute (jalur) terbaik yang akan dilalui [9].

Karena banyaknya gedung dan ruangan yang ada pada Universitas Universitas Haluoleo, mengakibatkan pengunjung kesulitan menemukan gedung dan ruangan yang dicari. Oleh karena itu dibutuhkan sistem yang dapat menunjukkan lokasi gedung dan ruangan beserta jalur terpendeknya, agar waktu pencarian lebih efisien. Penelitian [10] menggunakan algoritma Dijkstra pencarian rute gedung dan ruangan pada fakultas di Universitas Haluoleo.

Penelitian dengan judul "Perbandingan Algoritma Branch And Bound Dan Algoritma Genetika Untuk Mengatasi Travelling Salesman Problem (Tsp) (Studi Kasus Pt. Jne Semarang)" [11] mengkaji sebuah permasalahan pencarian solusi optimum untuk masalah Travelling Salesman Problem (TSP). Sedangkan [12] menerapkan algoritma Branch And Bound dalam menentukan jalur terpendek untuk melakukan pencarian penginapan dan hotel di Kota Kendari.

Dari beberapa penelitian sebelumnya yang telah dilakukan metode A* dapat digunakan sebagai algoritma pencarian dalam penelitian ini untuk mencari lokasi halte bus, karena dari beberapa penelitian menunjukkan akurasi yang hasil yang bagus untuk pencarian jarak terpendek, adapun metode yang telah dibandingkan dari beberapa jurnal yang dibaca oleh penulis yaitu metode *Depth First Search* dan *Best First Search*. Menunjukkan A* memiliki akurasi yang tinggi untuk pencarian jarak terpendek.

3. Metode Penelitian

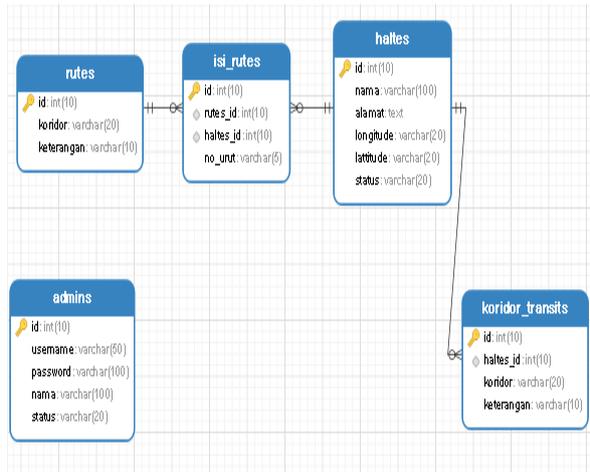
Beberapa langkah yang harus dilakukan dalam pengembangan sistem :

3.1 Analysis System

Analisa sistem merupakan hal yang terpenting dalam pembuatan sistem, tanpa adanya data yang benar maka sistem juga tidak akan menghasilkan solusi yang baik untuk permasalahan yang ada. Analisis sistem dilakukan dengan studi literatur, observasi dan wawancara.

3.2 Design System

Pada tahap ini merupakan tahap implementasi design yang meliputi interface maupun design model yang akan di implementasikan. Beberapa desain sistem yang dibuat, meliputi desain login, desain halaman halte admin, desain rute admin, desain *database*



Gambar 1. desain *database*

3.3 Development System

Dalam tahap ini merupakan tahap *development* atau merubah perilaku atau perhitungan A* menjadi sebuah kode pemrograman yang nantinya akan menjadi sebuah sistem. Di tahap ini kode pemrograman dibuat dengan android dan juga PHP. karena user admin akan melakukan input data halte dan rute melalui laptop / pc. Sedangkan user pengguna akan mengakses melalui *android* atau *smartphone*.

Tahap ini pembuatan kode pemrograman PHP dibuat menggunakan Framework CAKEPHP. Dan berikut adalah kode perhitungan A* untuk hasil development yang telah dibuat.

```

function garis_lurus($lat1, $lon1, $lat2, $lon2, $unit)
{
    $lat = $lat1 - $lat2;
    $lon = $lon1 - $lon2;
    $la = pow($lat, 2);
    $lo = pow($lon, 2);
    $to = $la + $lo;
    $total = sqrt($to);
    $ha = round($total,3);
    // var_dump($ha);
    // if ($ha >= 1) {
        $hasil = round(($ha * 111.32),2);
    }
    return $hasil;
}

function jarak_sebenarnya($lat1, $lon1, $lat2, $lon2)
{

```

```

$sp1 = "origins=".$lat1.", ".$lon1;
$sp2 = "&destinations=".$lat2.", ".$lon2;
$parameter = $sp1.$sp2;

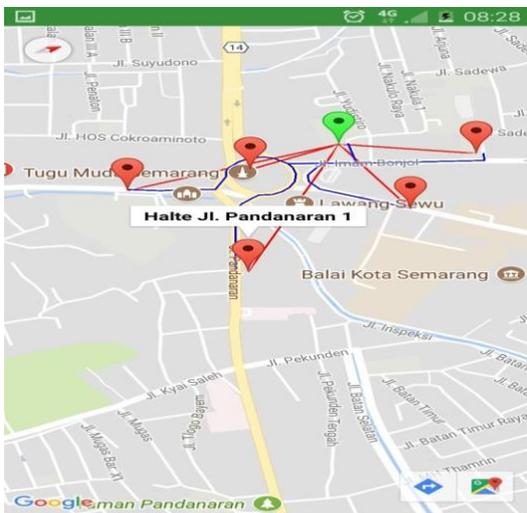
$url="https://maps.googleapis.com/maps/api/distance
matrix/json?units=imperial&".$parameter;

$ch = curl_init();
curl_setopt($ch,CURLOPT_SSL_VERIFYPEER,false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_URL, $url);
$result = curl_exec($ch);
curl_close($ch);

$result_array = json_decode($result);
$hasil = $result_array->rows[0]->elements[0]->distance-
>value / 1000;
//print_r();

// return $hasil;
return $hasil;
// echo $result;
}

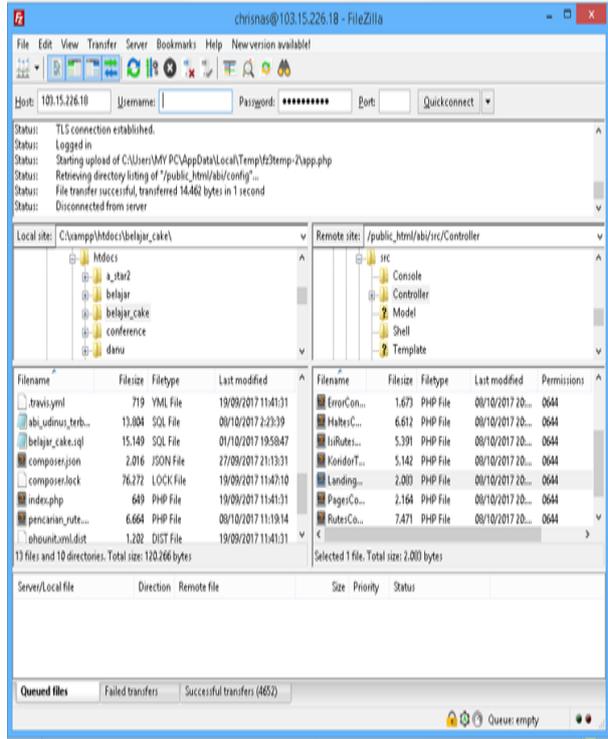
function A_star($lat1, $lon1, $lat2, $lon2, $unit)
{
    $x = garis_lurus($lat1, $lon1, $lat2, $lon2, $unit);
    $y = jarak_sebenarnya($lat1, $lon1, $lat2, $lon2);
    return $x + $y;
    # code...
}
    
```



Gambar 2. Perhitungan A* pencarian titik terdekat

3.4 Implementation System

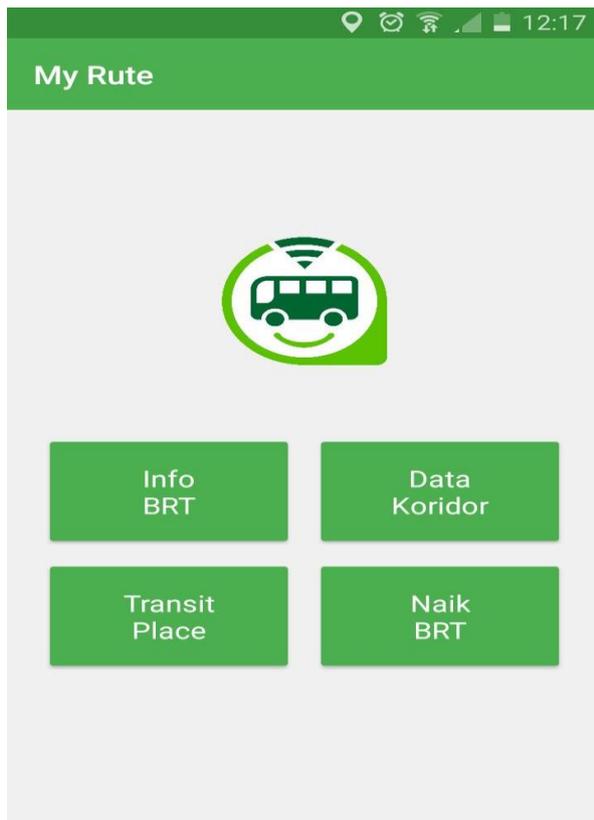
Tahap ini adalah tahap dimana aplikasi sudah jadi dan telah siap untuk dilakukan deployment. Juga merupakan tahap dimana penulis melakukan setting dan pemasangan aplikasi yang telah jadi. Lihat Gambar 3 – 8.



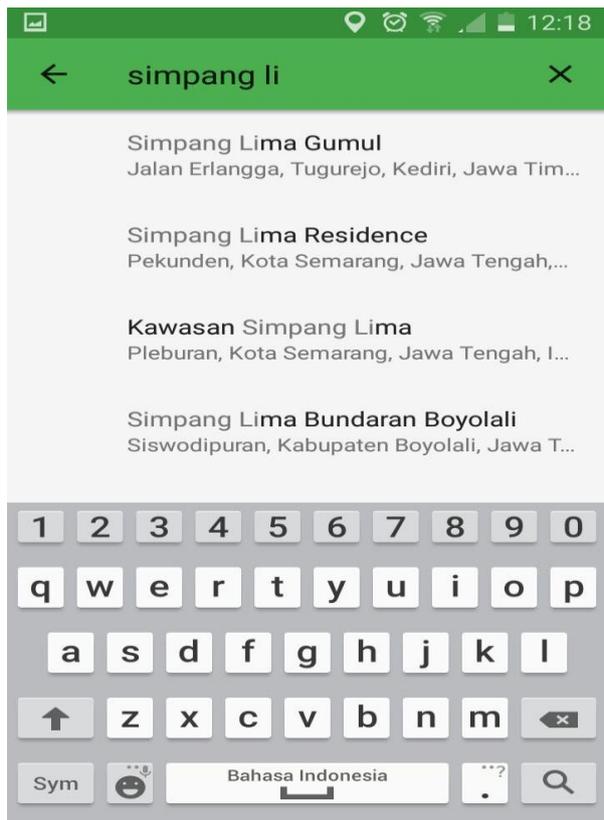
Gambar 3. Upload file ke server



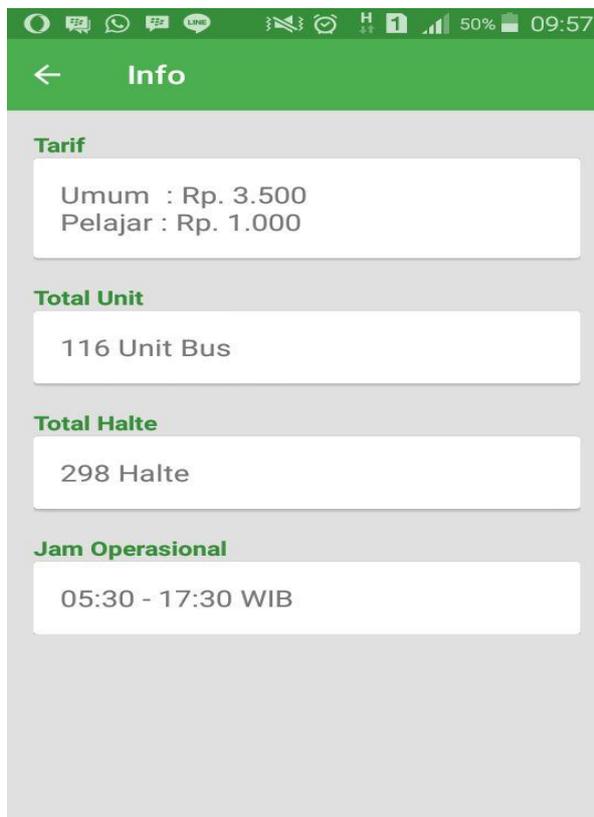
Gambar 4. Halaman Transit



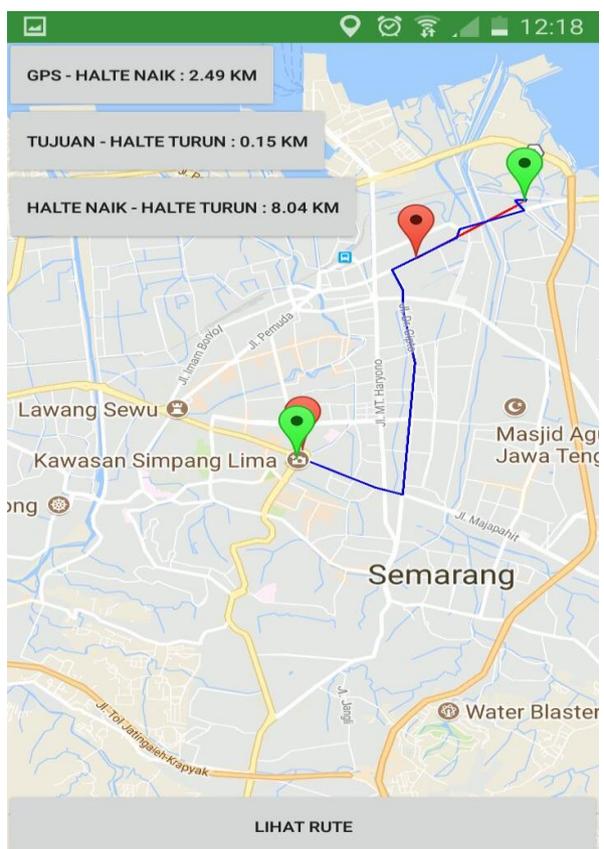
Gambar 5. Halaman Utama Aplikasi



Gambar 7. Pencarian Rute Turun



Gambar 6. Halaman Info



Gambar 8. Hasil Pencarian Titik Naik dan Turun Terdekat

3.5 Evaluation System

Pada tahap ini adalah tahap dilakukan evaluasi terhadap hasil penelitian yang telah dilakukan dan sistem yang telah dibuat. Lihat Tabel 1.

Tabel 1. Tabel pengujian A – Star

Kasus dan Hasil Pengujian				
No	Data Input	Target	Hasil Pengamatan	Kesimpulan
1	Input data gps	Menampilkan halte naik terdekat	Menampilkan halte naik terdekat	Berhasil
2	Mencari lokasi turun	Menampilkan halte turun terdekat	Menampilkan halte turun terdekat	Berhasil

4. Hasil dan Pembahasan

Implementasi pencarian halte naik dan turun terdekat menggunakan metode A*. Metode perhitungan A* dapat diperoleh dengan rumus :

$$f(n) = g(n) + h(n) \tag{1}$$

Dimana g adalah jarak garis lurus antara satu titik dengan titik lain. Dan h adalah jarak heuristic satu titik dengan titik lainnya. Dari hasil yang didapat maka dicari titik terendah dari titik awal yang nantinya akan menjadi titik terdekat.

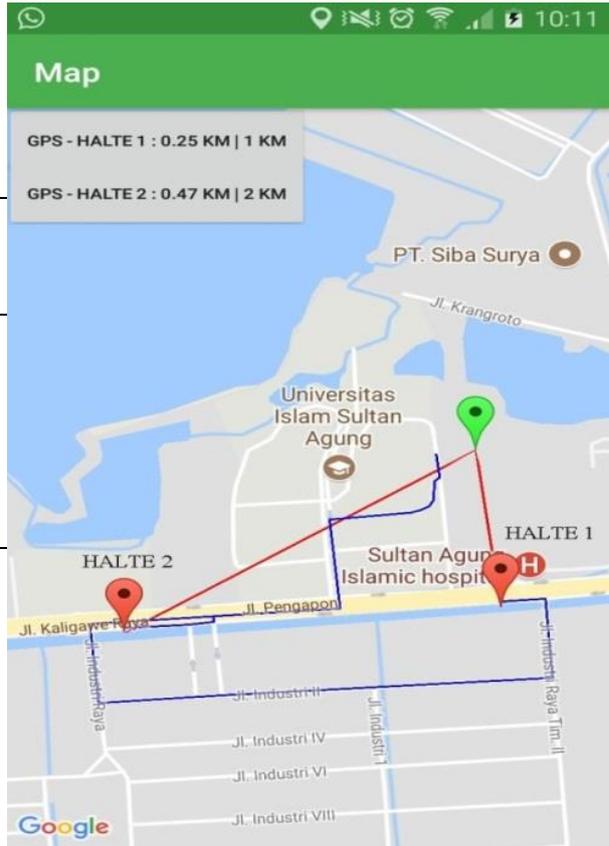
1. Implementasi Pencarian titik terdekat menggunakan A*

Dalam percobaan ini mencari data halte melalui tabel yang ada dari *database*. Diambil 2 sampel data yaitu halte 1 dan halte 2. Lalu metode A* akan memilih satu halte sebagai titik terdekat. Lihat Gambar 9.

Pada Gambar 9 titik hijau menjelaskan titik pengguna atau user dengan GPS yang menyala. Lalu data untuk halte satu nilai $g(n) = 0.25\text{KM}$ dan nilai $h(n) = 1\text{ KM}$. Sedangkan untuk halte dua memiliki nilai $g(n) = 0.47\text{KM}$ dan nilai $h(n) = 2\text{KM}$. Jika dihitung dengan menggunakan metode A*. Maka titik halte terdekat yang akan dipilih adalah halte satu.

2. Hasil Pencarian

Pada hasil pencarian di Gambar 10 di dapatkan hasil titik halte terdekat di halte satu dengan menggunakan metode A*. Dalam penelitian ini hasil nilai $g(n)$ diperoleh dari rumus 2.



Gambar 9. Hasil pencarian metode A*



Gambar 10. Hasil Pencarian

$Jarak = Jarak\ pada\ peta \times skala\ peta$ (2) 5.2 Saran

Sedangkan hasil nilai $h(n)$ diperoleh dari rumus:

$$jarak = \sqrt{(lattitude1 - lattitude2)^2 + (longitude1 - longitude2)^2} \quad (3)$$

Berikut adalah paparan evaluasi setelah hasil implementasi dilakukan oleh penulis.

1. Program yang dibuat telah berhasil lulus pengujian *blackbox* dan sudah sesuai dengan target yang diharapkan
2. Program yang dibuat telah sesuai mencari titik terdekat menggunakan pencarian A*.
3. Metode A* dapat digunakan untuk mencari jarak terdekat naik dan turun halte bus rapid transit.
4. Apabila terdapat titik koordinat yang tidak sesuai maka hasil perhitungan jarak garis lurus dan jarak asli akan berbeda, maka berubah juga hasil yang akan di dapat.

5. Kesimpulan

5.1 Simpulan

Simpulan yang dapat diambil dari aplikasi sistem pencarian halte BRT terdekat kota Semarang menggunakan metode A* berbasis android adalah :

1. Dalam penelitian ini dapat menentukan titik terdekat naik dan turun pada tujuan tertentu untuk membantu penumpang atau pengguna *Bus Rapid Transit*.
2. Aplikasi yang dibuat memberikan informasi halte terdekat dengan menggunakan metode A*.
3. Dari perhitungan yang dilakukan menggunakan metode A* dan dilakukan 20 percobaan pada lokasi yang berbeda, metode A* memiliki akurasi 100% dalam menentukan halte naik dan turun terdekat, apabila dalam keadaan *Global Positioning System* yang akurat.
4. Penggunaan *Smartphone* sangat menentukan titik halte terdekat, apabila *Global Positioning System* memberikan titik akurat maka halte terdekat yang dihasilkan akan akurat juga begitupun sebaliknya.

1. Dalam penelitian selanjutnya dapat dilakukan perbandingan metode lain selain A* jika memungkinkan di dapat akurasi yang lebih baik.
2. Metode A* dapat digunakan dalam penentuan rute terpendek untuk kasus yang lain.

6. Daftar Rujukan

- [1] K. Panji Wisnu Wirawan, Djalal Er Riyanto, 2016. "Pemodelan Graph Database Untuk Moda Transportasi Bus Rapid Transit", *Jurnal Informatika*, vol. 10, no. 2, pp. 1233–1243.
- [2] Firman Harianja, 2013. "Penerapan Algoritma a * Pada Permasalahan Optimalisasi Pencarian Solusi Dynamic Water Jug", *Jurnal Pelita Informatika Budi Darma*, Vol IV, Nomor 3, Agustus 2013, pp. 48–53.
- [3] M. Irsyad and E. Rasila, 2015. "Aplikasi Pencarian Lokasi Gedung dan Ruangan Universitas Islam Negeri Sultan Syarif Kasim Riau pada Platform Android Menggunakan Algoritma A-Star (A*)", *Jurnal CoreIT*, vol. 1, no. 2, Desember 2015, pp. 90–95.
- [4] A. Ayu, Y. Saputra, and A. Rahman, 2016. "Penerapan Algoritma A Star Dalam Pencarian Mesin ATM Terdekat di Palembang Berbasis Android", *Jurnal STMIK GI MDP*.
- [5] W. Mutia Purwati, Okti Firnawati, 2015. "Penerapan algoritma a* (a star) dalam optimasi penentuan halte transmisi di Palembang berbasis android, *Jurnal STMIK GI MDP*.
- [6] Badzrotul Mufida, 2016. "Implementasi Metode A* (A-Star) Untuk Npc Musuh Pada Game 3d Pembelajaran Kosakata Bahasa Arab", Skripsi, Jurusan Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, Malang.
- [7] A. Pamungkas, E. P. Widiyanto, and R. Angreni, 2014. "Penerapan Algoritma A* (A Star) Pada Game Edukasi The Maze Island Berbasis Android", *Jurnal STMIK GI MDP*.
- [8] Ifa Alif, 2015. "3d Wayang Adventure Game Untuk Pengenalan Budaya Wayang Nusantara Menggunakan A* Pathfinding Algorithm Sebagai Pembangkit Perilaku Pencarian Pada Npc", Skripsi, Jurusan Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, Malang.
- [9] Y. Syukriah, Falahah, and H. Solihin, 2016. "Penerapan algoritma a* (star) untuk mencari rute tercepat dengan hambatan", *Seminar Nasional Telekomunikasi dan Informasi.*, no. Selisik, pp. 219–224.
- [10] M. Y. Amden Junianto Jalu Marseno, 2015. "Panduan Pencarian Rute Gedung Dan Ruangan Pada Fakultas Di Universitas Halu Oleo Menggunakan Algoritma Dijkstra Berbasis Macromedia Flash", *Seminar Nasional semanTIK*, vol. 1, no. 2, pp. 45–56.
- [11] R. A. Ari Yulianto Nugroho, Amin Suyitno, 2016. "Perbandingan Algoritma Branch And Bound Dan Algoritma Genetika Untuk Mengatasi Travelling Salesman Problem (Tsp) (Studi Kasus Pt. Jne Semarang)", *Jurnal Math UNNES.*, vol. 3, no. 1, pp. 3–8.
- [12] L. B. A. Restu Hadi Saputra, Jumadil Nangi, 2017. "Penerapan Algoritma Branch And Bound Dalam Menentukan Jalur Terpendek Untuk Melakukan Pencarian Penginapan Dan Hotel Di Kota Kendari", *Seminar Nasional semanTIK*, vol. 3, no. 1, pp. 2–5.