

repository.ub.ac.id

PENGEMBANGAN MODUL *ROUTING* PADA PETA *OFFLINE OPEN STREET MAP* PADA APLIKASI ANDROID MENGGUNAKAN METODE ALGORITMA *A*(STAR)*

Aditya Bagus Setiawan¹⁾, Dr.Eng, Herman Tolle, S.T.,M.T²⁾,
Drs, Marji, M.T³⁾

Program Studi Informatika/ Ilmu Komputer
Program Teknologi Informasi dan Ilmu Komputer
Universitas Brawijaya, Malang 65145, Indonesia
email: adityabagussetiawan[at]outlook.co.id¹⁾, emang[at]ub.ac.id²⁾,
marji[at]ub.ac.id³⁾

ABSTRAK

Pada jaman modern saat ini penggunaan perangkat bergerak tidak terbatas hanya pada short message service (SMS) atau telephone saja namun pengguna perangkat bergerak dapat melakukan *explore* lebih jauh lagi pada perangkat bergerak yang dimilikinya salah satunya adalah penentuan posisi atau yang sering kita sebut sebagai *routing*. Fasilitas ini dirasa perlu ketika ingin menemukan jalur menuju tempat yang dituju dengan cepat dengan menggunakan perangkat bergerak dibandingkan dengan membaca peta.

Aplikasi yang dibangun berbasis *mobile* dengan sistem operasi yang digunakan adalah Android serta pengembangannya menggunakan Bahasa pemrograman Java. Semuanya diperoleh hasil sebuah aplikasi yang dapat menampilkan rute terpendek menuju tempat yang ingin dituju dengan menggunakan peta *offline Open Street Map* untuk wilayah Kota Malang.

Pada penelitian ini menggunakan Algoritma *A*(star)* untuk menentukan rute terpendek menuju tempat tujuan. Algoritma *A*(star)* dipilih karena dianggap mampu untuk memecahkan masalah perhitungan terpendek dari sebuah rute. Berdasar pengujian yang telah dilakukan Algoritma *A*(star)* akurasi mencapai 80% sehingga ini dapat digunakan sebagai acuan yang cukup akurat.

Kata kunci: Algoritma *A*(star)*, *Open Street Map*, Perangkat bergerak, rute terpendek.

ABSTRACT

In today's modern era use of mobile devices is not limited to the short message service (SMS) or telephone only but mobile users can explore further on its mobile devices one of which is the positioning or what we often refer to as routing. This facility is necessary when you want to find a path to the destination quickly by using mobile devices than reading a map.

Applications built based mobile operating system used is Android and its development using the Java programming language. All obtained results of an application that can display the shortest route to where you want to target by using offline maps Open Street Map for the city of Malang.

*In this study, using Algorithm A *(star) to determine the shortest route towards a destination place. Algorithm A *(star) were selected because they are able to issue memecahkan shortest calculation of a route. Based on the testing that was done Algorithm A *(star) reached 80% accuracy so that it could be used as a fairly accurate.*

Keywords: Algorithm A *(star), *Open Street Map*, Mobile devices, the shortest route.

1. PENDAHULUAN

Perangkat bergerak atau sering kita sebut dengan *smartphone* sudah sangat pesat perkembangannya tidak hanya digunakan sebagai alat komunikasi semata namun penggunaannya sudah kini sudah sepadan dengan sebuah PC ataupun laptop. Tidak semua

kalangan masyarakat dapat mengoperasikan atau bahkan memanfaatkan kecanggihan fasilitas perangkat bergerak yang dimiliki, salah satunya adalah memanfaatkan fasilitas *routing* yang dapat dilakukan oleh perangkat bergerak guna menuntun pengguna untuk menuju suatu tempat yang dituju dengan perhitungan rute yang akan ditempuh adalah rute yang tercepat.

Tujuan penelitian ini adalah untuk memudahkan para pengguna perangkat bergerak untuk menuju suatu tempat yang diinginkan pada suatu kota atau wilayah tertentu. Aplikasi yang dibangun merupakan aplikasi yang berbasis *mobile* dengan menggunakan sistem operasi Android.

Aplikasi ini berjalan dengan cara pemilihan rute tercepat atau terpendek dengan menerapkan Algoritma $A^*(star)$ pada peta *offline open street map*, Algoritma ini bertugas sabagai mesin penghitung atau kalkulasi dari *input* yang diberikan oleh pengguna perangkat bergerak sehingga menghasilkan *output* berupa rute perjalanan menuju tempat yang dituju dengan melalui jalur tercepat.

Algoritma $A^*(star)$ pertama kali dikenalkan pada tahun 1968 oleh Peter Hart, Nils Nilson dan Bertram Raphael (Heart et al., 1968). Dalam ilmu komputer Algoritma $A^*(star)$ (sering diucapkan dengan *A star*) merupakan salah satu Algoritma pencarian *graph* terbaik yang mampu menemukan jalur dengan biaya pengeluatau atau *cost* paling sedikit dari titik permulaan yang diberikan hingga titik tujuan yang diharapkan (Pearl,1948). Dipilihnya Algoritma $A^*(star)$ karena Algoritma ini dapat menemukan jalur tercepat termasuk dalam menentukan arah tujuan suatu tempat yang berada pada suatu daerah atau kota tertentu.

Selain Algoritma $A^*(star)$ terdapat Algoritma lain yaitu Algoritma Genetika yang fungsinya sama-sama dapat digunakan sebagai pemecahan masalah yang erat kaitannya dengan pencarian rute terpendek, jika dibandingkan dengan Algoritma Genetika maka Algoritma $A^*(star)$ yang lebih unggul dikarenakan Algoritma $A^*(star)$ tidak memerlukan persamaan matematika yang cukup banyak. Algoritma $A^*(star)$ dalam operasinya hanya menggunakan beberapa persamaan diantaranya fungsi *distance - plus - cost* (biasanya dinotasikan dengan $f(x)$) untuk menentukan urutan kunjungan pencarian node dalam *tree*. Gabungan jarak - plus -biaya yang merupakan penjumlahan dari dua fungsi yaitu fungsi *path - cost* (selalu dinotasikan dengan $g(x)$) yang dimungkinkan dengan nilai *heuristic* maupun

tidak), dan sebuah kemungkinan penerimaan atas perkiraan heuristik jarak titik ke tujuan (sinotasikan dengan $h(x)$). Fungsi *path - cost* $g(x)$ adalah jumlah biaya yang harus dikeluarkan dari node awal menuju node yang akan dituju. Dengan $h(x)$ bagian dari fungsi $f(x)$ yang harus dapat heuristik, yang mana tidak diperbolehkan untuk terlalu jauh memperkirakan jarak ke arah tujuan. Oleh karena itu aplikasi seperti *routing*, $h(x)$ mungkin mewakili garis lurus jarak ke titik tujuan. (Setiawan, 2015).

Dari uraian tersebut dirasa perlu dibuat sebuah aplikasi yang bertujuan bisa membantu pengguna perangkat bergerak untuk menuju suatu tempat dengan jalur yang tercepat dengan menerapkan Algoritma $A^*(star)$ dalam perhitungan rutenya. Dalam hal ini pemilihan perangkat bergerak sebagai medianya karena dirasa perangkat bergerak sebagai salah satu kebutuhan yang vital sebagai alat komunikasi selain itu dari segi desain dan ukuran yang menarik sehingga tidak terlalu sulit untuk dibawa kemanapun.

2. RUMUSAN MASALAH

Berdasar uraian pendahuluan diatas maka pada penelitian didapat beberapa rumuasan permasalahan diantaranya adalah

1. Bagaimana penerapan Algoritma $A^*(star)$ pada modul routing peta offline
2. Bagaimana rancangan dan implementasi modul routing dengan menggunakan metode $A^*(star)$ pada peta offline open street map pada aplikasi LBS Android
3. Bagaimana tingkat akurasi hasil implementasi metode Algoritma $A^*(star)$

3. LANDASAN KEPUSTAKAAN

3.1 Kajian pustaka

Sebelum melakukan penelitian ini dilakukan studi terhadap penelitian yang berkaitan dengan pencarian rute terpendek terhadap beberapa jurnal dari beberapa peneliti yang meneliti hal hampir serupa. Yang pertama jurnal yang berjudul Rancang Bangun Aplikasi Pencarian Rute Terpendek Untuk Menemukan SPBU Terdekat Di Kota Malang (Vika,2010), dalam jurnal ini penulis menggunakan Algoritma

Genetika sebagai penentu jalur tercepat namun penerapan ini dirasa masih kurang dikarenakan perhitungan yang sangat rumit dan panjang selain itu penggunaan Algoritma ini juga tidak dapat digunakan untuk *routing* secara offline.

Penelitian kedua adalah jurnal yang berjudul Perancangan dan Implementasi Algoritma A*(star) pada Aplikasi Angkot-Finder di Kota Bandung Untuk Smartphone Berbasis Android(Didik,2015), jurnal ini berisikan tentang jalur angkot dari suatu tempat menuju tempat lain dengan rute yang terpendek menurut perhitungan dari Algoritma A*(star) namun pada penelitian ini ruang lingkup yang digunakan hanya terbatas pada jalur-jalur angkot saja dan penelitian ini bersifat online.

Penelitian yang terakhir adalah jurnal yang berjudul Perancangan Aplikasi Dalam Implementasi *Shortest Path Finder* Di Kawasan Kampus Telkom University Berbasis Android dengan Menggunakan Algoritma A*(star)(Achmad,2013), jurnal ini berisikan tentang pencarian suatu tempat di wilayah *intern* kampus Telkom University dengan perhitungan Algoritma A*(star) sehingga menghasilkan rute atau jalan tercepat menuju tempat yang dituju. Ruang lingkup penelitian ini hanya terbatas pada kawasan kampus saja dan bersifat *online*.

3.2 Dasar Teori

Algoritma A*(star)

Algoritma ini dikenalkan pertama kali pada tahun 1968 oleh Peter Hart, Nils Nilson dan Bertram Raphael (Hart et al.,1968). Dalam sebuah ilmu komputer Algoritma A* (sering diucapkan A star) adalah salah satu algoritma pencarian *graph* yang terbaik dikarenakan algoritma ini memiliki keluaran atau output berupa jalur yang memiliki biaya pengeluaran paling sedikit dari titik awal yang diberikan hingga titik tujuan yang diharapkan (Pearl,1984).

Secara keseluruhan algoritma ini memiliki fungsi *distance* – plus – *cost* (yang dinotasikan dengan $f(x)$) yang berguna untuk menentukan urutan kunjungan pencarian ke setiap node di dalam sebuah *graph* jalan atau *tree*. Gabungan jarak – plus – biaya merupakan sebuah penjumlahan dari dua fungsi di antaranya fungsi *path* – *cost* (yang dinotasikan dengan $g(x)$), sehingga dimungkinkan bernilai heuristik atau tidak), dan sebuah kemungkinan perkiraan atas heuristik jarak menuju tujuan (yang dinotasikan dengan $h(x)$).

Sedangkan fungsi *path* – *cost* $g(x)$ merupakan jumlah biaya yang harus dikeluarkan mulai dari titik awal hingga titik tujuan, dengan $h(x)$ yang merupakan bagian dari fungsi $f(x)$ yang harus heuristik dan tidak diperbolehkan untuk terlalu jauh memperkirakan jarak menuju arah tujuan. Maka dari itu aplikasi seperti *routing*, $h(x)$ mungkin mewakili garis lurus jarak menuju titik tujuan(Setiawan,2015).

Terdapat beberapa terminologi pada Algoritma A*(star) diantaranya adalah : (seriawan, 2015)

1. Simpul (node) merupakan petak-petak kecil sebagai representasi dari area pencarian (*pathfinding*).
2. Sumpul asal/awal (*source node*) merupakan sebuah terminologi untuk posisi awal sebuah benda.
3. Simpul ahir/tujuan (*destination node*) merupakan tempat tujuan yang ingin dituju pada pencarian.
4. Simpul sekarang (*current node*) merupakan simpul terbaik sebelumnya yang dipilih dan menjadi titik awal acuan guna membangkitkan simpul tetangga (*adjacement*).
5. Simpul tetangga (*neighbor node*) merupakan simpul-simpul yang saling bertetangga dengan *current node*.
6. *Openlist* merupakan tempat menyimpan data simpul yang akan diakses dari simpul asal (*source node*) atau dari tetangga simpul sekarang (*current node*) yang belum pernah berada di *open list* atau *closed list*.
7. *Closed list* merupakan tempat untuk menyimpan simpul yang pernah menjadi simpul sekarang (*current node*).
8. *Came from* merupakan tempat yang menyimpan data tetangga dari suatu simpul, contoh *y came from x* yang artinya *neighbor node y* dari *current node x*.
9. *Walk ability* merupakan sebuah atribut yang menyatakan apakah sebuah simpul dapat atau tidak dilalui oleh *current node*.

Secara keseluruhan didapat sebuah persamaan dari Algoritma A*(star) sebagai berikut

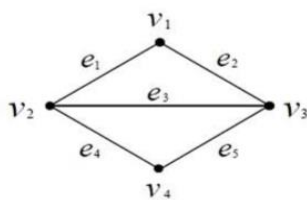
$$f(x) = g(x) + h(x) \quad (1)$$

pada persamaan (1) dijelaskan bahwa $g(x)$ merupakan biaya yang sebenarnya (*actual cost*), biasanya berupa jarak yang sudah pernah ditempuh tapi dalam sebuah kasus tertentu $g(x)$ dapat berupa penjumlahan antara jarak dengan biaya medan (*terrain cost*). Sedangkan $h(x)$ merupakan biaya heuristiknya (*heuristic cost*) berupa perkiraan jarak dari simpul sekarang (*current node*) menuju simpul tujuan (*destination node*). Dan $f(x)$ merupakan total biaya dari sebenarnya dan biaya heuristik.

Pada persamaan (1) diatas algoritma ini menggunakan fungsi biaya heuristik simpul x untuk menentukan urutan yang akan dilintasi dalam prioritas antrian (S.Rabin, 2000). Fungsi dari biaya dapat dibagi menjadi dua bagian yaitu fungsi biaya jalan dan fungsi biaya masa depan (S.Rabin, 2000).

Graf

Graf merupakan sekumpulan dari simpul dan garis dimana seluruh pasangan simpul tersebut dihubungkan oleh segmen garis, simpul ini bisa disebut *vertex* dan segmen garis disebut *edge* (Setiawan, 2015). Graf bisa digunakan untuk menampilkan objek-objek diskrit dan hubungan antara semua objek tersebut. Graf seringkali digunakan untuk menampilkan suatu jaringan misalnya jaringan jalan raya yang dimodelkan graft dengan kota sebagai simpul (*vertex*) dan jalan yang menghubungkan setiap kota sebagai sisi (*edge*) yang bobotnya (*weight*) adalah panjang jalan tersebut.



Gambar 1 Graf sederhana

Sumber : Setiawan (2014)

Pada Gambar 1 diatas merupakan sebuah gambar graf sederhana yang memiliki dua komponen dasar yaitu *vertex* dan *edge*. Sebuah

graf $G = (v, e)$. terdapat beberapa istilah yang berkaitan dengan graf diantaranya :

1. *Vertex* merupakan himpunan simpul/titik pada sebuah graf.
2. *Edge* merupakan himpunan garis yang menghubungkan tiap simpul/*vertex*.
3. *Adjacement* merupakan sebuah titik yang saling berdekatan, dua buah titik dikatakan berdekatan jika dua buah titik tersebut terhubung dengan sebuah sisi.
4. *Path* merupakan *walk* dengan setiap *vertex* yang berbeda. Sebuah *walk* didefinisikan sebagai urutan *vertex* dan *edge*.
5. *Cycle* merupakan siklus atau sirkuit lintasan yang berawal dan berakhir pada simpul yang sama.

Shortest path

Pada jurnal Achmad Mirizki Chilmi disebutkan bahwa lintasan terpendek (*shortest path*) adalah sebuah lintasan minimum yang diperlukan untuk mencapai suatu titik tujuan. Dalam pencarian lintasan terpendek masalah yang sering dihadapi adalah mencari lintasan mana yang akan dilalui sehingga didapat lintasan terpendek dari sebuah *vertex* yang lain. Terdapat beberapa macam persoalan lintasan terpendek diantaranya (Achmad, 2013) :

1. Lintasan terpendek antara dua buah *vertex*.
2. Lintasan terpdendek antara semua pasangan *vertex*.
3. Lintasan terpendek dari *vertex* tertentu ke semua *vertex* yang lain.
4. Lintasan terpendek antara dua buah *vertex* yang melalui beberapa *vertex* tertentu.

SQLite

SQLite adalah sebuah proses *library* yang digunakan untuk mengimplementasikan penyimpanan mandiri, *serverless*, tidak ada konfigurasi, mesin database SQL transaksional. Kode program untuk SQLite dalam domain publik dan dengan demikian bebas digunakan untuk tujuan apapun baik komersil atau pribadi. SQLite sekarang ini termasuk yang banyak ditemukan dalam sebuah proyek kecil maupun

proyek besar, tidak seperti kebanyakan database SQL lainnya. SQLite memiliki server terpisah, SQLite dapat membaca dan menulis langsung pada *file disk* biasa. Sebuah SQL database lengkap dengan beberapa *table*, *index*, *trigger*, dan *view* yang terkandung dalam sebuah *file disk* tunggal (Alfan, 2013).

Secara keseluruhan sebuah SQLite memiliki delapan fitur diantaranya (Alfan, 2013) :

1. *Serverless*, SQLite tidak memerlukan proses pada *server* atau sistem untuk menjalankannya melainkan hanya sebuah file yang diakses oleh *library* SQLite.
2. *Zero configuration* yang berarti tidak ada *server* yang melakukan *setup*.
3. *Cross platform* adalah semua instan database berada dalam sebuah file yang *cross-platformnya* tidak memerlukan administrasi.
4. *Self-contained* sebuah *library* mengandung keseluruhan dari sistem database yang langsung terintegrasi pada sebuah aplikasi program.
5. *Small runtime footprint* yang berguna untuk membangun database SQLite yang hanya membutuhkan kurang dari satu *megabyte library* (kode program) dan hanya membutuhkan beberapa *megabyte memory* saja.
6. *Transactional*, SQLite *transactional* akan memperbolehkan aksi penyimpanan melalui beberapa proses *thread*.
7. *Full featured* yang merupakan dukungan SQLite pada hampir seluruh bagian besar standar SQL92 (SQL2).
8. *Highly reliable* yang merupakan pengembang SQLite melalui kode program yang melewati proses *testing*.

Location Based Service (LBS)

Location bases service merupakan sebuah layanan informasi yang bisa diakses dengan menggunakan perangkat bergerak melalui sebuah jaringan dan mampu menampilkan posisi secara geografis dimana keberadaan sebuah perangkat bergerak tersebut (William, 2013).

Terdapat dua unsur utama pada LBS diantaranya adalah (William, 2013) :

1. *Location manager* yang merupakan penyedia perangkat bagi sumber atau *source* untuk *location bases service*.

2. *Location provider*, menyediakan teknologi pencarian lokasi yang digunakan oleh sebuah perangkat.

Open Street Map (OSM)

Open street map (OSM) merupakan sebuah alat untuk membuat dan berbagi informasi dalam bentuk peta. Siapapun dapat berkontribusi untuk mengembangkan OSM dan ribuan orang bebas menambahkan sebuah proyek setiap harinya. Pengguna OSM dapat memanfaatkan segala fasilitas khususnya dalam hal menambahkan detail informasi ke dalam peta. OSM bersifat digital sehingga membuatnya sangat berguna untuk pengembangan aplikasi LBS (HOT, 2012).

Mobile atlas creator (MOBAC)

MOBAC merupakan sebuah aplikasi yang bersifat open source sehingga dapat membuat peta secara *offline* untuk dipergunakan dan diterapkan pada sebuah aplikasi *mobile* sebagai penentu posisi. Dengan menggunakan MOBAC sebuah peta dapat disimpan sebagai satuan gambar PNG, file zip, SQLite dll. Sebagai sumbernya MOBAC menggunakan sebagian besar peta *online* yang *open source* seperti *Open Street Map* dan penyedia peta *online* lainnya (Mobac, 2008).

4. METODE PENELITIAN

Pada tahap ini akan dibahas mengenai langkah yang akan digunakan untuk membangun aplikasi ini. Pembangunan aplikasi ini terdiri dari beberapa tahap diantaranya adalah studi literatur, identifikasi, analisa data, pengembangan modul, implementasi, pengujian dan evaluasi. Seluruh tahapan tersebut dirangkai menjadi sebuah diagram alir dan dapat dilihat pada Gambar 2.



Gambar 2 Diagram alir metode penelitian

Berdasarkan diagram alir diatas tahapan yang akan dilakukan guna membangun aplikasi ini adalah:

1. Studi literatur, dalam tahap ini dilakukan untuk memperdalam dan memahami tentang teori kan konsep yang akan digunakan untuk membanun aplikasi “Pengembangan Modul Routing Peta Open Street Map pada Aplikasi Android Menggunakan Metode Algoritma A*(star)”. Beberapa metode yang dipelajari adalah :
 - a. Algoritma A*(star).
 - b. Pemrograman Android.
 - c. SQLite.
 - d. Peta *offline open street map* (OSM).
 - e. *Location Bases Service* (LBS).
2. Identifikasi, pada tahap ini digunakan untuk mengkaji dan memberikan batasan masalah yang akan dibahas sekaligus yang akan diimplementasikan dalam aplikasi yang akan dibangun, pada tahap ini pula akan dilakukan pengumpulan data yang berkaitan dengan konsep aplikasi yang akan dibangun. Dalam fase identifikasi ini pula akan menentukan keberhasilan dari setiap tahapan-tahapan berikutnya.
3. Analisis data, pada tahap ketiga ini merupakan tahapan dimana dilakukan analisis terhadap seluruh data yang telah terkumpul. Setelah semua data dikumpulkan maka akan dilakukan pengujian menggunakan Algoritma A*(star) terhadap selua data yang telah terkumpul.
4. Pengembangan modul routing, tahap ini merupakan tahap dimana semua data yang diperlukan telah terkumpul dan dinyatakan dapat digunakan dalam pembangunan aplikasi ini. Pengembangan rouing ini dilakukan dengan menggunakan Algoritma A*(star) sebagai solusi pemecahan masalah dalam penentuan jalur tercepat menuju suatu tempat yang diinginkan
5. Implementasi, tahap yang kelima adalah implementasi diamana apabila semua data yang diperoleh telah diproses dan pemecahan terhadap masalah yang muncul telah didapat makan akan masuk pada tahap ini. Pada tahap ini aplikasi yang diimplementasikan berdasar pada tahap desain yang telah disetujui, pada tahap ini pula diberlakukan bebrapa hal diantaranya *input*, *proses* dan *output*

yang sesuai dengan masukan yang diberikan oleh pengguna dan yang telah diproses oleh aplikasi.

6. Pengujian dan evaluasi, tahap ini dilakukan jika semua tahapan yang lain telah dipenuhi dan menghasilkan sebuah aplikasi tertentu. Pengujian dan evaluasi ini bertujuan untuk mengetahui kesalahan dan kekurangan yang terdapat pada aplikasi yang telah dibangun. Pengujian yang dilakukan menggunakan *black box* testing guna memastikan bhwa aplikasi yang telah dibangun sudah memenuhi kebutuhan yang telah didefinisikan pada bab perancangan.

5. PERANCANGAN

5.1 Gambaran umum sitem

Gambaran umum sistem merupakan representasi desain arsitektur yang dibuat secara umum. Tahai ini merupakan tahap awal perancangan sistem yang akan dibangun. Gambaran umum sistem ditunjukkan pada Gambar 3 dibawah ini.



Gambar 3. Gambaran Umum Sistem

Pengguna masuk pada aplikasi, setelah masuk pengguna akan dihadapkan pada tiga menu utama aplikasi diantaranya menu mulai, menu tentang, dan menu *exit*. Setelah pengguna memilih menu mulai maka layar akan akan menampilkan peta wilayah kota Malang dan pengguna dapat langsung melakukan set posisi. Set posisi dilakukan secara manual dengan cara memilih marker jalan yang ada pada peta.

5.2 Identifikasi aktor

Aktor yang berinteraksi penuh dan secara langsung pada perangkat lunak adalah pengguna aplikasi. Pengguna aplikasi ini mempunyai hak penuh pada seluruh fitur yang ada pada aplikasi.

5.3 Kebutuhan fungsional

Kebutuhan fungsional terdiri dari fitur-fitur yang disediakan oleh aplikasi dan berguna untuk memenuhi kebutuhan pengguna. Kebutuhan fungsional ditunjukkan dengan menggunakan penomoran SRS(*Software Requirement*

Spesification) dan digambarkan dengan use case seperti pada Tabel 1.

Tabel 1. Spesifikasi kebutuhan fungsional

Nomor SRS	Kebutuhan	Use Case
SRS_001	Aplikasi harus dapat menampilkan peta wilayah Kota Malang agar pengguna dapat melakukan set awal posisi secara manual pada setiap jalan yang ada di Kota Malang.	Melakukan set posisi.
SRS_002	Aplikasi harus dapat menampilkan rute jalan tercepat menuju tempat yang telah ditentukan dengan menggunakan Algoritma A*(star).	Mencari rute terpendek.

Gambar 4 merupakan use case dari aplikasi yang dibangun

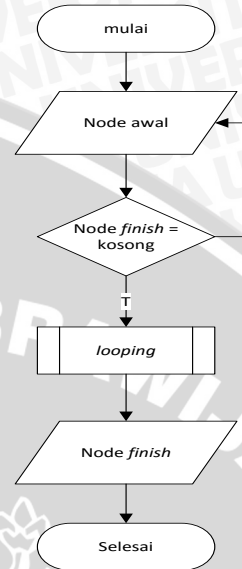


Gambar 4. Diagram Use Case

5.4 Perancangan Algoritma A*(star)

Algoritma A*(star) pada aplikasi ini adalah sebagai penentu rute terbaik dalam hal ini rute terpendek untuk menuju point atau node yang

dituju oleh pengguna. Proses ini dimulai ketika pengguna melakukan set posisi awal dan tujuan setelah itu Algoritma A*(star) akan melakukan perhitungan sesuai dengan tugasnya. Langkah-langkah Algoritma A*(star) dapat dilihat pada Gambar 5.

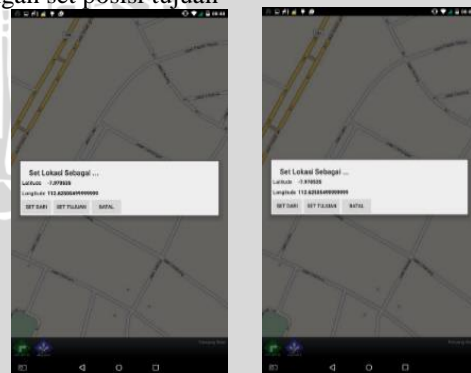


Gambar 5. Diagram alir Algoritma A*(star)

6. IMPLEMENTASI DAN PENGUJIAN

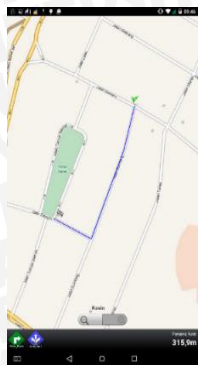
6.1 Implementasi

Gambar 6 merupakan implementasi dari set posisi awal dan set posisi tujuan yang dilakukan oleh pengguna aplikasi. Set posisi awal dilakukan dengan cara manual begitu juga dengan set posisi tujuan



Gambar 6. Halaman set posisi awal dan tujuan

Pada Gambar 7 ini merupakan hasil dari pencarian rute yang telah ditentukan oleh pengguna



Gambar 7. Implementasi halaman hasil

6.2 PENGUJIAN

Analisis hasil pengujian validasi fitur

Pengujian validasi dilakukan dengan memadupadankan kesesuaian hasil kerja sistem yang telah dibangun dengan seluruh daftar kebutuhan yang telah diidentifikasi pada bab perancangan. Berdasar hasil pengujian yang telah dilakukan dapat diketahui bahwa hasil implementasi sistem telah sesuai dengan semua fitur yang ada pada daftar kebutuhan .

Analisis hasil pengujian kompatibilitas

Pengujian ini dilakukan untuk memastikan agar setiap fitur yang ada berjalan dengan baik. Pengujian kompatibilitas dilakukan pada sistem operasi android versi 5.1 Lollipop. Dari hasil pengujian yang dilakukan diketahui bahwa implementasi aplikasi dapat dilakukan dengan baik.

Analisis hasil pengujian validasi hasil algoritma A(star)*

Pengujian validasi hasil dilakukan dengan menggunakan 17 titik awal dari suatu tujuan yang ingin dituju, titik awalnya diantara lain adalah node 1, node 12, node 7, node 4, node 5, node 3, node 2, node 11, node 15, node 10, node 8, node 13, node 3, node 17, node 6, node 16, dan node 9. Dari hasil pengujian yang telah dilakukan didapat kesimpulan jika hasil yang dihasilkan oleh perhitungan manual dan aplikasi memiliki kesamaan. Selain itu didapat kesimpulan juga bahwa tingkat akurasi yang dihasilkan oleh perhitungan Algoritma A*(star) mencapai lebih dari 80 % dikarenakan pemilihan jalur dan arah tujuan sudah tepat dan sesuai dengan yang diharapkan sekaligus jalur yang dipilih sudah memiliki cost yang terpendek.

7. PENUTUP

7.1 Kesimpulan

Berdasar hasil perancangan, implementasi dan pengujian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut :

1. Penerapan Algoritma A*(star) pada peta *offline* khususnya modul *routing* dirasa dapat membantu pengguna perangkat bergerak untuk menemukan tempat yang dituju dengan pemilihan jalur terpendek sehingga akan mempersingkat waktu perjalanan dan menghemat penggunaan daya baterai serta *bandwith* dari perangkat bergerak yang dimiliki serta tingkat akurasi yang dirasa cukup.
2. Rancangan dan implementasi modul *routing* dengan menggunakan Algoritma A*(star) secara umum mengarah kepada asas- asas *user friendly* sehingga pengguna aplikasi tidak merasa bingung dalam penggunaannya. Selain itu aplikasi ini dibangun dengan menggunakan SQLite sebagai media atau platform penyimpanan seluruh data yang diperlukan oleh aplikasi.
3. Tingkat akurasi yang dihasilkan oleh Algoritma A*(star) pada implementasi untuk menentukan jalur terpendek mencapai 80%. Akurasi jika dibandingkan dengan menggunakan algoritma lain sangat berbeda dikarenakan cara kerja Algoritma A*(star) ini mencoba seluruh kemungkinan jalur yang ada setelah itu akan ditentukan *cost* yang terkecil lalu dijadikan sebagai jalur tercepat.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi ini adalah sebagai berikut :

1. Perlunya dilakukan pengembangan lebih lanjut pada aplikasi khususnya mengatasi loading data yang lama saat proses *routing* saat titik awal dan tujuan yang ditentukan memiliki jarak yang jauh.
2. Perlunya pengembangan aplikasi dengan menggunakan format lain selain menggunakan pemrograman *Eclipse*.
3. Dari segi aplikasi perlu adanya pengembangan lebih lanjut dengan menambahkan beberapa faktor lain yang mendukung baik dari segi penerapan peta, algoritma dan arah jalan yang telah diperbaharui.

8. DAFTAR PUSTAKA

Alfan, Mochammad.,2013. *Analisis Performasi DBMS SQLite sebagai Mobile Database*. [e-book]. Universitas Telkom Bandung. S1. Tersedia di <https://openlibrary.telkomuniversity.ac.id/pustaka/95480/analisis-performansi-dbms-sqlite-sebagai-mobile-database.html>> [Diakses 10 Desember 2015]

Achmad, Marizki, C., 2013. *Perancangan aplikasi dalam implementasi shortest path finder di kawasan kampus IT Telkom berbasis android dengan Algoritma A**. S1. [e-book]. Fakultas Elektro dan Komunikasi, Universitas Telkom Bandung. Tersedia di <http://openlibrary.telkomuniversity.ac.id/pustaka/93152> [Diakses 3 Mei 2015]

Budi.,2009. Pengertian navigasi, [online]. Tersedia di <http://jawaratracker.com/pengertian-navigasi-dan-contohnya/>> [Diakses 3 Mei 2015].

Setiawan, Didik., 2015. *Perancangan dan implementasi Algoritma A* pada aplikasi angkot-finder di Kota Bandung untuk smartphone berbasis android*. S1. [e-book]. Fakultas Teknik, Universitas Telkom Bandung. Tersedia di <https://openlibrary.telkomuniversity.ac.id/pustaka/102733>> [Diakses 3 Mei 2015]

Hart, P. E.; Nilsson, N. J.; Raphael, B. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics SSC4 4 (2): 100-107. doi:10.1109/TSSC.1968.300136

HOT., 2012. *Mengumpulkan data spasial dengan OpenStreetMaps*. [online]. Tersedia di http://openstreetmap.id/docs/Unit1_Mengump

[ulkan data spasial dengan OSM Tingkat Dasar.pdf](#)> [Diakses 25 Mei 2015]

Kergall, John., 2014. "How to use the OSMDroid Library" [online]. Tersedia di <https://github.com/osmdroid/osmdroid/wiki/How-to-use-the-osmdroid-library>> [Diakses 1 Oktober 2015].

Mobac., 2008. "Mobile Atlas creator". [online]. Tersedia di www.mobac.sourceforge.net/> [Diakses 3 Mei 2015]

Novitasari, Vika., 2014., *Rancang Bangun Aplikasi Pencarian Rute Terpendek untuk Menemukan SPBU Terdekat di Kota Malang dengan Menggunakan Algoritma Genetik Berbasis Android*. S1. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Open Street Map Indonesia., 2014. *Pengenalan OpenStreetMap*. [online] Tersedia di www.slideshare.net/hotosmid/03-pengenalan-openstreetmap> [Diakses 5 Mei 2015]

Open Street Map., 2012. *OpenStreetMap*. [image online]. Tersedia di <http://www.openstreetmap.org/#map=18/-8.09917/112.17807>> [Diakses 25 Mei 2015]

Earl, Judea., 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley. ISBN 0-201-05594-5

S. Rabin., 2000 "A* Speed Optimizations." In *Game Programming Gems*, edited by Mark Deloura. Hingham, MA : Charles River Media.

Wiliam., 2003. *Location Based Service*, [online]. Tersedia di <http://supeeerblog.blogspot.com/2013/05/location-based-services-lbs.html>> [Diakses 25 Desember 2015]