PENERAPAN ALGORITMA A* SEARCH UNTUK PENCARIAN LINTASAN TERPENDEK MENUJU BEBERAPA LOKASI PADA PETA KOTA BANDUNG DENGAN BERBASIS ANDROID

Ardi Fadillah[1)], Dr. Wendi Zarman, M.Si.[2)]
[1), 2)]Jurusan Sistem Komputer Universitas Komputer Indonesia Bandung
[1]ardi.fadillah387@gmail.com, [2]wendizar@gmail.com

ABSTRAK

Persoalan pencarian rute terdekat di kota Bandung sangat diperlukan bagi wisatawan baik dalam maupun luar kota Bandung. Banyaknya jalan yang dapat ditempuh membuat pengguna mencari jalur tercepat untuk sampai ke tujuan. Untuk menyelesaikan masalah pencarian jarak terpendek dapat menggunakan algoritma A Search. Algoritma A* Search membandingkan jarak terdekat dari beberapa titik yang akan dilalui menuju titik tujuan, kemudian memilih titik yang memiliki nilai terkecil menuju tujuan. Penerapannya pada Android memungkinkan bagi pengguna yang memiliki mobilitas tinggi. Peta kota Bandung dimasukkan kedalam database program, sehingga pengguna tidak memerlukan koneksi internet untuk menggunakan aplikasi pencarian jalur terpendek ini.*

Kata Kunci: Jalur Terpendek, Algoritma A Search, Android*

1. PENDAHULUAN

Akhir-akhir ini pencarian rute terpendek menjadi masalah yang semakin penting, terutama untuk para wisatawan dalam dan luar kota. Hal ini menyebabkan kebutuhan untuk memperoleh informasi rute telah menjadi kebutuhan pokok masyarakat. Salah satu informasi yang dibutuhkan masyarakat terutama wisatawan dalam dan luar kota adalah kebutuhan informasi yang berkaitan dengan tempat-tempat umum, seperti hotel, *mall*, dan tempat wisata. Letak tempat-tempat umum yang berbeda-beda menyulitkan wisatawan dalam dan luar kota dalam mencapai tujuan. Mereka biasanya mencari rute terpendek agar dapat menempuh perjalanan secepat mungkin untuk dapat sampai ke tempat tujuan dan mereka biasanya mencari titik tujuan mana yang terkdekat dahulu untuk dicapai, sehingga tidak banyak memakan waktu serta biaya.

Pemilihan rute ini dapat menggunakan peta konvensional untuk melihat seluruh jalan raya yang terdapat di dalam kota, lalu memilih lintasan terpendek. Tetapi hal ini seringkali tidak dapat membantu secara maksimal karena banyaknya jalan raya yang ada sehinga menyebabkan banyaknya pilihan jalur yang dapat ditempuh. Serta dengan hanya melihat pada peta, sulit sekali untuk dapat secara langsung menentukan jalur tertentu adalah jalur yang terpendek karena kebanyakan terdiri dari jalan yang berkelok-kelok sehingga tidak dapat diperkirakan jarak tempuh pada jalur tersebut.

Salah satu metode penyelesaian permasalahan rute terpendek ini dengan menggunakan Android menggunakan metode *path tree* atau penandaan jalur yang dianggap sebagai lintasan terpendek dan dengan menggunakan algoritma A* *Search* sebagai alur program yang cukup efisien pada kasus pencarian lintasan terpendek. Penggunaan algoritma A* *Search* pada kasus ini bertujuan untuk menganalisa dan membandingkan hasil pencarian lintasan terpendek dengan aplikasi yang sebelumnya pernah dibuat oleh orang lain yang menggunakan algoritma yang berbeda.

Mengingat pengguna aplikasi yang sangat *mobile* sekarang ini, maka aplikasi ini dibuat untuk mengikuti keinginan pengguna dengan menerapkannya pada sistem operasi android. Umumnya aplikasi android memerlukan koneksi internet yang lancar, terutama untuk aplikasi berupa peta. Koneksi internet sangat dibutuhkan, namun terkadang jaringan yang jelek membuat penampilan peta tidak maksimal bahkan tidak dapat dipakai. Pembuatan aplikasi ini memudahkan pengguna, karena dibuat dan dapat digunakan tanpa memerlukan koneksi internet (*offline*).

2. TEORI PENUNJANG

***Graph* (graf)** bisa dibayangkan sebagai kumpulan obyek atau aktifitas. Sebagai contoh, rute bis kota dari satu terminal ke terminal lain, rute perjalanan seseorang dari satu tempat ke tempat lain, dan contoh-contoh lain yang bisa dijadikan sebagai suatu *graph*. Contoh di atas merupakan contoh klasik dari teori *graph* yang lebih dikenal dengan persoalan *travelling salesman problem* atau *shortest path problem*, yand pada prinsipnya mencari lintasan terpendek dari semua tempat yang harus dikunjungi, sehingga dapat menghemat waktu, tenaga, maupun biaya [1].

Lintasan terpendek adalah lintasan yang memiliki jarak terpendek yang diperlukan untuk mencapai suatu tempat dari tempat tertentu. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf yang berbobot, yaitu graf yang memiliki suatu nilai atau bobot.

A* (dibaca "A bintang" atau "A *star*") adalah algoritma pencarian graf/pohon yang mencari jalur dari satu titik awal ke sebuah titik akhir yang telah ditentukan. Algoritma A* menggunakan pendekatan heuristik h(x) yang memberikan peringkat ke tiap-tiap titik x dengan cara memperkirakan rute terbaik yang dapat dilalui dari titik tersebut.

A* juga dapat dijamin keoptimalannya untuk sembarang heuristik, yang berarti bahwa tidak ada satu pun algoritma lain yang mempergunakan heuristik yang sama akan mengecek lebih sedikit titik dari A*, kecuali ketika ada beberapa solusi parsial dimana dapat dengan tepat memprediksi ongkos jalur minimal [2]. Kompleksitas waktu dari A* sangat bergantung dari heuristik yang digunakannya. Pada kasus terburuk, jumlah titik yang diperiksa berjumlah eksponensial terhadap panjang solusi (lintasan terpendek), tetapi A* akan memiliki kompleksitas waktu polinomial apabila fungsi memenuhi kondisi berikut:

$$|h(x) - h^*(x)| \leq O(\log h^*(x) \quad \textbf{(2.1)}$$

Haversine Formula adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan garis lintang (*latitude*) dan garis bujur (*longitude*) [3]. Dapat dikatakan *haversine formula* digunakan untuk menghitung jarak antara dua titik pada permukaan bumi berdasarkan garis lintang (*latitude*) dan garis bujur (*longitude*) [3].

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk Eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. IBM menginvestasikan US$ 40 juta untuk pengembangannya. Sejak 5 November 2001, konsorsium Eclipse Foundation megambil alih pengembangan Eclipse lebih lanjut [4].

***Geocoding* (pencocokan lokasi)** merupakan suatu proses untuk menggabungkan satu lokasi fisik di bumi dengan lokasi logiknya. Informasi dari satu lokasi didapatkan dari perhitungan koordinat, sehingga didapatkan jarak, arah, dan lokasi yang tepat [5].

SQLite merupakan sebuah sistem manajemen basis data selasional yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat *public domain* yang dikerjakan oleh D. Richard Hipp [6].

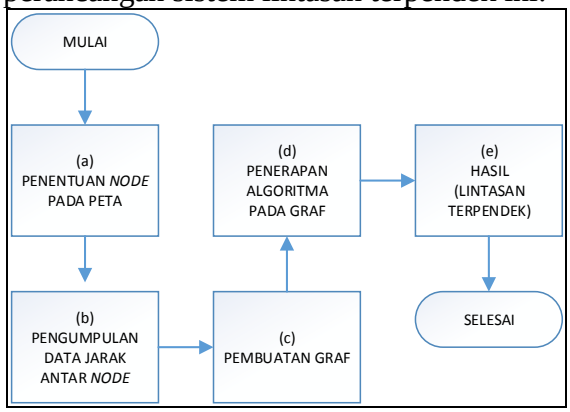
Skobbler (http://maps.skobbler.com/) adalah *map source freeware* dan *open source* yang memungkinkan kita mengunduh peta secara keseluruhan. Peta yang telah diunduh dapat digunakan, karena telah mencakup semua data spasial di dalam peta.

Android adalah sistem operasi yang bersifat *open source*, dan google merilis kodenya di bawah lisensi Apache [7].

Kota Bandung merupakan kota metropolitan terbesar di Jawa Barat sekaligus menjadi ibu kota provinsi tersebut. Kota ini terletak 140 km sebelah tenggara Jakarta dan merupakan kota terbesar ketiga di Indonesia setelah Jakarta dan Surabaya menurut jumlah penduduk [8].

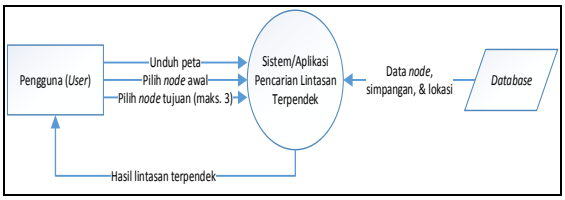
3. PERANCANGAN SISTEM

Pada tahap ini ditujukan untuk memperoleh gambaran mendasar mengenai desain perancangan sistem aplikasi pencarian lintasan terpendek yang akan dibuat. Berikut adalah diagram alir dalam perancangan sistem lintasan terpendek ini.



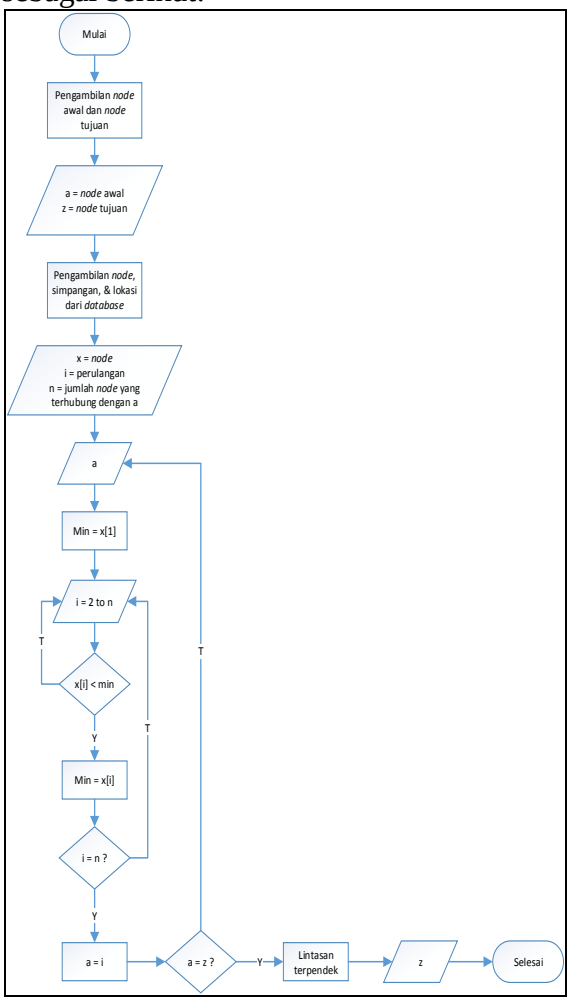
Gambar 3.1 Diagram Alir Sistem Pembuatan Aplikasi

Berikut adalah proses aplikasi sistem pencarian lintasan terpendek digambarkan dalam bentuk *Data Flow Diagram* (DFD):



Gambar 3.2 *Data Flow Diagram* (DFD) Proses Aplikasi Sistem Pencarian Lintasan Terpendek

Proses pengolahan *node* awal dan *node* tujuan yang telah dipilih dengan menggunakan algoritma A* *Search* adalah sebagai berikut:



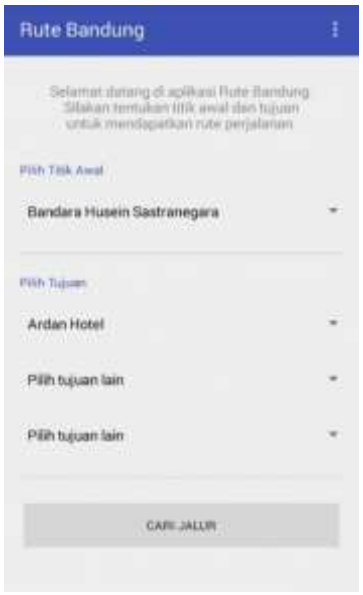
Gambar 3.3 Diagram Alir Proses Pengolahan Pencarian Lintasan Terpendek

4. IMPLEMENTASI DAN PENGUJIAN

Implementasi Aplikasi

a. Implementasi Antar Muka Pengguna (*User Interface*) Aplikasi

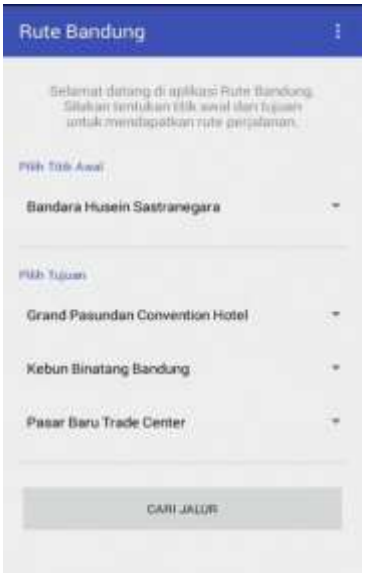
Berikut adalah tampilan antar muka pengguna (*user interface*) aplikasi.



Gambar 4.1 Tampilan Antar Muka Pengguna (*User Interface*) Aplikasi

b. Implementasi Pencarian Lintasan Terpendek Menggunakan Aplikasi

Berikut contoh tampilan pengambilan *node*/titik awal dan *node*/titik tujuan pada aplikasi.



Gambar 4.2 Tampilan Pengambilan *Node*/Titik Awal dan *Node*/Titik Tujuan pada Aplikasi

Setelah pengguna memilih *node*/titik awal dan *node*/titik tujuan, aplikasi akan memproses lintasan terpendek menuju ketiga *node*/titik tujuan dari *node*/titik awal dan menampilkannya pada peta. Berikut ini tampilan setelah *button* "Cari Jalur" di jalankan (*run*).



Gambar 4.3 Tampilan Lintasan Terpendek di Peta Setelah Diproses pada Aplikasi

Untuk mengetahui jalan apa saja yang dilalui dan berapa jaraknya, pengguna dapat menjalankan *menu* 1 "*View Option* (≡)" pada pojok kanan atas. Berikut ini tampilan dari *view option* berdasarkan pencarian di atas.

Pengujian Aplikasi

a. Pengujian *Alpha*

Pengujian *alpha* dilakukan dari sisi pengembang oleh seorang pengguna. Pengujian dilakukan terhadap aplikasi untuk memastikan bahwa aplikasi dapat berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. Pengujian *alpha* berfokus pada persyaratan fungsionalitas aplikasi.

Pada pengujian *alpha* akan dilakukan metode *black box testing* dan pengujian langsung aplikasi pencarian lintasan terpendek terhadap menu-menu yang tersedia pada aplikasi. Pada *black box testing* tidak terlalu memperhatikan struktur internal (*source code*) dari program, tetapi kepada kesalahan program.

Hasil pengujian menunjukkan bahwa aplikasi dapat berjalan di semua spesifikasi *Smartphone* Android yang diujikan. Aplikasi sudah dapat berjalan sesuai dengan yang diharapkan dan secara fungsionalitas,

sistem sudah dapat menghasilkan keluaran yang diharapkan.

b. Pengujian *Beta*

Pengujian *beta* dilakukan pada satu atau lebih pengguna oleh pemakai akhir perangkat lunak. Pengujian *beta* merupakan pengujian yang dilakukan secara objektif, dimana dilakukan pengujian secara langsung terhadap *user* dengan menggunakan kuesioner mengenai kepuasan atas *user* atas aplikasi yang dibangun. Adapun metode kuantitatif berdasarkan data dari *user*. Pengujian *beta* dilakukan dengan kuesioner dengan teknik *Simple Random Sampling*. Kuesioner disebarkan kepada 30 orang dari umur 20 tahun ke atas yang menggunakan *smartphone* Android dengan klasifikasi kota Bandung 10 orang, luar kota Bandung 15 orang, dan turis asing 5 orang.

Hasil pengujian menunjukkan bahwa pengguna sangat setuju:

1) aplikasi Rute Bandung mudah untuk diinstalasi;
2) antarmuka (*interface*) aplikasi Rute Bandung nyaman untuk digunakan;
3) aplikasi Rute Bandung mudah untuk digunakan atau dijalankan;
4) aplikasi Rute Bandung memudahkan pengguna mendapatkan informasi lintasan terpendek menuju *public transportation center*, hotel, tempat pariwisata, dan *mall*/pusat perbelanjaan; dan
5) aplikasi Rute Bandung sesuai dan akurat menunjukkan tempat-tempat *public transportation center*, hotel, tempat pariwisata, dan *mall*/pusat perbelanjaan.

5. KESIMPULAN DAN SARAN

Setelah melalui tahap implementasi dan pengujian sistem dilakukan maka dapat diambil beberapa kesimbulan sebagai berikut:

1) Aplikasi ini berhasil menemukan lintasan terpendek dari titik awal ke titik tujuan yang telah ditentukan oleh pengguna.
2) Aplikasi pencarian lintasan terpendek ini memberikan informasi (rekomendasi) tentang efisiensi jarak tempuh yang lebih baik dicapai terlebih dahulu oleh pengguna.
3) Aplikasi pencarian lintasan terpendek ini *mobile* (mudah dibawa ke mana saja), karena dijalankan di *smartphone* yang biasanya selalu dibawa oleh pengguna.
4) Aplikasi ini *user friendly* (mudah digunakan), karena pengguna dimudahkan dalam menggunakan aplikasi baik dari *user interface* dan penampilan peta serta lintasannya.

Adapun saran untuk mengoptimalkan aplikasi di masa mendatang adalah sebagai berikut:

1) Sebaiknya aplikasi dapat langsung digunakan tanpa harus mengunduh peta terlebih dahulu.
2) Hasil pencarian lintasan terpendek yang lebih efektif dapat diperoleh dengan melakukan penambahan parameter selain jarak tempuh, misalnya dengan menghitung waktu proses, biaya perjalanan, waktu tempuh, alternatif angkutan umum, peraturan lalu lintas, dan sebagainya.
3) Masih ada beberapa algoritma pencarian lintasan terpendek lainnya jika peneliti lain ingin membandingkan.
4) Perlu dilakukan pengembangan lebih lanjut untuk lokasi yang dicari dan wilayah pencarian.

6. DAFTAR PUSTAKA

[1] Joyner, David., Van, Nguyen Minh., & Cohen, Nathan. 2010. *Algorithmic Graph Theory*. GNU Free Documentation License.

[2] Riftadi, Mohammad. 2007. Variasi Penggunaan Fungsi Heuristik dalam Pengaplikasian Algoritma A*. *Makalah IF2251 Strategi Algoritmik Tahun 2007*. Bandung: Teknik Informatika Institut Teknologi Bandung (ITB).

[3] Chopde, Nitin R., & Nichat, Mangesh K. 2013. Landmark Based Shortest

Path Detection by Using A* and Haversine Formula. *International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013*. pp298-302.

[4] Fatimah, Wina Noviani. 2011. *Pengenalan Eclipse*. Diakses pada Maret 2015 dari *world wide web*: https:// wi01.files.wordpress.com/2011/02/ pengenalan-eclipse.pdf.

[5] Handayani, Dewi., Soelistijadi, R., & Sunardi. 2005. Pemanfaatan Analisis Spasial untuk Pengolahan Data Spasial Sistem Informasi Geografi. *Jurnal Teknologi Informasi DINAMIK Volume X No. 2 Mei 2005*. hal108-116.

[6] SQLite. Diakses pada Februari 2015 dari *world wide web*: http://id.wikipedia.org /wiki/SQLite.

[7] Safaat, Nazruddin. 2011. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.

[8] Kota Bandung. Diakses pada Mei 2015 dari *world wide web*: http://id.wikipedia.org/ wiki/Kota_Bandung.