

Semarang, 22 Juli 2013

**PENGGUNAAN ALGORITMA DIJKSTRA PADA APLIKASI SEARCHING
HOTEL DI KOTA SEMARANG**

Febri Anjar Pratama

Desi Purwanti K.

Jurusan Teknik Informatika – S1, Fakultas Ilmu Komputer

Universitas Dian Nuswantoro Semarang

Jl. Nakula I No. 5-11 Semarang Nomor Telepon (024) 3517261 No Fax 0243520165
Website: www.dinus.ac.id E-mail: sekretariat@dinus.ac.id anjar.simple@gmail.com

ABSTRAK

Dengan semakin canggihnya teknologi setiap tahunnya, maka semakin di kembangkannya aplikasi dengan kelebihan-kelebihan tertentu. Maka dari itu dengan ini dibuat adanya aplikasi dengan menerapkan Algoritma Dijkstra. Aplikasi pencarian Hotel dengan menerapkan algoritma dijkstra di dalamnya. Untuk memudahkan para wisatawan atau warga local Semarang untuk mempermudah dalam menemukan Hotel atau penginapan secara detail dan dengan di dukung dengan dijkstra dapat di cari lokasi terdekat dengan user. Maka dari itu peneliti akan membuat suatu aplikasi dengan judul “Penggunaan Algoritma Dijkstra pada Aplikasi Searching Hotel di Kota Semarang”. Dengan adanya aplikasi ini diharapkan dapat mempermudah para wisatawan dan masyarakat local dalam menemukan hotel atau penginapan terdekat dengan lebih lengkap dan spesifik. Untuk mendukung pembuatan aplikasi ini peneliti menerapkan Algoritma Dijkstra untuk mencari rute terpendek, Android versi 2.3 dan google maps api untuk menampilkan peta hotel.

Kata Kunci : navigasi, android, dijkstra, wisatawan, teknologi, hotel

Semarang, 22 Juli 2013

**PENGGUNAAN ALGORITMA DIJKSTRA PADA APLIKASI SEARCHING
HOTEL DI KOTA SEMARANG**

Febri Anjar Pratama

Desi Purwanti K.

Jurusan Teknik Informatika – S1, Fakultas Ilmu Komputer

Universitas Dian Nuswantoro Semarang

Jl. Nakula I No. 5-11 Semarang Nomor Telepon (024) 3517261 No Fax 0243520165
Website: www.dinus.ac.id E-mail: sekretariat@dinus.ac.id anjar.simple@gmail.com

ABSTRACT

With the growing sophistication of teknologi every year, then getting in kembangkannya applications with certain advantages. So from that with this application is made by applying the Dijkstra algorithm. Hotel search applications by applying Dijkstra's algorithm in it. To make it easier for tourists or local residents Semarang to facilitate in finding hotels or accommodation in detail and with the bearing of the Dijkstra can find the nearest location by user. So from that researchers will make an application under the title "The use of Dijkstra's algorithm on Searching Applications Hotel in Semarang City". The availability of these applications are expected to facilitate the tourists and the local community to find the nearest hotel or accommodation more complete and specific. To support these applications making researchers apply Dijkstra's algorithm to find the shortest route, the Android version 2.3 and google maps to show you a map hotel fire.

Keyword : : navigation, android, dijkstra, tourist, teknologi, hotels

I. PENDAHULUAN

Pada dasarnya aplikasi pencarian Hotel hanya menampilkan pencarian seperti sistem informasi dan letak-letak Hotel saja. Tidak ada metode untuk mencari rute terpendek dari user. Misalnya seperti menampilkan Hotel-hotel yang berada di sekitar tempat posisi user dengan jalur yang terdekat. Sehingga di sini akan di terapkan metode Algoritma Dijkstra yang bertujuan agar dapat mencari rute-rute terpendek dengan user. [1]

Permasalahan utama pencarian rute terpendek tentu saja mencari rute atau jalur terpendek yang memungkinkan. Namun untuk implementasinya, persoalan ini dapat dikembangkan lebih luas lagi diantaranya untuk mencari biaya minimum, dll. Intinya adalah mencari solusi yang paling efektif yang dapat diterapkan dalam persoalan yang dihadapi. [2]

Secara umum, terdapat lima *shortest path algorithm* yang sering digunakan, yaitu Algoritma Dijkstra, A* Search, Bellman-Ford, Johnson, dan Floyd-Warshall. Setiap algoritma memiliki karakteristik dan kelebihan yang berbeda-beda. Hal inilah yang menyebabkan tidak semua algoritma dapat diterapkan dalam aplikasi ini. Pada aplikasi ini, tujuannya untuk mencari satu titik rute terpendek namun user dapat menentukan titik awal sehingga titik awal dapat terletak dimana saja. Setiap perpindahan dari satu titik ke titik lain membutuhkan *cost*, di mana *cost* selalu bernilai positif. [3]

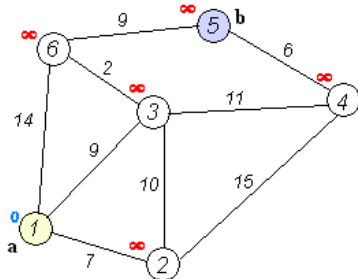
Algoritma Dijkstra merupakan sebuah *graph search algorithm* yang menyelesaikan *singlesource shortest path problem* di mana Dijkstra akan mencari jalur terpendek dari satu *start vertex* dengan cara memeriksa dan membandingkan setiap jalur. Walaupun demikian, Dijkstra dapat dimodifikasi sehingga dapat digunakan untuk mencari jalur terpendek dari setiap *vertex*. Untuk *sparse graph*, yaitu *graph* dengan jumlah edge yang lebih kecil dari V^2 , Dijkstra dapat memiliki *time complexity* yang lebih kecil (Cormen, dkk., 2001: 599). Sama seperti Dijkstra, Bellman-Ford menghitung *single-source shortest path* dan memiliki struktur yang mirip dengan Dijkstra, namun lebih lambat. Kelebihan dari algoritma ini adalah dapat digunakan pada *graph* dengan *negative edge weights*. Dengan demikian, Dijkstra akan lebih cocok diterapkan dalam aplikasi ini dibandingkan dengan Bellman-Ford. (Cormen, dkk., 2001: 588) [4]

Algoritma Dijkstra mengalami banyak pengembangan dan salah satunya adalah algoritma A* Search. A* Search memiliki performa yang lebih tinggi, di mana algoritma ini menggunakan *heuristic function*. *Heuristic* adalah studi mengenai metode dan aturan dalam penemuan solusi. [5]

Berdasarkan uraian diatas maka pada tugas akhir ini akan di terapkan algoritma dijkstra ke dalam aplikasi searching hotel dan diharapkan aplikasi ini bisa menjadi lebih baik dari yang sudah ada sebelumnya. [6]

Algoritma Dijkstra (Pencarian Jalur Terpendek)

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



Gambar 2.1 Contoh keterhubungan antar titik dalam algoritma Dijkstra

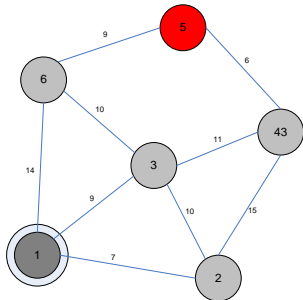
Pertama-tama tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu per satu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap. Inilah urutan logika dari algoritma Dijkstra:

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node “Belum terjamah” dan set node awal sebagai “Node keberangkatan”

3. Dari node keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi $6+2=8$. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node terjamah”. Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set “Node belum terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3

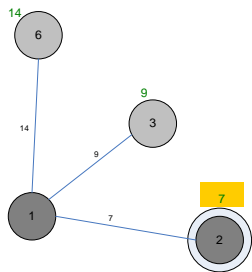
Dibawah ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

1. Node awal 1, Node tujuan 5. Setiap edge yang terhubung antar node telah diberi nilai



Gambar 2.2 Contoh kasus Dijkstra - Langkah 1

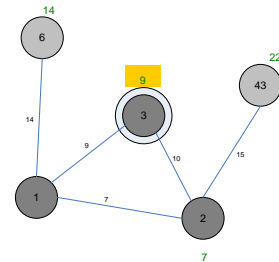
2. Dijkstra melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node 1), dan hasil yang didapat adalah node 2 karena bobot nilai node 2 paling kecil dibandingkan nilai pada node lain, nilai = 7 ($0+7$).



Gambar 2.3 Contoh kasus Dijkstra - Langkah 2

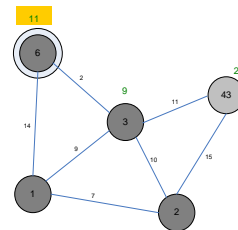
3. Node 2 diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan

node yang telah terjamah. Dan kalkulasi dijkstra menunjukkan bahwa node 3 yang menjadi node keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai 9 ($0+9$).



Gambar 2.4 Contoh kasus Dijkstra - Langkah 3

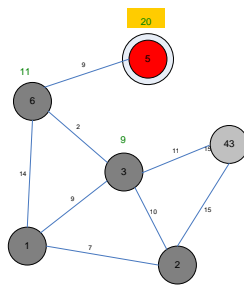
4. Perhitungan berlanjut dengan node 3 ditandai menjadi node yang telah terjamah. Dari semua node tetangga belum terjamah yang terhubung langsung dengan node terjamah, node selanjutnya yang ditandai menjadi node terjamah adalah node 6 karena nilai bobot yang terkecil, nilai 11 ($9+2$).



Gambar 2.5 Contoh kasus Dijkstra - Langkah 4

5. Node 6 menjadi node terjamah, dijkstra melakukan kalkulasi

kembali, dan menemukan bahwa node 5 (node tujuan) telah tercapai lewat node 6. Jalur terpendeknya adalah 1-3-6-5, dan nilai bobot yang didapat adalah 20 (11+9). Bila node tujuan telah tercapai maka kalkulasi dijkstra dinyatakan selesai.



Gambar 2.6 Contoh kasus Dijkstra -
Langkah 5

Google API

Google API bisa di katakan bagian dari Framework Google.

Google menyediakan berbagai API (Application Programming Interface) yang sangat berguna bagi pengembang web maupun aplikasi desktop untuk memanfaatkan berbagai fitur yang disediakan oleh Google seperti misalnya: AdSense, Search Engine, Translation maupun YouTube. API secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi atau web yang kita buat dengan fungsi-fungsi yang dikerjakan. Misalnya dalam hal ini Google API berarti kode program (yang disederhanakan) yang dapat kita

tambahkan pada aplikasi atau web kita untuk mengakses/menjalankan/memanfaatkan fungsi atau fitur yang disediakan Google. Misalnya saja kita bisa menambahkan fitur Google Map pada website kita.

Google API dapat dipelajari langsung melalui Google Code. Melalui Google Code kita dapat belajar tentang Google API dan dapat mengimplementasikan pada aplikasi web atau website yang kita kembangkan.

Ada banyak API yang disediakan oleh Google, beberapa diantaranya adalah:

- Language API : untuk memanfaatkan fitur translation yang dimiliki Google.
- Earth API : memanfaatkan fitur yang ada pada Google Earth
- Javascript API
- Maps API : memanfaatkan fitur yang ada pada Google Maps
- Search API : memanfaatkan fitur pencarian pada Google Search
- Visualization API : membuat grafik maupun chart dengan Google API
- YouTube API : memanfaatkan fitur yang ada pada YouTube misalnya untuk pencarian video

Salah satu cara mudah mempelajari Google API adalah dengan memanfaatkan Google AJAX APIs Playground. AJAX APIs playground adalah sebuah situs yang disediakan oleh Google bagi kita untuk mencoba secara langsung sejumlah

Google API yang berbasis AJAX (Asynchronous Javascript and XML). Karena berbasis AJAX maka tentunya semua kode program dalam sintaks Javascript yang bisa kita lihat, kopi dan paste secara langsung untuk digunakan pada website kita.

Dengan menggunakan Google AJAX API, kita bisa mengintegrasikan data pada website kita dengan API yang disediakan oleh Google.

GPS (Global Positioning System)

Global Positioning system (GPS) atau system pemosisi global menggunakan system yang digunakan menentukan posisi di permukaan bumi dengan sinkronisasi sinyal satelit.

Dengan bantuan GPS seseorang dapat mengetahui posisi objek yang diinginkan dengan bantuan perangkat yang memiliki sensor GPS di dalamnya. GPS bekerja ketika sejumlah satelit yang berada di orbit Bumi memancarkan sinyalnya ke Bumi kemudian sinyal tersebut ditangkap oleh sebuah alat penerima yang nantinya diubah menjadi informasi berupa titik lokasi dari alat penerima tersebut. Karena alat ini bergantung penuh pada satelit maka sinyal satelit merupakan hal yang penting untuk mendapatkan informasi posisi objek yang berupa titik koordinat. Untuk itu perlu diperhatikan hal yang dapat mengganggu sinyal satelit antara lain adalah kondisi geografis, alat yang menggunakan gelombang elektromagnetik, gedung dan sinyal yang memantul.

Keakuratan atau ketepatan menjadi perhatian utama dalam sistem ini untuk mendapatkan sebuah lokasi

atau koordinat. Tingkat akurasi pada GPS lebih sering dipengaruhi oleh faktor lingkungan di sekitarnya. Ketika alat GPS berada di sebuah lembah, maka tingkat akurasinya akan jauh lebih rendah daripada ada di puncak gunung

Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

- **Multi-platform:** Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- **Multi-language:** Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode

pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP:

Core platform

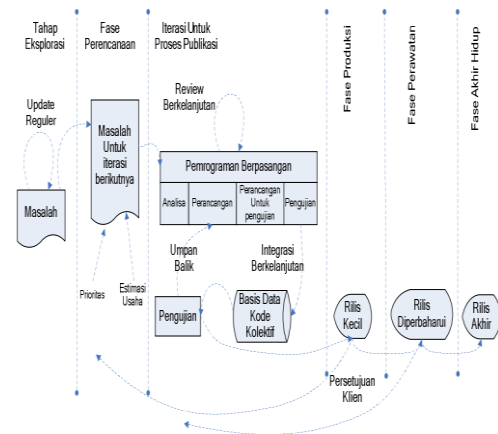
- OSGi
- SWT (Standard Widget Toolkit)
- JFace
- Eclipse Workbench

Secara standar Eclipse selalu dilengkapi dengan JDt (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-innya diimplementasikan dalam bahasa pemrograman Java. Konsep Eclipse adalah IDE yang terbuka (open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan

menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya. Salah satu situs yang menawarkan plug-in secara gratis seperti Eclipse downloads by project.

METODE PENELITIAN

Dalam tugas akhir ini penulis menggunakan model pengembangan sistem model *agile*. Sebuah “metodologi praktis” untuk memodelkan system perangkat lunak ber-teknik orientasi objek secara efektif. Metodologi AM terbentuk dari kumpulan-kumpulan praktis yang dipelihara oleh prinsip-prinsip dan nilai-nilai yang dapat diaplikasikan oleh perangkat lunak profesional dari waktu ke waktu.



Gambar 3.1 : Siklus fase Agile XP

Tahap – Tahap Pengembangan Sistem Tiga Tahapan khusus dalam Agile Modeling :

1. Tahap Pemodelan (*Modeling*).
2. Tahap Pengembangan (*Development*).
3. Tahap Praktis (*Practice*).

Selain itu juga *Agile* model memiliki beberapa fitur pada saat melakukan pengembangan perangkat lunak diantaranya adalah:

- 1) Iterasi yang cepat dan pengiriman software yang berfungsi secara regular memastikan kepuasan pelanggan.
- 2) Perubahan yang telat dapat ditangani dengan mudah dan juga diterima secara terbuka.
- 3) Perkembangan dinilai berdasarkan implementasi software
- 4) Komunikasi pelanggan dan pengguna ditekankan secara bertatap muka.

II. HASIL DAN PEMBAHASAN

Aplikasi ini dibuat sebagai media pembantu pencarian informasi tentang alamat hotel di kota Semarang. Batasan masalah dalam aplikasi ini adalah ditujukan untuk para wisatawan baik local maupun dari luar kota semarang. Aplikasi ini dibuat mudah untuk digunakan serta mampu memberikan informasi yang dibutuhkan oleh para pengguna. Tidak semua hotel dihadirkan di aplikasi ini, yang akan dihadirkan di aplikasi ini hanya hotel-hotel di kota semarang.

Aplikasi ini juga menampilkan jalan tercepat/terpendek yang bisa

5) Setiap pertemuan dengan anggota tim dilakukan secara bertatap muka.

6) Setiap anggota tim pengembang adalah orang yang berkomitmen dan bermotivasi tinggi serta kompeten dan dapat dipercaya.

Tahapan Praktis Agile

Tahapan Praktis Agile Modeling adalah :

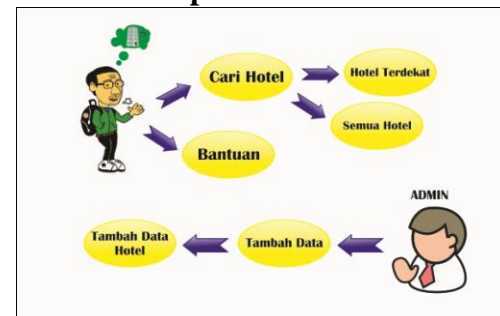
Nilai lebih dalam pemodelan Agile yang menitik beratkan pada realisasi di dunia nyata.

Aktivitas yang dilakukan adalah:

- Komunikasi lengkap dengan *stakeholder* .
- Kerjasama di antara *stakeholder* .

dilalui untuk mencari tempat yang akan dituju oleh pengguna, sehingga pengguna dapat mengetahui jalur yang akan ditempuh untuk mengunjungi tempat tersebut.

Arsitektur Aplikasi



Pada diagram *use case* diatas ada dia use case. Dimana *software* adalah *single user* dan user akan memilih aksi dimana user dapat menemukan dua buah aksi pilihan yang nantinya akan langsung ditangani oleh aplikasi.

Menu Utama

Merupakan awal dari aplikasi ini, didalamnya terdapat menu utama lainnya diantaranya cari kampus, tambah data, bantuan, keluar. User dapat memilih dengan menyentuh menu pada layar.

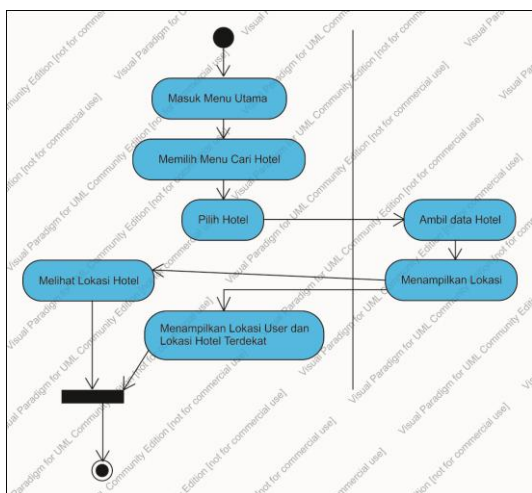
Cari Hotel

Merupakan inti dari program, yang menampilkan data hotel yang dituju, kemudian menampilkan hotel yang di pilih.

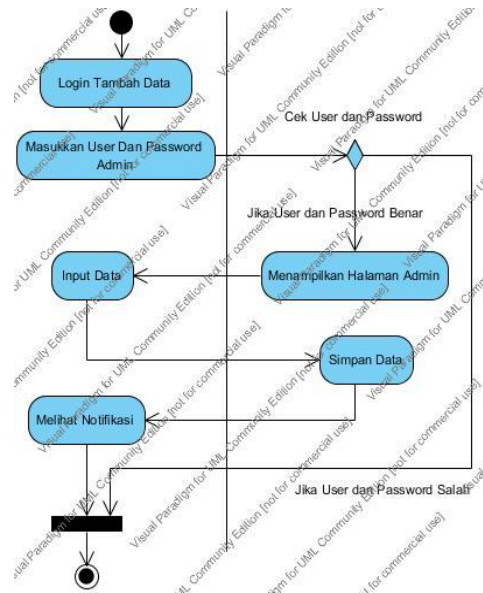
Tambah Data

Tambah Data yaitu menu dimana user dapat menambah data hotel

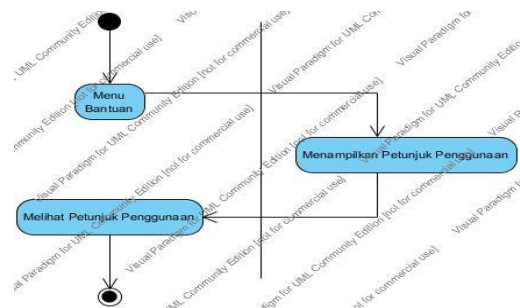
Activity Diagram.



Gambar 4.3 : Activity Cari Hotel



Gambar 4.4 : Activity Tambah Data



Gambar 4.5 : Activity Bantuan

III. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan analisis dan pembahasan pada bab sebelumnya, dapat ditarik simpulan yang berkaitan dengan perumusan masalah dan tujuan penelitian bahwa Penggunaan Algoritma Dijkstra pada Aplikasi Searching Hotel di kota Semarang

berbasis android dapat disimpulkan sebagai berikut :

1. Aplikasi ini dapat membantu pengguna untuk mengetahui letak tempat hotel di kota Semarang. Google Map juga tidak memiliki fasilitas untuk mengetahui letak hotel di kota Semarang, google map juga tidak menyediakan informasi-informasi rute terdekat menuju hotel.
2. Aplikasi ini dapat menunjukkan letak hotel penginapan, lebih menguntungkan dari hanya menggunakan peta konvensional. Aplikasi ini menampilkan letak lokasi hotel secara jelas sehingga pengguna dapat menuju lokasi yang di inginkan dengan mudah.

Saran

Dari beberapa kesimpulan yang telah diambil, maka didapatkan saran yang akan sangat membantu untuk pengembangan perangkat lunak ini selanjutnya. Setelah menyelesaikan tugas akhir ini, ada beberapa kekurangan yang tidak dapat diselesaikan pada tugas akhir ini. Beberapa kekurangan tersebut terangkum dalam saran di bawah ini, yaitu :

1. Aplikasi masih memiliki beberapa kekurangan di antaranya, jika user yang berasal dari luar kota dan tidak menggunakan kendaraan pribadi, maka di harapkan apabila aplikasi ini ingin di kembangkan nantinya karena aplikasi bersifat opensource, di tambahkan jalur

angkutan umum untuk menempuh lokasi kampus yang dituju. Agar nantinya dapat mempermudah pencarian lokasi hotel (bagi user yang tidak menggunakan kendaraan pribadi).

2. Keakuratan aplikasi ini harus ditingkatkan karena lokasi di aplikasi ini dengan lokasi sebenarnya berselisih kurang lebih 10 meter.
3. Untuk android dengan os jenis 4.0 *Ice Cream Sandwich* (ICS) ditambah *librarynya* sehingga bisa berjalan dengan baik.
4. Pada halaman Menu Utama bisa ditambahkan dengan tombol pencarian berdasarkan nama dan harga. Contohnya : user bisa mengetikkan hotel yang di cari agar lebih mempermudah dalam pencarian. Serta menambahkan database untuk berelasi sehingga bila pengguna ingin melakukan pencarian tempat bisa mengetikkan contoh Hotel Grand Candia tau mengetikkan harga kisaran berapa yang di inginkan.
5. Di harapkan aplikasi dapat dikembangkan sendiri oleh user, update aplikasi tidak harus menunggu dari pembuat aplikasi karena aplikasi bersifat opensource.

DAFTAR PUSTAKA

- [1]. Kusnadi, Rahmat. (1999) .*Geografi SMU kelas 1*. Bandung : Grafindo Media Pratama.
- [2]. Prahasta, Eddy (2009). Sistem Informasi Geografis Konsep-konsep Dasar : Informatika.
- [3]. Prahasta, Edy. (2005). *Sistem Informasi Geografis*. Bandung: C.V.Informatika.
- [4]. Riyanto, Prinali dan Hendi Inderlako. (2009). *Aplikasi Pengembangan SIG*. Yogyakarta : Gaya Media.
- [5]. Safaat, Nazarudin. (2011). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: C.V.Informatika.
- [6]. Satzinger JW, Jackson RB, Burd SD. 2007. System Analysis and Design in Changing a World, Fourth Edition. Canada: Thomson
- [7]. Robet Sangodrig, Karla Wagner. 2007. Minimum Spaning Tree