

# RANCANG BANGUN APLIKASI ANDROID PENCARIAN RUMAH SAKIT DI JAKARTA MENGGUNAKAN ALGORITMA DIJKSTRA

Cahyani Budihartanti<sup>1</sup>, Riswan Pandiangan<sup>2</sup>

Program Studi Sistem Informasi – STMIK Nusa Mandiri<sup>1</sup>

Program Studi Teknik Informatika – STMIK Nusa Mandiri<sup>2</sup>

<sup>1</sup>cahyani\_budihartanti@yahoo.com, <sup>2</sup>riswanpandiangan@gmail.com

**Abstrak** - Ketika kita berkunjung ke suatu tempat baru, adakalanya kita mengalami kesulitan untuk menemukan letak suatu tempat dengan benar, atau mungkin kita bisa saja keliru menemukan arah ketempat tersebut, sehingga bukannya sampai ketempat tujuan tetapi malah menemukan tempat berbeda. Jakarta adalah kota yang cukup luas, untuk menemukan suatu tempat terutama saat kondisi kritis bisa saja melelahkan dan butuh waktu lama, misalnya saja ketika kita sedang sakit atau mengalami kecelakaan. Oleh sebab itu untuk efisiensi pencarian rumah sakit di Jakarta, penulis membuat aplikasi pencarian rumah sakit di Jakarta, yang dikembangkan untuk smartphone berbasis Android menggunakan ADT Bundle dari eclipse. Pengguna akan semakin mudah menemukan lokasi rumah sakit dengan tepat dengan bantuan peta digital berbasis google maps yang langsung ditampilkan di perangkat pengguna, juga panduan arah untuk menuju lokasi tujuan yang dapat menunjukkan jalur yang terpendek untuk sampai dilokasi tujuan. Dengan bantuan Global Positioning System (GPS) yang berfungsi sebagai penunjuk lokasi, Location Based Service (LBS) yang menyediakan informasi berdasarkan letak geografis perangkat mobile, melalui visualisasi Google Maps, maka aplikasi ini akan mudah digunakan.

**Kata kunci** : rumah sakit, Google Maps, GPS, Jakarta

## I. PENDAHULUAN

Jakarta adalah kota yang cukup luas dan memiliki jalur transportasi yang banyak. masing-masing jalan memiliki angkutan umum tersendiri sehingga terkadang cukup membingungkan untuk mencari suatu tempat yang ingin dituju. hal ini juga berkaitan dengan pencarian rumah sakit yang terdekat, misalkan jika kita sedang mengalami keadaan darurat atau butuh pertolongan karena sakit, maka pencarian rumah sakit yang terdekat mungkin sulit dilakukan jika harus mencari-cari lokasi rumah sakit tersebut.

Teknologi memiliki peran penting di era modernisasi seperti pada saat ini, dimana teknologi telah menjadi bagian yang tidak dapat dipisahkan dalam kehidupan sehari-hari. Perkembangan teknologi saat ini telah merambah ke segala aspek kehidupan sehingga saat ini seolah masyarakat telah dimanjakan oleh adanya alat-alat yang dapat memberikan kemudahan dalam aktifitas sehari-hari. Salah satunya adalah handphone, yang ditandai lahirnya teknologi smartphone.

Dengan bantuan Global Positioning System (GPS) yang berfungsi sebagai penunjuk lokasi, Location Based Service (LBS) yang menyediakan informasi berdasarkan letak geografis perangkat mobile, melalui visualisasi Google Maps, maka aplikasi ini akan mudah digunakan. Aplikasi ini dibangun dengan pemrograman java Android menggunakan software ADT Bundle yang di dalamnya terdapat Eclipse sebagai editor bahasa pemrograman java, ADT sebagai plugin untuk Eclipse, dan SDK untuk kepentingan development aplikasi berbasis Android.

Dari permasalahan tersebut, Aplikasi pencarian rumah sakit terdekat merupakan ide yang cukup baik diimplementasikan untuk mengatasi hal tersebut. Oleh Sebab itu, penulis membuat sebuah aplikasi pencarian

rumah sakit terdekat di jakarta yang dijalankan pada smartphone berbasis sistem operasi android. aplikasi ini dirancang untuk memberikan informasi rumah sakit yang terdekat dan juga rute yang dapat diambil untuk sampai kerumah sakit tujuan pengguna dapat melihat informasi lokasi yang disajikan dalam peta digital berbasis google maps.

Pencarian jalur terpendek dalam aplikasi ini menggunakan algoritma Dijkstra. Algoritma ini digunakan dalam graf berarah dimana setiap titik dihubungkan oleh sisi yang memiliki bobot. Dengan memperhitungkan bobot pada setiap sisi, algoritma ini dapat digunakan untuk menentukan jalur terpendek dari suatu titik ke titik akhir tujuan. Algoritma Dijkstra merupakan suatu algoritma yang disebut juga sebagai single-source shortest path yang digunakan dalam menentukan jalur terpendek dari simpul sumber menuju simpul tujuan berdasarkan bobot pada sisi.

## II. METODOLOGI PENELITIAN

### 2.1. Metode Pengembangan Sistem

Dalam penelitian dibutuhkan sebuah aliran proses (workflow) yang dapat mengarahkan kerangka kerja menjadi lebih terstruktur. Hal tersebut juga berlaku dalam proses rekayasa perangkat lunak. Pada penelitian ini, penulis menggunakan metode waterfall (model air terjun) untuk pengembangan sistem. dimana semua proses Analisis sampai implementasi dan pemeliharaan sistem dibuat berurutan sesuai dengan langkah-langkah dalam metode ini.

#### a. Analisis Kebutuhan Software

Penulis menganalisis perencanaan yang dapat untuk mendukung pembuatan spesifikasi rancangan aplikasi merupakan salah satu

fungsi mendasar dari *action script*. Yang dimaksud adalah bagaimana membantu user untuk masuk kebagian-bagian yang sudah disesuaikan.

#### b. Desain

Dimulai dari merancang konsep yang kemudian dilanjutkan dengan pembuatan desain aplikasi. Proses desain menerjemahkan kebutuhan pengguna dalam sebuah aplikasi dalam sebuah dokumen aplikasi yang dapat diperkirakan kualitasnya sebelum proses *coding* dimulai. Program aplikasi yang dibuat ini adalah sebuah program yang berjalan pada perangkat *smartphone* dengan menggunakan sistem operasi android.

#### c. Testing

Pengujian sistem menggunakan *black box testing*, yang menganggap aplikasi sebagai sebuah kotak hitam dimana user mengabaikan sistem yang diadopsinya. *Black box testing* menitik beratkan pada kesesuaian suatu komponen terhadap spesifikasi.

#### d. Implementasi

Penerapan (*implementasi*) akan dicoba diterapkan dalam sebuah *handphone* / *smartphone* yang berbasis android. Dengan hal ini diharapkan untuk menghasilkan aplikasi pencarian rumah sakit di jakarta yang benar-benar berhasil terwujud sesuai harapan penulis

## 2.2. Metode Pengumpulan Data

### 1. Metode Studi Pustaka

Metode ini merupakan metode pengumpulan data dan referensi dari media cetak, gambar maupun elektronik yang bermanfaat dalam penyusunan.

### 2. Metode Observasi

Metode ini mendapatkan data dengan cara melakukan pengamatan baik terhadap tempat Rumah Sakit yang ditentukan dalam database yang dibuat dan juga aplikasi yang dikembangkan apakah sudah sesuai dengan yang yang diharapkan setelah database diimplementasikan kedalam aplikasi dan dilakukan evaluasi jika ditemukan informasi terbaru atau kekurangan dari aplikasi yang sudah dibuat.

## III. DASAR TEORI

### 3.1 Aplikasi

Aplikasi adalah sebuah media penunjang dalam sebuah object yang memiliki beberapa instruksi yang disusun sedemikian rupa sehingga dapat menghasilkan input dan output[1]. Pendapat lain mengatakan bahwa Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu[2]. Aplikasi adalah suatu program komputer yang dibuat untuk

mengerjakan dan melaksanakan tugas khusus dari pengguna.

### 3.2 Informasi

Informasi adalah data yang telah diproses/diolah ke dalam bentuk yang sangat berarti untuk penerimanya dan merupakan nilai yang sesungguhnya atau dipahami dalam tindakan atau keputusan yang sekarang atau nantinya[3]. Pada literatur yang lain dikatakan bahwa Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi[4].

### 3.3 Android

Android adalah sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi Linux, namun telah dimodifikasi[5]. Android pertama kali dikembangkan oleh sebuah perusahaan bernama Android Inc. Kemudian pada tahun 2005, Google mengakuisisi perusahaan ini sehingga industri ketika itu publik beranggapan akan muncul istilah gPhone dengan langkah Google tersebut.

Pada tahun 2007, Google dan beberapa perusahaan yang tergabung dalam Open Handset Alliance (Intel, Nvidia, Texas Instrument) mengembangkan sistem operasi Android dan resmi menjadi open-source.

Pada tahun 2008 Android SDK 1.0 diluncurkan dan ponsel G1 yang diproduksi oleh HTC menggunakan Sistem operasi tersebut. Pada tahun 2009 versi terbaru dari Sistem Operasi Android diluncurkan mulai dari versi 1.5 (Cupcake), versi 1.6 (Donut) dan versi 2.0 / 2.1 (Eclair). Hal ini didukung dengan lebih dari 20 Gadget yang menggunakan versi tersebut. Android merupakan sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache[6].

### 3.4 Unified Modeling Language (UML)

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek[7]. Dengan kata lain UML berarti bahasa pemodelan standar dan merupakan alat komunikasi yang konsisten dalam mensupport para pengembang saat ini[8].

Dari kedua pendapat tersebut, dapat disimpulkan bahwa UML adalah bahasa pemodelan sistem yang digunakan sebagai alat komunikasi antara pengembang dengan pengguna atau antar pengembang itu sendiri untuk pemrograman berorientasi objek (*Object Oriented Programming*).

### 3.5 Algoritma

Menurut Thomas H. Cormen “Algoritma adalah prosedur komputasi yang mengambil beberapa nilai atau kumpulan nilai sebagai input kemudian di proses sebagai output sehingga algoritma merupakan urutan langkah komputasi yang mengubah input menjadi

output”. Dalam penulisan ini, penulis menggunakan algoritma Dijkstra.[9]

Menurut Somantri dkk mengatakan bahwa “Algoritma Dijkstra dinamai sesuai dengan nama penemunya yaitu Edsger Dijkstra. Algoritma Dijkstra menggunakan prinsip *greedy*, dinamai pada setiap langkah dipilih sisi dengan bobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih”.

Input algoritma ini adalah sebuah graf berarah dan berbobot,  $G$  dan sebuah *source vertex*  $s$  dalam  $G$ .  $V$  adalah himpunan semua simpul dalam graf  $G$ . setiap sisi dari graf ini adalah pasangan *vertices*  $(u,v)$  yang melambangkan gubungan dari *vertex*  $u$  ke *vertex*  $v$ . himpunan semua *edge* disebut  $E$ . *Weights* dari *edges* dihitung dengan fungsi  $w: E \rightarrow [0, \infty)$ ; jadi  $w(u,v)$  adalah jarak *non-negatif* dari *vertex*  $u$  ke *vertex*  $v$ . *cost* dari sebuah *edge* dapat dianggap sebagai jarak antara dua *vertex*, yaitu jumlah jarak semua *edge* dalam *path* tersebut. Untuk sepasang *vertex*  $s$  dan  $t$  dalam  $V$ , algoritma ini menghitung jarak terpendek dari  $s$  ke  $t$ .

#### IV. ANALISIS DAN RANCANGAN SOFTWARE

Pada pembangunan aplikasi ini dibutuhkan spesifikasi minimal *hardware* sebagai berikut :

##### 1. Spesifikasi Perangkat Keras

Untuk membuat aplikasi pencarian rumah sakit berbasis android, spesifikasi perangkat keras (*hardware*) yang digunakan adalah :

- a. Prosesor : Intel Core i5
- b. RAM : 4GB DDR3
- c. Harddisk : 750GB HDD

##### 2. Spesifikasi Perangkat Lunak

Perangkat lunak untuk mendukung membuat aplikasi ini adalah:

- a. Instalasi *Java Development Kits* (JDK)
- b. Instalasi *Android Software Development Kits* (SDK)
- c. Instalasi Eclipse dan ADT Bundle

#### 4.1 Analisis Kebutuhan Software

Penulis mengidentifikasi kebutuhan dengan menganalisis masalah pencarian rumah sakit daerah DKI Jakarta saat orang membutuhkan informasi lokasi tetapi sulit mendapatkan lokasi dengan cepat. Sehingga untuk memudahkan pengguna, dibuatlah aplikasi pencarian rumah sakit dalam bentuk *mobile* berbasis android. Dimana Pengguna dapat menemukan lokasi terdekat menggunakan Aplikasi yang diinstall di Perangkat yang mereka gunakan.

Dalam perancangan sebuah aplikasi dibutuhkan suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. Untuk menggambarkan dan pendokumentasian sebuah sistem, penulis menggunakan UML.

#### 4.1.1. Identifikasi Masalah

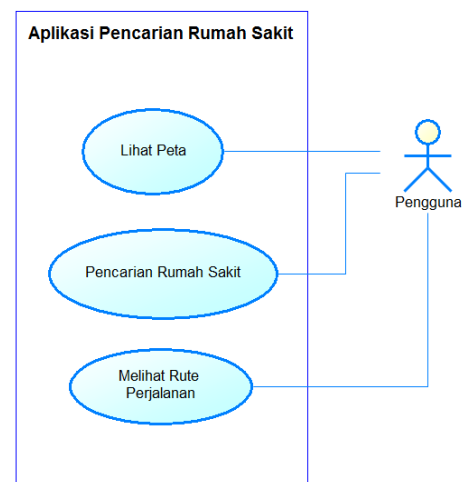
Pada penulisan ini permasalahan yang akan diteliti dalam pembuatan aplikasi pencarian rumah sakit berbasis peta ini adalah penentuan rute terpendek dari lokasi pengguna dan lokasi rumah sakit yang dituju menggunakan algoritma Dijkstra .Dalam aplikasi pencarian lokasi rumah sakit ini akan menampilkan hasil pencarian dalam bentuk peta digital yang dilengkapi dengan panduan arah untuk menemukan lokasi yang dimaksud.

#### 4.1.2. Analisis Kebutuhan

Setelah melakukan identifikasi dan mendefinisikan masalah penulis akan merumuskan kebutuhan untuk permasalahan yang dibutuhkan dalam bentuk UML.

##### 1. Use Case Diagram

*Use case diagram* sangat sesuai untuk menggambarkan siapa saja aktor yang terlibat dan fungsi-fungsi apa saja yang bisa dilakukan dalam sebuah aplikasi. Aktor dan fungsi-fungsi yang dirancang dalam aplikasi ini adalah sebagai berikut:

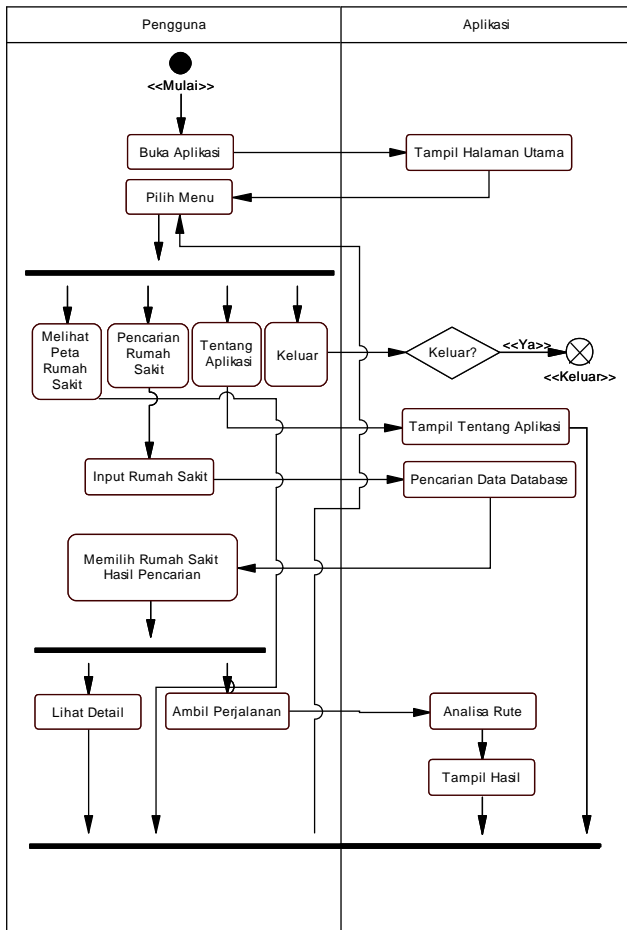


Gambar 1. Use Case Diagram

Pada gambar di atas dapat dilihat pengguna aplikasi sebagai aktor dapat melihat peta aplikasi, dapat melakukan pencarian rumah sakit pada aplikasi, dan dapat melihat perjalanan dari posisi pengguna ke rumah sakit yang dituju.

##### 2. Activity Diagram

Dalam menganalisis proses yang akan terjadi dalam sebuah aplikasi, penulis menggunakan *activity diagram*. *Activity diagram* dapat menggambarkan langkah-langkah yang dilalui untuk melakukan sebuah aksi di dalam aplikasi. Diagram rancangan dari aplikasi ini adalah sebagai berikut:

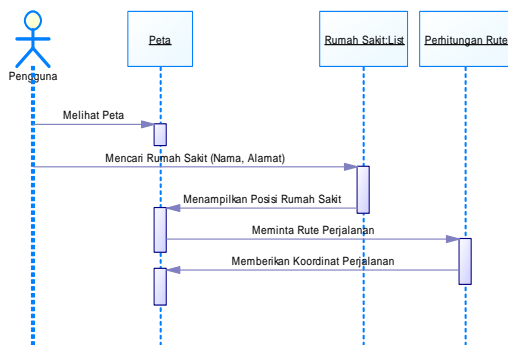


Gambar 2. Activity diagram

Pada gambar di atas dapat dilihat pada saat aplikasi pertama dijalankan, akan menampilkan sebuah halaman utama yaitu berbentuk pilihan menu yang dapat dipilih oleh pengguna aplikasi. Ketika pengguna memilih sebuah menu, aplikasi akan menampilkan hasil *interface* sesuai dengan pilihan pengguna dan pengguna dapat kembali ke menu awal agar dapat memilih menu lainnya.

3. Sequence Diagram

Untuk menjelaskan kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram* penulis menggunakan *sequence diagram*. *Sequence diagram* dari aplikasi ini adalah sebagai berikut:

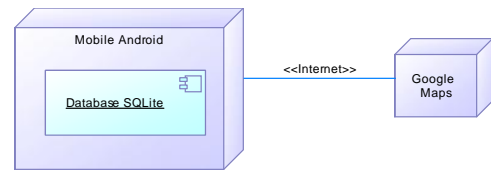


Gambar 3 Sequence diagram

Pada gambar di atas dapat dilihat ketika pengguna sebagai aktor ingin melihat peta maka peta akan langsung menampilkan peta lokasi pengguna. Pada saat pengguna mencari rumah sakit, aplikasi akan mengambil sebuah *list* objek rumah sakit yang berisi hasil pencarian rumah sakit. Pada saat pengguna melakukan *request* rute peta, peta akan mengirim permintaan ke proses perhitungan rute lalu rute akan mengembalikan hasil perjalanan untuk ditampilkan pada peta.

4. Deployment Diagram

*Deployment diagram* dibutuhkan untuk menggambarkan bentuk fisik dari sistem. *Deployment diagram* dari aplikasi ini adalah sebagai berikut:

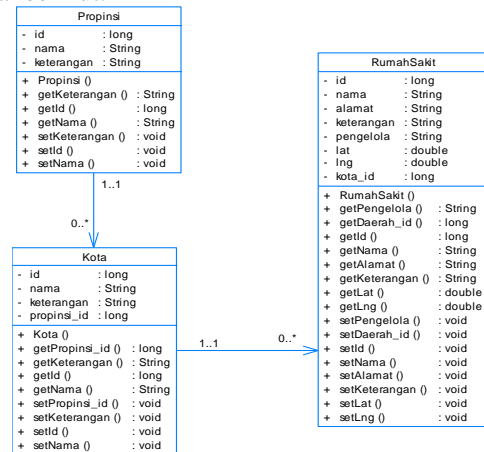


Gambar 4. Deployment diagram

Pada gambar di atas dapat dilihat, aplikasi akan disimpan di *mobile android* dan mengambil database lokal. Aplikasi akan memanfaatkan jaringan internet untuk mengambil *GoogleMap Service* untuk ditampilkan di layar android.

5. Class Diagram

Untuk menggambarkan struktur statis dari kelas dalam sistem dan menggambarkan atribut, operasi, dan hubungan antar kelas. *Class diagram* membantu memvisualisasikan struktur kelas-kelas dari suatu sistem. *Class diagram* dari aplikasi ini adalah sebagai berikut:



Gambar 5. Class diagram

Pada gambar di atas dapat dilihat rancangan *classmodel* yang digunakan untuk menampung data dari database.

4.2 Desain

4.2.1. Rancangan Algoritma

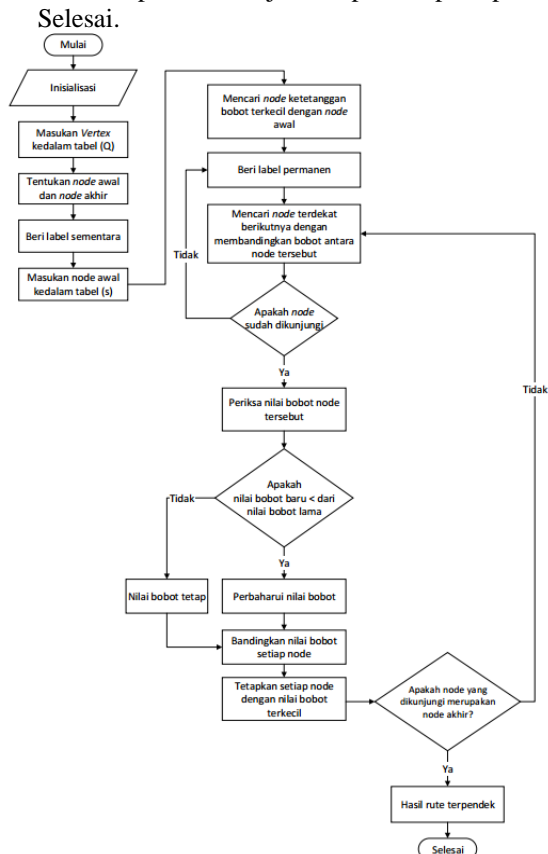
Algoritma Dijkstra adalah salah satu varian dari bentuk algoritma yang sudah populer dalam pencarian persoalan yang berkaitan dengan masalah optimisasi dan bersifat sederhana. Algoritma ini adalah salah satu algoritma yang dapat menyelesaikan permasalahan

untuk mencari lintasan yang terpendek dari vertex a sampai dengan vertex z dalam sebuah graf berbobot dimana bobot tersebut dalam bilangan bulat positif.

Proses algoritma Dijkstra dalam mencari lintasan terpendek dari suatu simpul ke semua pasangan simpul algoritma Dijkstra melalui sejumlah langkah yang menggunakan prinsip *greedy*. Selain matriks ketetanggaan M, algoritma ini menggunakan tabel S = [Si], dengan Si = 1, jika simpul i termasuk ke dalam lintasan terpendek dan sebaliknya Si = 0, jika simpul i tidak termasuk ke dalam lintasan terpendek dan juga tabel D = [di], dengan di = panjang lintasan dari simpul awal a ke simpul i. *Flowchart* proses algoritma Dijkstra dapat dilihat pada Gambar 6.

Pencarian jalur terpendek dengan algoritma Dijkstra pada Gambar 6 dapat dituliskan sebagai berikut:

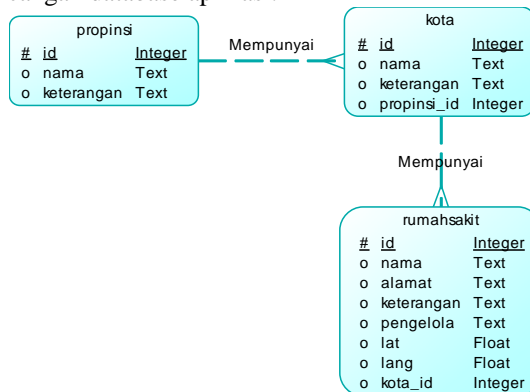
1. Inisialisasi
2. Ambil *node* masukkan dalam himpunan (Q)
3. Tentukan *node* awal dan *node* tujuan
4. Beri label sementara nilai tak hingga pada setiap *node* dan *node* awal diberi label (0)
5. Periksa nilai bobot terkecil pada *node* ketetanggaan dengan *node* awal
6. Hapus label sementara dan beri label permanen pada *node* yang dikunjungi
7. Periksa *node* berikut yang dikunjungi apakah *node* sudah pernah dikunjungi. Bandingkan bobot nilai baru dengan bobot nilai lama
8. Melakukan proses dari langkah 6, sampai didapatkan hasil proses jalur terpendek pada *node* akhir
9. Menampilkan hasil jalur terpendek pada peta



Gambar 6. *Flowchart* Algoritma Dijkstra

#### 4.2.2. Database

Dalam perancangan aplikasi ini, penulis menggunakan database untuk menyimpan lokasi rumah sakit. Database yang digunakan adalah database SQLite. SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat ACID-compliant (atomic, konsistensi basisdata, isolasi, dan durabilitas) dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. Berikut adalah rancangan database aplikasi:



Gambar 7. Rancangan Database

Pada gambar di atas dapat dilihat, aplikasi pencarian rumah sakit ini dibangun dengan satu database yang berisi tiga table yang saling berelasi. Propinsi dapat tidak mempunyai kota atau mempunyai lebih dari satu kota, sedangkan kota harus mempunyai satu propinsi. Kota dapat tidak mempunyai rumah sakit atau mempunyai lebih dari satu rumah sakit, sedangkan rumah sakit harus mempunyai satu kota.

#### 4.2.3. Software Architecture

Pada dasarnya aplikasi ini terdiri dari 2 bagian, yaitu aplikasi pencarian rumah sakit dan *library map direction* yang digunakan untuk penentuan rute terpendek menggunakan algoritma Dijkstra. Aplikasi pencarian rumah sakit merupakan halaman antar muka yang memungkinkan pengguna untuk meng-input nama atau alamat rumah sakit untuk mencari rumah sakit yang dituju menggunakan layar sentuh. Aplikasi akan menangkap inputan dari pengguna yang kemudian mengeksekusi *query* untuk mendapatkan hasil yang sesuai. Sedangkan *library map direction* merupakan sebuah *class library* yang menyediakan fungsi untuk mendapatkan rute terpendek dari posisi pengguna ke posisi rumah sakit yang dipilih oleh pengguna pada halaman antar muka peta. *Pseudocode* dari proses pencarian rumah sakit dan penggunaan *library map direction* adalah sebagai berikut:

```

Function ButtonCari onClick:
    Kota = SpinnerKota.GetValue
    Keyword = TextviewRumahSakit.GetValue
    Query = "SELECT * FROM rumahsakit
    WHERE kota=Kota AND nama
    LIKE '%Keyword%' OR alamat
    LIKE '%Keyword%'"
    Hasil = Execute(Query)
    Tampilkan(Hasil)
Function onRumahSakitSelected
    
```

```
LokasiAwal = LonLat(CurrentLocation.Lon,
CurrentLocation.Lat)
LokasiTujuan = LonLat(SelectedLocation.Lon, SelectedLocation.Lat)
Points = Direction.GetDirection(LokasiAwal,
LokasiTujuan)
For Points as Point
Map.DrawLine(Point)
```

**4.2.4. User Interface**

Tampilan antar muka aplikasi dibuat sederhana dan tidak banyak konten yang tidak diperlukan. Hal ini bertujuan agar pengguna dapat dengan mudah menggunakan aplikasi ini. Aplikasi terdiri dari halaman menu utama, halaman peta, halaman pencarian, halaman tentang aplikasi dan sebuah jendela popup yang pada halaman peta untuk menampilkan pilihan untuk pengguna apakah ingin mengambil rute atau melihat detail rumah sakit yang dipilih. Aplikasi tidak menyediakan halaman untuk mengelola lokasi atau koordinat rumah sakit untuk ditampilkan dalam bentuk *marker* di peta karena aplikasi sudah mempersiapkan lokasi rumah sakit selengkap mungkin. Berikut adalah tampilan antar muka aplikasi:



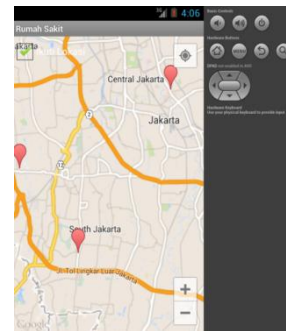
**Gambar 8. Tampilan Menu Awal**

Pada Tampilan awal aplikasi, ada empat menu yang dapat dipilih yang masing-masing menu diwakili oleh tombol pilihan berupa ikon / gambar yang menunjukkan fungsi dari setiap menu. Ada menu Peta digital, Pencarian, Tentang / Informasi Aplikasi dan Keluar, berikut dijelaskan fungsi dari masing-masing menu:

**1. Menu Peta Digital**

Diwakili oleh *icon* bergambar Peta. Jika dipilih maka akan menampilkan peta digital Google lengkap dengan atribut-atribut pendukungnya, juga akan menampilkan lokasi pengguna secara realtime (keadaan saat berlangsung). Kemudian dalam peta digital juga akan tampil lokasi-lokasi Rumah sakit yang terdekat dengan Pengguna saat tersebut. Agar

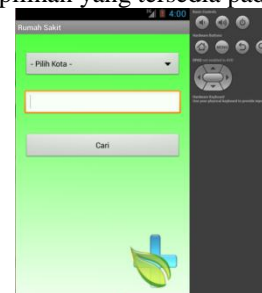
peta digital dapat menunjukkan lokasi pengguna, maka harus dipastikan hubungan/koneksi internet dalam kondisi aktif dan layanan lokasi berbasis GPS (*Global Positioning System*) di perangkat pengguna.



**Gambar 9. Tampilan Peta**

**2. Menu Pencarian**

Diwakili oleh *icon* kaca pembesar. Berfungsi untuk memilih kota yang tersedia di Jakarta, yaitu Jakarta Barat, Timur, Pusat, Selatan dan Utara. Pengguna bisa memilih salah satu kota yang terdekat dan memasukkan kata kunci untuk mencari Rumah Sakit yang tersedia. Jika kata kunci dikosongkan kemudian langsung memilih tombol cari, maka akan tampil semua pilihan yang tersedia pada kota tersebut.



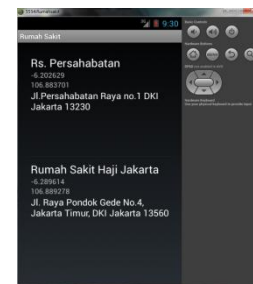
**Gambar 10. Tampilan Form Pencarian**

**3. Menu Tentang / Informasi**

Diwakili oleh *icon* i dalam bola hijau. Menampilkan informasi mengenai Aplikasi, Versi dan Hak Cipta.

**4. Menu Keluar**

Diwakili oleh *icon* X, berfungsi untuk pilahan keluar dari Aplikasi atau tidak. Jika memilih keluar maka Aplikasi akan tertutup, tetapi jika tidak maka Aplikasi akan tetap terbuka dan objek terakhir yang tampil tetap ditampilkan.



**Gambar 11. Tampilan Hasil Pencarian**

**4.3 Implementasi**

Aplikasi pencarian rumah sakit dibangun menggunakan *Android Development Tools (ADT) v22.3.0*. Aplikasi dapat berjalan pada *hardware* apapun yang menggunakan sistem operasi android. Spesifikasi sistem operasi android untuk dapat beroperasi minimum pada android 2.2 (Froyo) dan maksimum pada android 4.4.2 (Kitkat), sedangkan proses *compiling* dilakukan menggunakan android 4.4.2.

**4.4 Testing**

Untuk mengetahui tingkat keakurasian hasil yang diperoleh dari aplikasi dan mengurangi *bug* yang mungkin muncul pada aplikasi perlu dilakukan uji coba. Uji coba dilakukana menggunakan pengujian *white box* dan *black box*.

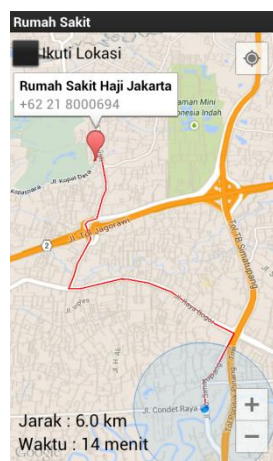
**4.4.1 Pengujian white box**

Pengujian *white box* yang dilakukan adalah untuk menguji struktur kontrol program. *Test case* dilakukan untuk memastikan bahwa semua statemen pada program yang telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji. Pengujian yang akan dilakukan dapat dilihat pada tabel 1.

**Tabel 1 Pengujian White box**

**4.4.2 Pengujian black box**

Pengujian ini dilakukan untuk menguji fungsi dari algoritma Dijkstra yang digunakan. Pengujian yang dilakukan menggunakan alur dari *activity diagram* pada Gambar IV-2. Aktivitas yang diuji adalah pada saat pengguna mengambil perjalanan lalu aplikasi menganalisis rute dan menampilkan hasil.



**Gambar 12. Rute pengguna ke RS Haji Jakarta**

Gambar 12. Pengujian pertama ke lokasi RS Haji Jakarta dari titik awal dengan koordinat (-6.302375,106.8561737) dan titik akhir dengan koordinat (-6.289613976571911, 106.88927806913853). Perhitungan rute berhasil

dilakukan dan menghasilkan jarak 6 km dengan waktu 14 menit.

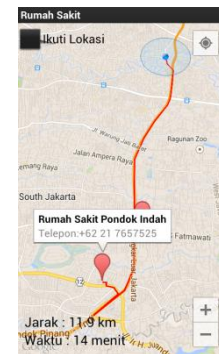


**Gambar 13. Rute pengguna ke RSUP Fatmawati**

Gambar 13. Pengujian kedua ke lokasi RSUP Fatmawati dari titik awal dengan koordinat (-6.302375,106.8561737) dan titik akhir dengan koordinat (-6.294330889508361,

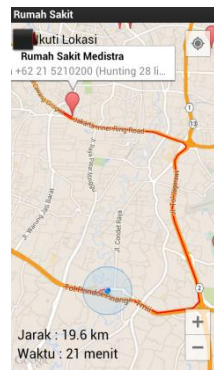
No	Input	Hasil
1	Memilih tombol Peta	Peta Digital Terbuka dengan lokasi Pengguna terlihat
2	Memilih tombol Cari	Tampil Combo Box memilih Wilayah dan Text filed kata Kunci
3	Memilih tombol Tentang	Tampil Informasi Mengenai Aplikasi
4	Memilih tombol Keluar	Tampil Pilihan Ya untuk Keluar atau Tidak
5	Kata kunci Pencarian	Jika input sesuai maka akan tampil Data alamat yang tersedia
6	Memilih hasil pencarian	Lokasi Rumah sakit ditampilkan di Peta Digital

106.79650615900755). Perhitungan rute berhasil dilakukan dan menghasilkan jarak 7.6 km dengan waktu 10 menit.



**Gambar 14. Rute pengguna ke RS Pondok Indah**

Pengujian ketiga ke lokasi RS Pondok Indah dari titik awal dengan koordinat (-6.302375,106.8561737) dan titik akhir dengan koordinat (-6.284052869870353, 106.78143583238125). Perhitungan rute berhasil dilakukan dan menghasilkan jarak 11.9 km dengan waktu 14 menit.



**Gambar 15. Rute pengguna ke RS Medistra**

Pengujian keempat ke lokasi RS Medistra dari titik awal dengan koordinat (-6.3028451, 106.8547695) dan titik akhir dengan koordinat (-6.239915834682788, 106.83383211493492). Perhitungan rute berhasil dilakukan dan menghasilkan jarak 19.6 km dengan waktu 21 menit.

Dari hasil pengujian di atas, dapat dilihat pencarian rute sudah diimplementasikan dan dapat berjalan dengan baik.

## V. KESIMPULAN

Berdasarkan hasil implementasi aplikasi pencarian rumah sakit dan pembahasan pada bab-bab sebelumnya, maka kesimpulan yang diperoleh adalah:

1. Aplikasi yang dibuat sudah dapat memenuhi kebutuhan untuk melakukan proses pencarian rumah sakit dari *mobile device* dengan memberikan posisi yang cukup akurat.
2. Aplikasi pencarian rumah sakit dapat menampilkan rute jalan ke rumah sakit dari dengan

posisi user serta jaraknya sekarang ini dalam bentuk peta.

3. Dari hasil pengujian yang dilakukan diketahui bahwa fungsi-fungsi pada aplikasi pencarian rumah sakit dapat berjalan dengan benar dan sesuai kriteria dalam pembuatan.

## VI. DAFTAR PUSTAKA

- [1] Jogiyanto HM, 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- [2] Tim Redaksi Kamus Besar Bahasa Indonesia, 2009. *Kamus Besar Bahasa Indonesia*. Jakarta : Balai Pustaka.
- [3] Davis, Gordon B, 2002. *Kerangka Dasar: Sistem Informasi Manajemen, Bagian I Pengantar. Seri Manajemen No. 90-A. Cetakan Kedua Belas*, Jakarta: PT. Pustaka Binawan Pressindo.
- [4] Kusrini dan Andri Koniyo, 2007. *Tuntunan Praktis Membangun Sistem Informasi Akuntansi dengan Visual Basic & Microsoft SQL Server*. Yogyakarta: Penerbit Andi.
- [5] Suprianto, Dodit & Rini Agustina, 2012. *Pemrograman Aplikasi Android: step by step membuat aplikasi android untuk Smartphone dan Tablet*. Yogyakarta: MediaKom.
- [6] "Android Overview". Open Handset Alliance. Diakses 2014-07-15
- [7] Shalahuddin, Muhammad & Rosa Ariani S, 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- [8] Widodo, Prabowo Pudji & Herlawati, 2011. *Menggunakan UML*. Bandung: Informatika.
- [9] Cormen, Thomas H , 2009. *Introduction to Algorithms*, 3<sup>rd</sup> Edition. MIT Press Publisher