PEMBUATAN APLIKASI PETA RUTE BUS TRANS JOGJA BERBASIS *MOBILE GIS* MENGGUNAKAN *SMARTPHONE* ANDROID

Danang Budi Susetyo, Andri Suprayogi, S.T, M.T *, M. Awaluddin, S.T, M.T *

Program Studi Teknik Geodesi Fakultas Teknik, Unversitas Diponegoro

Jl. Prof. Sudarto SH, Tembalang Semarang Telp. (024) 76480785, 76480788

ABSTRAK

Sebagai salah satu kota pariwisata, ada beberapa komponen yang harus dipenuhi oleh Yogyakarta untuk mempermudah wisatawan, sehingga mereka dapat merasa nyaman untuk menikmati suguhan wisata di Yogyakarta. Salah satu komponen itu adalah transportasi. Transportasi yang baik meningkatkan mobilitas wisatawan yang menjadikan mereka tidak perlu repot untuk mengakses seluruh tempat wisata di Yogyakarta.

Sejak Maret 2008, Dinas Perhubungan, Komunikasi, dan Informatika Daerah Istimewa Yogyakarta sudah memberlakukan moda transportasi bus Trans Jogja. Trans Jogja sudah beroperasi dengan cukup baik dan memberikan kemudahan kepada masyarakat Yogyakarta sendiri maupun wisatawan. Namun ketersediaan informasi mengenai pedoman penggunaan Trans Jogja yang mudah diakses masyarakat masih sangat minim. Masih sangat sedikit informasi rute dan *shelter* yang tersedia baik secara manual maupun internet.

Peningkatan popularitas *smartphone* Android merupakan opsi menarik untuk menjadikannya sebagai *platform* informasi bus Trans Jogja. Aplikasi berbasis *mobile* GIS adalah salah satu pilihan yang banyak digunakan dalam pembuatan aplikasi Android. Dengan teknologi *Global Positioning System* (GPS) yang dikombinasikan dengan *Location Based Service* (LBS) melalui visualisasi pada Google Map, kita dapat mendapatkan informasi berdasarkan letak geografis pada perangkat *mobile*. Aplikasi ini dikembangkan menggunakan *Framework Android SDK*, bahasa pemrograman java dan PHP, *MySQL* sebagai basis data, dan Google Map.

Kata Kunci : Trans Jogja, Aplikasi, GIS, *Mobile GIS*, GPS, LBS, Android

ABSTRACT

As one of the tourist city, there are several components that must be fullfilled by Yogyakarta to facilitate the tourist to make them have a pleasant travel in Yogyakarta. One of the component is transportation. A Good transportation improve mobility that makes them easy to access all of recreation place in Yogyakarta.

Since March 2008, The Transportation, Communication, and Information Department of Yogyakarta impose Trans Jogja bus transportation. Trans Jogja has operated quite well and provide facilities to the people of Yogyakarta and the tourists. However, the availability of information on the guidelines for the use of Trans Jogja that easily accessible for the people in Yogyakarta is still rare. There is only little information about the routes and shelters that available either manually or internet.

The increasing popularity of Android smartphones is an interesting option to make it as an information platform Trans Jogja bus. Mobile GIS-based applications is one option that is widely used in the manufacture of Android applications. With the technology of Global Positioning System (GPS) combined with Location Based Service (LBS) through visualization on a Google Map, we can obtain geographic information on mobile devices.

***Keywords**: Trans Jogja, Application, GIS, Mobile GIS, GPS, LBS, Android*

I. PENDAHULUAN

Yogyakarta menjadi salah satu andalan kota pariwisata di Indonesia. Salah satu prasarana yang menjadi unsur penting dalam peningkatan kualitas sebuah kota terutama dalam pemenuhan syarat kota wisata yang baik adalah sarana transportasi. Saat ini, kota Yogyakarta mempunyai sebuah moda transportasi yang cukup ideal karena cukup murah dan mencakup tempat-tempat strategis di Yogyakarta, yaitu bus Trans Jogja.

Saat ini, Trans Jogja sudah cukup populer sebagai alternatif transportasi umum utama di Yogyakarta. Manajemen pelayanannya pun sudah cukup baik, hampir seperti busway Trans Jakarta. Namun, informasi mengenai peta rute yang tersedia masih sedikit, baik di media *website* maupun dalam aplikasi *mobile*. Dari permasalahan yang sudah dipaparkan di atas, peneliti ingin membangun sebuah aplikasi berbasis sistem operasi Android yang dapat menjadi salah satu media untuk mengakses informasi mengenai rute Trans Jogja per *shelter* dan koridor serta untuk mengetahui lokasi *shelter* terdekat dari suatu posisi dengan memanfaatkan kombinasi metode *Location Based Service*, pemanfaatan GPS, dan internet. Android dipilih karena saat ini kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis Android.

II. TINJAUAN PUSTAKA

II.1 Bus Trans Jogja

Trans Jogja adalah sebuah sistem transportasi bus cepat, murah dan ber-AC di seputar Kota Yogyakarta. Trans Jogja merupakan salah satu bagian dari program penerapan Bus Rapid Transit (BRT) yang dicanangkan Departemen Perhubungan. Sistem ini mulai dioperasikan pada awal bulan Maret 2008 oleh Dinas Perhubungan Pemerintah Provinsi DIY.

Perencanaan Trans Jogja cukup mendesak karena sistem transportasi Yogyakarta dan sekitarnya sebelumnya dinilai tidak efisien. Kondisi kinerja dan transportasi umum perkotaan semakin parah, rata-rata *Load Factor* (LF) pada tahun 2004 hanya 27,22%. Selain itu, pertumbuhan kendaraan pribadi semakin tidak terkendali, sementara konstruksi jalan terbatas (dishub-diy.net). Oleh karena itulah, Trans Jogja menjadi moda transportasi yang dirasa sangat perlu.

II.2 *Mobile GIS*

Teknologi GIS (*Geographic Information System*) mengalami perkembangan yang sangat pesat. Teknologi itu diantaranya adalah *mobile* GIS dimana GIS yang tadinya hanya digunakan di dalam lingkungan kantor menjadi semakin fleksibel dan mampu digunakan di luar kantor secara *mobile*. *Mobile* GIS merupakan sebuah integrasi cara kerja perangkat lunak/ keras untuk pengaksesan data dan layanan geospasial melalui perangkat bergerak via jaringan kabel atau nirkabel.

Secara umum, *mobile* GIS diimplementasikan pada dua area aplikasi utama yaitu Layanan Berbasis Lokasi (*Location Based Service*) dan GIS untuk kegiatan lapangan (*Field Based GIS*). Berikut ini hal-hal yang berkenaan dengan aplikasi *mobile* GIS (Riyanto, 2010):

1. Diimplementasikan pada perangkat bergerak dengan keterbatasan ruang penyimpanan, memori, dan resolusi.
2. Dapat diimplementasikan secara mandiri (*stand alone*) dengan menyimpan data dalam perangkat bergerak (untuk aplikasi sederhana), atau disesuaikan dengan arsitektur *server*-nya.
3. Kemampuan aplikasi *mobile* GIS, seperti:
   a. Menampilkan atau melakukan navigasi.
   b. Mengidentifikasi.
   c. Pencarian atau *query*.
   d. Memodifikasi nilai atribut.
   e. Pemberian tanda atau *redline*.
   f. Memodifikasi geometri.
   g. Mengintegrasikan dengan data kantor.
4. Terdapat dua jenis data, yaitu koleksi data (*data collection*) dan navigasi (*navigation*). Adapun kelebihan sistem koleksi data dengan mobile GIS adalah sebagai berikut:
   a. Dapat diintegrasikan dengan perangkat GPS, *rangefinder*, dan kamera digital.
   b. Sistem koleksi data sangat efisien, yaitu hanya dengan *point* dan *click*.
   c. Data spasial dikelola dalam dataset referensi.

II.3 *Assisted Global Positioning System* (A-GPS)

A-GPS (*Assisted* GPS) adalah sebuah teknologi yang menggunakan sebuah *server* bantu untuk mempercepat waktu yang diperlukan dalam menentukan sebuah posisi menggunakan perangkat GPS, yaitu dengan memberi tahu unit GPS satelit mana saja yang sebaiknya layak untuk langsung didengarkan daripada

harus mendeteksi seluruh satelit yang ada sehingga dapat mengurangi waktu yang dibutuhkan secara signifikan untuk menentukan posisi saat ini yang juga disebut sebagai *Time to First Fix* (TTFF).

Dalam mode A-GPS, perangkat penerima selular cukup mengambil sekilas dari sinyal satelit dan mentransmisikannya ke sebuah *tower* selular yang meneruskannya ke sebuah *server* bantu yang melakukan kalkulasi yang diperlukan untuk menghitung sebuah *position fix*. *Server* itu kemudian mengirim kembali hasil perhitungan posisi ke perangkat penerima selular tersebut (Riyanto, 2010).

II.4 *Location Based Service* (LBS)

LBS (*Location Based Services*) sebenarnya adalah salah satu nilai tambah dari layanan selular GSM. LBS bukanlah sistem, tetapi merupakan layanan yang menggunakan sistem tambahan penunjang sistem GSM. Sistem ini menggunakan prinsip dasar triangulasi. Jadi, prinsipnya tidak jauh beda dengan sistem GPS, hanya saja fungsi satelit digantikan oleh BTS.

Perbedaan antara LBS dan GPS adalah pemroses posisi. Pada peralatan GPS, pengguna lah yang mengukur dan mengolah suatu posisi. Sistem *back-end* satelit hanya memberikan info posisi satelit, kecepatan, dan waktu. Sedangkan pada sistem LBS, yang melakukan kalkulasi posisi adalah *back-end* sistem GSM, bukan *handset* pengguna. Informasi posisi akan dicatat oleh BTS yang terdekat kemudian data dikirim ke sistem LBS untuk dikalkulasi dan dikirimkan ke *channel* yang dituju.

III. METODOLOGI PENELITIAN

III.1 Alat dan Bahan

1. Perangkat keras (*hardware*), yang terdiri dari:
   a. Seperangkat laptop dengan spesifikasi *Processor Intel(R) Core(TM)2 Duo* 2.00 GHz, *Hardisk* 250 GB, RAM 3.00GB
   b. GPS *Handheld*
   c. Kamera digital
   d. Handphone Android (*Samsung Galaxy Gio GingerBread* 2.3.4 *version*)
2. Perangkat lunak (*software*), yang terdiri dari:
   a. XAMPP 1.7.4
   b. *Java Development Kit (JDK)*
   c. Eclipse Helios
   d. *Android Development Tools Rev 19*
   e. *Android Software Development Kit (SDK) 11*
   f. Notepad ++
   g. Mozilla Firefox
   h. MS Visio 2007
3. Data yang digunakan terdiri dari:
   a. Data koordinat GPS tiap *shelter*, diperoleh dari pengukuran GPS *handheld*
   b. Data rute bus Trans Jogja, diperoleh dari Dinas Perhubungan, Komunikasi, dan Informatika Daerah Istimewa Yogyakarta (DIY)

III.2 Pelaksanaan

1. Instalasi Program

Sebelum memulai melaksanakan penelitian ini, terlebih dahulu harus disiapkan *software-software* yang dibutuhkan seperti yang sudah disebutkan di atas dengan urutan sebagai berikut:

1. Instalasi *Java Development Kit* (*JDK*). Android adalah aplikasi yang dikembangkan dengan basis bahasa pemrograman java, sehingga sebelum melakukan *coding* aplikasi, komputer harus sudah terinstal program java.
2. Membuka Eclipse Helios sebagai IDE (*Integrated Development Environtment*) yang akan digunakan dalam *coding* aplikasi Android.
3. Instalasi ADT atau *plugins* Eclipse. *Plugins* ini yang membuat Android SDK yang sudah kita miliki dapat dihubungkan dengan IDE Eclipse.

4. Instalasi Android SDK. SDK (*Software Development Kit*) ini digunakan sebagai alat bantu dan API dalam mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman java.
5. Membuat AVD (*Android Virtual Device*) yang merupakan emulator untuk menjalankan program aplikasi.

2. Perancangan Program

Perancangan program pada tahap ini adalah menentukan hasil akhir dari aplikasi yang akan dibuat, yaitu perancangan sistem aplikasi, *database*, *user interface*, hingga *activity* yang dapat diakses oleh *user*.

3. Pembuatan *Database*

Pembuatan *database* dilakukan pada *software* XAMPP 1.7.4 setelah sebelumnya dilakukan perancangan komponen yang akan dimasukkan di dalamnya.

4. Koneksi *Database* dengan Aplikasi

Setelah *database* selesai dibuat, kita harus membuat *coding* PHP untuk mengkoneksikan *database* tersebut dengan aplikasi. Pembuatan PHP dilakukan dengan menggunakan *Notepad* ++.

5. Integrasi dengan Google Map

Dalam *mobile GIS* menggunakan *operating system* Android, kita perlu mendaftarkan sebuah kunci yang dikenal dengan API *Key* agar kita dapat mengakses Google Map dengan *coding* aplikasi. Kunci ini didapat berdasarkan *generate* otomatis dari SDK Android yang diinstal.

6. *Coding* Aplikasi

Coding aplikasi dimulai dengan membuat project menggunakan bahasa pemrograman java pada Eclipse. Langkah pertama yang harus dilakukan adalah membuat *new android project*, dimana akan terbentuk sebuah *project.* Dalam *project* tersebut terdapat *activity* yang terletak pada *package* yang dibuat pertama kali. Setiap *activity* merupakan sesuatu yang ditujukan untuk meng-*handle* berbagai macam hal yang dilakukan oleh *user*. Semua kelas *activity* harus sesuai dengan `<activity>` yang dideklarasikan dalam suatu paket AndroidManifest.xml. Untuk tampilan *User Interface* (UI) dapat diatur pada file .xml.

7. Uji Coba dengan Emulator

Setelah selesai membuat *project*, sebelum diinstal di *device* program sebaiknya dicoba terlebih dahulu di emulator. Emulator ini adalah *Android Virtual Machine* (AVD) yang merupakan tempat untuk menjalankan aplikasi di komputer. AVD berjalan di *Virtual Machine*.

8. Implementasi

Implementasi pada tahap ini berupa aplikasi *server* ke *hosting*, instalasi ke *handset* Android, dan analisis kesesuaian trayek.

IV. HASIL DAN PEMBAHASAN

IV.1 Tampilan Aplikasi

Berikut ini adalah hasil tampilan aplikasi GeoTrans pada handphone Samsung Galaxy Gio dengan versi Android 2.3.4 (*GingerBread*):

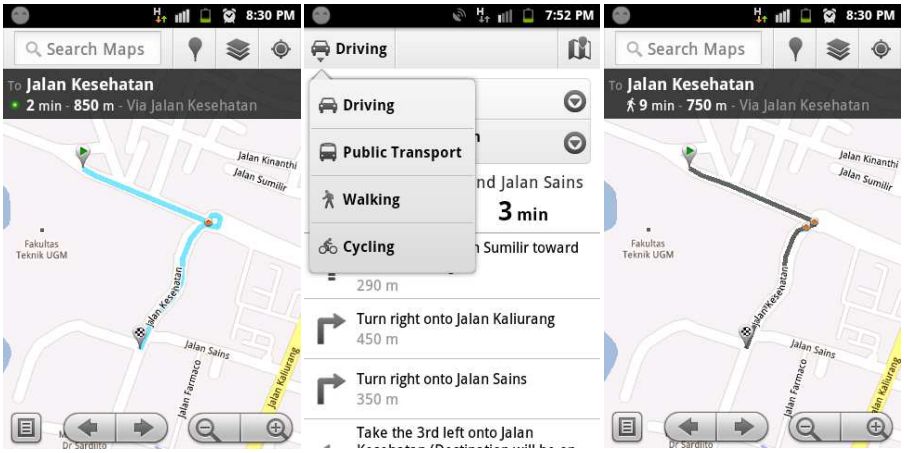
1. Tampilan Menu *Nearest*

Hasil yang ditampilkan adalah *listview shelter-shelter* yang terdekat dari lokasi *user* berdasarkan koordinat yang sudah didapatkan dari *request* ke *server* API GeoTrans. Urutan *shelter* didapatkan dari kalkulasi jarak berdasarkan koordinat pengguna dan koordinat *shelter*. Pada proses ini, aplikasi akan mencari koordinat berdasarkan nilai LAC dan CID dari BTS-BTS terdekat, kemudian melakukan *request* koordinat ke *server* API Google. Pemanfaatan GPS pada aplikasi ini adalah alternatif jika aplikasi tidak menerima nilai koordinat latitude dan longitude dari *server* Google (Trisnawati, 2012).

Gambar 4.1 Tampilan Menu *Nearest*

2. Tampilan *Direction*

Direction merupakan fungsi lanjutan dari menu *Nearest*, dimana Google Map dapat menunjukkan arah dari lokasi pengguna menuju *shelter* yang ingin dituju.



Gambar 4.2 Tampilan Menu *Direction*

3. Tampilan Menu *Destination*

Menu ini merupakan menu yang membantu *user* untuk menentukan *shelter* yang harus dituju. Menu ini berisi *shelter-shelter* yang berada di dekat tempat-tempat penting di Yogyakarta dan dilengkapi *direction* dari *shelter* ke lokasi yang dimaksud. Prinsipnya hampir sama dengan menu *Nearest*.

Gambar 4.3 Tampilan Menu *Destination*

4. Tampilan Menu *Input*

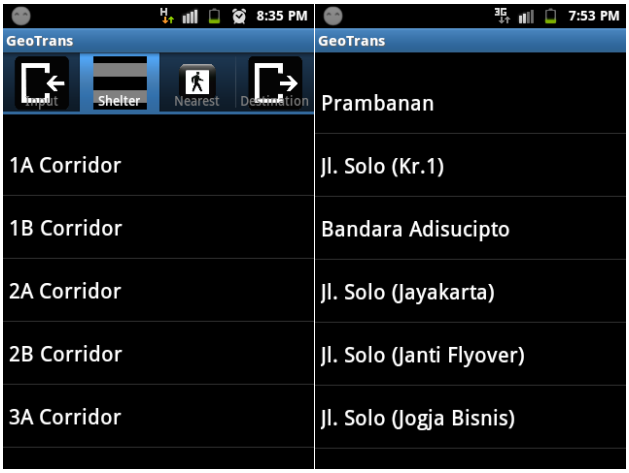
Menu ini menyediakan fungsi untuk mengetahui *shelter-shelter* yang dilewati dari jalur yang menghubungkan *shelter* awal dan *shelter* tujuan. Melalui menu tersebut pengguna dapat mengetahui koridor bus

berapa yang harus dia gunakan dan di mana saja dia harus transit untuk mencapai tempat tujuannya. Menu ini juga dilengkapi dengan *autocomplete*, sehingga ketika kita mengetik beberapa kata saja maka aplikasi akan mengeluarkan opsi-opsi yang berhubungan dengan kata yang diketikkan.

Gambar 4.4 Tampilan Menu *Input*

5. Tampilan Menu *Shelter*

Menu ini lebih berfungsi untuk mengetahui keseluruhan jalur dari rute bus Trans Jogja. Tampilan pertama dari menu ini adalah *listview* enam koridor yang tersedia, yaitu 1A, 1B, 2A, 2B, 3A, dan 3B. Jika salah satu koridor dalam *listview* itu kita klik maka aplikasi akan meloncat ke *listview shelter-shelter* yang dilewati oleh koridor tersebut.



Gambar 4.5 Tampilan Menu *Shelter*

IV.2 Analisis

Ada lima parameter yang digunakan dalam analisis, yaitu spesifikasi sistem operasi Android, tipe *device* yang digunakan, jaringan untuk mengakses internet, analisis algoritma Dijkstra, dan analisis *haversine* formula.

1. Spesifikasi sistem operasi Android

Instalasi aplikasi GeoTrans berhasil pada Android versi *GingerBread* dan *Ice Cream Sandwich*, namun gagal diinstal pada versi *Froyo*. Hal itu dikarenakan sejak awal *project* aplikasi sudah disetting pada versi minimal 2.3.3, sehingga aplikasi tidak akan bisa terinstal pada Android versi di bawah *GingerBread*.

2. Tipe *device* yang digunakan

Tipe *device* di sini adalah merk handphone Android yang digunakan untuk mengetes aplikasi. Pada penelitian ini, aplikasi dicoba pada handphone merk Samsung dengan berbagai tipe (Galaxy Young, Galaxy Pocket, Galaxy Mini, Galaxy Mini Dua, Galaxy Gio, Galaxy Ace, Galaxy Ace Plus, Galaxy Wonder, dan Galaxy SIII) serta LG Nexus A7. Aplikasi dapat berjalan pada seluruh percobaan.

3. Jaringan untuk mengakses internet

Parameter ini digunakan untuk menguji kecepatan akses aplikasi kepada *database* yang dikirim dari *server*. Tiga tipe koneksi yang digunakan adalah WLAN, 2G, dan 3G. Berikut merupakan hasil kecepatan akses untuk fungsi membuka aplikasi untuk pertama kali, pencarian *shelter* terdekat, dan *direction* pada Google Map.

a. Membuka aplikasi untuk pertama kali

Tabel 4.1 Kecepatan Akses untuk Membuka Aplikasi Pertama Kali

Koneksi	Waktu yang diperlukan (detik)					Rata-rata
	1	2	3	4	5	
WLAN	1,73”	1,74”	1,79”	1,90”	1,88”	1,808”
2G	5,11”	6,31”	6,30”	7,77”	7,40”	6,578”
3G atau di atasnya	1,60”	1,92”	1,80”	1,77”	1,65”	1,748”

b. Pencarian *shelter* terdekat

Tabel 4.2 Kecepatan Akses untuk Pencarian *Shelter* Terdekat

Koneksi	Waktu yang diperlukan (detik)					Rata-rata
	1	2	3	4	5	
WLAN	2,24”	1,86”	1,95”	1,95”	1,91”	1,982”
2G	4,58”	3,88”	5,40”	6,00”	6,04”	5,180”
3G atau di atasnya	1,67”	1,66”	1,65”	1,59”	1,86”	1,686”

c. *Direction* pada Google Map

Tabel 4.3 Kecepatan Akses untuk *Direction* pada Google Map

Koneksi	Waktu yang diperlukan (detik)					Rata-rata
	1	2	3	4	5	
WLAN	2,80”	2,64”	2,69”	2,36”	2,56”	2,610”
2G	7,25””	4,65”	6,35”	7,34”	7,34”	6,586”
3G atau di atasnya	3,98”	3,63”	3,55”	3,85”	3,88”	3,778”

Dari hasil pengujian menu aplikasi yang harus menggunakan internet, dapat disimpulkan bahwa jaringan 2G adalah yang membutuhkan waktu paling lama untuk mengakses *database* dari *server*. Hal itu dikarenakan kecepatan akses jaringan 2G (EDGE) secara teori sekitar 384 kpbs, lebih lambat daripada jaringan 3G atau di atasnya (WCDMA atau HSDPA) yang secara teori kecepatannya bisa mencapai 480 kbps.

Pada jaringan WLAN ada perbedaan hasil pada menu *direction* dengan dua menu lainnya. Pada menu *direction* WLAN lebih cepat daripada 3G ke atas, sedangkan pada dua menu lainnya 3G ke atas lebih cepat daripada WLAN. Hal ini dikarenakan kecepatan WLAN setiap waktunya bisa berbeda-beda, tergantung dari banyaknya pengguna pada jaringan tersebut.

4. Analisis Algoritma Dijkstra

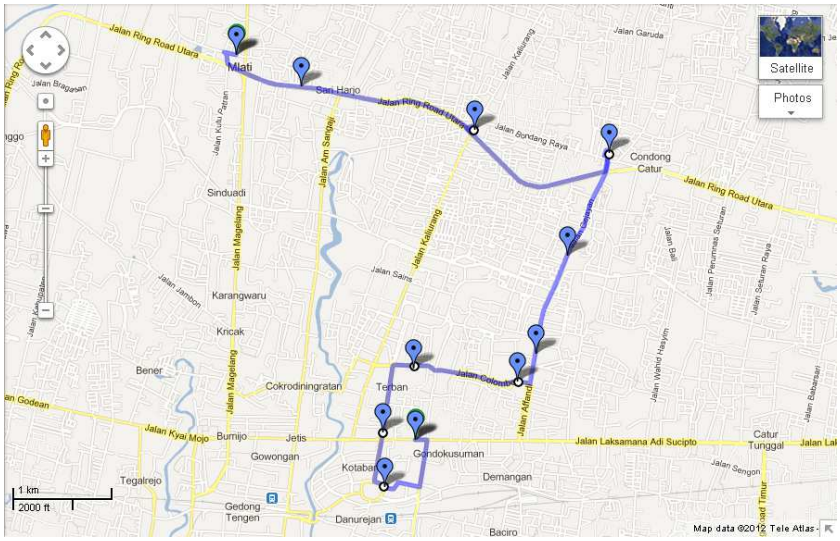
Pada analisis ini dilakukan pengujian terhadap menu *Input* dengan masukan *shelter* asal Terminal Jombor dan *shelter* tujuan Sudirman 1. Pengujian dilakukan dengan mengecek efektivitas rute yang diberikan oleh aplikasi GeoTrans. Hasil rute yang diberikan adalah sebagai berikut:

Tabel 4.4 Hasil Pencarian Rute dari Terminal Jombor ke Sudirman 1

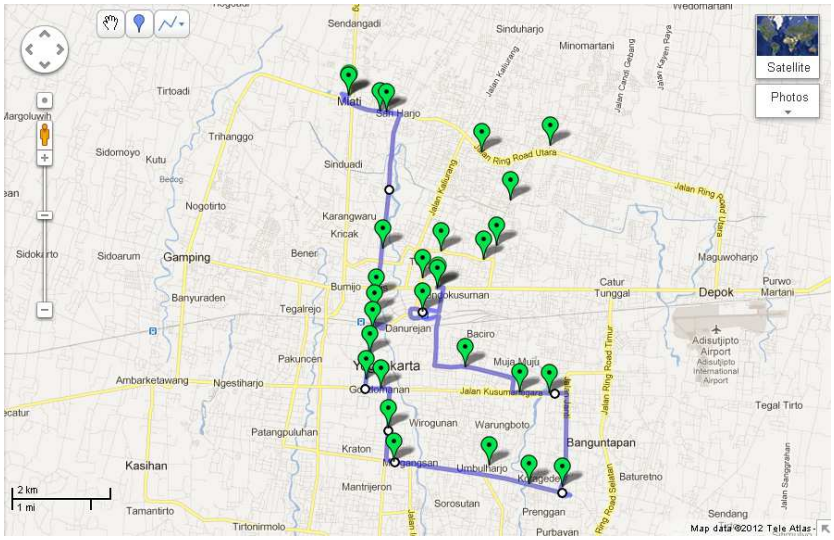
Shelter	**Koridor**
Terminal Jombor	2A
Ring Road Utara (Monjali 1)	2A
Ring Road Utara (Monjali 1)	2B
Ring Road Utara (Kentungan)	2B

Terminal Condong Catur	2B
Susteran Novisiat	2B
Sanata Dharma	2B
Jl. Colombo (Samirono)	2B
Jl. Colombo (Panti Rapih)	2B
Cik Di Tiro 1	2B
Yos Sudarso	2B
Yos Sudarso	2A
Sudirman 1	2A

Pada koridor 2A, sebenarnya kita dapat menemukan bahwa dua *shelter* masukan (Terminal Jombor dan Sudirman 1) berada dalam satu jalur. Dalam artian, kita dapat hanya menggunakan bus 2A saja untuk pergi dari Terminal Jombor ke Sudirman 1. Namun karena algoritma Dijkstra menggunakan prinsip rute terpendek dari *vertex* awal ke *vertex* tujuan, maka aplikasi akan memilih transit di suatu *shelter* yang memungkinkan untuk menuju *shelter* tujuan dengan cara yang lebih singkat. Untuk percobaan ini, pengguna disarankan transit di *shelter* Ring Road Utara (Monjali 1) dan pindah ke bus 2B, lalu transit lagi di *shelter* Yos Sudarso dan pindah lagi ke bus 2A sebelum sampai di *shelter* Sudirman 1. Berikut merupakan perbandingan jarak dari rute yang dipilihkan aplikasi dengan rute yang hanya mengambil koridor 2A saja.



Gambar 4.6 Hasil Rute yang Dipilihkan Aplikasi



Gambar 4.7 Hasil Rute dengan Menggunakan Koridor 2A

Dari peta tersebut kita bisa mengetahui bahwa dengan *shelter* asal dan tujuan yang sama, rute yang dibuat oleh aplikasi jauh lebih pendek sekalipun kita harus berpindah bus dua kali daripada jika kita hanya menggunakan satu koridor. Hal itu karena algoritma Dijkstra memberikan persamaan yang memberikan hasil

rute terpendek dengan mempertimbangkan arah. Algoritma Dijkstra dapat memilih transit karena terdapat rute lain yang arahnya sudah dimasukkan dalam *database* rute dan menuju *shelter* yang dikehendaki.

5. Analisis *Haversine Formula*

Analisis *haversine formula* dilakukan dengan membandingkan jarak yang tertera pada hasil *listview shelter* di menu *Nearest* dengan jarak sebenarnya di lapangan. Pada analisis ini, penulis menggunakan aplikasi GeoTrans di dekat *shelter*, lalu jarak yang tertera di aplikasi pada *shelter* yang bersangkutan disesuaikan kebenarannya dengan posisi di lapangan. Berikut daftar *shelter* yang dijadikan sampel.

Tabel 4.15 Hasil Jarak pada Menu *Nearest* untuk 5 Sampel *Shelter*

No.	*Shelter*	Jarak
1.	Jl. Kaliurang (Kopma UGM)	0,23 km
2.	Pertanian UGM	0,25 km
3.	Jl. Colombo (Samirono)	0,20 km
4.	Urip Sumoharjo	0,05 km
5.	Sudirman 1	0,21 km
Kesalahan Rata-rata		0,188 km

Pada tahap analisis ini, diketahui bahwa kesalahan jarak rata-rata pada menu *Nearest* adalah 0,188 km atau 188 m. Kesalahan tersebut diakibatkan oleh dua faktor:

a. Faktor yang pertama adalah penggunaan LBS (BTS) sebagai opsi awal penentuan posisi pada *mobile* GIS dibandingkan dengan menggunakan GPS (satelit). *Device* cenderung memilih BTS (dalam *coding* menggunakan `NETWORK_PROVIDER`) dengan menggunakan metode *Observed Time Difference of Arrival* (OTDOA) *positioning* untuk penentuan posisi dan baru menggunakan GPS (satelit) sebagai *second choice* ketika respon dari BTS = 0. Seperti kita ketahui, tingkat akurasi BTS lebih rendah daripada GPS, dimana secara teori akurasi OTDOA mencapai 30-300 m, jauh di bawah GPS yang mencapai 1-10 m (*outdoor*) atau 30 m (*indoor*). Karena tingkat akurasi BTS yang lebih rendah itu maka penentuan posisi *device* sangat mungkin tidak tepat pada posisi sebenarnya di lapangan.

b. Faktor kedua adalah karena Google Map tidak menawarkan akurasi dalam penentuan posisi *device*, namun hanya memberikan letak pada peta berdasarkan radius tertentu. Dalam artian, posisi *user* pada Google Map bisa jadi tidak berada di posisi yang tepat, namun selama masih dalam cakupan area posisi sebenarnya, Google Map masih dapat mentolerir. Untuk membandingkan akurasi A-GPS dan LBS pada Google Map, penulis melakukan pengujian dengan mengukur posisi suatu titik dengan menggunakan GPS *handheld*, aplikasi A-GPS, dan Google Map.

Tabel 4.16 Perbandingan Selisih Antar Sistem (dalam meter)

No.	Selisih *Handheld* dan A-GPS	Selisih A-GPS dan Google Map	Selisih *Handheld* dan Google Map
1.	9.879845343	194.9593008	191.6994694
2.	24.53735815	0.912419311	25.13567031
3.	8.412281141	2.966368993	6.439399118
4.	12.35786895	1.000952546	11.42385614
5.	15.6200032	7.337648942	20.21155194

Dari hasil pengukuran, pada dasarnya terlihat bahwa akurasi A-GPS maupun Google Map relatif cukup baik untuk digunakan dalam keperluan navigasi. Namun karena layanan LBS menggunakan BTS yang tergantung pada sinyal *provider*, akurasinya dapat berubah-ubah. Kita bisa lihat pada pengukuran pertama yang selisihnya mencapai 191,699 m, serta pengukuran nomor 2 dan 5 yang masing-masing memiliki kesalahan 25,135 m dan 20,211 m. Namun selama masih dapat digunakan untuk keperluan navigasi, maka akurasi yang tinggi dapat diabaikan.

V. PENUTUP

V.1 Kesimpulan

Berdasarkan hasil pembuatan aplikasi GeoTrans 1.0, diperoleh beberapa kesimpulan sebagai berikut:

1. Pembuatan aplikasi *mobile* GIS Android GeoTrans 1.0 menggunakan *software* XAMPP, Eclipse, dan Notepad ++. XAMPP digunakan dalam pembuatan *database*, Eclipse digunakan untuk *coding* bahasa pemrograman java, dan Notepad ++ digunakan untuk membuat *coding* PHP. Sedangkan proses menampilkan peta pada Google Map dan pengambilan *database* dilakukan secara *online* digunakan *hosting* dari *server*.
2. Aplikasi GeoTrans 1.0 dapat berjalan pada versi *GingerBread* ke atas dengan koneksi paling cepat menggunakan jaringan 3G ke atas (WCDMA atau HSDPA). Kesalahan rata-rata jarak pada menu *Nearest* adalah 0,188 km yang diakibatkan oleh dua faktor, yaitu penggunaan LBS (BTS) sebagai opsi awal penentuan posisi pada *mobile* GIS dibandingkan dengan menggunakan GPS (satelit) dan Google Map tidak menawarkan akurasi dalam penentuan posisi *device*, namun hanya memberikan letak pada peta berdasarkan radius tertentu.

V.2 Saran

Untuk pengembangan aplikasi lebih lanjut agar makin memberikan manfaat untuk *user* ada beberapa hal yang dapat dijadikan masukan. Sebagian besar merupakan saran yang diberikan oleh responden yang sudah menggunakan sendiri aplikasi GeoTrans. Saran-saran tersebut yaitu:

1. Perbaikan di menu *Nearest* karena *user* masih sering mengalami masalah di sana yang diakibatkan oleh koneksi.
2. Perlu adanya *update* agar aplikasi dapat selalu relevan dengan kondisi bus Trans Jogja dan selalu dapat digunakan sebagai panduan.
3. Perbaikan dari segi *interface* agar lebih menarik.

DAFTAR PUSTAKA

1. Abidin, Hasanuddin Z. 2007. *Penentuan Posisi dengan GPS dan Aplikasinya*. Jakarta: PT Pradnya Paramita
2. Safaat, Nazruddin. 2011. *ANDROID Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika Bandung
3. Hariyanto, Bambang. 2011. *Esensi-esensi Bahasa Pemrograman Java*. Bandung: Informatika Bandung
4. Prahasta, Eddy. 2005. *Konsep-konsep Dasar Sistem Informasi Geografis*. Bandung: Informatika Bandung
5. Riyanto, 2010. *Membuat Sendiri Aplikasi Mobile GIS Platform Java ME, Blackberry & Android*. Yogyakarta: ANDI
6. Riyanto, 2010. *Sistem Informasi Geografis Berbasis Mobile*. Yogyakarta: Gava Media
7. Trisnawati, Arifah. 2012. *Aplikasi Peta Kuliner Kota SemarangBerbasis Mobile GIS pada Smartphone Android.* Universitas Diponegoro: Semarang.
8. Yin, Hongying. 2003. *Location Based Service*. Helsinki University of Technology: Helsinki

Situs Web:

1. Atunggal, Dedi. 2006. *Global Positioning System*. http://dediatunggal.blogspot.com
2. Departemen Pertanian. *Pengenalan PHP*. http://pusdatin.deptan.go.id
3. Elinux. 2011. *Android Architecture*. http://elinux.org
4. ESRI Developer Network. *Developing Mobile Applications*. http://edndoc.esri.com
5. http://developer.android.com
6. http://dishub-diy.net
7. http://id.wikipedia.org
8. Navi Gadget. *A-GPS (Assisted GPS).* http://NaviGadget.com
9. Prihadi, Susetyo Dwi. 2012. *Android Terus Meraksasa*. http://inet.detik.com
10. Trans Jogja. *Rute Trayek Bus Trans Jogja*. http://transjogja.com
11. Waldura, Renauld. 2007. *Dijkstra's Shortest Path Algorithm in Java*. http://renaud.waldura.com