

Aplikasi Perencanaan Perjalanan Kota Bandung dengan Memanfaatkan OpenStreetMap XML dan .NET Compact Framework di PDA dengan Teknologi Pencarian Rute Algoritma A*

Andi Wahyu Rahardjo Emanuel, Aditya Wiraguna

Jurusan S1 Teknik Informatika

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164

Email: andi.wre@eng.maranatha.edu

Abstract

Map is a tool to help people on finding a location. Reading a map is often difficult for some people, the nature of available maps that are available in the shape of books and big sheet of paper made it inconvenient to use. Merging technologies to solve these mapping problems can be done using PDA technologies. This mapping system in PDAs will make people find it easier to find a specified location. The goal of this application is to help people find places and streets in City of Bandung and also the available routes to the location. Using PDAs on accessing the application made it simple and convenient to use. The system also provides a way of updating place and street data using the internet. This application is built using C#, thus requiring the designated PDA platform to have .NET Compact framework 2.0. This application is using OpenStreetMap xml data format to define place and street, because limited capabilities of .NET Compact Framework to process xml data this application use db4o database engine to store the xml data. Features that are available on this application are searching, A search algorithm, street route, home, cache and pre-processing.*

Keywords : *A* Search, Shortest Path, OpenStreetMap, db4o*

1. Pendahuluan

Mengunjungi kota atau daerah tertentu yang tidak kita ketahui seluk beluk jalan di tempat tersebut menjadikan peta atau petunjuk jalan sebagai alat bantu pencapaian tujuan. Bagi sebagian orang membaca peta merupakan suatu hal cukup sulit dilakukan, terutama untuk menelusuri tempat yang dituju dari posisinya berada. Selain itu penggunaan peta biasa, baik yang berbentuk buku maupun lembaran kertas besar terasa kurang praktis, bahkan untuk beberapa daerah diperlukan peta yang selalu *up to date* misalnya dengan dibukanya jalur baru atau perubahan nama jalan, mengharuskan kita untuk membeli peta baru paling tidak dalam setiap tahunnya.

Tujuan dari aplikasi ini yaitu membantu pengguna dalam mencari tempat dan jalan di Kota Bandung serta informasi rute jalan yang dapat ditempuh. Penggunaan PDA sebagai perangkat aplikasi yang praktis dan mudah untuk dibawa. Selain itu system ini juga memungkinkan pengguna untuk memperbaharui data jalan dan tempat lewat update system di internet sehingga peta ini dapat terus digunakan.

2. Perancangan Aplikasi

Aplikasi berfungsi untuk membantu pengguna dalam menemukan rute perjalanan dari lokasi tertentu ke lokasi tujuan. Pengguna dapat mencari tempat atau jalan dengan memasukkan kata kunci, setelah pencarian selesai aplikasi akan menampilkan hasil dari pencarian. Pengguna dapat menyimpan data lokasi rumah pengguna untuk memudahkan proses pencarian rute dari rumah pengguna atau menuju rumah pengguna. Fitur-fitur yang terdapat dalam aplikasi ini yaitu:

- **Penelusuran**

Fitur ini membantu pengguna untuk mencari lokasi atau jalan tertentu dengan memasukkan kata kunci. Hasil dari dari penelusuran berupa daftar tempat atau jalan yang sesuai dengan kata kunci beserta perkiraan jarak yang harus ditempuh dari posisi pengguna. Contoh: jika pengguna bermaksud untuk mencari lokasi Universitas Kristen Maranatha, maka pengguna memasukkan kata kunci “maranatha” dan memilih kategori pencarian “universitas”.

- **Pencarian jalur terdekat dengan Algoritma A***

Pencarian A* adalah algoritma best-first search yang banyak digunakan untuk pencarian rute jalan. A* mengevaluasi suatu node dengan mengkombinasikan nilai $g(n)$ (nilai yang ditempuh dari node awal ke node tersebut) dengan $h(n)$ (nilai dari node ke tujuan).

- **Rute jalan**

Fitur ini membantu pengguna untuk mencari rute jalan dari tempat asal ke tempat tujuan dengan jarak terdekat. Rute ditampilkan pada peta dengan warna garis yang berbeda dengan garis lain. Fitur ini menggunakan fitur pencarian A* Contoh: jika pengguna bermaksud untuk mencari rute jalan dari Universitas Kristen Maranatha menuju Bandung Trade Center, maka pengguna memilih lokasi Universitas Kristen Maranatha sebagai lokasi asal, dan memilih lokasi Bandung Trade Center sebagai lokasi tujuan kemudian menekan tombol cari.

- **Rumahku**

Fitur ini membantu pengguna untuk mencari rute jalan dari rumah pengguna ke tempat tujuan, atau dari lokasi tertentu ke rumah pengguna. Rute ditampilkan pada peta dengan warna garis yang berbeda dengan garis lain. Fitur ini memudahkan pengguna karena pengguna tidak perlu memilih tempat asal. Contoh: jika pengguna bermaksud untuk mencari rute jalan dari Universitas Kristen Maranatha menuju rumah pengguna, maka pengguna memilih Universitas Kristen Maranatha pada peta dan memilih menu menuju rumahku.

- **Cache**

Fitur ini menyimpan rute jalan yang pernah dicari oleh pengguna ke dalam memori penyimpanan, sehingga ketika pengguna melakukan pencarian dengan rute yang sama waktu pemrosesannya akan lebih singkat. Contoh: ketika pengguna untuk pertama kalinya mencari rute jalan dari Universitas Kristen Maranatha menuju rumahku sistem akan menyimpan data pencarian tersebut. Ketika pengguna mencari rute jalan dari Universitas Kristen Maranatha menuju rumahku lagi, sistem

akan mengambil data pencarian yang telah si simpan sebelumnya sehingga pencarian akan lebih cepat.

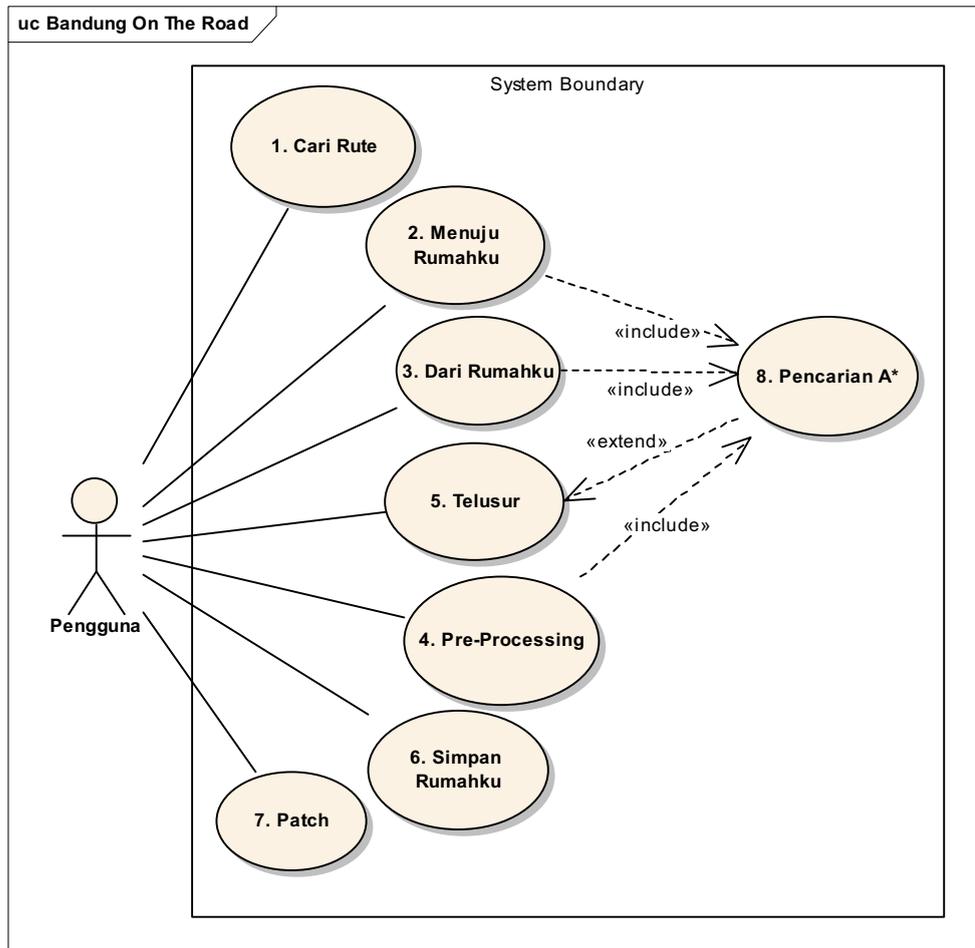
- *Pre-processing*

Fitur ini memungkinkan sistem untuk melakukan pencarian rute jalan dari rumah pengguna menuju lokasi-lokasi yang terdapat pada peta dan menyimpannya di dalam *cache* tanpa disadari oleh pengguna. Fitur ini membantu mengurangi waktu pemrosesan yang dibutuhkan untuk mencari rute jalan.

- *Patch Data*

Fitur ini memungkinkan pengguna untuk memperbaharui data jalan dan lokasi pada aplikasi dengan menerapkan *patch* yang dapat diambil dari internet. *Patch* yang diambil dari internet merupakan berkas *executable* untuk Pocket PC sehingga pengguna hanya perlu mengeksekusi berkas *patch* tersebut.

Use case diagram yang menggambarkan fitur-fitur diatas adalah sebagai berikut:



Gambar 1. Use Case Diagram Aplikasi

Persyaratan-persyaratan antarmuka dengan pengguna adalah sebagai berikut:

- Antarmuka berupa aplikasi Pocket PC.
- Antarmuka dengan pengguna berupa tampilan peta Kota Bandung. Peta dapat diperbesar atau diperkecil sesuai dengan keinginan pengguna.
- Pengguna dapat mencari lokasi jalan dengan memasukkan nama jalan yang dicari.
- Pengguna dapat memilih lokasi pada peta dengan melakukan *Tap & Hold* pada peta.
- Pengguna dapat memilih lokasi di peta sebagai titik awal atau titik tujuan.
- Rute jalan hasil pencarian ditampilkan pada peta dengan warna yang berbeda.

Aplikasi ini terdiri dari dua buah form, yaitu:

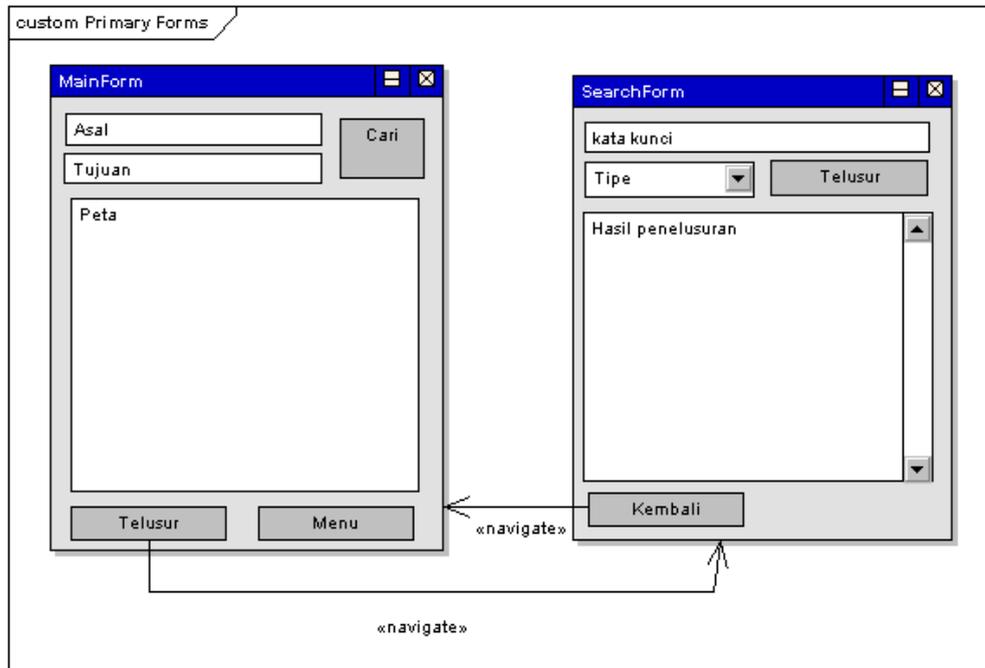
- MainForm

Form ini merupakan *form* yang ditampilkan ketika pengguna menjalankan aplikasi. *Form* ini berisikan gambar peta yang digunakan oleh pengguna untuk memilih lokasi awal dan tujuan.

- SearchForm

Form ini ditampilkan jika pengguna menekan tombol telusur pada *form* MainForm. *Form* ini menyediakan fasilitas penelusuran data kepada pengguna.

Berikut adalah gambar dari desain antarmuka pengguna:



Gambar 2. Desain Antar Muka Aplikasi

Aplikasi dibangun dengan menggunakan bahasa pemrograman C#, sehingga untuk menjalankan aplikasi dibutuhkan .NET Compact Framework 2.0 pada PDA pengguna. Sistem operasi minimum yang dibutuhkan untuk menjalankan .NET Compact Framework 2.0 adalah Pocket PC 2003, akan tetapi kebanyakan dari perangkat Pocket PC 2003 hanya memiliki kapasitas RAM sebesar 32 MB sehingga penulis menyarankan agar menggunakan perangkat dengan system operasi Windows Mobile 5.0 dan untuk menyimpan data jalan dan lokasi digunakan aplikasi basis data berorientasi objek Db4o.

Perkiraan spesifikasi perangkat keras minimum yang diperlukan untuk menjalankan aplikasi adalah PDA dengan processor 400 MHz dan RAM 64 MB. Dengan spesifikasi sedemikian aplikasi dirasakan masih berjalan kurang cepat, terutama saat melakukan pencarian yang tidak pernah dilakukan sebelumnya, sedangkan apabila aplikasi sudah berjalan cukup lama, maka sudah terdapat history pencarian yang memungkinkan pencarian lokasi dengan lebih cepat.

Rencana pembagian modul yang akan digunakan dalam pembuatan aplikasi perencanaan perjalanan Kota Bandung adalah terdiri dari modul db4o, modul OpenStreetMap, modul AStar, modul GUI.

3. OpenStreetMap

OpenStreetMap adalah proyek yang bertujuan untuk membuat dan menyediakan data geografi seperti peta jalan secara gratis. Data xml OpenStreetMap terdiri atas node, segment dan way. Berikut adalah contoh dari data OpenStreetMap:

```
<osm version='0.4' generator='JOSM'>
<node id='29356797' timestamp='2007-05-20 23:28:37' user='adhitya' visible='true'
lat='-6.88252843906513' lon='107.581485470605' />
<node id='29390964' timestamp='2007-05-20 23:25:45' user='adhitya' visible='true'
lat='-6.88563015251891' lon='107.581047734094'>
  <tag k='name' v='Universitas Kristen Maranatha' />
  <tag k='amenity' v='university' />
</node>
.
.
<segment id='25393523' timestamp='2007-05-20 23:28:37' user='adhitya'
visible='true' from='29356797' to='29391280' />
.
.
<way id='4623249' timestamp='2007-06-03 01:11:43' user='Adhitya Wiraguna'
visible='true'>
  <seg id='25393523' />
  <seg id='25393524' />
  <seg id='25393225' />
.
.
  <tag k='highway' v='secondary' />
  <tag k='name' v='Surya Sumantri' />
</way>
</osm>
```

Penjelasan dari masing-masing elemen:

- *Node*

Elemen ini mendefinisikan persimpangan atau lokasi. *Node* memiliki atribut utama id, lon (longitude), dan lat (latitude).

- *Segment*

Elemen ini mendefinisikan hubungan antar *node*. *Segment* memiliki atribut utama id, from (node asal), dan to (node tujuan).

- *Way*

Elemen ini mendefinisikan jalan. *Way* hanya memiliki satu atribut utama yaitu id. *Way* terdiri atas satu atau lebih segment.

- *Tag*

Elemen ini dapat mendefinisikan nama atau kategori dari lokasi. Elemen ini juga dapat mendefinisikan nama jalan, kategori jalan, dan arah dari jalan tersebut. Elemen ini dapat menjadi anak dari elemen *node*, *segment*, dan *way*.

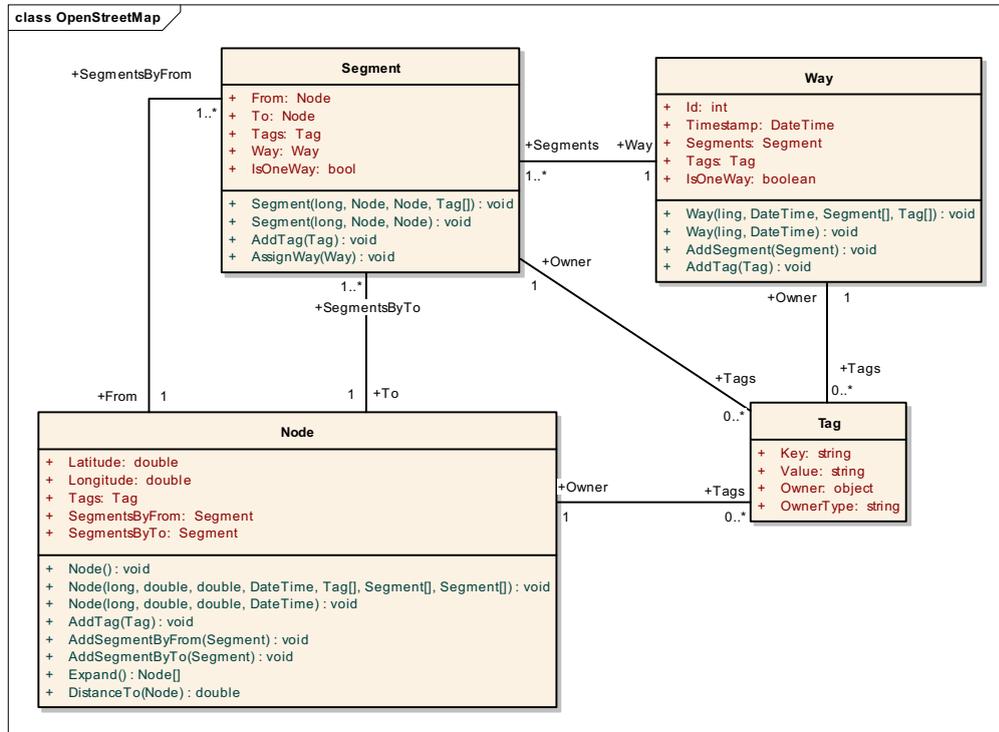
Untuk melakukan pengeditan peta, penulis menggunakan *tool* potlatch editor yang dapat diakses pada alamat <http://www.openstreetmap.org/edit.html>. Berikut adalah *screenshot* dari Potlatch editor:



Gambar 3. Proses Pembuatan Peta

Tool Osmarender dapat digunakan untuk mengubah format data OSM menjadi data gambar dengan format SVG. Penulis membuat *class-class* yang sesuai dengan elemen-elemen dari data OSM agar data dapat disimpan pada basis data db4o.

Berikut adalah *class diagram* yang mendefinisikan format data OSM:



Gambar 4. Class Diagram Aplikasi

4. Db4o

Db4o adalah *object database* yang dapat digunakan pada bahasa pemrograman JAVA dan .NET. Aplikasi db4o bersifat open source sehingga dapat digunakan secara gratis untuk kepentingan non-komersial. Aplikasi ini dapat diambil pada situs <http://www.db4o.com>. Db4o cocok untuk digunakan pada aplikasi yang tidak membutuhkan *database administrator*. Berikut adalah contoh perintah db4o untuk membuka berkas data, dan menyimpan data

- Membuka berkas data dengan nama data.db.

```
IObjectContainer data = Db4oFactory.OpenFile("data.db");
```

- Menyimpan data node dengan id 29356797.

```
node = new Node(29356797, -6.88252843906513, 107.581485470605);
```

```
data.Set(node);
```

Db4o menyediakan beberapa cara untuk melakukan *query* pada basis data, yaitu:

- *Query-By-Example*

Teknik ini menggunakan objek *template* sebagai kriteria. Berikut adalah contoh kode *Query-By-Example* untuk mengambil node dengan id 29356797:

```
nodeProto = new Node(29356797, 0, 0, null, null, null);
```

```
IObjectSet result = data.Get(fromProto);
```

- *Native Queries*

Teknik merupakan metoda *query* utama dari db4o. Berikut adalah contoh kode *Native Query* untuk mengambil node dengan id 29356797:

```
IList<Node> result = _data.Query<Node>(delegate(Node node){
```

```
    return node.Id == 29356797;
```

```
});
```

- *Soda API*

Metoda ini merupakan *low level query API* pada db4o. Metode ini cenderung sulit untuk digunakan. Berikut adalah contoh kode *Soda API* untuk mengambil node dengan id 29356797:

```
IQuery query = _data.Query();
```

```
query.Constraint(typeof(Node));
```

```
query.Descend("_id").Constraint(29356797);
```

```
IObjectSet result = query.Execute();
```

5. Pencarian A* (AStar)

Pencarian A* adalah algoritma *best-first search* yang paling banyak digunakan untuk mencari rute terdekat. Pencarian A* mengevaluasi *node* dengan mengkombinasikan nilai $g(n)$ (nilai yang dibutuhkan untuk mencapai *node* n dari *node* awal) dan $h(n)$ (nilai yang dibutuhkan untuk mencapai *node* tujuan dari *node* n tersebut):

$$f(n) = g(n) + h(n).$$

Karena $g(n)$ adalah nilai yang harus ditempuh dari *node* awal ke *node* n, dan $h(n)$ adalah perkiraan nilai paling kecil yang harus ditempuh dari *node* n ke *node* tujuan, maka $f(n)$ adalah perkiraan nilai terkecil yang harus ditempuh dari *node* awal ke *node* tujuan melalui n.

Pencarian A* akan menemukan solusi yang optimal jika $h(n)$ lebih kecil dari nilai sebenarnya untuk mencapai *node* tujuan. Penulis menggunakan jarak garis lurus antar node sebagai nilai $h(n)$. Pencarian A* pada kasus ini akan menemukan solusi yang optimal karena jarak garis lurus antara suatu lokasi / persimpangan ke lokasi / persimpangan lainnya lebih kecil dari jarak sebenarnya.

Berikut adalah *pseudo code* dari algoritma pencarian A*:

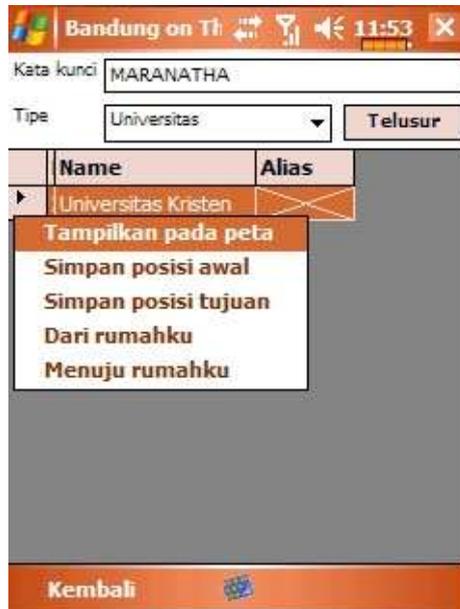
```
function AStar-Search(problem, fringe) returns a solution or failure
fringe ← Insert(Make-Node(Initial-State[problem]), fringe)
loop do
  if Empty?(fringe) then return failure
  node ← Remove-Lowest(fringe)
  if Goal-Test[problem](State[node]) then return Solution(node)
  fringe ← Insert-All(Expand(node, problem), fringe)
```

6. Hasil Implementasi

Hasil implementasi dari aplikasi yang sudah dibuat sebelumnya dan dipasang pada perangkat PDA dapat dilihat pada *screenshot* dari tampilan aplikasi sebagai berikut:



Gambar 5. Tampilan Main Form Aplikasi



Gambar 7. Tampilan Pencarian (Search Form)



Gambar 7. Pencarian Rute

Aplikasi juga telah melalui proses pengujian secara *blackbox* dengan hasil yang baik. Pengujian pengguna dilakukan dengan mencobakan kepada 30 orang dan mendapatkan tanggapan yang positif dengan kritik terbesar terutama masalah kecepatan.

7. Simpulan dan Saran

Aplikasi ini merupakan aplikasi PDA yang bersifat *stand alone*, sehingga untuk menjalankan aplikasi tidak dibutuhkan koneksi internet. Aplikasi ini menggunakan perhitungan jarak garis lurus antara lokasi awal dengan lokasi tujuan. Pengguna dapat memperbaharui data pada aplikasi dengan mengambil *patch* dari internet dengan menggunakan PC atau PDA.

Setelah aplikasi dikembangkan dan dijalankan pada perangkat PDA, diketahui bahwa masalah kecepatan masih menjadi kendala utama. Meskipun aplikasi sudah dilengkapi dengan fitur pre-processing yang akan melakukan pencarian selama PDA tidak dipergunakan oleh pengguna, namun apabila pengguna mencari lokasi yang tidak ada dalam hasil pencarian pre-processing tersebut akan memakan waktu yang cukup lama (mendekati 1 menit). Saran pengembangan yang dimungkinkan adalah penggunaan format gambar SVG (Scalable Vector Graphic) sebagai pengganti format JPG yang dipakai sekarang, dan juga memisahkan lapisan gambar peta dengan gambar nama jalan / tempat sehingga meskipun tampilan peta dibuat dengan resolusi lebih rendah tampilan nama jalan dan nama tempat masih dapat terbaca dan aplikasi dapat mengalami peningkatan kecepatan.

Daftar Pustaka

- [Db406] db4o. (n.d.). *db4o Developer Community - Reference* -. Retrieved June, 2006, from: <http://developer.db4o.com/Resources/view.aspx/Reference>.
- [Gun00] Gunnerson, E. (2000). *A Programmer's Introduction to C#*. New York: Apress.
- [Ope07] OpenStreetMap. (n.d.). *Main Page - OpenStreetMap*. Retrieved June, 2007, from: OpenStreetMap Wiki: <http://wiki.openstreetmap.org>
- [Rus03] Russel, S., & Norvig, P. (2003). *Artificial Intelligence A Modern Approach (2nd ed.)*. New Jersey: Prentice Hall.