

**API LOCATION (JSR 179):
STANDAR PENENTUAN POSISI UNTUK TELEPON
SELULER BERKEMAMPUAN JAVA**

Andi Sunyoto

Dosen STMIK AMIKOM Yogyakarta

Asbtraksi

JSR 179 API Location dapat menghasilkan informasi posisi seperti halnya perangkat penentu posisi lainnya, yaitu Global Positioning System (GPS). Informasi posisi yang dihasilkan dapat diolah untuk mekanisme layanan menemukan stasiun pengisian bahan bakar terdekat, restoran terdekat, memantau anak, dan masih banyak lagi. Aplikasi sosial networking beberapa sudah memanfaatkan penentuan posisi.

API location (JSR 179) dalam J2ME masuk ke optional package javax.microeditin.location yang memiliki kemampuan untuk menenghasilkan informasi posisi sebuah peralatan. API Location menghasilkan informasi informasi lokasi secara fisik yang dapat digunakan untuk landmark yang dapat disimpan.

Keywords: *J2ME, API, Location, GPS*

Pendahuluan

Location Base Services (LBS) adalah suatu mekanisme layanan yang menyediakan informasi tentang lokasi. Location based services memanfaatkan lokasi dari device untuk menyediakan informasi mengenai lokasi. Pertanyaan utama yang dijawab oleh Location Based Service adalah “Dimana saya?” . Untuk lebih spesifiknya adalah “Dimana letak device?”.

Perangkat yang terkenal untuk menginformasikan posisi adalah *Global Positioning System (GPS)*. GPS menggunakan teknologi satelit yang dikelola oleh departemen pertahanan Amerika Serikat untuk dapat menginformasikan posisi. Saat ini ada 24 satelit yang digunakan untuk

penentuan posisi. Informasi posisi juga dapat diperoleh dari perangkat yang kita gunakan sehari-hari, yaitu telepon seluler.

GPS mempunyai akurasi yang lebih tinggi dibanding telepon seluler. Akurasi berkisar 5 sampai 30 meter, sedangkan untuk telepon seluler tingkat akurasinya bervariasi mulai 500 meter sampai 20 meter. Penginformasian posisi menggunakan seluler dapat digunakan jika tidak membutuhkan tingkat akurasi yang tinggi.

Informasi posisi dapat dikembangkan untuk mekanisme layanan membantu menemukan stasiun pengisian bahan bakar terdekat, restoran terdekat, memantau anak, dan masih banyak lagi. Aplikasi sosial networking beberapa sudah memanfaatkan penentuan posisi.

Jurnal ini akan membahas penentuan posisi menggunakan perangkat telepon seluler, dengan harapan tidak memerlukan peralatan yang mahal, karena dapat memanfaatkan perangkat yang telah kita punya, yaitu telepon seluler.

Pembahasan

1. Penentuan Posisi Alat

Penentuan posisi sebuah perangkat menggunakan metode posisi *real-time*. Akurasi posisi yang dihasilkan tergantung metode yang digunakan. Lokasi dapat nyatakan dalam bentuk data spasial atau deskripsi teks. Posisi spasial dapat nyatakan menggunakan *latitude-longitude-altitude*. *Latitude* dinyatakan dengan nilai 0-90 derajat utara atau selatan dari garis katulistiwa dan *longitude* dengan nilai 0-180 derajat timur atau barat dari *prime meridian*. *Prime meridian* adalah sebuah garis virtual yang melewati sebuah kota bernama Greenwich di Inggris. *Altitude* menyatakan tinggi permukaan dari permukaan laut. Sedangkan teks biasanya digunakan untuk menyatakan jalan, kota, kode pos dan lain-lain. Berikut metode penentuan posisi yang sering digunakan:

- Menggunakan jaringan telepon seluler. *Cell ID* dapat digunakan mengidentifikasi *Base Transceiver Station* (BTS) sedangkan device berhubungan dengan BTS. Sebuah telepon seluler

mempunyai akurasi radius 2 sampai 20 kilometer. Teknik lain dapat memiliki akurasi 150 meter.

- Menggunakan satelit. GPS adalah salah satu teknik yang digunakan teknologi ini. Agar dapat menggunakan metode ini pengguna harus menggunakan *GPS receiver*. Akurasi metode ini mencapai 4 sampai 40 meter.
- Menggunakan *short-range position beacon*. Metode ini digunakan untuk area yang tidak terlalu luas. Metode ini dapat menggunakan teknologi bluetooth.

Beberapa aplikasi tidak membutuhkan akurasi yang tinggi, misalnya dengan toleransi 30 meter, tetapi aplikasi lain memerlukan layanan informasi posisi yang harus akurat.

2. API Location J2ME

API Location (JSR 179) dalam J2ME masuk ke optional package `javax.microedition.location` yang memiliki kemampuan untuk menghasilkan informasi posisi sebuah peralatan. API Location menghasilkan informasi lokasi secara fisik yang dapat digunakan untuk *landmark* yang dapat disimpan.

JSR 179 membutuhkan *Connected Device Configuration (CDC)* atau *Connected Limited Device Configuration (CLDC)* versi 1.1. Namun CLDC tidak cukup memadai, karena tidak dapat melakukan perhitungan *floating-point* yang digunakan untuk merepresentasikan koordinat dan pengukuran yang lain.

Platform hardware menentukan method apa yang disupport. Aplikasi dapat meminta menyediakan macam karakteristik, seperti derajat akurasi minimal.

Class abstrak `Location` menghasilkan hasil lokasi. Masing-masing obyek menghasilkan koordinat, kecepatan, time stamp.

Koordinat direpresentasikan masing-masing oleh dua kelas:

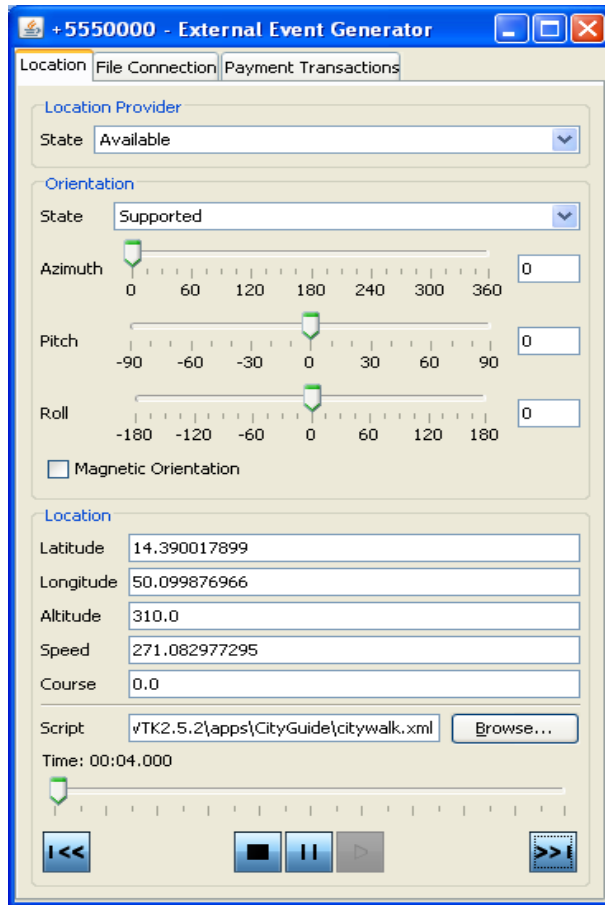
- Sebuah object `Coordinate` menunjukkan sebuah titik *latitude* dan *longitude* dalam derajat, dan *altitude* dalam meter.

- Sebuah obyek `QualifiedCoordinates` berisi *latitude*, *longitude* dan *altitude* dan juga menunjukkan tingkat akurasi, direpresentasikan dalam area radius.

Berikut potongan program yang menunjukkan bagaimana menghasilkan lokasi sebuah perangkat.

```
...
Criteria cr= new Criteria();
cr.setHorizontalAccuracy(500);
LocationProvider lp= LocationProvider.getInstance(cr);
Location l = lp.getLocation(60);
Coordinates c = l.getQualifiedCoordinates();
if(c != null ) {
    // Use coordinate information
    double lat = c.getLatitude();
    double lon = c.getLongitude();
}
...
```

Simulasi lokasi dapat dibuat pada emulator, sehingga pada saat membuat program seolah-olah seperti peralatan yang menghasilkan posisi. Pada saat emulator sedang dijalankan pilih menu *MIDlet – External Event* dari window emulator kemudian pilih tab location.



Gambar 1.

Pada bagian *Location* dapat diisikan nilai *latitude*, *longitude*, *altitude*, *speed*, dan *course*. Aplikasi yang menggunakan *API Location* dapat menerima nilai ini pada emulator sebagai nilai sebuah lokasi. Sun Java Wireless Toolkit CLDC menyediakan file untuk lebih

memperjelas perpindahan posisi dalam bentuk file XML bernama waypoint.

```
<waypoints>
```

```
  <waypoint time="1500" latitude="14.389796708964603"  
  longitude="50.09985002736201" altitude="310" />
```

```
  <waypoint time="5500" latitude="14.390283327582706"  
  longitude="50.099909292742595" altitude="310" />
```

```
  <waypoint time="5500" latitude="14.390282428592135"  
  longitude="50.100262519964886" altitude="310" />
```

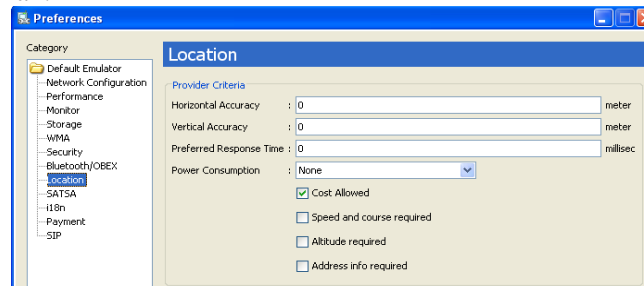
```
  <waypoint time="1500" latitude="14.390793552257932"  
  longitude="50.10032395846041" altitude="310" />
```

```
</waypoints>
```

Ukuran *altitude* dalam satuan meter, dan nilai waktu dalam *millisecond*. Gunakan text editor untuk merubah skrip lokasi dan load dengan menekan tombol Browse, kemudian dapat menggunakan tombol navigasi playing, pausing, dan stoping untuk menuju ke semua skrip.

3. Location Provider

API Location J2ME menyediakan sumber informasi lokasi. Karena terdapat security untuk mengakses method ini maka disediakan SecurityException jika tidak tersedia informasi posisi yang diinginkan.

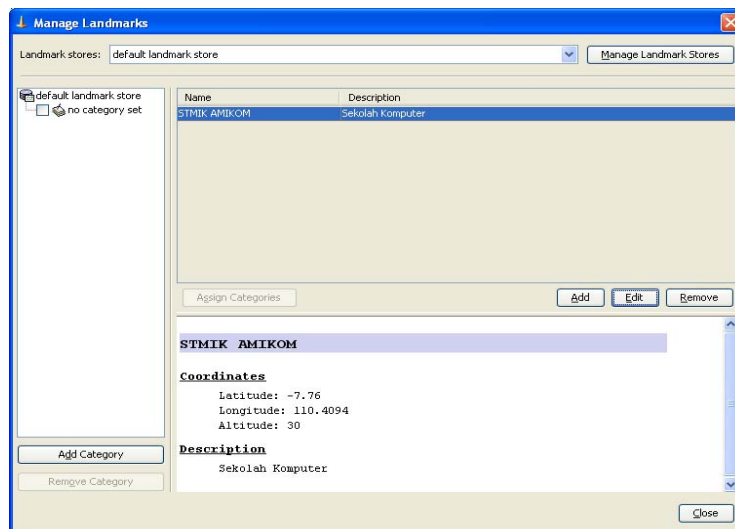


Gambar 2. Konfigurasi Location Provider

4. Landmarks

Sebuah *landmark* adalah sebuah lokasi yang diasosiasikan dengan nama dan deskripsi. *Landmark* dapat disimpan didalam database, dimana dapat *dishare* diantara aplikasi J2ME. Landmark dapat disimpan secara periodik menggunakan lokasi: misal rumah, office, restoran favorit dan lain sebagainya.masing-masing direpresentasikan menggunakan *instance Landmark*, dan database menggunakan LandmarkStore. Landmark dapat dibuat berulang dan diberi nama group atau kateogri seperti bioskop, museum atau customer.

Jika aplikasi membutuhkan fitur kompas dapat memanfaatkan *class Orientation* yang merepresentasikan azimuth yaitu sudut yang diambil dari utara.



Gambar 3. Konfigurasi Landmark pada Wireless Toolkit

Pada *Landmark Manager* minimal terdapat satu *landmark*. Jika akan menambah dapat menggunakan tombol *Add*. Dalam

landmark manager dapat menambah atau mengedit data *landmark*. Selain itu *landmark* dapat dikategorikan.

5. Security

Berikut table nama-nama permission dan method yang dilindungi method tersebut:

Nama Permission	Method yang dilindungi
javax.microedition.location.Location	LocationProvider.getLocation(), LocationProvider.setLocationListener()
javax.microedition.location.Orientation	Orientation.getOrientation()
javax.microedition.location.ProximityListener	LocationProvider.addProximityListener()
javax.microedition.location.LandmarkStore.read	LandmarkStore.getInstance(), LandmarkStore.listLandmarkStores()
javax.microedition.location.LandmarkStore.write	LandmarkStore.addLandmark(), LandmarkStore.deleteLandmark(), LandmarkStore.removeLandmarkFromCategory(), LandmarkStore.updateLandmark()
javax.microedition.location.LandmarkStore.category	LandmarkStore.addCategory(), LandmarkStore.deleteCategory()
javax.microedition.location.LandmarkStore.management	LandmarkStore.createLandmarkStore(), LandmarkStore.deleteLandmarkStore()

6. Location Listener

API Location J2ME juga diberi fitur sebuah *location listener* yang diintegrasikan dengan bentuk instan *Landmark provider*. Untuk memanfaatkan *listener* ini harus mengimplement *interface* *LocationInterface*. Dengan setting interval tertentu method *locationUpdate()* dapat dipanggil berulang-ulang oleh aplikasi.

7. Proximity Listener

Method `proximityEvent` akan dipanggil jika device berada dalam jangkauan koordinat. Koordinat yang menjadi acuan harus didaftarkan dulu.

Kesimpulan

Pemanfaatan API Location J2ME dapat membangun aplikasi yang dapat memberikan informasi posisi yang dapat dipasangkan pada perangkat mobil seperti telepon seluler atau PDA. Aplikasi yang akan dibangun harus diperhitungkan tingkat akurasi.

Daftar Pustaka

- Sing Li, Jonathan Knudsen., "Beginning J2ME: From Novice to Professional-Third Edition" ., Apress, 2005
- Yu Feng and Jun Zhu., "Wireless Java Programming with Java 2 Micro Edition" ., SAMS, 2001
- Kimmo Löytänä., "Enabling Location-Based Applications with J2ME" ., Forum.Nokia.com., _____
- James Keogh., "J2ME: The Complete Reference" ., McGraw-Hill/Osborne., 2003
- Nico Goeminne., "Developing Location Based Services: Introducing the Location API for J2ME" .,
<http://mobiforge.com/developing/story/developing-location-based-services-introducing-location-api-j2me>
- Qusay H. Mahmoud., "J2ME and Location-Based Services" .,
[http://developers.sun.com/mobility/apis/articles/location/.](http://developers.sun.com/mobility/apis/articles/location/),
March 2004
- Sun Microsystems., "Using the Location API" .,
[http://java.sun.com/products/sjwtoolkit/wtk2.5.2/docs/UserGuide-ide-html/location.html.](http://java.sun.com/products/sjwtoolkit/wtk2.5.2/docs/UserGuide-ide-html/location.html), 2007