

Prototipe Aplikasi Lacak Posisi Berbasis Java 2 Micro-EditionTeguh Susyanto ⁴⁾**Abstract**

Location-Based Service (LBS) is a technology based on device location information service called the Global Positioning System (GPS). Along with many types of mobile phones or handheld devices with GPS support has been widely available in the market with a relatively affordable price, this is quite advantageous, especially to be developed into effective technology-based location using a GPS device. This technology has been widely implemented to support specific business areas such as detection tools for security systems, the route guide in the transportation and tourism, and also for other fields that use geographic location systems. In this paper the author build a prototype application Tracking Position using GPS on a mobile device. That can later be exploited for the benefit in the field of transportation and security. This application is developed using Java 2 Micro Edition platform and JSR-179 Location API, as a simulation of detection of the existence of tools that can be monitored via computer by using the Internet Networks.

Keyword : Location-Based System, GPS, Lacak Posisi, J2ME

I. PENDAHULUAN

Location-Based Service (LBS) merupakan suatu teknologi layanan informasi lokasi berdasarkan perangkat yang bernama *Global Positioning System (GPS)*. Perangkat GPS ini menggunakan teknologi satelit yang dikelola oleh Departemen Pertahanan Amerika Serikat untuk dapat menginformasikan posisi. Ada 24 satelit yang dioperasikan untuk menginformasikan lokasi di permukaan bumi.

Saat ini banyak jenis ponsel atau perangkat handheld dengan dukungan GPS telah banyak tersedia dipasaran dengan harga yang relatif terjangkau, hal ini cukup menguntungkan terutama untuk dikembangkan menjadi teknologi tepat guna berbasis lokasi dengan memanfaatkan kecanggihan perangkat GPS. Teknologi ini berpotensi untuk menjangkau banyak bidang bisnis di antaranya bidang

⁴⁾ Staf Pengajar STMIK Sinar Nusantara Surakarta

pertanahan, pertanian, jasa transportasi dan pariwisata, bidang keamanan dan lain-lain. Sedangkan bentuk-bentuk aplikasi yang memanfaatkan teknologi Location-Based Service di antaranya pendeteksian alat untuk sistem keamanan, pemandu rute perjalanan dalam jasa transportasi maupun pariwisata, dan pengukuran tanah untuk bidang pertanahan dan sebagainya.

Dalam penelitian ini penulis mencoba membangun prototipe aplikasi Pelacakan Posisi dengan menggunakan perangkat GPS pada seluler yang nantinya dapat dimanfaatkan untuk kepentingan dalam bidang transportasi dan pariwisata. Aplikasi ini dikembangkan dengan menggunakan platform *Java 2 Micro Edition* dan *JSR-179 Location API*, sebagai sebuah simulasi deteksi keberadaan alat yang dapat dipantau melalui komputer dengan menggunakan media jaringan internet.

II. JAVA 2 MICROEDITION (J2ME)

Perangkat software yang dikembangkan oleh Sun Microsystem Co., yang memungkinkan perangkat micro seperti Handphone, PDA, maupun perangkat kecil lain, mampu menjalankan aplikasi-aplikasi khusus berbasis java pada perangkat tersebut. Sebagian besar vendor-vendor produk seluler telah menyertakan lingkungan software java (*Java Virtual Machine*), sehingga perangkat seluler tersebut dapat menjalankan aplikasi-aplikasi yang dibuat dengan berbasis java. Vendor-vendor seperti Nokia, Siemen, Motorola, RIM, Palm telah banyak menyematkan dukungan perangkatnya pada aplikasi Java.

J2ME memiliki komponen-komponen yang terdiri dari Java Virtual Machine (JVM), Java API (Application Programming Interface) dan Paket-paket opsional lain untuk pengembangan aplikasi java. Bagian paling pokok pada J2ME adalah bagian *configuration*, *Profile* dan *Optional Packages*.

Lapisan Konfigurasi (*Configuration*)

Terdapat dua macam konfigurasi pada lapisan ini, yaitu *Connected Limited Device (CLDC)* dan *Connected Device Configuration (CDC)*. CLDC merupakan konfigurasi pada perangkat yang memiliki karakteristik dengan prosesor 16 bit atau 32 bit dan memori yang kecil antara 160 KByte sampai dengan 512KByte. Implementasi java pada perangkat ini umumnya berupa KVM, yaitu Java Virtual Machine yang berukuran puluhan kilo byte saja. Sedangkan CDC, merupakan bentuk konfigurasi pada perangkat yang relatif memiliki memori yang lebih besar seperti perangkat PDA (Personal Digital Assistance), dan implementasi CDC dapat menggunakan JVM Standard (Java SE) terutama untuk alat yang bermemori 2 Mbyte atau lebih. Untuk

konfigurasi ini banyak dijumpai pada peralatan seperti PDA atau TV Set Top Boxes.

Lapisan Profil (*Profile Layer*)

Profile adalah ekstensi dari configuration, yaitu dengan menambahkan sekumpulan API tambahan yang umum digunakan untuk kebutuhan tertentu. Profile juga mendefinisikan life-cycle aplikasi, antarmuka pengguna dan akses ke fitur lainnya dari sebuah device.

J2ME memiliki beberapa bagian profil diantaranya : Mobile Information Device Profile (MIDP), Foundation Profile (FP), Personal Profile (PP), dan Personal Digital Assistance (PDA).

Optional Packages

CLDC dan CDC dapat dilengkapi dengan optional packages untuk kebutuhan tertentu, seperti misalnya Bluetooth, Location, Wireless Messaging atau Web Service dan lainnya.

MIDlet

Aplikasi Midlet memiliki satu atau beberapa kelas yang diturunkan dari kelas MIDlet dan MIDlet menyediakan tiga metode dasar yaitu *startApp()*, yang berfungsi untuk inisialisasi obyek dan display, metode kedua yaitu *pauseApp()* berguna untuk menghentikan aplikasi secara sementara. Sedangkan metode yang terakhir yaitu *destroyApp()* yang berfungsi untuk menghentikan aplikasi midlet.

Location API untuk J2ME

JSR-179 Location API merupakan spesifikasi yang didefinisikan di dalam *Optional Package* J2ME yang dimanfaatkan untuk mengembangkan aplikasi *Location-Based Service* (LBS) untuk perangkat bergerak. API ini dirancang oleh *Java Community Process* (JCP) untuk menghasilkan informasi tentang lokasi geografis dari sebuah terminal dengan aplikasi berbasis Java. Location API ini dapat dimanfaatkan pada perangkat yang mendukung J2ME Connected Limited Device Configuration (CLDC) v1.1 dan juga J2ME Connected Device Configuration (CDC).

Paket `javax.microedition.location` mengandung kelas dasar untuk *request* dan memperoleh sebuah hasil lokasi perangkat GPS, meliputi koordinat, kecepatan dan timestamp.

Koordinat direpresentasikan oleh dua kelas berikut :

- Obyek **Coordinates** menunjukan sebuah titik *latitude* dan *longitude* dalam derajat, dan *altitude* dalam meter.
- Obyek **QualifiedCoordinates** mengandung derajat *latitude* dan *longitude* dan besaran *altitude* sekaligus menunjukan tingkat akurasi, yang direpresentasikan dalam area radius.

III. IMPLEMENTASI PROGRAM DAN PEMBAHASAN

Aplikasi Lacak Posisi yang dibangun terdiri dari Aplikasi Pemantau Koordinat Alat dan Aplikasi Client Penentu Koordinat, yang mana aplikasi Pemantau Posisi Alat dimanfaatkan untuk menyimpan setiap perubahan posisi alat secara *realtime* yaitu dengan menerima pesan yang dikirimkan oleh Penentu Posisi Alat dengan menggunakan media komunikasi internet via GPRS/EDGE/3G. Setiap perubahan koordinat akan ditampilkan dengan menggunakan antarmuka peta lokasi dalam bentuk web-based.



Gambar 1 Skema Aplikasi Pelacakan Posisi

Aplikasi Penentu Koordinat (Client-side)

Aplikasi penentu posisi koordinat ini dibangun dengan lingkungan pengembangan J2ME dengan memanfaatkan *JSR-179 Location API* sedangkan target *Configuration* dan *Profile* menggunakan MIDP2.0 dan CLDC1.1.

Atribut data yang diolah pada aplikasi terdiri dari ID (Email), Lokasi Koordinat (*latitude*, *longitude* dalam satuan derajat dan *Altitude* dalam satuan meter), dan Interval waktu (detik). Atribut data ID(Email), digunakan untuk memberikan identitas perangkat yang

dipantau oleh sistem. Data posisi koordinat perangkat diperoleh dari informasi yang diberikan oleh satelit penerima sinyal GPS dengan menggunakan rutin-rutin dari API Location. Sedangkan atribut interval difungsikan untuk mengatur waktu pemicuan proses pemerolehan titik koordinat pada perangkat sekaligus untuk memicu pengupdatean informasi koordinat ke aplikasi server via protokol HTTP.

Proses pemerolehan posisi koordinat pertama kali dengan mendefinisikan konfigurasi nilai dari obyek `LocationProvider` dengan periode interval waktu tertentu.

```
sec = (tfInterval.getString()!=null)?
Integer.parseInt(tfInterval.getString()):5;

//mulai pemrosesan request data GPS
new Thread(new Runnable(){
    public void run(){
        locationProvider.setLocationListener(LacakPosisiMidlet.this, sec,
        -1,
        -1);
    }
}).start();
```

Penentuan inisialisasi metode `setLocationListener()` dari obyek `LocationProvider` akan memicu eksekusi metode `locationUpdated()` secara periodik sesuai dengan interval yang telah ditentukan. Dengan demikian metode `locationUpdated()` dapat difungsikan untuk meminta informasi posisi koordinat kepada obyek `Location`. Adapun potongan source code programnya adalah sebagai berikut :

```
public void locationUpdated(LocationProvider provider, Location
location){
    if (location != null && location.isValid()){
        QualifiedCoordinates qc = location.getQualifiedCoordinates();
        siLokasi.setText(
            "\nLat: "+qc.getLatitude()+"\n"+
            "Lon: "+qc.getLongitude()+"\n"+
            "Alt: "+qc.getAltitude()+"\n"
        );

        //update posisi saat ini ke database server
        .....
    }
}
```

Dari data posisi yang diperoleh dari satelit GPS kemudian dikirimkan ke aplikasi server untuk merubah posisi terkini dari perangkat, yaitu dengan menggunakan kelas `HttpConnection`.

```
HttpConnection http = null;
try{
    String url = getAppProperty("AppServer")+ "?email="+emailstr+
        "&lat=" + qc.getLatitude()+
        "&lon=" + qc.getLongitude()+
        "&alt=" + qc.getAltitude();
    http = (HttpConnection) Connector.open(url);
    http.setRequestMethod(HttpConnection.GET);
    int rc = http.getResponseCode();
}catch(Exception e){
    e.printStackTrace();
}
```

Aplikasi Pemantau Lokasi (Server-side)

Di sisi server dibangun beberapa modul aplikasi yang terdiri **updatelokasi.php** yang difungsikan untuk menyimpan posisi koordinat terakhir dari terminal ke dalam database. Sedangkan antarmuka untuk menampilkan posisi titik koordinat terminal memanfaatkan pustaka dari Google Maps API. Adapun untuk menampilkan posisi titik koordinat (latitude, longitude dan altitude) terminal menggunakan format data berbasis JSON.

```
<?php
//namafile : viewloc.php

$cn = mysql_connect("localhost", "root", "");
mysql_select_db("test", $cn);
$query = "SELECT email, lat, lon, alt FROM lokasi ";
$hasil=mysql_query($query, $cn);
$info = '{"info" : [';
while ($baris = mysql_fetch_row($hasil)){
    $info .= sprintf("{\"email\" : \"%s\", \"lat\" : \"%s\", \"lon\" : \"%s\", \"alt\" : \"%s\"}", $baris[0], $baris[1], $baris[2], $baris[3]);
}
$info = substr($info, 0, strlen($info)-1);
$info .= "]}";
header('Content-Type: text/plain');

echo $info;
?>
```

Hasil pemrosesan file modul di atas akan diproses oleh obyek `GMap2` (pustaka *Google Maps API*) yaitu untuk menentukan titik lokasi dari perangkat dalam peta Google Maps. Inisialisasi obyek `GMap2` untuk menggambarkan peta posisi adalah sebagai berikut ini :

```

function load(){
  if (GBrowserIsCompatible()) {
    map = new GMap2(document.getElementById("map"));
    map.setCenter(new GLatLng(0,0), 14);
    map.addOverlay(new GStreetviewOverlay());
    map.scrollWheelZoomEnabled();
    setInterval('updateLocation()',3000);
  }
}

```

Instan obyek GMap2 diberikan identitas dengan nama “map” dengan memunculkan koordinat latitude dan longitude pada titik default (0,0). Untuk ukuran zoom peta adalah 14, kemudian diatur interval waktu tiga (3) detik untuk update lokasi secara periodik.

```

function stateChanged() {
  if (xmlHttp.readyState==4 || xmlHttp.readyState=="complete"){
    for (k=0; k<marker.length; k++){
      if (marker[k] != null) map.removeOverlay(marker[k]);
    }

    var json = eval('(' + xmlHttp.responseText + ')');

    var point=new Array();
    for (i=0; i<json.info.length; i++){

      point[i] = new GLatLng(Number(json.info[i].lat),
        Number(json.info[i].lon));
      marker[i] = new GMarker(point[i]);
      marker[i].bindInfoWindowHtml(json.info[i].email);
      map.addOverlay(marker[i]);
      if (i==0){
        map.panTo(point[i]);
      }
    }
  }
}

```

Data lokasi dari database kemudian dibentuk ke dalam format JSON (email, latitude, longitude dan altitude) dan diinputkan sebagai parameter pada obyek GMap2, hal ini berfungsi untuk menggeser penanda posisi koordinat obyek pada peta secara berkala sesuai dengan interval waktu yang telah ditentukan.

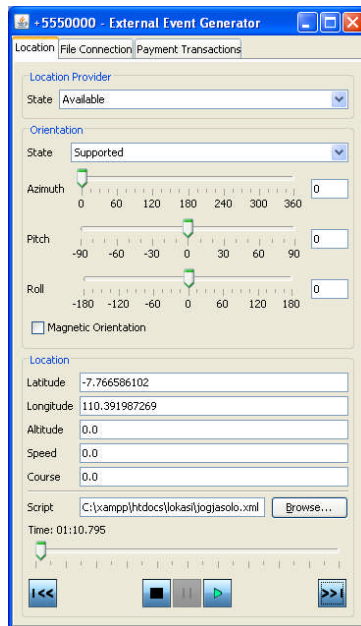
Pengujian Aplikasi

Untuk membuktikan aplikasi yang dibangun dapat berjalan sesuai dengan yang telah direncanakan dimanfaatkan tool emulator dari sisi aplikasi client dan webserver lokal untuk menjalankan

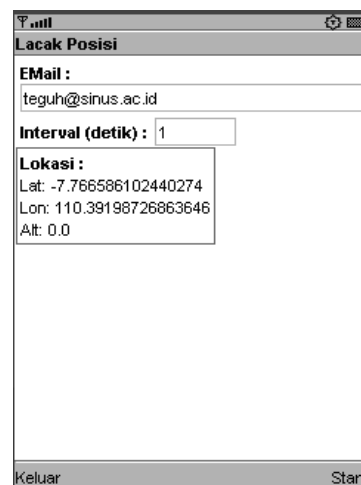
aplikasi server PHP. Sedangkan untuk data emulasi data koordinat posisi dibuat simulasi data koordinat dalam format XML.

```
<waypoints>
  <waypoint time="0" latitude="-7.758598263186935"
longitude="110.39562463760376" altitude="0" />
  <waypoint time="6466" latitude="-7.759363667168736"
longitude="110.39536714553833" altitude="0" />
  <waypoint time="8183" latitude="-7.7602353755621625"
longitude="110.39483070373535" altitude="0" />
  <waypoint time="6786" latitude="-7.7609795154436725"
longitude="110.39442300796509" altitude="0" />
  .....
</waypoints>
```

Uji coba aplikasi dilakukan dengan menjalankan aplikasi client dengan memanfaatkan data koordinat emulasi kemudian secara bersamaan aplikasi server dijalankan untuk memantau informasi posisi koordinat yang selalu bergerak seiring dengan perubahan posisi koordinat dari terminal. Berikut ini hasil *capture* tampilan program terlihat pada Gambar 2 dan 3.

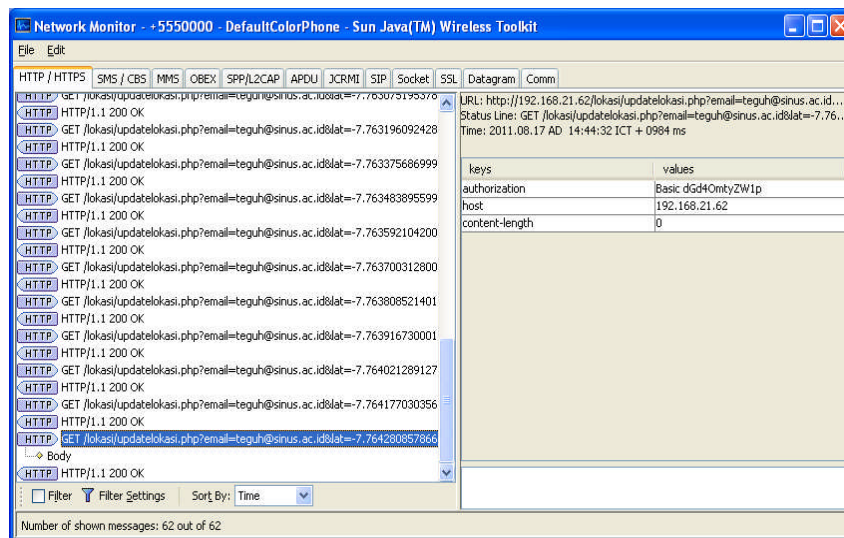


Gambar 2 Emulasi Data Posisi



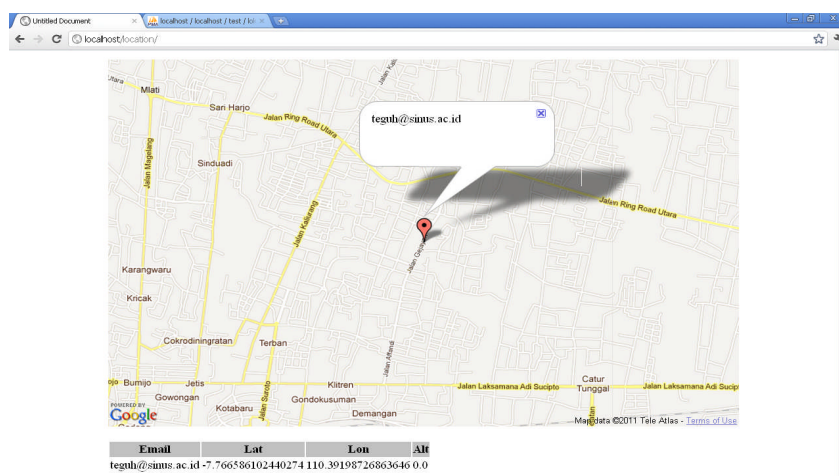
Gambar 3 Aplikasi Midlet (Client)

Setiap perubahan posisi koordinat akan dikirimkan ke aplikasi server melalui jaringan internet. Hasil rekaman gambar proses koneksi via protocol HTTP dapat dibaca pada gambar Gambar 4.



Gambar 4 Network Monitor

Perubahan data posisi perangkat yang telah simpan dalam database server 9Gambar 5) diamati pada antarmuka Peta pada aplikasi Pemantau posisi berbasis web yang dinstall dikomputer lokal (<http://localhost/location/index.html>).



Gambar 5 Aplikasi Sisi Server Pemantau Posisi

IV. PENUTUP

Hasil prototipe pembuatan sistem pelacakan posisi memanfaatkan dua antarmuka sebagai obyek yang menentukan posisi (perangkat seluler) dan antarmuka pemantau posisi (komputer desktop). Dua antarmuka ini mempertukarkan informasi posisi koordinat spasial melalui media jaringan internet.

Rekomendasi pengembangan prototipe aplikasi pelacakan posisi lebih lanjut yaitu dapat ditambahkan fitur perekaman jejak posisi alat.

Daftar Pustaka

- Andi Sunyoto, 2009, *API Location (JSR 179): Standar Penentuan Posisi untuk Telepon Seluler Berkemampuan Java*, JURNAL DASI ISSN: 1411-3201 Vol. 10
- Anonim, *Google Maps API Concepts*, <http://code.google.com/apis/maps/documentation/javascript/v2/basics.html>, diakses tanggal 18 Februari 2010
- Jonathan Knudsen, 2007, *Kicking Butt with MIDP and MSA: Creating Great Mobile Applications*, Prentice Hall
- JSR-179 Expert Group, 2003, *JSR-179: Location API for Java™ 2 Micro Edition v1.0 Final Release*, Nokia Cooperation.
- Kevin L Sally, 2006, *Develop a location-based service application using JSR 179*, IBM Cooperation
- Nico Goeminne., *Developing Location Based Services: Introducing the Location API for J2ME*, <http://mobiforge.com/developing/story/developing-locationbased-services-introducing-location-api-j2me> , diakses tanggal 17 februari 2010
- Qusay H. Mahmoud, 2001, *Learning Wireless Java*, O'Reilly.
- Qusay H. Mahmoud, 2004, *J2ME and Location-Based Services*, <http://developers.sun.com/mobility/apis/articles/location/> , diakses pada tanggal 17 februari 2011.