

# PERANCANGAN DAN PEMBUATAN APLIKASI TRACKER POSITION DAN ALERT PADA MOBILE DEVICE BERBASIS ANDROID YANG DAPAT DIPANTAU VIA WEBSITE

Anton Wahyudi<sup>1</sup>, Justinus Andjarwirawan<sup>2</sup>, Andreas Handojo<sup>3</sup>  
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra  
Jl. Siwalankerto 121 – 131 Surabaya 60236  
Telp. (031) – 2983455, Fax. (031) - 8417658

E-mail: m26409091@john.petra.ac.id<sup>1</sup>, justin@petra.ac.id<sup>2</sup>, handojo@petra.ac.id<sup>3</sup>

**ABSTRAK:** Saat ini jaman sudah sangat maju terutama di bidang Ilmu Pengetahuan dan Teknologi, sehingga menjadikan hubungan antar manusia menjadi lebih dekat dan mudah serta tidak dibatasi oleh jarak karena adanya *device* atau *smartphone* yang semakin canggih dan populer. Android merupakan salah satu sistem operasi yang ditanamkan dalam *device* dan *smartphone* tersebut. Menyadari semakin pentingnya *device* tersebut untuk menunjang aktivitas sehari - hari maka dibuatlah aplikasi untuk melacak serta mengontrol *device* yang dapat dilakukan oleh pemilik apabila terjadi kehilangan.

Aplikasi ini dapat mengambil lokasi geografis pada *device*. Informasi posisi yang dihasilkan dapat digunakan untuk memantau serta mencari *device* jika terjadi kehilangan dan didukung dengan fitur – fitur untuk mengontrol *device* dari jarak jauh melalui suatu *website* atau dengan mengirimkan SMS, sehingga membantu mempermudah pemilik untuk mencari keberadaan *device* tersebut.

Informasi posisi yang didapat dari GPS / A-GPS pada *device* akan ditampilkan pada *website* menggunakan Google Maps. Dari hasil pengujian, informasi posisi dipengaruhi oleh ketersediaan layanan operator, kekuatan sinyal operator, dan ketersediaan kemampuan *device*.

**Kata Kunci:** *Location based service, Tracker, Android.*

**ABSTRACT:** Today we are in the modern age especially in Science and Technology, that make people in the world can make a good and nice relationship even they are separate by distance and time, with a device or smartphone which now are more advanced. Android is one of the many operating system in smartphone and device. As we see how important our device now in this age, to support our daily activity, then created an application to seek or track and control owner's device that can be used by them if they lost their device.

This application can take the geographic location on your device. That information can seek and monitor the lost device and supported by another features that can control the device remotely by using the web that have been prepared for this application or by sending SMS from it so it makes the owners easier to locate and find their lost device.

The device position information that obtained from GPS / A-GPS on device will be displayed to web using google maps. From the test results, the position information is depend on the provider service availability, signal strength, and the availability of their device.

**Keywords:** Location based service, Tracker, Android.

## 1. PENDAHULUAN

Saat ini jaman sudah sangat maju terutama di bidang Ilmu Pengetahuan dan Teknologi, sehingga menjadikan hubungan antar manusia menjadi lebih dekat dan mudah serta tidak dibatasi oleh jarak karena adanya *device* atau *smartphone* yang semakin canggih dan populer.

Perkembangan tersebut terjadi karena adanya evolusi pada *operating system* yang ditanamkan pada *device* tersebut, salah satunya yang sedang populer saat ini adalah Android. *Operating system* ini dikembangkan oleh Google dan memiliki banyak keunggulan dibandingkan dengan *operating system* lain yang telah terlebih dahulu digunakan pada *device* generasi sebelumnya. Salah satu kelebihan utama terletak pada sifatnya yang *open source*, sehingga semua orang dapat bebas mengembangkannya secara gratis. Seiring dengan kelebihan dan jumlah aplikasi yang semakin banyak mengakibatkan jumlah *user* Android pun meningkat dengan drastis. Pada umumnya *device* tersebut sering digunakan untuk menghilangkan kejenuhan dengan memainkan *game* yang ada dan digunakan untuk mendukung aktivitas, sehingga perannya semakin sulit dipisahkan dari kehidupan sehari - hari.

Menyadari semakin pentingnya *device* tersebut untuk menunjang aktivitas sehari - hari maka dibutuhkan sebuah aplikasi yang dapat melacak serta memberi tahu posisi *device* kepada pemilik apabila terjadi kehilangan. Selain itu pada aplikasi tersebut juga terdapat fitur tambahan yang dapat membuat *device* berbunyi untuk mengetahui jika lokasinya berada di sekitar dan fitur untuk menampilkan pesan ke layar *device* tersebut dengan bantuan dari *website* yang dapat diakses melalui koneksi *internet* terdekat.

## 2. LANDASAN TEORI

### 2.1 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat *software* untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan *hardware, software, dan*

telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilis perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di pihak lain, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi *software* dan *open platform* perangkat seluler.

Pada masa saat ini sebagian besar *vendor-vendor smartphone* sudah memproduksi *smartphone* berbasis Android, seperti HTC, Motorola, Samsung, LG dan masih banyak *vendor* lainnya. Hal ini disebabkan karena Android adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh *vendor* manapun [1].

### 2.2.1 Arsitektur Android

Arsitektur Android terdiri dari: *Applications*, *Application Framework*, *Libraries*, *Android Runtime* dan *Kernel Linux*. Arsitektur lengkap *platform* ini dapat dilihat pada Gambar 1 [2].



Gambar 1. Arsitektur Android

### 2.2.2 The Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine (DVM)*. Android berjalan di dalam DVM bukan di *Java Virtual Machine (JVM)*. DVM adalah “*register based*” sementara JVM adalah “*stack based*”, DVM. *Dalvik Virtual Machine* menggunakan *kernel Linux* untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual Machine* untuk menjalankan aplikasi [3].

### 2.2.3 Fundamental Aplikasi Android

Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan *data file resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di-*package* oleh *tools* yang dinamakan *apt tools* ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File itulah yang kita sebut dengan aplikasi, dan nantinya dapat kita *install* di perangkat

*mobile*. Ada beberapa jenis komponen pada aplikasi Android, yaitu:

- *Activities*

Suatu *activity* akan menyajikan *user interface (UI)* kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface (UI)* saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain kita dapat melakukannya dengan satu *event*, misalnya click tombol, memilih opsi atau menggunakan *triggers* tertentu. Secara hirarki sebuah *windows activity* dinyatakan dengan *method Activity setContentView()*. *ContentView* adalah *object* yang berada pada *root* hirarki.

- *Service*

*Service* tidak memiliki *Graphic User Interface (GUI)*, tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan *music*, *service* mungkin memainkan *music* atau mengambil *data* dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya, *media player* sedang memutar lagu dari *list* yang ada, Aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu, atau menulis sms sambil *player* sedang jalan. Untuk menjaga *music* tetap di jalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada *thread* utama dari *proses* aplikasi.

- *Broadcast Receiver*

*Broadcast Receiver* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh *camera*, atau perubahan referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi *broadcast* misalnya memberikan informasi pada aplikasi lain bahwa ada *data* yang telah diunduh ke perangkat dan siap untuk digunakan.

*Broadcast Receiver* tidak memiliki *user interface (UI)*, tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya.

- *Content Provider*

*Content provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti *database SQLite*. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta (*Map*), atau aplikasi yang

membutuhkan untuk mengakses *data* kontak dan navigasi, maka disinilah fungsi *content provider* [4].

### 2.2.4 Google Cloud Messaging

*Google Cloud Messaging* (GCM) adalah salah satu *service* yang ada pada *operating system* Android. *Service* ini dapat membantu pengembang mengirimkan data dari *server* menuju ke aplikasi Android. Pesan yang dikirimkan adalah pesan singkat yang memberitahu aplikasi bahwa ada data baru yang harus diambil dari *server* atau pesan yang berupa *payload data* (Pesan yang dapat langsung dikonsumsi oleh aplikasi). GCM dapat menangani semua aspek dari antrian pesan dan mengirimkannya ke aplikasi yang sedang berjalan di *device target*. GCM dapat digunakan secara gratis tidak peduli berapa jumlah pesan yang dibutuhkan dan tidak ada kuota yang ditentukan [5].

### 2.2.5 Google Maps

*Google Maps* adalah layanan aplikasi peta *online* yang disediakan oleh Google secara gratis. Layanan peta *Google Maps* secara resmi dapat diakses melalui situs <http://maps.google.com>. Pada situs tersebut dapat dilihat informasi geografis pada hampir semua permukaan di bumi kecuali daerah kutub utara dan selatan. Layanan ini dibuat sangat interaktif, karena di dalamnya peta dapat digeser sesuai keinginan pengguna, mengubah level *zoom*, serta mengubah tampilan jenis peta.

*Google Maps* mempunyai banyak fasilitas yang dapat dipergunakan misalnya pencarian lokasi dengan memasukkan kata kunci, kata kunci yang dimaksud seperti nama tempat, kota, atau jalan, fasilitas lainnya yaitu perhitungan rute perjalanan dari satu tempat ke tempat lainnya [6].

### 2.2.6 Google Maps API

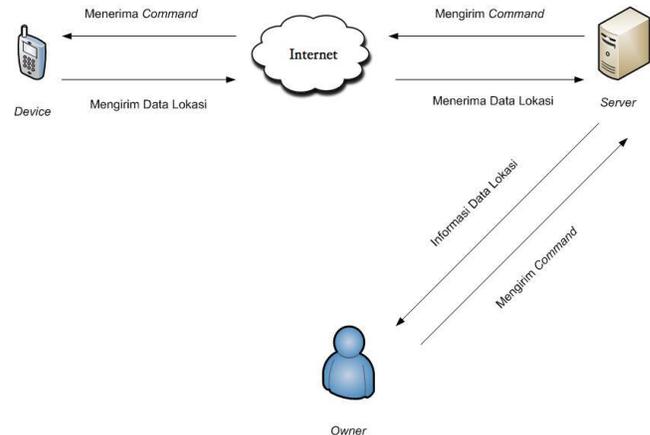
*API* atau *Application Programming Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya *API* ini, maka memudahkan programmer untuk “membongkar” suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. *API* dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan sistem *function*. Proses ini dikelola melalui *operating system*. Keunggulan dari *API* ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh *Google Maps* yang terdiri dari *HTML*, *Javascript* dan *AJAX* serta *XML*, memungkinkan untuk menampilkan peta *Google Maps* di *website* lain.

Google juga menyediakan layanan *Google Maps API* yang memungkinkan para pengembang untuk mengintegrasikan *Google Maps* ke dalam *website* masing-masing dengan menambahkan data *point* sendiri. Dengan menggunakan *Google Maps API*, *Google Maps* dapat ditampilkan pada *website* eksternal. Agar aplikasi *Google Maps* dapat muncul di *website* tertentu, diperlukan adanya *API key*. *API key* merupakan kode unik yang digenerasikan oleh Google untuk suatu *website* tertentu, agar *server Google Maps* dapat mengenali [6].

## 3. DESAIN SISTEM

### 3.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem berisi tentang arsitektur dari sistem yang akan diterapkan pada aplikasi. Perancangan arsitektur sistem pada aplikasi ini dapat dilihat pada Gambar 2.



Gambar 2. Perancangan Arsitektur Sistem

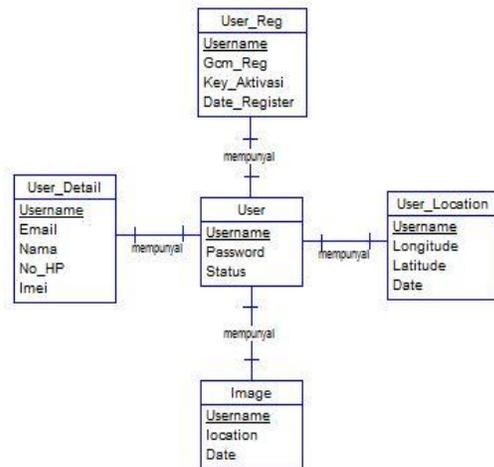
Pada Gambar 2 dapat dilihat bahwa *server* berperan penting sebagai perantara antara *owner* dengan *device*. *Owner* mengirimkan *command* yang untuk dilakukan oleh *device* melalui *server*, kemudian *server* akan meneruskan *command* tersebut ke *device* yang dituju melalui koneksi internet. Ketika *command* telah tiba pada *device*, maka *device* akan mengirimkan data yang diminta ke *server* melalui koneksi internet sehingga *owner* dapat menerima data yang diinginkan tersebut.

### 3.2 Garis Besar Aplikasi Pada Device

Garis besar aplikasi yang di *install* pada *device* dapat dilihat pada Gambar 3.



desain *database* aplikasi. Ada dua jenis model data, yaitu *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM). CDM adalah model yang memperlihatkan entitas dan relasinya. Sedangkan PDM adalah model yang memperlihatkan gambaran *database* secara *detail*, dimana hubungan antar tabel telah terlihat jelas. Desain CDM dapat dilihat pada Gambar 5.



Gambar 5. ERD Conceptual Data Model (CDM)

## 4. IMPLEMENTASI SISTEM

### 4.1 Implementasi Perangkat Lunak

Pembuatan aplikasi pada skripsi ini terdiri dari dua bagian utama, yaitu aplikasi pada *server* dengan PHP, AJAX, dan Javascript, serta aplikasi pada *device* dengan Eclipse IDE yang menggunakan Java.

*Server* berfungsi sebagai salah satu media bagi *owner* untuk mengirimkan *command* serta menyediakan informasi yang diminta dari Android untuk ditampilkan pada *website*. Pertukaran data antara *server* dan *client* (Android) menggunakan metode HTTP. Sedangkan Eclipse IDE berfungsi menghubungkan antara *owner* dengan keseluruhan fitur/fungsi yang dimiliki oleh aplikasi berbasis Android. Pembuatan aplikasi Android pada Eclipse IDE dijalankan pada Android dengan *minimum* SDK 14 (Ice Cream Sandwich).

### 4.2 Aplikasi Server

Aplikasi pada *server* digunakan untuk merespon permintaan dari *client* (Android) serta berfungsi untuk media menampilkan informasi yang diinginkan oleh *owner*. Aplikasi pada *server* ini berupa *website* dan *function* untuk mengakses *database*.

### 4.3 Aplikasi Client Android

Pada *class* aplikasi *client* Android terdapat beberapa komponen utama pada aplikasi, yaitu *Activity* yang berhubungan dengan *user interface* aplikasi, *Service* yang berguna untuk

menjalankan *function* pada *background* sehingga tidak membutuhkan *user interface*.

## 5. PENGUJIAN SISTEM

### 5.1 Spesifikasi Device Pada Pengujian

Aplikasi diujicobakan pada *mobile device* Samsung Galaxy Tab 2 10.1 (P5100) OS Android 4.1.1 dengan RAM 1 GB, *mobile device* Samsung Galaxy Tab 7.7 (P6800) OS Android 4.0.3 dengan RAM 1 GB, *mobile device* Samsung Galaxy Tab 7.0 (P6200) OS Android 4.0.3 dengan RAM 1 GB.

### 5.2 Pengujian Fitur Pada Aplikasi

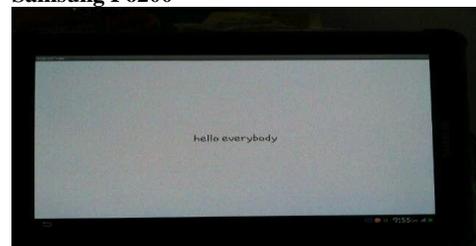
Pada bagian ini akan dibahas mengenai pengujian fitur yang tersedia dari aplikasi pada beberapa *device* yang berbeda. Pengujian fitur pada aplikasi dapat dilihat pada tabel berikut ini :

Tabel 1. Tabel Daftar Pengujian

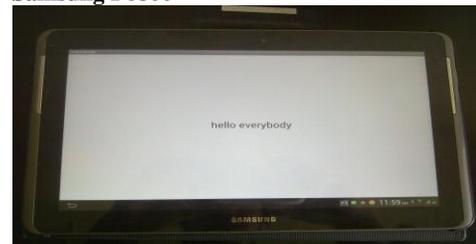
No.	Jenis Pengujian	Keterangan
1	Pengujian <i>Send Alert</i>	Gambar 6
2	Pengujian <i>Lock Screen</i>	Gambar 7
3	Pengujian <i>Hide Icon</i>	Gambar 8
4	Pengujian <i>Take Picture</i>	Gambar 9



Samsung P6200



Samsung P6800



Samsung P5100

Gambar 6. Pengujian *Send Alert*



Samsung P6200



Samsung P6800



Samsung P5100

Gambar 7. Pengujian Lock Screen



Samsung P6200



Samsung P6800



Samsung P5100

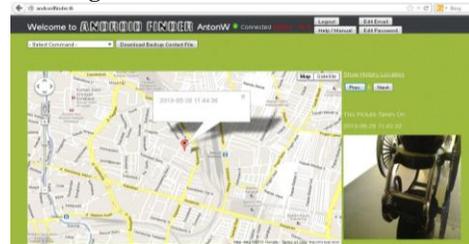
Gambar 8. Pengujian Hide Icon



Samsung P6200



Samsung P6800



Samsung P5100

Gambar 9. Pengujian Take Picture

## 6. KESIMPULAN

Dari hasil pengujian aplikasi, dapat disimpulkan beberapa hal sebagai berikut:

- Aplikasi pada *device* dapat mengambil koordinat lokasi geografis *device* tiap satuan waktu tertentu.
- Keberhasilan aplikasi membaca koordinat lokasi geografis dipengaruhi oleh ketersediaan layanan operator, kekuatan sinyal operator, kekuatan sinyal GPS, dan ketersediaan fitur pada *device*.
- Koordinat geografis dapat dibaca dengan akurat berdasarkan pada hasil pengujian keakuratan koordinat geografis pada kota Surabaya.
- Aplikasi membutuhkan sinyal dan koneksi data yang stabil agar proses pengiriman data koordinat ke *server* berhasil.
- Aplikasi dan semua fitur dapat berjalan lancar pada *device* yang sudah di *root* dan tidak di *root*.
- Pengiriman *command* via *website* yang menggunakan GCM membutuhkan waktu cukup lama dan tidak konsisten bergantung kepada jaringan internet.
- Aplikasi pada *device* tidak membutuhkan banyak *resource* baterai.

## 7. DAFTAR PUSTAKA

- [1] Safaat, Nazruddin. (2011). *Pemrograman aplikasi mobile smartphone dan tablet pc*.
- [2] Meier, Reto (2009). *Professional Android™ Application Development*.
- [3] Zakas, N. C. , Fawcett, J. & McPeak, J. (2007) . *Proffesional ajax* (2nd ed). Indianapolis: Wiley Publishing, Inc.
- [4] Rogers, Rick; Lombardo, John; Mednieks, Zigurd; Meike, Blake (2009). *Android Application Development*.
- [5] Android. (2006). *Software Development Kit* Retrieved January 10, 2013, from <http://developer.android.com/google/gcm/index.html>.
- [6] Amri, Syaiful. (2011). *Membangun Sistem Navigasi di Surabaya Menggunakan Google Maps API*. Retrieved October 23, 2012, from [www.eepis-its.edu/uploadta/downloadmk.php?id=1563](http://www.eepis-its.edu/uploadta/downloadmk.php?id=1563).