

# Perancangan dan Pembuatan Aplikasi Pencarian dan Navigasi Fasilitas Umum Berbasis Android

Dimas Adiputra Gunawan<sup>1</sup>, Gregorius Satia Budhi<sup>2</sup>, Rolly Intan<sup>3</sup>  
Program Studi Teknik Informatika. Fakultas Teknologi Industri Universitas Kristen Petra

Siwalankerto 121-131, Surabaya

Email : w3nh4u@gmail.com<sup>1</sup>, greg@petra.ac.id<sup>2</sup>, rintan@petra.ac.id<sup>3</sup>

## ABSTRAK

Saat ini *smartphone* berkembang dengan sangat pesat. Salah satunya *smartphone* dengan sistem operasi Android. Saat ini semakin banyak para pengguna yang menggunakan *smartphone* dengan sistem operasi Android ini. Hal tersebut juga ditunjang dengan perkembangan aplikasi yang tersedia untuk Android. Kita dapat melihat pada Google Play, sangat banyak aplikasi Android yang tersedia dan dapat digunakan oleh pengguna.

Salah satu contoh aplikasi yang cukup banyak diminati adalah aplikasi dengan fitur pencarian tempat - tempat yang sering dikunjungi. Misalnya rumah makan, rumah sakit, hotel, dan sebagainya. Tapi masih sangat kurang aplikasi yang menyediakan fitur tersebut untuk di Indonesia.

Aplikasi CityNavi ini dibuat dengan dasar sistem operasi Android. Dengan menggunakan aplikasi ini, pengguna dapat mencari tempat - tempat seperti rumah sakit, hotel dan SPBU. Selain itu juga, pengguna dapat memberi komentar, nilai, menambah data tempat yang baru, serta mengunggah foto.

Dan setelah dilakukan uji coba, aplikasi ini bisa mendapatkan lokasi pengguna dengan menggunakan GPS dari ponsel pengguna dan dapat mencari lokasi fasilitas umum yang berada di sekitar pengguna.

## Kata Kunci

*Point of Interest, PhoneGap, Navigation, Android.*

## ABSTRACT

Recently smartphone is developing rapidly. One of them is the smartphone with Android as the operating system. Nowadays many people are using this Android operating system smartphone. It is also supported by developed application's available for Android. We see at Google Play, for example, there are many Android applications available that can be accessed by users.

For example, one of the application which is many people are interested in the applications with point of interest searching features, such as, restaurants, hotels, hospitals, etc. As a matter of fact, in Indonesia, is still lack of application with those features.

This CityNavi application was made based on the Android operating system. By using this application, users can search point of interests. They can also give comment, score, adding data for a new place, and upload photos.

And after the trials, this application can get user's location by using GPS from user's smartphone and search for point of interest near user's location.

## Keywords

*Point of Interest, PhoneGap, Navigation, Android.*

## 1. PENDAHULUAN

*Location Based Service* (LBS) merupakan sebuah jenis aplikasi yang banyak dipakai dan diperlukan oleh pengguna ponsel. Melalui LBS, dapat dihasilkan aplikasi pencarian *Point of Interest*, aplikasi navigasi, aplikasi pencari ponsel, dan lain sebagainya.

Kegunaan dari aplikasi CityNavi adalah pencarian *point of interest* dan memberikan arah jalan kepada pengguna untuk mencapai lokasi yang diinginkan. Pada aplikasi CityNavi, dapat dilihat juga berapa jarak yang ditempuh oleh pengguna untuk mencapai lokasi dari *point of interest* yang diinginkan.

Selain itu, pengguna juga dapat memberi komentar, memberi nilai, menambahkan data *point of interest* yang belum terdapat dalam *database* CityNavi, serta dapat menambahkan foto ke dalam *gallery*.

Pembuatan aplikasi CityNavi menggunakan *framework PhoneGap*. Sehingga meskipun dibuat dengan basis Android, namun aplikasi CityNavi dapat dikembangkan pada sistem operasi selain Android.

## 2. LANDASAN TEORI

### 2.1 PhoneGap

PhoneGap adalah salah satu *framework* yang *open source* yang memungkinkan aplikasi berbasis *web* (html) menjadi aplikasi *native*. PhoneGap menyediakan *framework* untuk membuat aplikasi *native* untuk semua jenis *platform* sistem operasi *smartphone* atau *tablet* yang ada (Android, iOS, Blackberry, Windows Phone 7, Symbian, dll.) [1].

### 2.2 Location Based Service (LBS)

*Location Based Service* (LBS) adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang digunakan.

Dua unsur utama LBS [1] :

1. *Location Manager (API Maps)*  
Menyediakan *tools / source* untuk LBS, *Application programming Interface (API Maps)* menyediakan fasilitas untuk menampilkan, memanipulasi peta beserta fitur - fitur lainnya seperti tampilan satelit, jalan (*streets*), maupun gabungannya (*hybrid*).
2. *Location Providers (API Location)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. API *Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini, *Track* gerakan / perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

### 2.3 Spherical Law of Cosines

Perhitungan jarak antara 2 posisi dalam peta, digunakan teori *Spherical Law of Cosines*. Data posisi yang digunakan adalah nilai dari *longitude* dan *latitude* dari pengguna dan tempat yang dipilih.

Rumus dari *Spherical Law of Cosines* yang digunakan dalam perhitungan *longitude* dan *latitude* dapat dilihat pada Segmen Program 1 [2].

Segmen Program 1. *Spherical Law of Cosines*

```
d = acos (sin(lat1).sin(lat2) + cos(lat1).cos(lat2).cos(lon2-lon1)).6371
```

Jika ditulis ke dalam bahasa *javascript*, rumus tersebut akan menjadi seperti pada Segmen Program 2.

Segmen Program 2. *Spherical Law of Cosines* dalam *javascript*

```
var R = 6371; // km
var d = Math.acos(Math.sin(lat1)*Math.sin(lat2) +
    Math.cos(lat1)*Math.cos(lat2) *
    Math.cos(lon2-lon1) * R;
```

### 2.4 jQuery Mobile

*jQuery Mobile* adalah satu *set plugin* dan *widget* dari *jQuery* yang bertujuan untuk menyediakan API dengan berbagai macam *platform* untuk membuat aplikasi *web mobile*. *jQuery Mobile* merupakan *framework* berbasis HTML5 / CSS3 dan *javascript* untuk aplikasi *web* yang dijalankan di perangkat *mobile*.

Untuk menjalankan *jQuery Mobile* di *smartphone* bisa dilakukan dengan dua cara, yaitu [3] :

1. Memanggil *jQuery Mobile* dari *server* (*file jQuery Mobile* berada di *server*).
2. Memanggil *jQuery Mobile* dimana *jQuery Mobile* sudah berada dalam *package* aplikasi *project* yang dijalankan.

### 2.5 JavaScript Object Notation (JSON)

JSON adalah sebuah format *data-interchange* yang ringan. JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, *Java*, *JavaScript*, *Perl*, *Phyton*, dan sebagainya [4].

JSON dibangun dalam dua struktur, yaitu :

1. Beberapa pasangan dari nama / nilai. Dalam beberapa bahasa pemrograman biasa disebut dengan istilah *object*, *record*, *struct*, *tabel hash*, *key list* atau *associative array*.

2. Nilai – nilai tersusun secara *ordered list*. Biasa disebut dengan *array*, *vector*, *list* atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang universal, bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON.

### 2.6 Hypertext Preprocessor (PHP)

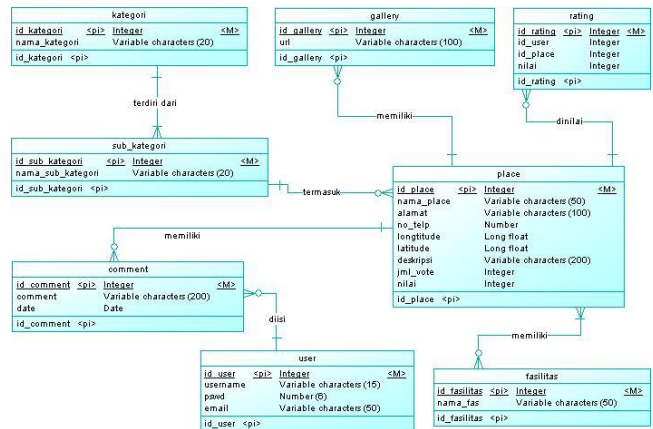
PHP dikenal sebagai bahasa pemrograman yang kodenya dijalankan di sisi *server*. Dengan demikian kode aslinya tidak akan terlihat pada klien (*browser*). PHP banyak dipakai dalam membuat aplikasi *web* antara lain karena dengan dukungan pustaka yang lengkap dan mudah digunakan pada berbagai *platform*. Dengan menggunakan PHP, koneksi ke *database server* juga mudah dilakukan [5].

## 3. DESAIN DATABASE

Perancangan *Conceptual Data Model* dapat dilihat pada Gambar 1.

1. Dalam *Conceptual Data Model* terdapat 8 tabel, yaitu :

1. kategori : menampung data kategori yang ada.
2. sub\_kategori : menampung data *sub-kategori* dari kategori yang ada.
3. *user* : menampung semua data pengguna yang mendaftar pada *CityNavi*.
4. *comment* : menampung data komentar yang diberikan oleh pengguna.
5. *gallery* : menampung data *url* dari setiap foto yang diunggah oleh pengguna.
6. *rating* : menampung data nilai – nilai terhadap suatu tempat yang diberikan oleh pengguna.
7. fasilitas : menampung data fasilitas yang dimiliki oleh suatu tempat.
8. *place* : menampung data dari semua tempat *point of interest* yang ada di dalam aplikasi *CityNavi*.



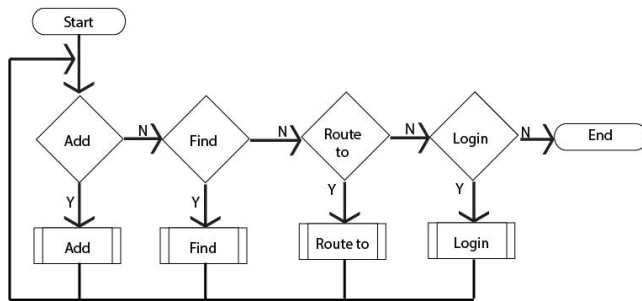
Gambar 1. *Conceptual Data Model*

## 4. ALUR PROGRAM

Secara umum, alur aplikasi *CityNavi* dapat dilihat pada Gambar 2. Pada saat pertama kali aplikasi *CityNavi* dibuka, halaman awal

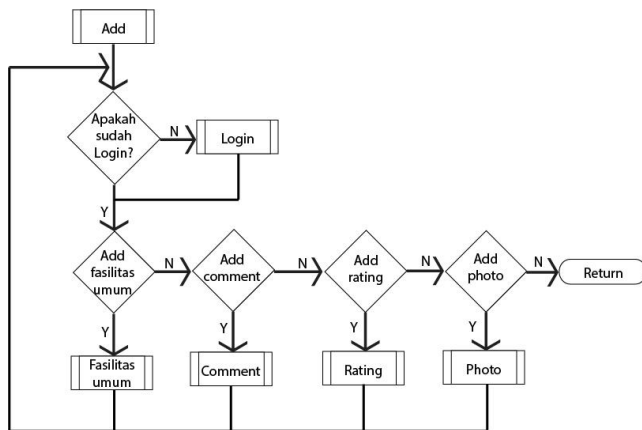
berisi sebuah peta yang menunjukkan posisi pengguna saat itu. Terdapat 5 menu dalam CityNavi, yaitu :

1. *Refresh* : berfungsi untuk memperbaharui halaman awal. Jika pengguna berpindah lokasi, maka lokasi pada peta akan berpindah saat tombol *refresh* ditekan.
2. *Login* : berfungsi bagi para pengguna yang telah mendaftar dan ingin menggunakan fitur – fitur yang membutuhkan *login* terlebih dahulu.
3. *Add* : fitur untuk menambahkan data baru ke dalam *database*. Pengguna dapat menambahkan data tempat baru yang belum terdaftar di dalam *database* melalui tombol ini.
4. *Find* : fitur untuk mencari tempat yang berada disekitar pengguna.
5. *Route to* : fitur untuk menuntun pengguna ke lokasi yang telah dipilih oleh pengguna.



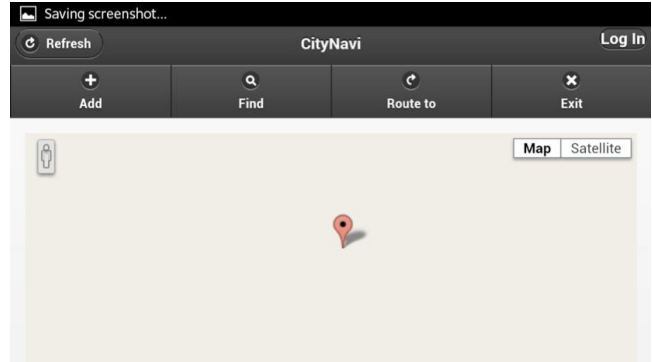
Gambar 2. Flowchart alur CityNavi secara umum

Untuk *flowchart* menu *add*, terbagi lagi dalam beberapa pembagian, pembagian tersebut dapat dilihat pada Gambar 3.



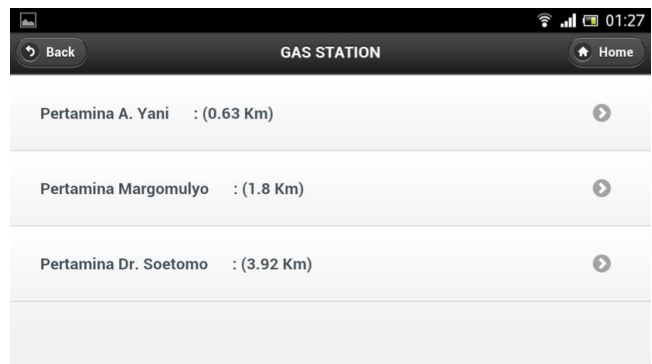
Gambar 5. Flowchart alur *add* secara umum

Sedangkan untuk tampilan halaman utama pada aplikasi CityNavi dapat dilihat pada Gambar 4.



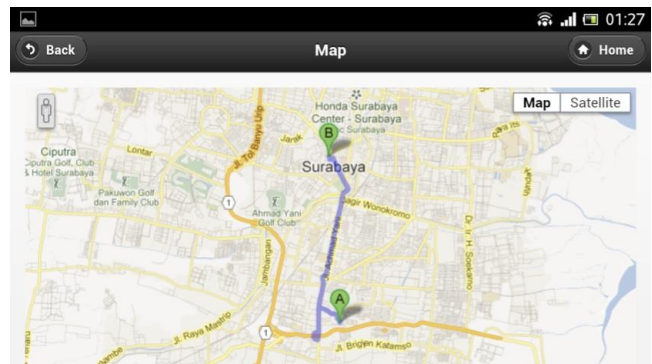
Gambar 4. Halaman Utama CityNavi

Pengguna dapat juga mencari tempat fasilitas umum yang berada di sekitar pengguna. Saat melakukan pencarian, aplikasi CityNavi menampilkan tempat fasilitas umum apa saja yang berada di sekitar pengguna dan berapa jauh jarak antara pengguna dengan tempat tersebut. Hasil pencarian tersebut dapat dilihat pada Gambar 5.



Gambar 5. Hasil Pencarian Fasilitas Umum

Setelah mengetahui fasilitas umum apa saja yang berada disekitar pengguna, pengguna juga dapat melihat peta lokasi dimana fasilitas umum tersebut berada dan jalan mana yang bisa ditempuh oleh pengguna untuk dapat mencapai lokasi tersebut, seperti terlihat pada Gambar 6. Pada peta tersebut menunjukkan A sebagai lokasi pengguna dan B sebagai lokasi tujuan yang dituju.



Gambar 6. Rute Lokasi Tujuan

## 5. PENGUJIAN

### 5.1 Pengujian pada Ponsel

Pengujian dilakukan dengan beberapa jenis ponsel dan versi Android yang berbeda. Hasil dari pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian

Ponsel	Versi Android	Hasil
Sony Xperia Acro S	4.0.4 (Ice Cream Sandwich)	OK
Samsung Galaxy W	2.3.6 (Gingerbread)	Pemilihan hasil pencarian tempat hanya 1 kali
Samsung Galaxy Tab 7.7 P6800	3.2 (Honey Comb)	OK
Samsung Galaxy Tab 2 7.0 P3100	4.0.4 (Ice Cream Sandwich)	OK
Axioo Vigo	2.3.5 (Gingerbread)	Pemilihan hasil pencarian tempat hanya 1 kali
HTC One X	4.1.1 (Jelly Bean)	OK
Samsung Galaxy Tab 7.7 P6800	4.1.2 (Jelly Bean)	OK

Pada tabel hasil pengujian dapat dilihat, pada saat aplikasi dijalankan pada ponsel dengan versi Android Gingerbread, aplikasi hanya berjalan lancar pada saat pertama kali percobaan. Pada saat kembali ke halaman utama dan memilih hasil pencarian, pengguna hanya bisa melihat hasil pencarian yang ada. Pengguna tidak dapat masuk ke dalam menu dari hasil pencarian tersebut. Jika pengguna mencoba untuk menekan dan memilih hasil pencarian tersebut, tidak akan terjadi apa – apa.

Sedangkan pada ponsel dengan versi Android Honey Comb ke atas, Ice Cream Sandwich dan Jelly Bean, aplikasi berjalan dengan lancar dan tidak didapati ada masalah seperti yang terjadi pada Gingerbread.

### 5.2 Pengujian Native dan PhoneGap

Pengujian ini dilakukan untuk membuktikan mana yang lebih responsive, aplikasi yang dibuat secara Native atau PhoneGap. Pengujian dilakukan menggunakan ponsel Sony Xperia Acro S, dengan aplikasi sederhana yang hampir sama. Aplikasi hanya berisi tombol yang jika ditekan, akan membuka halaman baru. Di

sini pengujian yang dilakukan adalah melihat berapa waktu yang dibutuhkan untuk membuka halaman baru tersebut. Hasil dari pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Native dan PhoneGap

Pengujian	Native	PhoneGap
1	352 ms	560 ms
2	400 ms	608 ms
3	384 ms	752 ms
4	368 ms	512 ms
5	306 ms	688 ms

Dari hasil pengujian dapat dilihat, di mana aplikasi yang dibuat dengan PhoneGap membutuhkan waktu yang lebih lama untuk membuka halaman yang baru. Hal ini dikarenakan pada PhoneGap berbasis web, sehingga saat akan membuka halaman baru, PhoneGap akan me-load dahulu halaman yang ingin ditampilkan. Sedangkan Native, langsung memanggil dan memprosesnya.

## 6. KESIMPULAN

1. Dibutuhkan koneksi internet untuk menampilkan peta dari google map.
2. Waktu yang dibutuhkan dalam proses pencarian dan penambahan tergantung pada koneksi internet.
3. CityNavi dibuat dengan menggunakan PhoneGap, sehingga jika dibandingkan dengan aplikasi yang dibuat secara *native*, CityNavi akan terasa kurang *responsive*.
4. Dapat berjalan dengan ukuran layar yang berbeda tanpa merusak tampilan yang telah dibuat, terbukti dengan pengujian pada beberapa ponsel dengan ukuran layar yang berbeda.

## 7. DAFTAR PUSTAKA

- [1]. Lunny, Andrew. (2011). *PhoneGap Beginners Guide*. Brimingham : Packt Publishing Ltd.
- [2]. *Calculate distance, bearing, etc. Between 2 Latitude/Longitude points*. Retrieved Juni 14, 2013. from <http://www.yourhomenow.com/house/haversine.html>.
- [3]. Reid, Jon. (2011). *jQuery Mobile*. Sebastopol: O'Reilly Media, Inc.
- [4]. *JSON*. Retrieved Maret 09, 2013. from <http://www.json.org>
- [5]. Kadir, Abdul. (2011). *Buku Pintar jQuery dan PHP*. Yogyakarta: MediaKom.