

# Aplikasi Pencarian Lokasi Fasilitas Umum Berbasis Foursquare APIv2 pada Sistem Operasi Android

Nur Rokhman<sup>\*1</sup>, Iqnatius Dimas Nugroho<sup>2</sup>

<sup>1,2</sup>Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta  
e-mail: <sup>\*1</sup>nurrokhman@gmail.com

## Abstrak

*Sebuah smartphone umumnya dilengkapi dengan Global Positioning System (GPS). Pengguna smartphone disamping dapat mengetahui lokasi dirinya, umumnya juga ingin mengetahui lokasi sekitarnya. Foursquare merupakan salah satu jejaring sosial yang menyediakan layanan berbasis lokasi. Foursquare memiliki fitur check-in, untuk menandai lokasi pengguna.*

*Dalam penelitian ini akan dikembangkan aplikasi pada perangkat dengan sistem operasi Android yang dapat mencari lokasi fasilitas umum di sekitar pengguna dengan memanfaatkan teknologi layanan berbasis lokasi. Aplikasi ini memanfaatkan data dari Foursquare.*

*Hasil pengujian terhadap aplikasi yang dibangun menunjukkan filter data dan sistem auto check-in berjalan dengan baik sehingga duplikasi data dalam Foursquare dapat diminimalkan.*

**Kata kunci**— Layanan berbasis lokasi, Android, Foursquare, Fasilitas umum

## Abstract

*A smartphone is equipped with Global Positioning System (GPS). A smartphone user may know the location itself, and usually want to know the surrounding location. Foursquare is a social network that provide location-based services. Foursquare has a check-in feature to mark the location of the user.*

*This research develops applications on devices with Android operating system that can find location of public facilities around the user by using location-based services technology. This application uses Foursquare data.*

*The test results showed that the application can filter data and the check-in systems running properly such that duplication of data in Foursquare can be minimized.*

**Keywords**— Location Based Services (LBS), Android, Foursquare, Public Facility

## 1. PENDAHULUAN

Perkembangan *smartphone* dewasa ini sangat pesat. Salah satu fitur *smartphone* adalah *Global Positioning System* (GPS) yang memungkinkan pengguna mengetahui koordinat lokasi. GPS berperan dalam memberi koordinat lokasi yang digunakan untuk mengikatkan data spasial ke dalam sistem koordinat dan data global [1]. GPS dapat dimanfaatkan untuk pengembangan aplikasi berbasis lokasi (*Location Based Services/LBS*). Aplikasi ini membantu pengguna memperoleh informasi lokasi.

Aplikasi LBS dengan metode *Cell Identification* (Cell ID) telah dibangun untuk layanan umum kota Yogyakarta[2]. Aplikasi ini dibagi menjadi *client* yang diakses melalui telepon seluler dan server yang diakses melalui *web*. Telepon seluler (ponsel) memperoleh lokasi pengguna melalui Cell ID dari sinyal ponsel. Cell ID adalah nomor unik yang umum digunakan untuk mengidentifikasi *Base Transceiver Station* (BTS) dalam jaringan GSM atau dalam *Location Area Code* (LAC). Data yang digunakan adalah data layanan dan BTS yang ditambahkan melalui *web*. Data layanan meliputi nama dan alamat. Aplikasi ini dibangun pada *Symbian S60 3<sup>rd</sup> edition feature pack 1* yang terdapat pada *mobile phone* Nokia.

Pencarian lokasi fasilitas umum terdekat pada perangkat mobile dengan penentuan posisi user menggunakan Cell ID telah dibangun untuk ponsel Sony Ericsson[3]. Sistem mengambil data Cell ID berformat XML dari basis data laboratorium Ericsson untuk mendapatkan informasi lokasi. Selanjutnya sistem meminta data pada *Google Maps* untuk menampilkan lokasi pengguna pada peta. Setelah itu, sistem mengambil lokasi fasilitas umum dari basis data sistem.

Pada penelitian ini digunakan *smartphone* dengan sistem operasi Android yang ber-*platform open source*. Beberapa penelitian tentang Android telah dilaksanakan, antara lain penggunaan teknologi *OpenStreetMap* (OSM) untuk Sistem Informasi Geografis (SIG) fasilitas pelayanan umum berbasis Android di kota Palu[4]. Aplikasi ini merupakan aplikasi pencarian rute terdekat dengan *plug-in* hasil pengembangan OSM. Namun demikian, aplikasi yang dikembangkan memiliki masalah dengan kompatibilitas pada beberapa sistem operasi.

Aplikasi pencarian fasilitas umum berbasis lokasi pada platform android dibuat untuk Bandung[5]. Aplikasi dibagi menjadi 2 yaitu *client* pada *smartphone* dan sebuah server *web*. Penentuan lokasi menggunakan dengan GPS atau Cell ID dan koneksi GPRS/EDGE/UMTS untuk melakukan pertukaran data pada *web* server. Aplikasi dibangun pada *platform* Android versi 2.1 dan dapat dioperasikan pada *smartphone* dengan Android versi 2.1 dan 2.2.

Foursquare merupakan salah satu jejaring sosial yang menggunakan teknologi LBS yang dapat diakses melalui *smartphone*. Foursquare memberi kebebasan kepada pengembang aplikasi melalui *Application Programming Interface* (API). Pengguna yang tidak memiliki akun Foursquare dapat menggunakan aplikasi dengan memanfaatkan *end-point* pada Foursquare API untuk pencarian lokasi.

Penelitian tentang Foursquare menghasilkan aplikasi “Auto Check-in lite” dengan menggunakan Cell ID untuk penentuan lokasi[6]. *SQLite* digunakan untuk penyimpanan data dan pengaturan *proxy* untuk *check-in* pada tempat-tempat yang bersifat pribadi, serta dapat dilakukan *back-up* pada pengaturannya. Aplikasi bernama “Checked” dengan Cell ID untuk penentuan posisi *WiFi* atau perangkat GPS yang ada dibuat dengan fitur tambahan *check-in* secara otomatis pada Foursquare tanpa harus melakukan pencarian lokasi terlebih dahulu[7].

*Location Based Service* (LBS) merupakan layanan informasi yang memanfaatkan kemampuan penunjuk lokasi pada piranti bergerak dan dapat diakses melalui jaringan *mobile*. International Open Geospatial Consortium (OGC) mendefinisikan LBS sebagai layanan IP nirkabel yang menggunakan informasi geografis untuk melayani pengguna ponsel. LBS merupakan kombinasi dari 3 teknologi (1) *mobile communication system*, (2) piranti telepon genggam, dan (3) SIG dengan basis data spasial.

Penentuan lokasi pengguna ponsel pada LBS dapat dilakukan dengan 3 macam cara [8], yaitu :

#### 1. Metode *Basic Positioning*

Pada metode ini, penentuan lokasi didasarkan pada daerah geografis yang tercakup oleh sebuah sel . Ketika ponsel terhubung ke BTS, ponsel diasumsikan berada dalam sel BTS.

#### 2. Metode *Enhanced Positioning*

Pada metode ini, sebuah ponsel harus aktif terhadap 3 BTS. Jarak ponsel terhadap masing-masing BTS ditentukan berdasar waktu yang ditempuh oleh sebuah sinyal dari ponsel ke masing-masing BTS. Selanjutnya dengan triangulasi, posisi ponsel dapat ditentukan.

### 3. Metode *Advanced Positioning*

Pada metode ini dilakukan pengukuran waktu tiba dari sebuah sinyal yang dikirim dari tiga buah satelit GPS. Hal ini berarti handset harus memiliki fasilitas untuk mengakses GPS.

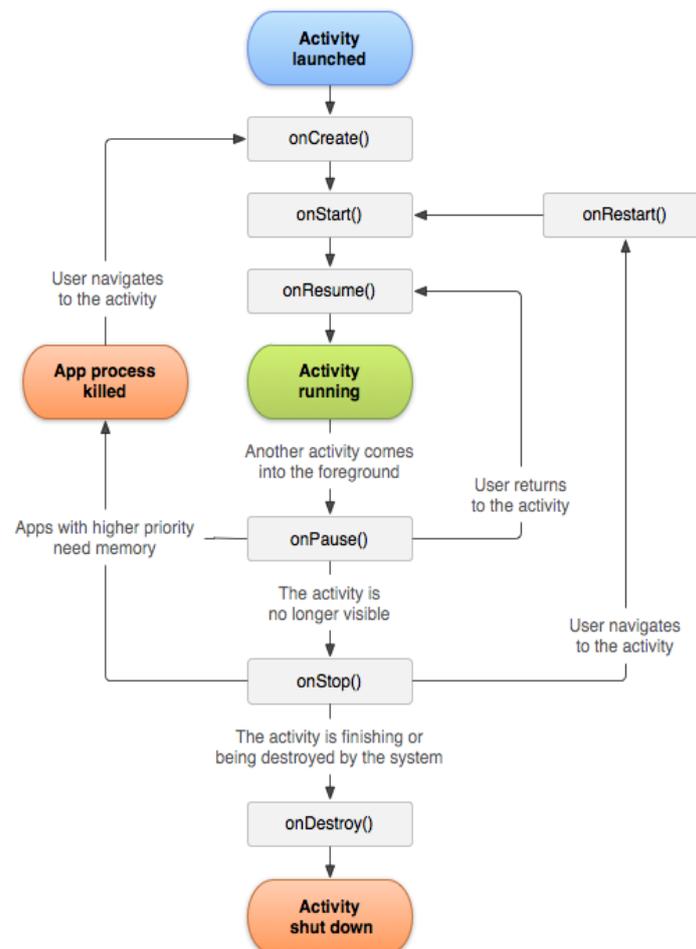
Android adalah sebuah sistem operasi untuk *smartphone*. Sistem ini dibangun dari 5 komponen utama [9] yaitu *Applications*, *Application Framework*, *Libraries*, *Android Runtime* dan *Linux Kernel*. Sistem operasi Android dikemas dengan beberapa aplikasi inti yaitu *email client*, program SMS, kalender, peta, browser, kontak dan lainnya. Semua aplikasi tersebut dibangun dengan bahasa Java. Dengan menyediakan pengembangan platform terbuka, Android memfasilitasi pengembang untuk membangun aplikasi. Pengembang diberi kebebasan untuk memanfaatkan perangkat keras, mengakses informasi lokasi, menjalankan *background services*, mengatur alarm, menambahkan notifikasi pada *status bar*, dan masih banyak lagi.

Para pengembang Android memiliki sepenuhnya terhadap *framework API* seperti yang digunakan aplikasi inti. Arsitektur aplikasi didesain untuk mempermudah penggunaan kembali komponen-komponen tersebut. Setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lainnya dapat menggunakan kemampuan tersebut. Android menyediakan paket library C/C++ yang digunakan oleh bermacam-macam komponen pada sistem Android. Beberapa library inti yang dapat digunakan antara lain *System C library*, *media libraries*, *surface manager*, *LibWebCore*, *SGL*, *3D libraries*, *FreeType* dan *SQLite*.

Setiap proses yang dilayani Android disajikan dalam *activity*. *Activity* adalah komponen aplikasi Android yang memberikan sebuah layar kepada pengguna untuk berinteraksi seperti menelpon, mengambil gambar, mengirim email atau melihat peta [9]. Pada setiap *activity* diberi sebuah frame untuk menggambarkan antarmuka *activity*. Siklus hidup *activity* dapat dilihat pada Gambar 1.

*Google Maps* adalah layanan dari Google untuk menampilkan citra digital dari dunia dengan menggunakan API *Google Maps*. *Google Maps* menyediakan gambar resolusi tinggi dari satelit. API *Google Maps* merupakan aplikasi antar muka dari *Google Maps* yang dapat diakses lewat *javascript* sehingga *Google Maps* dapat ditampilkan pada halaman web. Dalam pengolahan data, Google menyediakan API lain untuk mendukung penggunaan API *Google Maps* yaitu API *Direction Service* dan API *Google GeoCode*. Google menyediakan fitur *Google Maps* pada *Google APIs Add-On* untuk Android. *Google APIs Add-On* merupakan ekstensi dari *Software Development Kit (SDK) Android Development Environment* untuk membantu pengembangan aplikasi

Layanan *Google direction* menggunakan format *Keyhole Markup Language (KML)* yang merupakan format data untuk menampilkan data geografis. Untuk mendapatkan data tersebut pengguna cukup mengirim lokasi awal dan lokasi akhir. Server akan mengirim format data KML yang memuat arah yang diperlukan dalam bentuk koordinat-koordinat titik lokasi pada jalur di peta. Data ini dapat diolah sehingga akan muncul tampilan arah pada peta.



Gambar 1. Siklus-hidup dari *Activity* Android

Foursquare adalah aplikasi LBS pada *platform* piranti bergerak yang membuat kota-kota mudah untuk diakses dan menarik untuk dijelajah. Foursquare merupakan pelopor jejaring sosial yang menggunakan teknologi LBS.

Foursquare sangat populer sebagai media publikasi suatu lokasi agar diketahui oleh masyarakat. Data – data yang digunakan bersumber dari pengguna. Desain Foursquare sangat membantu para pengembang aplikasi untuk berpartisipasi dalam pengembangan aplikasi Foursquare itu sendiri.

Foursquare menggunakan *library s2* dari Google untuk melakukan pencarian *geo-indexing*. *PostGIS* dan himpunan data *geonames.org* digunakan untuk mengubah alamat *geocode* menjadi koordinat sehingga lokasi pada peta dapat ditampilkan dan dapat dilakukan pencarian berbasis lokasi.

Sampai Februari 2012, Foursquare menggunakan Google Maps sebagai media pemetaan, baik dalam aplikasi *smartphone* maupun web. Sejak 29 Februari 2012, Foursquare memindah peta yang digunakan dari *Google Maps* ke *MapBox* yang didukung oleh *OpenStreetMap*. Untuk aplikasi *smartphone*, Google masih tetap menggunakan *Google Maps*. Perubahan hanya dilakukan pada media pemetaan pada web dari Foursquare.

Foursquare menyediakan beberapa fitur dan komponen utama sehingga foursquare dikategorikan sebagai jejaring sosial yang unik. Foursquare dapat dihubungkan dengan jejaring sosial lain seperti Facebook dan Twitter. Disamping memiliki beberapa komponen utama pada jejaring sosial, Foursquare juga memiliki fitur dan komponen lain seperti :

### 1. Venue

*Venue* berisi informasi tentang nama tempat, alamat lokasi, nomor telepon, kategori lokasi, berapa kali tempat ini dikunjungi dan komentar pengunjung. Pengguna dapat membuat *venue* dan menghapus *venue*. Pengguna tidak dapat menghapus *venue* yang tidak dibuatnya. Pengguna dapat mencari *venue*. Pada pencarian biasa, Foursquare akan memberikan lokasi yang diinginkan sesuai dengan permintaan. Jika pengguna melakukan pencarian sesuai dengan kategori tertentu, maka *venue* yang sesuai dengan kategori tersebut akan ditampilkan. Foursquare juga merekomendasikan *venue* yang sedang ramai dikunjungi (*trending*), sedang ada acara tertentu (*event*) atau yang memiliki tawaran khusus (*spesial*). Pengguna dapat melakukan klaim ke Foursquare untuk suatu *venue* yang dibuatnya sesuai dengan ketentuan dan peraturan yang berlaku dalam Foursquare.

### 2. Check-in

Setelah mendapatkan *venue* yang sesuai, pengguna dapat melakukan *check-in* untuk menunjukkan lokasi pengguna berada. Pengguna lain yang telah berbagi koneksi, dapat memberi komentar terhadap *check-in*. Setiap pengguna melakukan *check-in*, Foursquare menyimpan data *check-in* ke *history* pengguna. Disamping itu, pengguna akan mendapatkan poin. Poin digunakan sebagai dasar peringkat dalam *leaderboard* antar koneksi.

### 3. Shout

*Shout* merupakan bentuk ekspresi pengguna. *Shout* dapat dilakukan tanpa melakukan *check-in*. *Shout* yang dilakukan biasanya mengenai suatu tempat, makanan apa yang disajikan, bagaimana rasanya, dengan siapa mereka di sana dan lain sebagainya. Apabila pengguna menghubungkan Foursquare dengan jejaring sosial lain, maka setiap *shout* dilakukan, juga akan dilakukan pembaruan status pada jejaring sosial lainnya.

### 4. Mayorship

Foursquare menghitung setiap *check-in* yang dilakukan oleh pengguna. Apabila pengguna sering melakukan *check-in* pada suatu *venue*, maka pengguna akan menjadi *mayor* pada *venue* tersebut.

### 5. Badges

*Badges* merupakan penghargaan bagi pengguna yang berhasil melakukan misi. *Badges* mempunyai ikatan dengan suatu lokasi. Foursquare membagi *badge* dalam 3 golongan yaitu *Foursquare badges*, *Expertise badges*, *partner badges*. *Foursquare badges* merupakan *badges* asli yang dikeluarkan Foursquare. *Expertise badges* diberikan kepada pengguna yang telah beberapa kali *check-in* pada *venue* dengan kategori tertentu. *Partner badges* ditawarkan oleh perusahaan tertentu yang bekerja sama dengan Foursquare.

### 6. Super User (SU)

Status SU diberikan kepada pengguna yang sering *check-in* atau memasukkan informasi tempat baru ke dalam Foursquare. Foursquare menyediakan tiga tingkat SU, masing-masing dengan fasilitas berbeda.

Foursquare menyediakan API untuk mengakses semua data yang digunakan pada aplikasi Foursquare. Dengan API, pengembang dapat menambah data tertentu pada basis data foursquare, mengolah, dan mengembangkan suatu aplikasi unik dari data-data tersebut. Saat ini Foursquare menyediakan API versi 2 untuk mengakses data dari Foursquare. Method ini berupa *url*. Hasil dari server berupa *JavaScript Object Notation* (JSON). JSON merupakan format pertukaran data yang ringan dan cepat. Tidak hanya mudah dibaca oleh mesin, format data ini mudah dibaca manusia. JSON selanjutnya diolah menjadi data yang dibutuhkan. Dengan data ini, pengembang dapat membuat aplikasi yang diinginkan.

API Foursquare dapat diakses dengan *OAuth 2.0*. *OAuth 2.0* merupakan pengembangan protokol *OAuth* untuk menjaga data dari pengguna agar tetap aman pada saat menggunakan API. Penggunaan Foursquare APIv2 dimulai dengan pendaftaran aplikasi. Setelah pendaftaran, pengembang mendapat kunci untuk melakukan otentikasi terhadap API Foursquare. Kunci

terdiri dari 2 komponen yaitu *client id* dan *client secret*. Beberapa *Endpoints* yang tidak membutuhkan informasi spesifik pengguna seperti pencarian *venue* diperbolehkan menggunakan akses tanpa identifikasi pengguna. Akses ini dilakukan tanpa menggunakan *OAuth*.

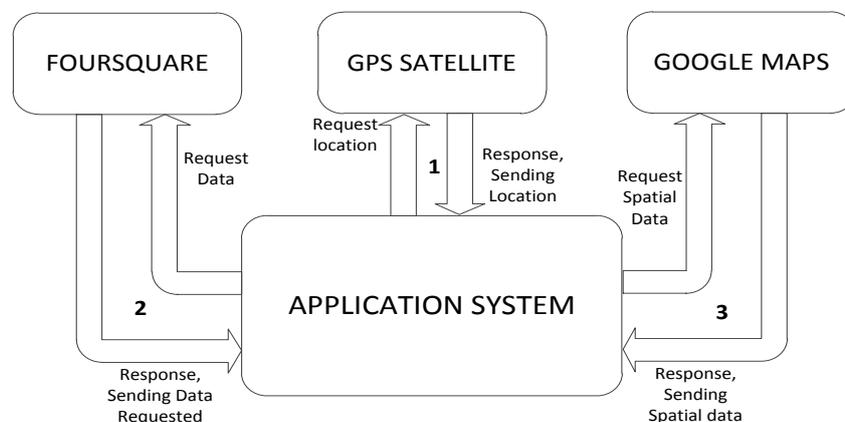
## 2. PERANCANGAN SISTEM

Sistem yang dirancang merupakan sebuah aplikasi yang memanfaatkan GPS, Foursquare dan *Google Maps*. Sistem akan mendeteksi lokasi dari satelit GPS. Selanjutnya sistem mencari data di sekitar pengguna dengan menggunakan data dari Foursquare. Setelah kategori ditentukan pengguna, sistem melakukan permintaan data ke server Foursquare. Data yang diminta akan direspon oleh server Foursquare. Untuk menyajikan data pada peta, sistem meminta data lokasi pengguna dari server *Google Maps*. Setelah data yang diminta diterima, peta dan informasi data yang diminta siap untuk ditampilkan. Gambar 2 dan 3 menunjukkan desain arsitektur dan diagram *use case* sistem ini.

Pada saat mulai, sistem menampilkan *splash screen* untuk melakukan pencarian lokasi pengguna. Pada awal pencarian, sistem memeriksa keberadaan GPS dan jaringan yang dapat digunakan untuk mencari lokasi. Jika kedua perangkat tidak memungkinkan maka akan digunakan lokasi terdahulu sebagai lokasi pengguna. Bila salah satu atau kedua perangkat dapat digunakan, sistem akan meminta koordinat lokasi dari pengguna. Setelah berhasil didapatkan, lokasi akan disimpan untuk digunakan dalam proses selanjutnya. Jika dalam waktu 20 detik lokasi pengguna tidak didapatkan, sistem akan mengambil koordinat lokasi sebelumnya. Pembatasan waktu pencarian lokasi dilakukan agar pengguna tidak menunggu terlalu lama untuk dapat melakukan pencarian lokasi. Sistem akan melakukan perbandingan waktu terhadap kedua lokasi tersebut. Lokasi yang memiliki update terbaru akan digunakan sebagai lokasi pengguna. Setelah proses selesai, *splash screen* akan berhenti dan menu utama akan ditampilkan. Detail proses ini disajikan pada Gambar 4.

Setelah pencarian selesai, sistem menampilkan menu utama. Proses pencarian dilakukan pada menu *Public Facilities* dan *Free Search* serta pada *service* dari *auto check-in*. Sistem membaca lokasi pengguna dan mengatur sistem. Setelah itu sistem akan memeriksa kategori atau masukan yang telah dipilih pengguna. Selanjutnya sistem mengirim permintaan data-data yang sesuai dengan kategori atau masukan tersebut. Jika koneksi terpenuhi, sistem melakukan permintaan data. Apabila koneksi tidak terpenuhi, tampilan pesan kesalahan.

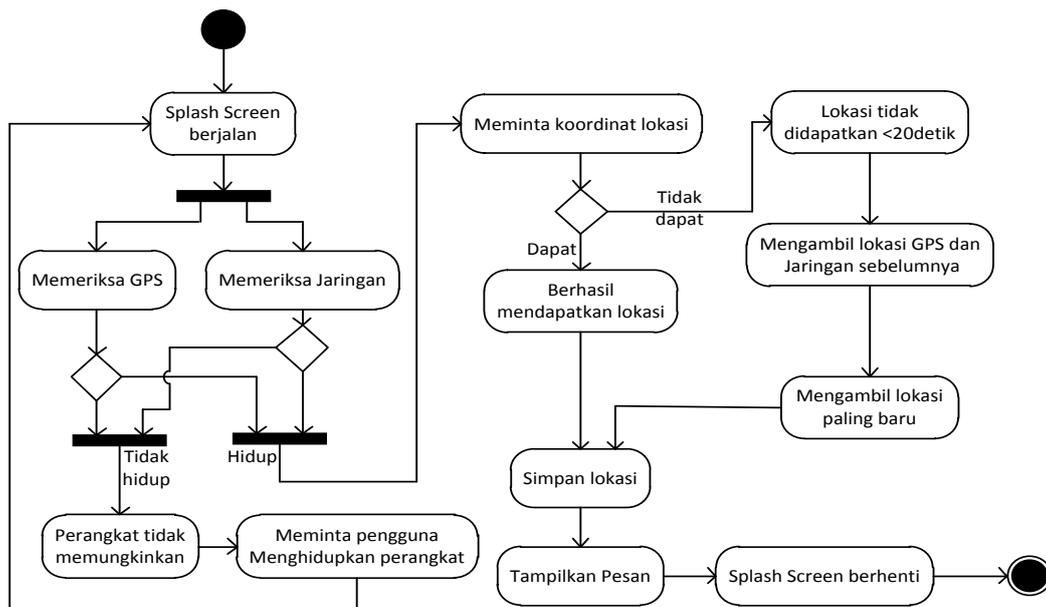
Selanjutnya server akan memberikan informasi apakah data tersebut ada atau tidak. Jika data-data yang diminta ada, data tersebut diolah dan ditampilkan sesuai dengan pengaturan sistem. Apabila data tidak ada, sistem akan menampilkan pesan kesalahan. Gambar 5 menunjukkan proses pencarian data.



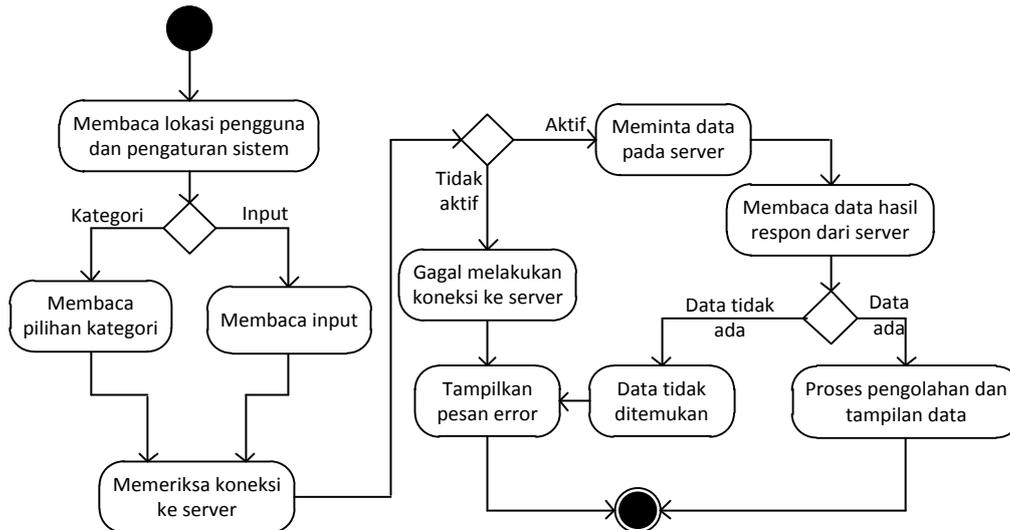
Gambar 2. Arsitektur sistem



Gambar 3. Diagram Use Case

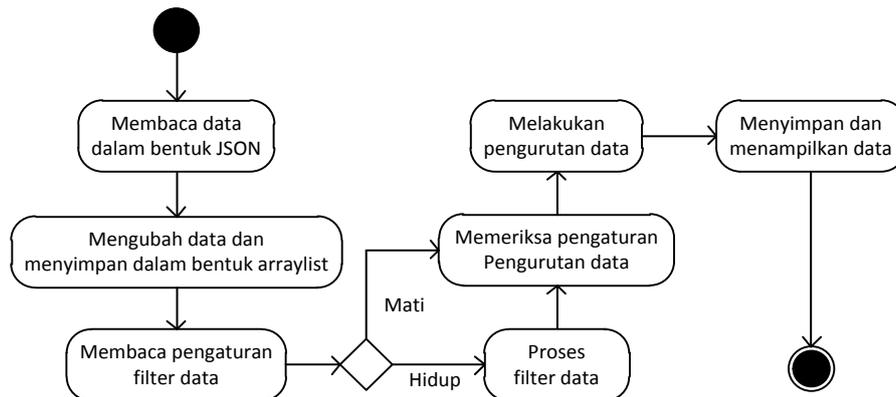


Gambar 4. Diagram aktivitas proses pencarian lokasi



Gambar 5. Diagram aktivitas proses pencarian data

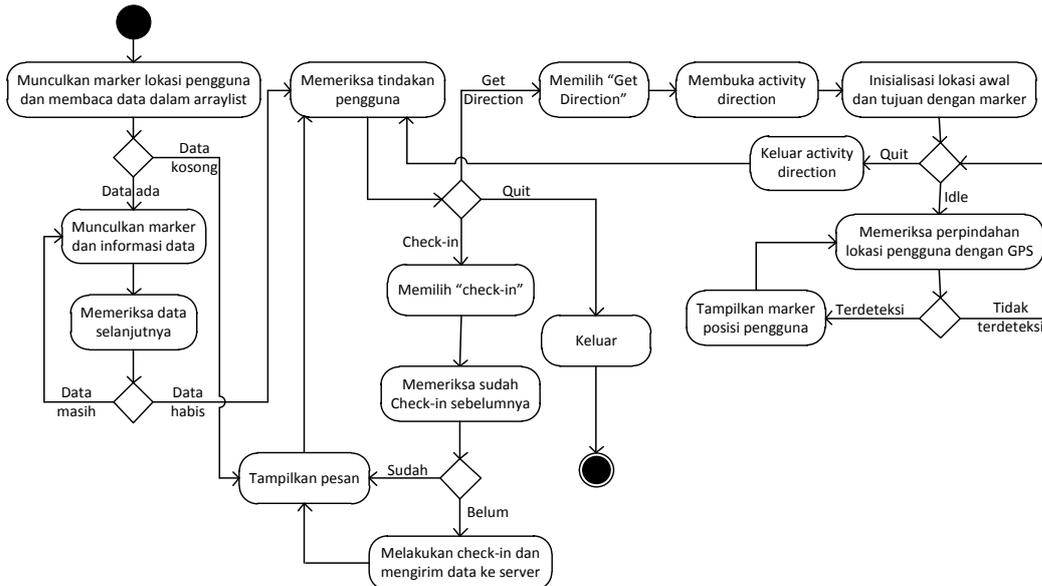
Proses pengolahan dan pencarian data dimulai apabila permintaan data dipenuhi server. Data dalam bentuk JSON diolah dengan menggunakan *JSONparser*. Kemudian data disimpan dalam bentuk *arraylist*. Sistem akan membaca pengaturan yang dilakukan pengguna yaitu pengurutan dan penyaringan data. Setelah data diolah sesuai dengan pengaturan sistem, data ditampilkan dalam bentuk *list*. Gambar 6 menunjukkan proses pengolahan dan tampilan data.



Gambar 6. Diagram aktivitas proses pengolahan dan tampilan data

Data hasil pencarian ditampilkan dalam bentuk *list*. *List* dapat dipilih untuk ditampilkan ke dalam peta. Apabila pengguna melakukan *login* terlebih dahulu pada menu *auto check-in*, pengguna dapat melakukan *check-in* dengan cara menekan selama 2 detik *item* pada *list* yang diinginkan. Pengguna akan diminta untuk melakukan *shout* terlebih dahulu sebelum data untuk melakukan *check-in* dikirimkan ke server. Jika berhasil melakukan *check-in*, sistem akan menampilkan pesan informasi dari server mengenai *check-in* yang dilakukan. Jika tidak, pesan error akan ditampilkan. Gambar 7 menunjukkan proses tampilan data pada peta.

Ketika pengguna memilih salah satu pilihan pada *list* atau melihat data-data pada peta, sistem akan membungkus *arraylist* dan mengirim ke *activity* untuk menampilkan peta. Dalam awal proses tampilan data pada peta, sistem akan menampilkan lokasi pengguna dan membaca data yang dikirimkan. Jika terdapat data, sistem akan memunculkan *marker* dan informasi mengenai lokasi tersebut ke dalam bentuk *overlay* pada peta. Apabila data kosong, pesan error akan ditampilkan. Setelah data berhasil dipopulasikan, sistem akan menunggu tindakan selanjutnya dari pengguna. Ketika *marker* dipilih, informasi mengenai lokasi tersebut akan muncul sama seperti informasi pada *list* sebelumnya.



Gambar 7. Diagram aktivitas proses tampilan data pada peta

Terdapat 3 menu pilihan pada *marker* yaitu (1) *close*, (2) *get direction* dan (3) *check-in*. *Close* digunakan untuk menutup jendela informasi yang ditampilkan. Pada *get direction*, sistem akan membuka *activity* baru untuk menampilkan rute dari lokasi pengguna menuju lokasi yang diminta. Sistem meminta data *direction* dalam format *kml* pada server google maps. Setelah server mengirim data yang diminta sistem akan membaca dan mengolah data tersebut. Data ini akan ditampilkan pada peta dalam bentuk garis sesuai dengan rute yang dituju. Jika perangkat GPS memungkinkan, sistem akan menampilkan *update* lokasi pengguna setiap detiknya untuk mengetahui lokasi pengguna apakah sudah sesuai dengan rute yang diberikan. Pada pilihan *check-in*, sistem akan memeriksa apakah pengguna telah melakukan *check-in* pada tempat ini. Jika belum, maka pengguna diminta untuk melakukan *shout* dan sistem akan meminta data yang diperlukan untuk melakukan *check-in*. Data akan dikirim untuk selanjutnya ditampilkan pesan informasi mengenai *check-in* yang dilakukan.

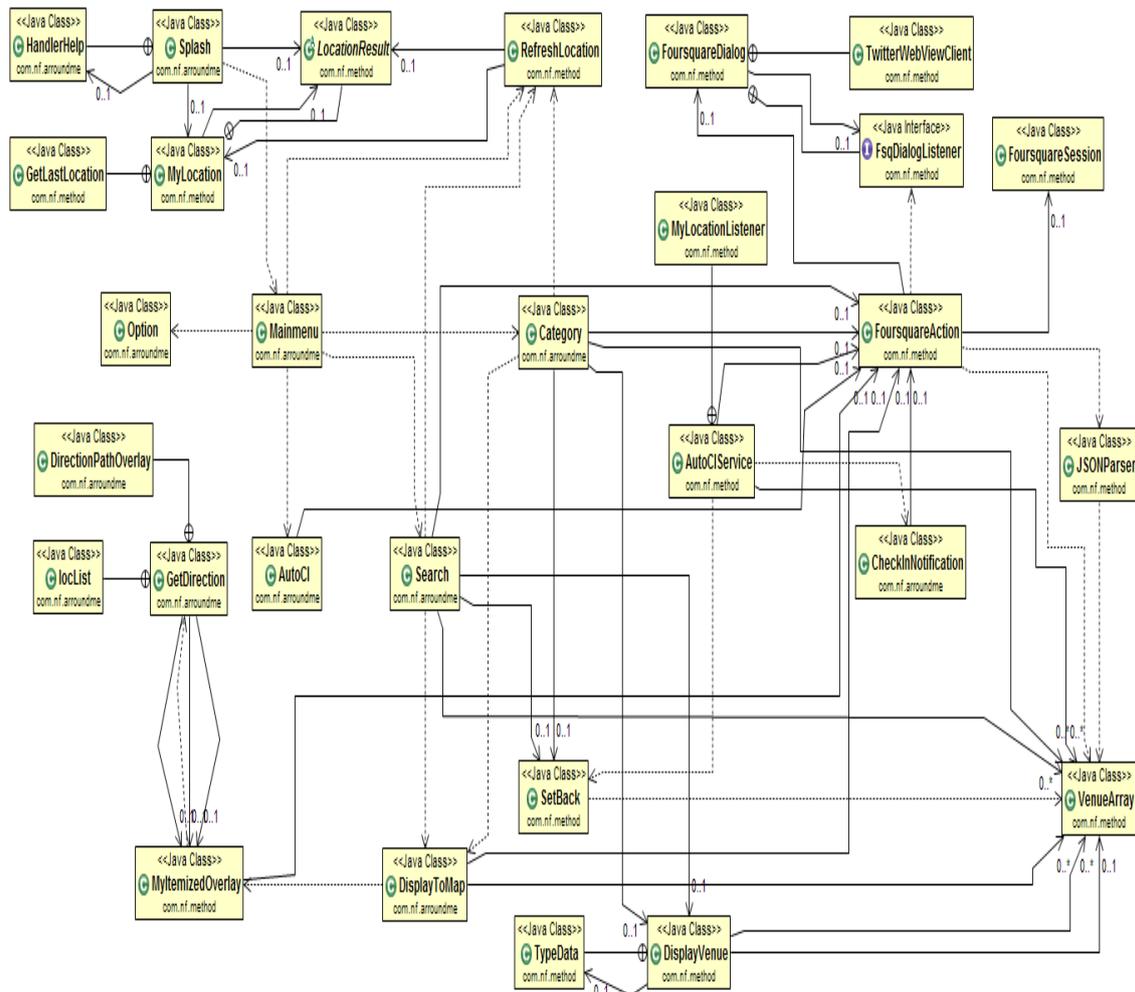
Sistem yang dibangun terbagi atas 2 *package* yaitu *package com.nf.arroundme* yang merupakan kumpulan *activity* untuk tampilan antar muka dan *package com.nf.method* yang merupakan kumpulan fungsi-fungsi kebutuhan sistem.

Antarmuka sistem ditangani oleh 9 kelas *activity* yang saling berhubungan. Setiap *activity* dibantu oleh beberapa kelas fungsi (lihat Gambar 8). Pada saat start, sistem memanggil kelas *Splash* yang dibantu *handler* untuk mendapatkan lokasi pengguna yang diproses oleh *MyLocation* dengan bantuan *LocationResult*. Setelah berhasil mendapatkan lokasi, sistem akan menuju ke kelas *MainMenu*. *MainMenu* terhubung dengan kelas *Search*, *Category*, *AutoCI*, *Option* dan sebuah menu *item RefreshLocation* untuk memperbarui lokasi pengguna. Kelas *Search* menangani proses pencarian tempat dengan menggunakan input pengguna. Kelas *Category* menangani proses pencatatan tempat sesuai dengan tempat dan fasilitas umum yang dipilih. Kelas *AutoCI* menangani fitur *auto check-in* apabila *login* dilakukan dan *AutoCIService* dijalankan. Kelas *RefreshLocation* yang dibantu *MyLocation* merupakan menu *item* untuk memperbarui lokasi pengguna.

Semua fungsi untuk melakukan koneksi ke server, pengambilan data dari server, login, dan pengolahan data dipusatkan pada kelas *FoursquareAction*. Pada proses login *FoursquareAction* membutuhkan kelas *Foursquare Dialog* untuk menangani halaman login pengguna. Kelas *FoursquareSession* menyimpan sesi pengguna. Kelas *JSONParser* mengubah format data JSON ke bentuk *arraylist* dengan nama *VenueArray*.

Setelah data diubah, sistem menampilkan data dengan bantuan kelas *DisplayVenue* dan *Setback* yang ditampilkan pada kelas *Search* dan *Category*. Kelas *DisplayVenue* bertujuan untuk

menampilkan informasi data dengan format yang telah ditentukan dengan bantuan kelas *TypeData*. Kelas *Setback* bertujuan untuk mengolah data apakah *filter* data akan dilakukan dan tampilan data akan diurutkan. Kelas *DisplayToMap* menangani populasi data pada peta yang dibantu oleh kelas *MyItemizedOverlay*. Kelas *GetDirection* menangani tampilan rute dari lokasi pengguna ke lokasi tujuan yang ditentukan pengguna. Kelas *DirectionPathOverlay* digunakan untuk membantu menampilkan jalur rute pada peta dalam bentuk garis dan kelas *loclist* digunakan untuk menampilkan marker lokasi pengguna secara *update* sesuai lokasi yang diterima GPS.

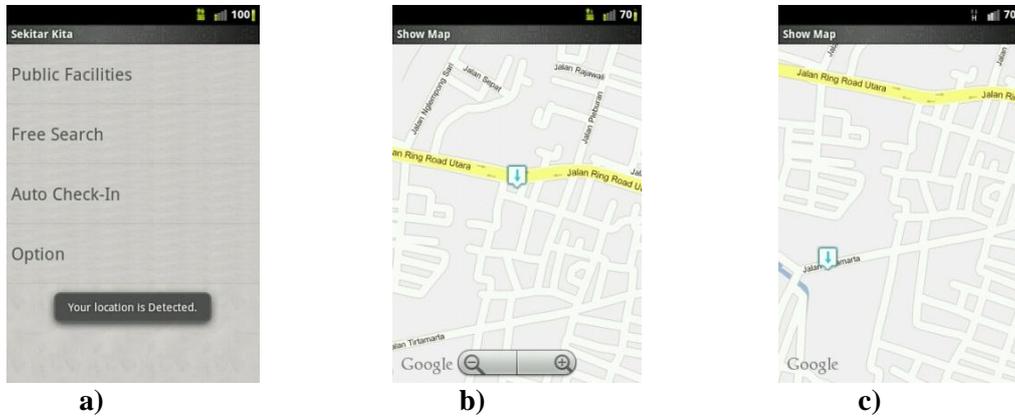


Gambar 8. Class diagram dari Sistem

### 3. HASIL DAN PEMBAHASAN

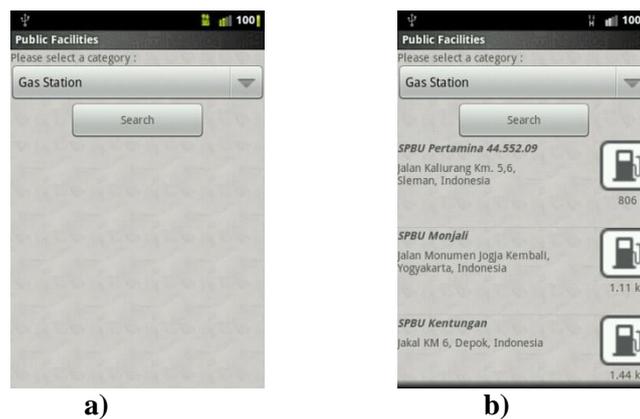
Pada awal aplikasi dijalankan, sistem akan meminta lokasi pengguna dengan menggunakan perangkat GPS atau jaringan yang ada. Lokasi pengguna lebih akurat didapatkan oleh GPS. Gambar 9 menunjukkan permintaan lokasi pengguna. Pengguna dapat melakukan pencarian fasilitas umum dengan memilih kategori yang telah disajikan. Terdapat 16 pilihan kategori yaitu "Gas Station", "Bus Station", "Train Station / Reservation", "Airports / Ticket Agent", "Religious Center", "Police Station", "Post Office", "Hospital", "Bank", "Food", "Cafe", "School", "Great Outdoor", "Hotel", "Government Building" dan "Shop / Market". Gambar 10 menunjukkan tampilan pencarian fasilitas umum.

Pengguna dapat melakukan pencarian dengan *input* sesuai keinginan pengguna pada menu *Free Search*. Pada menu ini terdapat sebuah kolom untuk mengisikan tempat yang dicari. Pada Gambar 11, pengguna ingin mencari lokasi “mie ayam” di sekitar pengguna.

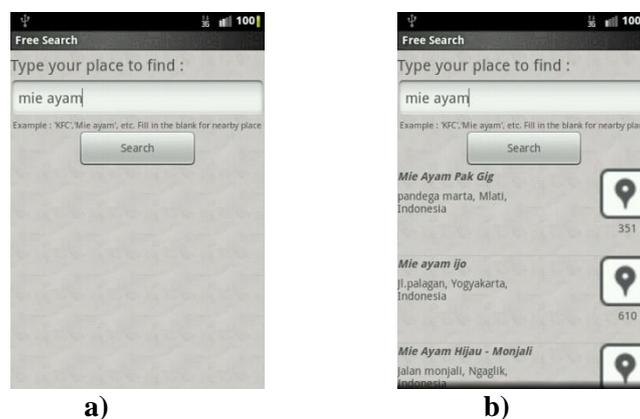


Gambar 9.

(a) Pencarian lokasi berhasil, (b) Lokasi dengan GPS, (c) Lokasi dengan jaringan GSM



Gambar 10. (a) Tampilan pilihan pencarian. (b) Tampilan hasil pencarian



Gambar 11. (a) Tampilan pemasukan data pencarian, (b) Tampilan hasil pencarian

#### 4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi dan pengujian terhadap sistem yang dibangun dapat disimpulkan bahwa aplikasi pencarian lokasi fasilitas umum dengan layanan teknologi LBS pada sistem operasi Android dapat dibangun dengan menggunakan Foursquare APIv2.

#### DAFTAR PUSTAKA

- [1] Prahasta, E., 2009, *Sistem Informasi Geografis Konsep-Konsep Dasar (Perspektif Geodesi dan Geomatika)*, Informatika, Bandung.
- [2] Zaky, R.A., 2010, *Aplikasi Location Based Service Menggunakan Metode Cell Identification (Studi kasus layanan umum kota Yogyakarta)*, *Skripsi*, Jurusan Matematika, Universitas Gadjah Mada, Yogyakarta.
- [3] Rachmawati, A., 2011, *Pencarian Lokasi Fasilitas Umum Terdekat Pada Perangkat Mobile Dengan Penentuan Posisi User Menggunakan Cell ID*, *Tugas Akhir*, Institut Teknologi Sepuluh November, Surabaya.
- [4] Mantjoro, Y., 2012, *Penggunaan Teknologi OpenStreetMap Untuk GIS Fasilitas Pelayanan Umum Berbasis Android (Studi Kasus Kota Palu)*, *Tugas Akhir*, Institut Teknologi Sepuluh November, Surabaya.
- [5] Nugraha, A., 2011, *Rancang Bangun Aplikasi Fasilitas Umum Berbasis Lokasi Pada Platform Android Studi Kasus Kota Bandung*, *Skripsi*, Universitas Komputer Indonesia, Bandung.
- [6] RMWthings, 2011, *Android things*, <http://android.rwmthings.com/latest/>, diakses 9 April 2012.
- [7] Walkbase, 2012, *Checked F.A.Q.*, <http://checkedapp.com/faq/>, diakses 9 April 2012.
- [8] Adam Cp, 2008, *Aplikasi Layanan Berbasis Lokasi untuk Memberikan Informasi Sarana Umum*, [http://digilib.ittelkom.ac.id/index.php?option=com\\_content&view=article&id=44:lbs-location-based-service&catid=17:sistem-komunikasi-bergerak&Itemid=14](http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=44:lbs-location-based-service&catid=17:sistem-komunikasi-bergerak&Itemid=14), diakses 8 Maret 2012.
- [9] Felker, D., dan Dobbs, J., 2011, *Android : Application Development for Dummies*, Wiley Publishing, Inc., Indiana.