

# Implementasi Push Notification dan Location Based Service Pada Aplikasi Smart Rekomendasi Wirausaha Untuk Pedagang Makanan Keliling

Eko Budi Setiawan<sup>1</sup>, Wahyu Saputra<sup>2</sup>, Angga Setiyadi<sup>3</sup>

Program Studi Teknik Informatika, Universitas Komputer Indonesia, Bandung, Indonesia

eko@email.unikom.ac.id

way.saputra@gmail.com

angga.setiyadi@email.unikom.ac.id

Diterima 9 Maret 2019

Disetujui 24 Juni 2019

**Abstract**— Food mobile traders, especially in the city of Bandung, are currently experiencing difficulties in obtaining permanent buyers and customers. 86 % traders interviewed felt that their income was increasingly decreasing due to the many buyers using online applications. The traders difficult to get buyers and it was difficult to determine the direction of his destination when trading around. Likewise, buyers who want to buy free merchandise find it difficult because they do not know the schedule of the traveling merchants and are not aware of the traders themselves. In this case, the traveling food traders and buyers need an application container that can facilitate the transaction process. Based on the results of alpha and beta tests that have been carried out at the implementation and testing stages, it can be concluded that the development of smart applications recommended by mobile food traders using push notification and location-based service based on android can make it easier for mobile traders to get buyers, and buyers are facilitated in knowing the position of the nearest traveling merchant.

**Index Terms**— Food mobile traders, Recommendation, Android, LBS, Google Maps API

## I. PENDAHULUAN

Perkembangan teknologi saat ini semakin pesat, khususnya kemajuan dibidang teknologi informasi terutama pada teknologi *mobile* dan internet. Penggunaan aplikasi *mobile* dan web dikatakan lebih efektif dan efisien karena adanya kemudahan dalam pengaksesan dan penyampaian informasi. Saat ini masih banyak para pedagang yang sering berdagang keliling. Pedagang keliling merupakan seorang pedagang yang menjajakan dagangan dengan cara menawarkan dagangannya dari satu tempat ke tempat lain.

Pedagang keliling khususnya di Kota Bandung sedang mengalami kesulitan dalam mendapatkan pembeli dan pelanggan tetap. Terbukti 26 dari 30 pedagang keliling yang diwawancarai merasakan bahwa pendapatannya semakin hari semakin

berkurang dikarenakan pembeli sudah banyak menggunakan aplikasi *online*. Pak Dedi berusia 39 tahun salah satu seorang pedagang lumpia basah mengatakan bahwa merasa kesulitan dalam mendapatkan pembeli serta sulit menentukan arah tujuan saat berdagang keliling. Begitupun para pembeli yang ingin membeli dagangan keliling merasa kesulitan karena tidak mengetahui jadwal pedagang makanan keliling dan tidak mengetahui adanya pedagang itu sendiri.

Dari penelitian Sihabudin [1], untuk mendapatkan informasi mengenai promosi produk baru, diskon, dan *event-event* yang diadakan, dan pencarian lokasi toko-toko yang terutama bagi pelanggan yang luar kota yang sedang berkunjung ke kota Bandung dapat menggunakan *google cloud messaging* untuk menjalankan sebuah *push notification, location based service* untuk menentukan lokasi yaitu *API maps* dan *API location, google direction API* untuk menghitung arah antar lokasi, *web service* sebagai *selfcontaining* dan *JSON* sebagai format pertukaran data ringan.

Penelitian lainnya dari Mohd. Siddik dan Akmal Nasution [2] disimpulkan bahwa aplikasi android dapat memberikan gambaran secara individu terhadap penggunaan teknologi dan teknologi *push notification* ini dapat memberikan kontribusi dalam pelayanan pemberian informasi secara *real time* bagi pengguna.

Sering kali pedagang pada berdagang di suatu tempat mendapati cuaca yang kurang bagus atau bahkan hujan. Pola iklim dan cuaca di Indonesia yang tidak beraturan dan ekstrim akan mengganggu [3]. Tentu akan menyebabkan kerugian pada pedagang tersebut, apa lagi pedagang yang berkeliling jauh. Oleh sebab itu dengan menggunakan teknologi *OpenweatherMaps API* akan memberikan prakiraan cuaca sebelum berdagang. Data yang diperoleh dari *OpenweatherMaps* berupa informasi mengenai lokasi, cuaca, deskripsi cuaca, ikon, *humadity*, tekanan angin,

suhu udara, kecepatan angin, arah angin, dan awan yang tersimpan dalam format JSON [4].

Menurut penelitian Risa dan Eko [5] aplikasi *mobile* yang dibuat dapat menampilkan info cuaca atau suhu menggunakan *OpenweatherMap API* dimana data diproses oleh *OpenWeatherMap* dengan cara menyediakan data perkiraan cuaca dan peta cuaca online yang akurat, seperti awan atau curah hujan.

Dari penelitian lain yang ditulis oleh Jefferson Setiawan dkk [6] menyimpulkan, aplikasi yang dibangun dapat membantu pengguna untuk mendapatkan informasi objektif dan *real time*. Aplikasi juga dapat memperhitungkan umpan balik dari kegiatan-kegiatan untuk ditampilkan.

Saat ini semua orang menggunakan berbagai jenis *smartphone* khususnya Android. Hal itu seturut dengan kemampuan, motivasi, keinginan serta kebutuhan masyarakat terhadap kegunaan dari pada media tersebut [7]. Dari hasil kuesioner yang diberikan kepada 70 responden baik itu dari kalangan pedagang keliling dan pembeli, didapatkan sebanyak 100% adalah pengguna *smartphone* dan 91,4% menggunakan *smartphone* android.

Berdasarkan latar belakang yang telah diuraikan didapatkan sebuah metode *push notification* sangat membantu dalam memberikan informasi yang objektif dan *real time*. Dalam hal ini para pedagang makanan keliling maupun pembeli sangat membutuhkan sebuah wadah aplikasi yang dapat mempermudah proses transaksi. Penelitian ini dilakukan dengan harapan dapat membantu dan memudahkan para pedagang makanan keliling dan pembeli yang ada di Kota Bandung.

## II. TINJAUAN PUSTAKA

### A. Push Notification

*Push notification* merupakan suatu komunikasi jaringan, yaitu *server* akan mengirimkan pesan pemberitahuan ke *client* jika ada perubahan data, sehingga *client* tidak perlu melakukan proses *request* data tiap periode untuk mengambil data pemberitahuan. Pada sistem operasi android proses *push notification* dapat memanfaatkan layanan *Google Cloud Messaging* (GCM) untuk mengirim pesan atau notifikasi yang disediakan oleh Google [8].

### B. Location Based Service

*Location based service* merupakan layanan informasi yang dapat diakses menggunakan *mobile devices*, yang dilengkapi kemampuan untuk mengetahui keberadaan lokasi dari pengguna perangkat dan kemampuan memberikan informasi mengenai layanan yang tersedia berdasarkan lokasi

mereka pada saat itu. Menurut Schiller J, *location based service* dapat didefinisikan sebagai "layanan yang mengintegrasikan lokasi perangkat *mobile* atau posisi dengan informasi lain sehingga dapat memberikan nilai tambah bagi pengguna" [9]. Dua unsur utama dari *location based services* adalah :

1. *Location Manager (API Maps)*
2. *Location Providers (API Location):*

Lokasi, perpindahan, kedekatan dengan lokasi tertentu dapat ditentukan melalui *location manager* [10].

### C. GPS

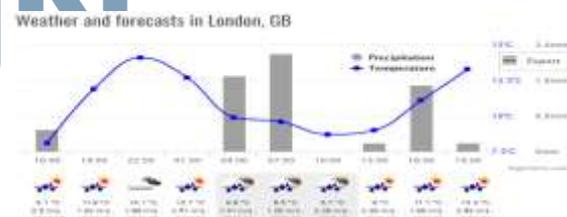
*Global positioning system* atau yang biasa disingkat dengan GPS adalah alat navigasi elektronik yang menerima informasi dari 4 - 12 satelit sehingga GPS bisa memperhitungkan posisi dimana kita berada di Bumi. Satelit GPS tidak mentransmisikan informasi posisi kita, yang ditransmisikan satelit adalah posisi satelit dan jarak penerima GPS kita dari satelit [11].

### D. Application Programming Interface (API)

API adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. Pemanggilan fungsi ke suatu situs tertentu akan menghasilkan respon yang berbeda kepada pengguna untuk membangun aplikasi *enterprise* di dalam websitenya [12].

### E. OpenWeatherMap API

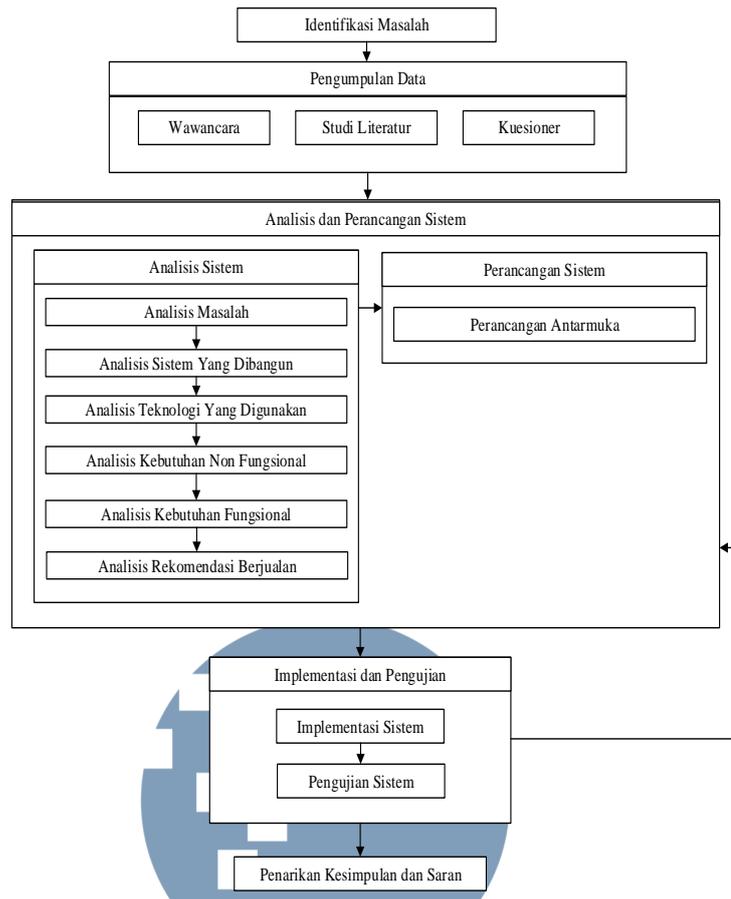
*Openweathermap* adalah layanan online yang menyediakan data cuaca terkini, termasuk data prakiraan dan data historis terkini untuk para pengembang layanan web dan aplikasi *mobile*. Contoh data cuaca pada *openweathermap* dapat dilihat pada gambar 1 berikut :



Gambar 1 Contoh Data Cuaca Pada Openweathermap  
Sumber Gambar : <https://openweathermap.org/>

## III. METODOLOGI PENELITIAN

Metode penelitian yang digunakan adalah metode deskriptif. Tujuan dari penelitian deskriptif ini adalah untuk membuat deskripsi, gambaran secara sistematis, faktual dan akurat mengenai fakta-fakta, sifat-sifat serta hubungan antar fenomena yang diselidiki [13]. Kerangka kerja pada penelitian ini dapat dilihat pada gambar 2 berikut :



Gambar 2 Kerangka kerja penelitian

#### IV. HASIL DAN PEMBAHASAN

Pada tahapan ini terdiri dari tahapan penelitian yang dilakukan. Tahapan tersebut terdiri dari aturan bisnis sistem, teknologi yang digunakan serta hasil implementasi dari aplikasi yang dibangun.

##### A. Aturan Bisnis Sistem

Berikut adalah aturan bisnis sistem :

1. Pedagang
  - a. Daftar sebagai pedagang setelah *login* sebagai pembeli
  - b. Setiap pembeli dapat mendaftar sebagai pedagang
  - c. Pedagang dapat mengelola dagangan setelah mendaftar sebagai pedagang
  - d. Pedagang dapat mendapatkan rekomendasi rute dari hasil histori dagangan selama 1 minggu yang lalu
  - e. Pedagang dapat membeli dan melihat pedagang lainnya
  - f. Pedagang hanya bisa membeli dagangan sesama pedagang ketika tidak ada orderan dari pembeli
  - g. Pedagang akan menerima pembayaran secara langsung dari pembeli setelah orderan selesai
  - h. Pedagang akan mendapatkan *review* dari pembeli setelah orderan selesai

##### 2. Pembeli

- a. Daftar *user* dapat melalui akun google, facebook dan manual
- b. Pembeli dapat melihat pedagang terdekat
- c. Pembeli dapat melihat pedagang makan dan minuman terdekat
- d. Pembeli dapat mendaftar sebagai pedagang, jika ingin membuka dagangan
- e. Pembeli tidak dapat membuat orderan ketika sudah order kepada satu pedagang hingga orderan selesai
- f. Pembeli membayar langsung kepada pedagang ketika orderan selesai dilakukan oleh pedagang
- g. Pembeli dapat memberikan *review* kepada pedagang setelah orderan selesai

##### B. Teknologi Yang Digunakan

Teknologi yang akan dipakai dalam pembangunan aplikasi ini adalah teknologi GPS yang dimanfaatkan untuk mendapatkan titik koordinat sebuah objek dan diterjemahkan ke dalam peta digital. Teknologi lain di dalam aplikasi ini adalah teknologi yang memanfaatkan public API (*Application Programming Interface*), yaitu sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh *programmer* saat membangun perangkat lunak untuk sistem operasi tertentu [14].

API yang digunakan yakni *Google Maps Android API*, *Google Maps Direction API*, *Google Places API* dan *OpenWeatherMap API* adalah *platform* untuk membangun aplikasi yang tersedia [15]. Pada *Google Maps Android API* akan memanfaatkan API dari Google tersebut untuk dipakai sebagai peta digital di dalam aplikasi yang dibangun. Analisis teknologi bertujuan untuk mengetahui teknologi apa saja yang akan dipakai di dalam sebuah sistem yang dibangun. Adapun teknologi yang digunakan pada sistem yang dibangun antara lain :

#### 1. GPS

Teknologi yang digunakan pada aplikasi yang dibangun adalah teknologi GPS dengan tipe A-GPS. Teknologi ini akan menentukan posisi smartphone dengan akurasi yang tinggi. Alasan akurasi yang tinggi tersebut menjadi dasar memilih A-GPS sebagai GPS yang digunakan pada aplikasi yang dibangun. Pemanfaatan A-GPS pada aplikasi ini berguna untuk mendapatkan lokasi smartphone lebih cepat karena didukung oleh operator telekomunikasi yang digunakan pada *smartphone*. Penggunaan A-GPS ini tentu melibatkan biaya berlebih karena menggunakan koneksi internet untuk menggunakannya.

#### 2. Google Maps API

API (*Application Programming Interface*) yang disebut oleh Google akan digunakan di dalam sistem yang dibangun. Kegunaan API ini adalah untuk mendapatkan layanan peta digital yang akan dipakai pada aplikasi. Pada API ini, layanan yang dapat digunakan selain menampilkan peta digital di dalam aplikasi adalah penggunaan marker, poligon, dan *overlay* ke peta dasar, serta mengubah tampilan area peta tertentu ke pengguna.

#### 3. OpenWeatherMap API

*OpenweatherMap* digunakan untuk mendapatkan data prakiraan cuaca suatu tempat. Adapun cara kerja *OpenweatherMap API* pada aplikasi yang dibangun yaitu :

1. Sistem akan mengirimkan permintaan kepada *OpenweatherMap API* melalui internet.
2. *OpenweatherMap* akan menerima request dari pengguna.
3. Sistem akan mendapat kan balasan dari *OpenweatherMap* berupa data JSON.
4. Pengguna akan menerima data prakiraan cuaca.

#### 4. Facebook API

Pada aplikasi yang dibangun ini juga menggunakan Facebook API untuk memungkinkan pengguna login menggunakan akun facebook tanpa melakukan registrasi terlebih dahulu. Dengan begitu pengguna bisa *login* dengan lebih mudah. Facebook API juga digunakan untuk menentukan rekomendasi berdasarkan kesukaan pengguna pada akun facebook masing masing.

Adapun cara kerja Facebook API pada aplikasi yang dibangun yaitu :

1. Pengguna menggunakan API *key* untuk terhubung ke facebook API.
2. Sistem meminta *request* ke facebook API melalui API *key*
3. Facebook mengirimkan *response* berupa data JSON.

#### 5. Firebase Database

*Firebase realtime database* atau disebut juga *firebase database* adalah *database* yang di-*host* di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap *client* yang terhubung. Ketika membuat aplikasi lintas-*platform* dengan SDK Android, iOS, dan JavaScript, semua *client* akan berbagi sebuah instance *realtime database* dan menerima update data terbaru secara otomatis.

#### 6. Firebase Cloud Messaging

*Firebase Cloud Messaging* (FCM) adalah solusi *server push* untuk aplikasi android. *Server push* maksudnya adalah fitur yang digunakan apabila aplikasi *server* ingin mentrigger aplikasi *mobile*. Google menyediakan fasilitas FCM yang bisa digunakan untuk aplikasi android maupun iOS.

#### C. Analisis Rekomendasi

Rekomendasi pada aplikasi ini berdasarkan dari histori data lokasi pembeli saat pedagang berjualan. Sistem akan mencocokkan alamat atau nama lokasi dari histori pembeli. Contoh studi kasus adalah sebagai berikut.

Pak Ali merupakan seorang pedagang cuangki dari Sekeloa. Pak Ali setiap harinya berdagang dari Komplek Sekeloa ke Komplek Tubagus Ismail sampai ke Komplek Sadang Serang. Menurut pak Ali dan hasil beberapa survei kriteria dan tempat berdagang didapatkan informasi sebagai berikut :

#### 1. Hari Pertama

Data penjualan hari pertama dapat dilihat pada tabel 1 berikut :

Tabel 1 Data Penjualan Hari Pertama

Nama Tempat	Waktu (WIB)	Jumlah Pembeli (per hari)
Tubagus Ismail	07.00 – 09.00	5
Sekeloa	09.00 – 12.00	10
Sadang Serang	15.00 – 18.00	4

Dari 3 tempat diatas diketahui bahwa komplek Sekeloa merupakan pembeli terbanyak.

#### 2. Hari Kedua

Data penjualan hari kedua dapat dilihat pada tabel 2 berikut :

Tabel 2 Data Penjualan Hari Kedua

Nama Tempat	Waktu (WIB)	Jumlah Pembeli (per hari)
Tubagus Ismail	07.00 – 09.00	11
Sekeloa	09.00 – 12.00	4
Sadang Serang	15.00 – 18.00	6

Dari 3 tempat diatas diektahui bahwa komplek Tubagus Ismail merupakan dengan pembeli terbanyak.

3. Hari Ketiga

Data penjualan hari ketiga dapat dilihat pada tabel 3 berikut :

Tabel 3 Data Penjualan Hari Ketiga

Nama Tempat	Waktu (WIB)	Jumlah Pembeli (per hari)
Tubagus Ismail	07.00 – 09.00	9
Sekeloa	09.00 – 12.00	5
Sadang Serang	15.00 – 18.00	7

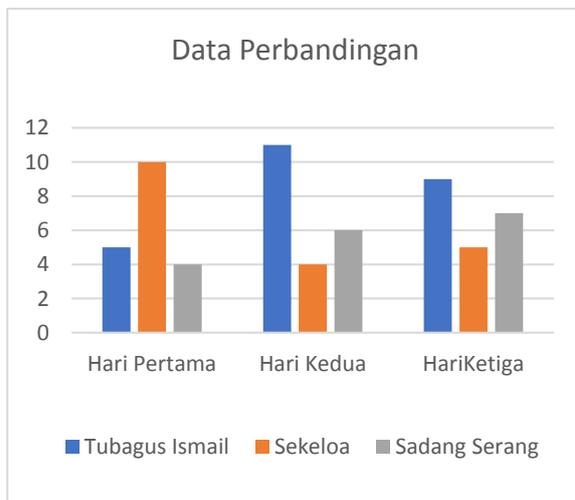
Dari tiga tempat diatas komplek Tubagus Ismail dengan pembeli terbanyak.

Hasil total data keseluruhan dapat dilihat pada tabel 4 berikut :

Tabel 4 Data Keseluruhan

Nama Tempat	Waktu (WIB)	Jumlah Pembeli (per hari)
Tubagus Ismail	07.00 – 09.00	25
Sekeloa	09.00 – 12.00	19
Sadang Serang	15.00 – 18.00	17

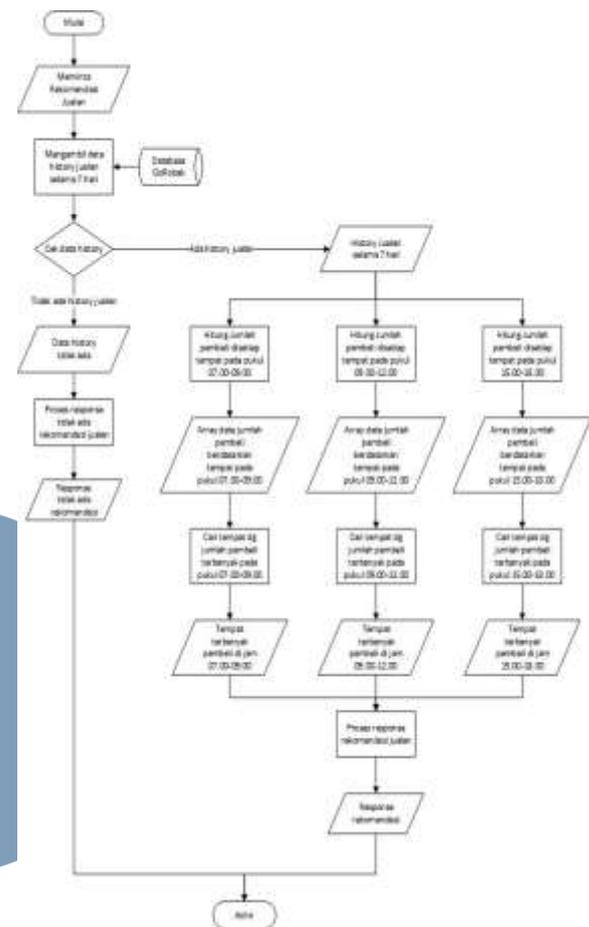
Dari hasil tabel data keseluruhan dan grafik yang ada pada gambar 3, terlihat bahwa grafik batang berwarna biru lebih dominan maka dapat disimpulkan bahwa komplek Tubagus Ismail dengan pembeli terbanyak yang akan direkomendasikan kepada pedagang untuk berjualan.



Gambar 3 Data Perbandingan

D. Alur Rekomendasi

Alur dari rekomendasi secara lengkap yang ada pada aplikasi dapat dilihat pada gambar 4 dibawah ini:



Gambar 4 Alur Rekomendasi

Berikut merupakan contoh implementasi yang dilakukan :

1. Pembeli yang melakukan pembelian kepada pedagang dengan tap tombol *checkboxlist* (ikon centang) pada halaman percakapan pemesanan akan dicatat nama lokasinya dengan menggunakan *class Geocoder*. Dari *class geocoder* ini bisa didapatkan alamat atau nama lokasi dari *latitude* dan *longitude* yang dimasukkan pada parameter. Karena setiap perpindahan *device* android *latitude* dan *longitudenya* berbeda, maka data alamat atau nama lokasi inilah yang diambil untuk diolah menjadi informasi rekomendasi tempat yang paling banyak pembelinya untuk pedagang.

Berikut merupakan contoh *function* geocoder untuk mendapatkan alamat atau nama lokasi dari *latitude* dan *longitude* :

```

fun Context.getTempat(latitude: Double, longitude: Double): String? {
    var addresses = mutableListOf<Address>()
    val geocoder = Geocoder(this, Locale.getDefault());
    try {
        addresses = geocoder.getFromLocation(latitude, longitude, 1)
        return if (addresses.size > 0) {
            val address = addresses.get(0).getAddressLine(0)
            val city = addresses.get(0).getLocality()
            val state = addresses.get(0).getAdminArea()
            val country = addresses.get(0).getCountryName()
        }
    }
}
    
```

```

        val postalCode = addresses.get(0).getPostalCode()
        val knownName = addresses.get(0).getFeatureName()
        address + ", " + city + ", " + state;
    } else {
        ""
    }
}
} catch (e: IOException) {
    e.printStackTrace()
}
return null
}
    
```

Contoh pemanggilannya:  
 alamatGet = getTempat(latitude, longitude)

Misal :

latitude = -6.8868245

longitude = 107.6131064

Maka diperoleh nama alamat/lokasi = Jl. Ir. H. Djuanda No.167, Lb. Siliwangi, Coblong, Kota Bandung, Jawa Barat

2. Ketika penjual menekan tombol mulai berkeliling dari aplikasi, maka sistem akan melakukan proses pencarian tempat yang paling banyak pembelinya dari histori data lokasi pembeli saat pedagang berjualan. Sistem akan mencocokkan alamat atau nama lokasi dari histori pembeli.

Misalkan terdapat data lokasi pembeli seperti pada tabel 5 berikut.

Tabel 5 Tabel Data Pembeli

Nama Tempat	Jumlah Pembeli (per hari)
Komplek A	5
Komplek B	10
Komplek C	4

Dilihat pada tabel 5 didapatkan bahwa Komplek B merupakan pembeli terbanyak yang nantinya akan di rekomendasikan kepada pedagang.

3. Setelah data rekomendasi tersebut didapat maka selanjutnya adalah mencari *latitude* dan *longitude* dari alamat atau nama lokasi yang direkomendasikan dengan menggunakan *class geocoder* kembali.

Berikut merupakan contoh untuk mendapatkan *latitude* dan *longitude* dari *geocoder*.

```

        val geocoder = Geocoder(this@SetLocationMapsActivity)
        try {
            addressList = geocoder.getFromLocationName(lokasi, 1)
        } catch (e: IOException) {
            e.printStackTrace()
        }
        var address: Address? = null
        try {
            address = addressList.get(0)
        } catch (e: NullPointerException) {
            e.printStackTrace()
        }
        if (address != null) {
            val latLng = LatLng(address.getLatitude(), address.getLongitude())
        }
    
```

Misal :

lokasi = "Jl. Ir. H.Djuanda No.167, Lb. Siliwangi, Coblong, Kota Bandung, Jawa Barat"

Maka

latLng.latitude = -6.8868245

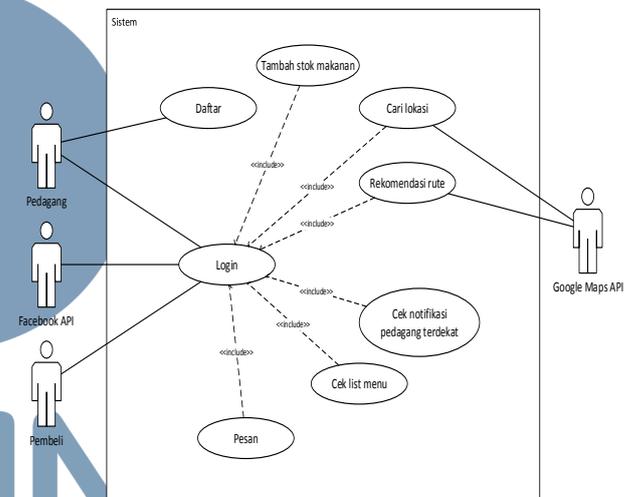
latLng.longitude = 107.6131064

4. Setelah diperoleh *latLng* (*latitude* dan *longitude*) dari lokasi yang direkomendasikan, selanjutnya akan ditampilkan posisi lokasinya pada *maps* sekaligus meminta perkiraan cuaca lokasi yang direkomendasikan tersebut dari *openweather* dengan mengirim *latitude* dan *longitude*.

5. Setelah perkiraan cuaca diperoleh, selanjutnya aplikasi akan memunculkan informasi alamat lokasi dan perkiraan cuaca dilokasi tersebut.

E. Use Diagram

*Use case diagram* digunakan untuk menggambarkan interaksi antara aktor dengan aktifitas yang terdapat pada sistem yang dibangun. Dengan menggambarkan *use case diagram*, fungsional-fungsional yang terdapat pada sistem dapat dilihat secara singkat. Adapun *use case diagram* pada sistem *mobile* yang dibangun dapat dilihat pada Gambar 5 berikut :



Gambar 5 Use Case Diagram

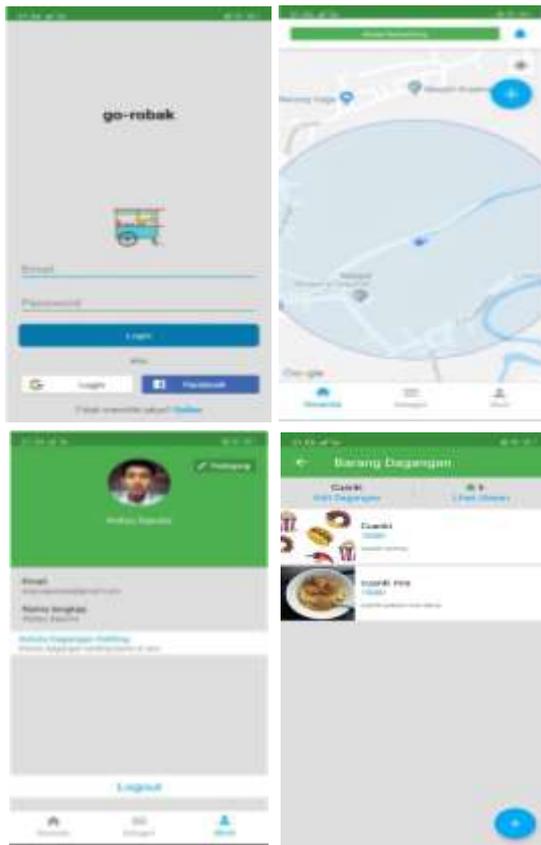
F. Implementasi Sistem

Tahap implementasi sistem merupakan tahap penjeremahan perancangan berdasarkan hasil analisis. Dimasukan ke dalam bahasa pemrograman tertentu serta penerapan sistem yang akan dibangun pada sebuah aplikasi. Aplikasi ini diimplementasikan menggunakan spesifikasi *device* yang ada pada tabel 6.

Tabel 6 Implementasi perangkat lunak komputer dan *smartphone*

Komputer	
Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 10 64-bit
Web Browser	Google Chrome
IDE	Android Studio
Text Editor	Notepad++
Smartphone	
Perangkat Lunak	Spesifikasi
Sistem Operasi	Android 6.0
Processor	Snapdragon 625
Layar	5,5 inch

Hasil tampilan antarmuka dari aplikasi *mobile* yang dibangun dapat dilihat pada gambar 6 berikut :



Gambar 6 Implementasi Tampilan Antarmuka

Berdasarkan hasil pengamatan dan pengujian secara fungsional dari sistem yang telah diimplementasikan, dapat dijelaskan hasil seperti berikut :

1. Aplikasi sudah bisa melakukan login baik itu dengan menggunakan *username* dan *password* yang telah didaftarkan sebelumnya, maupun *login* dengan menggunakan akun *google* atau *facebook*. Hasil tersebut mengartikan bahwa penggunaan *facebook API* dan *Google API* telah berjalan dengan baik.
2. Ketika pedagang keliling mengaktifkan fitur mulai berkeliling, maka pada aplikasi calon pembeli mendapatkan notifikasi bahwa pedagang telah memulai berjualan pada hari tersebut. Fitur ini memang terkesan sederhana, tetapi sangat membantu bagi calon pembeli sehingga diketahui secara pasti apakah pada hari tersebut pedagang yang ditunggu berjualan atau tidak.
3. Aplikasi juga telah memberikan notifikasi kepada calon pembeli apabila ada pedagang yang sedang ditunggu sudah berada dengan jarak radius 100 meter dari posisi calon pembeli. Saat ini jarak tersebut telah ditentukan secara *hardcore* dalam *script*, namun untuk penelitian selanjutnya akan dikembangkan kembali

sehingga pengguna bisa mengubah besaran jarak notifikasi yang diinginkan.

4. Berdasarkan hasil implementasi nomor 2 dan 3 sebelumnya, hasil tersebut mengartikan bahwa fitur dari *Firebase Cloud Messaging (FCM)* telah berhasil diimplementasikan dengan baik.

5. Apabila calon pembeli tidak mengaktifkan fitur notifikasi, masih dapat melihat pedagang makanan dan minuman yang lokasi berjualannya berjarak radius 100 meter dari lokasi calon pembeli. Hal ini sangat menguntungkan bagi para pedagang untuk mempromosikan barang dagangannya sehingga pada ujungnya diharapkan dapat meningkatkan pendapatan dari setiap pedagang keliling.

6. Aplikasi ini sudah memberikan rekomendasi tempat berjualan yang paling sesuai bagi para pedagang keliling berdasarkan *history* penjualan dari minggu sebelumnya. Dengan demikian para pedagang dapat dimudahkan terutama untuk menentukan rute mana terlebih dahulu yang akan digunakan sebagai lokasi berjualan. Penentuan rute tersebut berdasarkan *history* penjualan dari minggu sebelumnya.

7. Selain rekomendasi rute, pedagang keliling juga diberikan informasi prakiraan cuaca dari lokasi tujuan berjualan. Fitur ini berjalan dengan menggunakan *OpenWeather API*. Informasi prakiraan cuaca tersebut dapat dilihat seperti pada gambar 7 berikut.



Gambar 7 Implementasi Prakiraan Cuaca

Prakiraan cuaca tersebut sesuai dengan prakiraan ditempat lokasi pedagang berjualan. Prakiraan tersebut memberikan informasi setiap 3 jam. Dengan demikian, para pedagang keliling dapat mempersiapkan dengan keadaan cuaca yang diperkirakan.

#### G. Pengujian Sistem

Pengujian merupakan bagian yang penting dalam siklus pengembangan perangkat lunak. Hasil pengujian dari setiap fungsional yang ada pada *usecase diagram* yang diuji mendapatkan kesimpulan bahwa setiap pengujian berhasil dilakukan tanpa ada kesalahan.

Sedangkan pengujian kepada pengguna dilakukan kepada 30 pedagang makanan keliling, dengan hasil sebesar 88,39 % dari total pedagang makanan keliling merasa terbantu dengan adanya aplikasi ini untuk mendapatkan pembeli maupun pelanggan. Selain itu para pedagang makanan keliling dapat mengetahui informasi tempat berdagang yang strategis, informasi cuaca, serta jadwal dan rute perjalanan yang paling baik untuk berdagang. Suasana ketika pengujian kepada para pedagang keliling dapat dilihat pada gambar 8.



Gambar 8 Tahapan Pengujian Kepada Pengguna

Sewaktu implementasi dan pengujian sistem kepada para pedagang makanan keliling, para pengguna sangat antusias dengan adanya aplikasi ini serta memberikan *feedback* positif dalam hal memberikan saran untuk kemudian menjadi fitur tambahan dalam pengembangan aplikasi selanjutnya.

Namun demikian, dari hasil implementasi dan pengujian yang dilakukan, didapatkan juga beberapa kendala ketika dioperasikan. Kendala tersebut seperti di beberapa tempat berjualan sinyal internet dan GPS tidak berjalan dengan baik sehingga dapat mengurangi akurasi dari informasi yang diberikan. Selain itu, dikarenakan aplikasi ini menggunakan sensor GPS yang diharuskan aktif sepanjang hari, menjadikan *smartphone* yang digunakan jadi lebih boros baterai. Hal tersebut membuat para pedagang menyiapkan *power* tambahan seperti baterai cadangan atau juga mempersiapkan *power bank* disetiap aktifitas berjualan.

## V. SIMPULAN

Adapun kesimpulan yang didapatkan dari hasil penelitian ini adalah aplikasi ini membantu pedagang dalam berdagang keliling ke tempat-tempat yang ada di Kota Bandung. Selain bagi pedagang, aplikasi ini juga bermanfaat bagi para calon pembeli untuk mengetahui informasi dari pedagang keliling yang sedang ditunggunya.

## UCAPAN TERIMA KASIH

Peneliti mengucapkan terimakasih kepada Kemenristekdikti yang telah mendanai dan mendukung proses penelitian ini melalui skema Penelitian Terapan tahun pertama (2019) melalui kontrak penelitian tahun jamak tahun anggaran 2019 dengan nomor kontrak 2898/L4/PP/2019 dan nomor kontrak 045/SP/LPPM/UNIKOM/III/2019.

## DAFTAR PUSTAKA

- [1] S. Ahmad, "Pembangunan Mobile Application Store Alisha Fancyshop Memanfaatkan Sistem Push Notification Dan Location Based Service (LBS) Berbasis Android," Jurnal Ilmiah Komputer dan Informatika (KOMPUTA), pp. 45-54, 2016.
- [2] M. Siddik and A. Nasution, "Perancangan Aplikasi Push Notification Berbasis Android," JURTEKSI (Jurnal Teknologi dan Sistem Informasi), vol. 4, no. 2, pp. 149-154, 2018.
- [3] M. A. Kahfi, S. Arifin and A. Rahmadiansah, "Perancangan Prediktor Cuaca Maritim Berbasis Logika Fuzzy Menggunakan User Interface Android," JURNAL TEKNIK POMITS, vol. 2, no. 2, pp. 324-328, 2013.
- [4] T. E. Rahman and I. W. Mustika, "Pengembangan Aplikasi Bergerak untuk Mendeteksi Tingkat Kemacetan Lalu Lintas dan Cuaca Memanfaatkan Google Maps API, OpenWeatherMap API, dan GPS," pp. 1-6.
- [5] R. Herdianto and E. B. Setiawan, "Penggunaan Smartphone Android sebagai Alat Analisis Kebutuhan Kandungan Nitrogen pada Tanaman Padi," JNTETI, vol. 7, no. 3, pp. 273-280, 2018.
- [6] J. Setiawan, E. Kristianto and F. , "Implementasi Push Notification Pada Informasi Perkuliahan Dan Kegiatan Mahasiswa Berbasis Android," Jurnal Teknik dan Ilmu Komputer, vol. 4, no. 14, pp. 211-219, 2015.
- [7] S. R. Juraman, "Pemanfaatan Smartphone Android Oleh Mahasiswa Ilmu Komunikasi Dalam Mengakses Informasi Edukatif," vol. 3, no. 1, 2014.
- [8] C. Dewi and K. N. Pramono, "Pembuatan Aplikasi Pencatatan Servis Mobil di PT. Armada International Motor Berbasis Android," JNTETI, vol. 4, no. 4, 2015.
- [9] A. Fauzi, "Penerapan Location-Based Service Pada Layanan Informasi Budaya Indonesia Di Perangkat Mobile," Faktor Exacta, vol. 8, no. 3, pp. 250-260, 2015.
- [10] I. Bahari and E. Wijaya, "Pembangunan Aplikasi Pemandu Dan Rekomendasi Gerbang Tol Berbasis Mobile Android".
- [11] I. Ziad, "Rancang Bangun Pelacak Lokasi Dengan Teknologi GPS," Jurnal Teknologi dan Informatika, vol. 3, no. 1, pp. 1-14, 2013.
- [12] R. Rachmawan, "Rancang Bangun Aplikasi Perangkat Bergerak berbagi foto Berbasis Android menggunakan API Facebook, Flickr dan Picasa," Jurnal Teknik POMITS, vol. 1, no. 1, 2012.
- [13] M. Nazir, Metode Penelitian, Bogor: Ghalia, 2013.
- [14] S. Widiyanto, "Pengembangan Aplikasi City Guide Berbasis Android," Jurnal STT STIKMA Internasional, vol. 7, no. 1, pp. 52-60, 2016.
- [15] K. Arun and M. G. Nayagam, "Building Applications with Social Networking API's," Int. J. Advanced Networking and Applications, vol. 5, no. 5, pp. 2070-2075, 2014.