

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Beberapa penelitian yang menggunakan pencarian antara lain sebagai berikut :

1. Penelitian tentang pencarian lokasi pernah dilakukan sebelumnya oleh Vitaloka Nuristyana(2014). Aplikasi yang dibuat dapat menampilkan informasi berupa lokasi titik koordinat tambal ban dengan titik koordinat pengguna.
2. Penelitian tentang pencarian lokasi pernah dilakukan oleh Hardika Catur Saputra(2015). Aplikasi yang dibuat dapat menampilkan informasi berupa gambar, nama desa, *rute*, dan lokasi Desa Wisata di wilayah Yogyakarta.
3. Penelitian pencarian lokasi oleh Ilham(2014). Aplikasi yang dibuat merupakan aplikasi yang dapat menampilkan informasi wisata candi, alam, pantai, sejarah,kuliner di Yogyakarta.
4. Chairul Anam (2014) membuat penelitian yang bertujuan untuk menunjukkan arah jalur menuju lokasi Pelanggan. Aplikasi ini akan menampilkan menentukan posisi Pengguna dan menampilkan lokasi Pelanggan pada *map* dengan menggunakan *Google Maps* dan terdapat fitur untuk mengetahui kemacetan jalan pada jam tertentu.
5. Edo Surya Putra (2014) membangun aplikasi pencarian lokasi toko oleh- oleh Yogyakarta dengan metode layanan berbasis lokasi pada *platform android* yang

dapat menampilkan lokasi toko oleh-oleh terdekat dengan pengguna, menampilkan rute dan jarak menuju toko oleh-oleh serta dilengkapi fasilitas *call dialer*.

Terdapat perbedaan antara aplikasi tersebut dengan aplikasi yang dibuat dalam tugas akhir ini, yaitu antara lain *User Interface* dan aplikasi yang akan dibuat dapat mencari dan menampilkan posisi dan jarak petugas – petugas tambal ban panggilan dari posisi pengguna berada radius 15 km. Pencarian petugas tambal ban panggilan bersifat *real time*.

Tabel 2.1 Tinjauan Pustaka

NO	Penulis	Objek	Bahasa Pemrograman	Output
1	Vitaloka Nuristyana (2014)	Tambal Ban	Java	Informasi Lokasi Tambal Ban Di Yogyakarta
2	Ilham k (2014)	Tempat Wisata	Java	Informas Gambar, Nama Desa Dan Rute Lokasi Desa Wisata Di Yogyakarta
3	Hardika Catur Saputra (2015)	Desa Wisata & Kuliner	Java	Informasi Wisata DI Yogyakarta
4	Chairul Anam (2014)	Lokasi Pelanggan Internet Isp	Java	Menampilkan Lokasi Pelanggan Internet Isp (Studi Kasus PT. Data Dinamika Utama Yogyakarta)
5	Edo Surya Putra (2014)	Toko Oleh – Oleh	Java	Menampilkan Informasi Lokasi Toko oleh – oleh di Yogyakarta.
6	Rancangan Yang di usulkan dari peneliti	Tambal Ban	Java	Menampilkan Marker Petugas Tambal Ban Keliling Di Yogyakarta Secara Real Time.

## **2.2 Dasar Teori**

### **2.2.1 Location Based Service (LBS)**

Location Based Service (LBS) atau layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi (Vitaloka Nuristyana,2014, STMIK AKAKOM Yogyakarta).Dua unsur utama *Location Based Service (LBS)* adalah :

#### **1. Location Manager(API Maps)**

Menyediakan *tools* atau *source* untuk *LBS, Application Programming Interface(API Maps)* menyediakan fasilitas menampilkan memanipulasi maps atau peta beserta fitur – fitur lainnya seperti tampilan satelit, *street*(jalan), maupun gabungannya.

#### **2. Location Provides(API Location)**

Menyediakan teknologi pencarian lokasi yang digunakan oleh *device*(Perangkat). *API location* berhubungan dengan data *GPS (Global Positioning System)* dan data lokasi *real – time. API location* berada pada paket *android* yaitu dalam paket *android.location*. Dengan *Location Manager*, penulis dapat menentukan lokasi saat itu juga(*real-time*).

### **2.2.2 Global Positioning System(GPS)**

*Global Positioning System(GPS)* merupakan sistem koordinat global yang dapat menentukan koordinat posisi benda dimana saja di bumi baik koordinat lintang(*latitude*) dan bujur(*longitude*), maupun ketinggiannya. Sistem *GPS* dapat

memberikan data koordinat global karena didukung oleh informasi dari 24 satelit yang ada pada ketinggian orbit sekitar 11.000 mil di atas bumi. Satelit – satelit tersebut terbagi dalam 6 bidang orbit yang berada dengan masing – masing bidang orbit diisi oleh 4 satelit. Dengan konfigurasi seperti ini, maka setiap titik di bumi selalu dapat ditentukan koordinatnya oleh *GPS* setiap 24 jam penuh sehari.

Setiap satelit *GPS* memancarkan sinyal – sinyal gelombang *mikro*. *GPS receiver* menggunakan sinyal satelit yang diterima untuk melakukan triangulasi posisi dengan cara mengukur lama perjalanan waktu sinyal dikirimkan dari satelit, kemudian mengalihkannya dengan kecepatan cahaya untuk menentukan secara tepat berapa jauh dirinya dengan satelit.

Dengan mengunci minimum 3 sinyal dari satelit yang berbeda, maka *GPS receiver* dapat menghitung posisi tetap sebuah titik yaitu koordinat posisi lintang dan bujur (*latitude* dan *longitude*). Penguncian sinyal ke – 4 membuat pesawat pengirim *GPS* dapat menghitung posisi ketinggian titik tersebut terhadap permukaan laut (*Altitude*). *GPS receiver* akan terus menjaga dan mengunci sinyal satelit yang diperlukan untuk melakukan triangulasi secara bersama dan paralel. Dengan sistem ini, informasi yang diterima selalu *up to date*. (Hardika, 2015)

### **2.2.3 Google Maps**

Merupakan sebuah layanan peta dunia *virtual* yang berbasis web yang disediakan oleh *Google*. Layanan ini gratis dan dapat ditemukan di <http://maps.google.com>.

Google Maps menawarkan peta yang dapat digeser (*panned*), diperbesar (*zoom in*), diperkecil (*zoom out*), dapat diganti beberapa mode ( *map, satelllite, hybrid* dan lain – lain), fitur pencarian rute (*direction*), dan juga pencarian tempat(*place*). (Hardika, 2015).

### **2.2.4 Java**

*Java* adalah sebuah bahasa pemrograman yang dikembangkan oleh perusahaan sun Microsystem yang digunakan pada *chip – chip embedded* untuk *device intelegent consumer electronic*. *Java* digunakan dalam programming karenamemiliki keunggulan – keunggulan yang tidak dimiliki bahasa lain (Wahana,2008). *Java* tidak bergantung pada *platform* yang artinya *java* dapat dijalankan pada sembarang komputer dan bahkan pada sembarang sistem operasi.

### **2.2.5 XML (eXtensible Markup Language)**

*XML* adalah kumpulan aturan – aturan yang mendefinisikan suatu sintak yang digunakan untuk menjelaskan dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan *tag*. *XML* adalah bahasa *markup* yang digunakan untuk mengolah *meta data* (informasi tentang data) yang menggambarkan struktur dan tujuan

data yang terdapat dalam *document XML*, namun bukan menggambarkan format tampilan tersebut.(Hardika, 2015).

### **2.2.6 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 menciptakan aplikasi mereka. (Safaat, 2012)

### **2.2.7 Firebase**

*Firebase* adalah *Backend as a Service* (BaaS) yang saat ini dimiliki oleh *Google*. *Firebase* merupakan solusi yang ditawarkan oleh *Google* untuk mempermudah pengembangan aplikasi *mobile*. Dua fitur menarik dari *Firebase* adalah *Firebase Remote Config* dan *Firebase Real Time Database*. Selain itu juga terdapat fitur pendukung untuk aplikasi yang memerlukan *push notification* yaitu *Firebase Notification Console*. *Firebase Database* merupakan penyimpanan basis data *nonSQL* yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data itu antara lain *String*, *Long*, dan *Boolean*. Data pada *Firebase Database* disimpan sebagai objek *JSON tree*. Tidak seperti basis data *SQL*, tidak ada tabel dan baris pada basis data *non-SQL*. Ketika ada penambahan data, data tersebut akan menjadi *node* pada struktur *JSON*. *Node* merupakan simpul yang berisi data dan bisa memiliki cabang-cabang berupa *node* lainnya yang berisi data pula. Proses pengisian suatu data ke *Firebase Database* dikenal dengan istilah *push*. Selain *Firebase Database*, *Firebase*

menyediakan beberapa layanan lainnya yang juga dimanfaatkan dalam pengembangan aplikasi ini. Layanan tersebut antara lain *Firebase Authentication*, *Storage*, dan *Cloud Messaging*. pengembangan aplikasi, layanan lainnya yang digunakan pada pengembangan aplikasi adalah *Firebase Storage*. Layaknya sebuah penyimpanan awan, *Firebase Storage* memungkinkan pengembang untuk mengunggah atau mengunduh sebuah berkas.

### **2.2.8 Otentikasi (Authentication)**

Otentikasi (*Authentication*) adalah proses untuk memastikan bahwa kedua ujung koneksi dalam keadaan benar atau sama. Seperti *password* pada umumnya, syarat agar otentikasi berhasil adalah *password* yang dikirimkan *client* harus sama dengan *password* yang disimpan di *server*. Dengan alasan keamanan jarang sekali server menyimpan password user dalam bentuk plain-text. Biasanya server menyimpan password user dalam bentuk hash sehingga tidak bisa dikembalikan dalam bentuk plain-text. Jadi syarat otentikasi berhasil di atas bisa diartikan sebagai hasil penghitungan hash dari *password* yang dikirim klien harus sama dengan nilai hash yang disimpan dalam server.

### **2.2.9 One Time Password**

*One time password* atau OTP adalah kode verifikasi sekali pakai berupa angka yang dikirimkan lewat SMS dan *email*. Kode OTP berupa 6 digit angka yang akan hangus dalam 2 sampai 5 menit dan hanya bisa digunakan sekali. Kode ini digunakan untuk proses autentikasi *login*.

Kode OTP adalah salah satu bagian dari *two way authentication* yang menambah *layer* proteksi data Anda di internet. Umumnya, kode OTP digunakan oleh aplikasi yang mempunyai sifat rahasia dan personal seperti aplikasi dompet digital dan *mobile banking*. Namun, kode OTP juga bisa digunakan untuk proses verifikasi transaksi *eCommerce marketplace*. Proses verifikasi OTP digunakan untuk menghindari masalah seperti pencurian data maupun penyalahgunaan akun.

Kode OTP mempunyai karakteristik diantaranya :

1. Hanya bisa digunakan satu kali.
2. Biasanya berbentuk angka dan karakter.
3. Tidak bisa digunakan dalam durasi waktu yang panjang.
4. Umumnya digunakan untuk melindungi transaksi uang *online*.
5. Bersifat sangat rahasia.

### **2.2.10 Geofire**

*GeoFire* adalah pustaka *JavaScript open-source* yang memungkinkan untuk menyimpan dan menampilkan sekumpulan item berdasarkan lokasi geografisnya. *GeoFire* menggunakan *Firebase* untuk penyimpanan data, memungkinkan hasil 18 *query* diperbarui secara waktu nyata saat perubahan. *GeoFire* melakukan lebih dari sekadar mengukur jarak antar lokasi, itu selektif hanya memuat data dekat lokasi tertentu, menjaga aplikasi tetap ringan dan responsif, bahkan dengan dataset yang sangat besar (Wenger, 2014). *GeoFire* dirancang sebagai *add-on* ringan untuk *Firebase*. Untuk menjaga semuanya tetap sederhana, *GeoFire* menyimpan data dalam

formatnya sendiri dan lokasinya sendiri di dalam basis data *Firestore*. Ini memungkinkan format data yang ada dan aturan keamanan untuk tetap tidak berubah sementara masih menyediakan solusi mudah untuk *query geo*.

### **2.2.11 Web Service**

*Web Service* adalah aplikasi sekumpulan data(*database*), perangkat lunak(*software*) atau bagian dari perangkat lunak yang dapat diakses secara *remote* oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum, *web service* dapat diidentifikasi dengan menggunakan *URL* seperti hanya *web* pada umumnya.

### **2.2.12 Unified Markup Language (UML)**

*UML* adalah sebuah bahasa yang telah menjadi standar dalam industri visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standard untuk merancang model sebuah sistem. Pemodelan sistem yang akan digunakan pada pembuatan aplikasi ini adalah :

- *Use Case Diagram*, menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem(*actor*). Diagram ini menunjukkan fungsionalitas suatu sistem dan bagaimana sistem berinteraksi dengan dunia luar.
- *Sequence Diagram*, menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *usecase*. *Sequence diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *usecase*.

- *Activity Diagram*, memodelkan alur kerja sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena dapat memodelkan sebuah alur kerja dari suatu aktivitas ke aktivitas lainnya .
- *Class Diagram*, memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain ( *data logical view*) dari suatu sistem. Memiliki 3 area pokok, yaitu : nama kelas, atribut, dan *methode* atau *operation*. (Yuli, 2015).