

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Daerah Istimewa Yogyakarta (DIY) adalah salah satu daerah otonom setingkat propinsi yang ada di Indonesia. DIY memiliki berbagai predikat, seperti kota perjuangan, kota kebudayaan, kota pelajar dan kota pariwisata (Pemerintah Kota Yogyakarta,2007), (Adrisijanti,2007), (Susilo,2009), (Bernas Jogja,2010). Sebagai kota berpredikat pariwisata, kota DIY memiliki investasi dan permintaan perjalanan wisata yang cukup tinggi (Arliani,2009), (Suyanto,2006).

DIY dengan berbagai predikat kotanya memiliki banyak tempat yang bisa dikunjungi dengan banyak pilihan transportasi yang tersedia (Sugiyanto et.al.,2011), (Munawar,2006). Namun dalam kehidupan sehari-hari sering kali kita mengalami kendala dalam mencari lokasi suatu tempat dimana daerah tersebut belum pernah kita kunjungi, oleh karena itu diperlukan suatu aplikasi tentang *city directory*(Ackman et.al, 2011) yang dapat membantu pengguna dalam pencarian lokasi suatu tempat (Kushwaha et.al, 2011), (Filjar et.al, 2008), (Burak et.al, 2004), (Vrcek et.al, 2009) dan dapat menampilkan peta yang memberikan informasi tentang tempat yang dicari (Massengill,2010), (Nguyen,2009), (Hoar,2009), (Lin,2009), (Yu,2010), (Schmid,2010).

Ada beberapa penelitian tentang aplikasi pencarian lokasi suatu tempat seperti *city directory*. Liarokapis et.al(2007), Umlauf et.al(2003), Schwinger et.al(2002), Kenteris et.al(2011) telah melakukan penelitian terhadap aplikasi

mobile pencarian lokasi yang dapat memandu dan memberikan informasi tentang suatu tempat pariwisata yang bisa diakses kapan saja dan dimana saja. Kenteris et.al,(2011) dalam penelitian yang berjudul *Mytilene E-guide: a multiplatform mobile application tourist guide exemplar* membahas tentang aplikasi *mobile* untuk turis yang *multiplatform* serta dapat diakses secara online maupun offline. Shu(2010) dalam penelitiannya yang berjudul *City Guide over Android* membahas tentang *mobile city guide* dengan menggunakan platform Android dan menjelaskan prototipe dari *city guide* tersebut. Tan et.al(2009) dalam penelitiannya membahas tentang bagaimana cara mengatasi kurangnya pemahaman akan konteks dan tujuan wisatawan sehingga dapat memenuhi kebutuhan pengguna.

Berdasarkan kajian pustaka diatas, penulis mengambil topik yang menggabungkan penelitian diatas dengan mengangkat lokal konten di Yogyakarta tentang *mobile city directory* berbasis Android yang akan memberikan informasi tempat-tempat di Yogyakarta.

Beberapa penelitian tentang aplikasi tentang pariwisata yang ada di Indonesia adalah Perancangan dan Implementasi Aplikasi *Mobile Bandung Guidance* Berbasis Teknologi *Location Based Service* Menggunakan Platform BlackBerry (Ferinata), aplikasi ini digunakan untuk memberikan informasi kepada pengguna tentang info lokasi-lokasi penting di Bandung serta lokasi terdekat dari posisi pengguna beserta rute jalannya. Kemudian ada Rancang Bangun Aplikasi Sistem Informasi *Context Aware* untuk Pariwisata berbasis Pengguna Menggunakan Teknologi *Bluetooth* (Amaliah,2008), penelitian ini

membuat aplikasi sistem informasi *context aware* untuk pariwisata dengan menggunakan teknologi *bluetooth*. Kemudian penelitian tentang Perancangan dan Pembuatan Aplikasi Sistem Informasi Geografis berbasis Web untuk Penunjuk Perjalanan Wisata di Surabaya (Hidayati,2006), dan Aplikasi Informasi dan Rute Pariwisata di kota Pacitan dengan menggunakan J2ME (*Java 2 Micro Edition*) (Hakim,2011). Tabel dibawah akan menunjukkan spesifikasi perbandingan aplikasi yang telah ada dengan aplikasi yang akan dibuat oleh penulis.

Tabel 1. Perbandingan Spesifikasi Aplikasi

Fungsionalitas	Ferinata	Amaliah	Hidayati	Hakim	Penulis
Pencarian lokasi	√		√	√	√
Peta Lokasi	√		√	√	√
Assisted GPS	√				√
Jarak	√		√		√
Rute	√		√	√	√
Transportasi					√
Info Umum	√	√	√	√	√

2.2 Landasan Teori

2.2.1 Layanan Berbasis Lokasi

Sebuah layanan berbasis lokasi adalah layanan informasi atau hiburan yang dapat diakses dengan perangkat *mobile* melalui jaringan selular. Sistem Layanan Berbasis Lokasi, atau lebih dikenal dengan *Location-Based Services* (LBS), menggabungkan antara proses dari layanan *mobile* dengan posisi geografis dari penggunanya. Posisi target, di mana sebuah target bisa jadi adalah pengguna *Location-Based Services* itu sendiri atau entitas lain yang tergabung dalam suatu layanan.

Ada 2 tipe layanan yang bisa digunakan dalam *Location-Based Services* untuk memperoleh posisi pengguna, yaitu dengan menggunakan posisi sel jaringan atau dengan GPS maupun aGPS. Dari kedua cara ini akan didapatkan posisi pengguna dalam bentuk koordinat latitude dan longitude. Latitude adalah representasi dari arah Utara-Selatan, sedangkan longitude adalah representasi dari arah Timur-Barat. Selain dari sisi posisi pengguna, *Location-Based Services* juga bisa dilihat dari sisi layanan yang diberikan. Dari sisi layanan yang diberikan, *Location-Based Services* bisa dibagi menjadi 2 yaitu *reactive Location-Based Services* dan *proactive Location-Based Services*. *Reactive Location-Based Services* adalah layanan yang hanya aktif jika ada aksi yang dilakukan pengguna. Layanan yang hanya akan memberi jika ada permintaan dari pengguna. Sedangkan *proactive Location-Based Services* merupakan layanan yang akan selalu member informasi kepada pengguna walaupun pengguna tidak melakukan permintaan terhadap layanan.

2.2.2 GPS

2.2.2.1 Pengertian GPS

Global Positioning System (GPS) merupakan suatu kumpulan satelit dan sistem kontrol yang memungkinkan sebuah penerima GPS untuk mendapatkan lokasinya di permukaan bumi 24 jam sehari . Setiap satelit ini mengelilingi bumi sekitar 12000 mil atau 19.300 km, mengelilingi bumi 2 kali setiap harinya. Orbit satelit-satelit ini diatur sedemikian rupa sehingga pada setiap saat, suatu tempat di bumi akan dijangkau minimal 4 satelit (Gintoro, et.al 2010).

2.2.2.2 Cara Kerja GPS

Sistem ini menggunakan sejumlah satelit yang berada di orbit bumi, yang memancarkan sinyalnya ke bumi dan ditangkap oleh sebuah alat penerima. Ada tiga bagian penting dari sistim ini, yaitu bagian kontrol, bagian angkasa, dan bagian pengguna.

a. Bagian Kontrol

Seperti namanya, bagian ini untuk mengontrol. Setiap satelit dapat berada sedikit diluar orbit, sehingga bagian ini melacak orbit satelit, lokasi, ketinggian, dan kecepatan. Sinyal-sinyal sari satelit diterima oleh bagian kontrol, dikoreksi, dan dikirimkan kembali ke satelit. Koreksi data lokasi yang tepat dari satelit ini disebut dengan data ephemeris, yang nantinya akan di kirimkan kepada alat navigasi kita.

b. Bagian Angkasa

Bagian ini terdiri dari kumpulan satelit-satelit yang berada di orbit bumi, sekitar 12.000 mil diatas permukaan bumi. Kumpulan satelit-satelit

ini diatur sedemikian rupa sehingga alat navigasi setiap saat dapat menerima paling sedikit sinyal dari empat buah satelit. Sinyal satelit ini dapat melewati awan, kaca, atau plastik, tetapi tidak dapat melewati gedung atau gunung. Satelit mempunyai jam atom, dan juga akan memancarkan informasi 'waktu/jam' ini. Data ini dipancarkan dengan kode '*pseudo-random*'. Masing-masing satelit memiliki kodenya sendiri-sendiri. Nomor kode ini biasanya akan ditampilkan di alat navigasi, maka kita bisa melakukan identifikasi sinyal satelit yang sedang diterima alat tersebut. Data ini berguna bagi alat navigasi untuk mengukur jarak antara alat navigasi dengan satelit, yang akan digunakan untuk mengukur koordinat lokasi. Kekuatan sinyal satelit juga akan membantu alat dalam penghitungan. Kekuatan sinyal ini lebih dipengaruhi oleh lokasi satelit, sebuah alat akan menerima sinyal lebih kuat dari satelit yang berada tepat di atasnya (bayangkan lokasi satelit seperti posisi matahari ketika jam 12 siang) dibandingkan dengan satelit yang berada di garis cakrawala (bayangkan lokasi satelit seperti posisi matahari terbenam/terbit).

Ada dua jenis gelombang yang saat ini dipakai untuk alat navigasi berbasis satelit pada umumnya, yang pertama lebih dikenal dengan sebutan L1 pada 1575.42 MHz. Sinyal L1 ini yang akan diterima oleh alat navigasi. Satelit juga mengeluarkan gelombang L2 pada frekuensi 1227.6 Mhz. Gelombang L2 ini digunakan untuk tujuan militer dan bukan untuk umum.

c. Bagian Pengguna

Bagian ini terdiri dari alat navigasi yang digunakan. Satelit akan memancarkan data almanak dan *ephemeris* yang akan diterima oleh alat navigasi secara teratur. Data almanak berisikan perkiraan lokasi (*approximate location*) satelit yang dipancarkan terus menerus oleh satelit. Data *ephemeris* dipancarkan oleh satelit, dan valid untuk sekitar 4-6 jam. Untuk menunjukkan koordinat sebuah titik (dua dimensi), alat navigasi memerlukan paling sedikit sinyal dari 3 buah satelit. Untuk menunjukkan data ketinggian sebuah titik (tiga dimensi), diperlukan tambahan sinyal dari 1 buah satelit lagi.

Dari sinyal-sinyal yang dipancarkan oleh kumpulan satelit tersebut, alat navigasi akan melakukan perhitungan-perhitungan, dan hasil akhirnya adalah koordinat posisi alat tersebut. Makin banyak jumlah sinyal satelit yang diterima oleh sebuah alat, akan membuat alat tersebut menghitung koordinat posisinya dengan lebih tepat.

Karena alat navigasi ini bergantung penuh pada satelit, maka sinyal satelit menjadi sangat penting. Alat navigasi berbasis satelit ini tidak dapat bekerja maksimal ketika ada gangguan pada sinyal satelit. Ada banyak hal yang dapat mengurangi kekuatan sinyal satelit:

1. Kondisi geografis, seperti yang diterangkan diatas. Selama kita masih dapat melihat langit yang cukup luas, alat ini masih dapat berfungsi.

2. Hutan. Makin lebat hutannya, maka makin berkurang sinyal yang dapat diterima.
3. Air. Jangan berharap dapat menggunakan alat ini ketika menyelam.
4. Kaca film mobil, terutama yang mengandung metal.
5. Alat-alat elektronik yang dapat mengeluarkan gelombang elektromagnetik.
6. Gedung-gedung. Tidak hanya ketika didalam gedung, berada diantara 2 buah gedung tinggi juga akan menyebabkan efek seperti berada di dalam lembah. Sinyal yang memantul, misal bila berada diantara gedung-gedung tinggi, dapat mengacaukan perhitungan alat navigasi sehingga alat navigasi dapat menunjukkan posisi yang salah atau tidak akurat.

2.2.3 Google Maps

Google Maps adalah sebuah jasa peta virtual gratis dan online yang disediakan oleh Google, google maps ini dapat diakses di <http://maps.google.com/>.

Google Maps API merupakan aplikasi *interface* yang dapat diakses lewat *javascript* agar Google Maps dapat ditampilkan pada halaman web yang sedang kita bangun. Untuk dapat mengakses Google Maps, Kita harus melakukan pendaftaran *Api Key* terlebih dahulu dengan data pendaftaran berupa nama domain web yang kita bangun.

2.2.4 Android

2.2.4.1 Sejarah Android

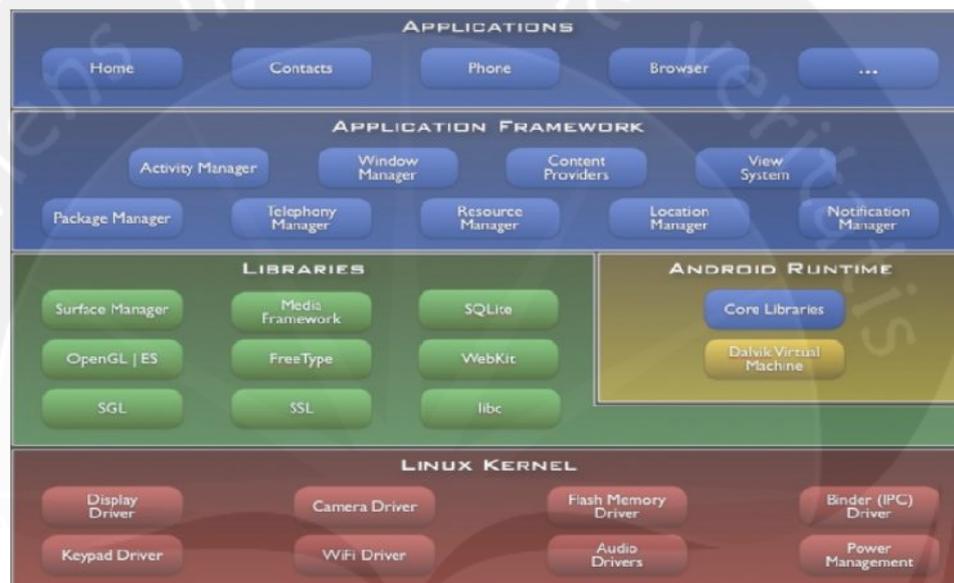
Menurut situs resmi Android (<http://developer.android.com>) dan Lessard et.al(2010) serta Bharati et.al(2010) Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Terdapat beberapa versi pada sistem operasi Android, mulai dari versi 1.5 (CupCake), versi 1.6 (Donut), versi 2.1 (Eclair), versi 2.2 (Froyo), versi 2.3 (GingerBread), versi 3.0 (HoneyComb), versi 4.0 (Ice Cream Sandwich) hingga versi yang terbaru yaitu versi 4.1 (Jelly Bean).

2.2.4.2 Arsitektur Android

Karena teknologi yang digunakan merupakan pengembangan dari sistem operasi yang ada dalam Linux maka sistem operasi Android juga mempunyai arsitektur yang tersusun atas beberapa layer. Gambar berikut ini menjelaskan arsitektur yang digunakan sistem operasi Android.



Gambar 2.1. Arsitektur Sistem Operasi Android (Android, 2012)

Layer-layer yang tersusun dalam arsitektur Android tersebut adalah sebagai berikut:

- a. *Applications Layer:* Android akan menggabungkan dengan serangkaian aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.
- b. *Application Framework Layer:* Dengan menyediakan sebuah platform pengembangan yang terbuka, pengembang Android menawarkan kemampuan untuk membangun aplikasi yang sangat kaya dan inovatif.

Pengembang bebas untuk mengambil keuntungan dari perangkat keras, akses informasi lokasi, menjalankan layanan latar belakang, mengatur alarm, tambahkan pemberitahuan ke status bar, dan masih banyak lagi. Pengembang memiliki akses penuh ke API *framework* sama yang digunakan oleh aplikasi inti. Arsitektur aplikasi ini dirancang untuk menyederhanakan penggunaan kembali komponen; aplikasi apapun dapat mempublikasikan kemampuan dan aplikasi lain makadapat menggunakan kemampuan mereka (terbatasi pada batasan keamanan yang diberikan oleh framework). Mekanisme yang sama memungkinkan komponen yang akan diganti oleh pengguna.

- c. *Libraries Layer*: Android mempunyai satu set pustaka C/C++ yang digunakan oleh berbagai komponen sistem Android. Kemampuan ini ditujukan bagi pengembang aplikasi melalui *framework* aplikasi Android.
- d. *Android Runtime Layer*: Android mempunyai satu set perpustakaan inti yang menyediakan sebagian besar fungsi yang tersedia di perpustakaan inti dari bahasa pemrograman Java. Setiap aplikasi Android berjalan dalam prosesnya sendiri, dengan contoh sendiri dari mesin *Dalvik virtual*. Dalvik telah ditulis sehingga perangkat dapat menjalankan beberapa VMs efisien. VM Dalvik mengeksekusi file dalam Dalvik *executable* (.dex) format yang dioptimalkan untuk jejak memori minimal. VM bersifat terdaftar, dan berjalan kelas dikompilasi oleh *compiler* Java yang telah ditransformasikan ke dalam format “.dex” oleh peralatan "dx" yang termasuk di dalamnya. VM Dalvik bergantung pada kernel Linux untuk

fungsionalitas dasar seperti threading dan manajemen tingkat rendah memori.

- e. *Linux Kernel Layer*: Android bergantung pada Linux versi 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *network stack*, dan model pengemudi. *Kernel* juga bertindak sebagai lapisan abstraksi antara *hardware* dan seluruh *software stack*.

2.2.5 *City Directory*

City Directory adalah daftar orang-orang yang bekerja dan tinggal di kota tertentu. Direktori biasanya diatur menurut abjad nama kemudian alamat jalan. Direktori juga akan berisi nama-nama pejabat kota dan informasi lainnya tentang kota. *City Directory* mempunyai informasi kunci dari tempat tinggal, bisnis dan institusi dari sebuah kota. Isi dari *city directory* bisa bervariasi dari kota ke kota, negara bagian dan penerbit ke penerbit, namun biasanya dapat berisi item seperti, sejarah kota, direktori jalan, peta kota dan sebagainya (Ackman et.al, 2011), (Lin,2009), Shu(2010).