

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Fasilitas Umum**

Menurut Kamus Besar Bahasa Indonesia (KBBI) Fasilitas umum adalah barang yang dikuasai negara, dibiayai sebagian atau seluruhnya oleh anggaran dan belanja negara yang pemakaiannya atau peruntukannya oleh pemerintah atau negara bagi umum.

Adapun pengertian fasilitas umum merupakan fasilitas yang diadakan untuk kepentingan umum. Dikatakan fasilitas umum karena keberadaan wadah atau tempat yang bersifat mempermudah atau memperlancar terpenuhinya kebutuhan-kebutuhan bersama dari kelompok atau komunitas tertentu, misalnya di bidang keamanan, komunikasi, rekreasi, pendidikan, kesehatan, administrasi publik, religius dan sosial-budaya. (Hendar Putranto, 2012).

#### **2.2. *Augmented Reality***

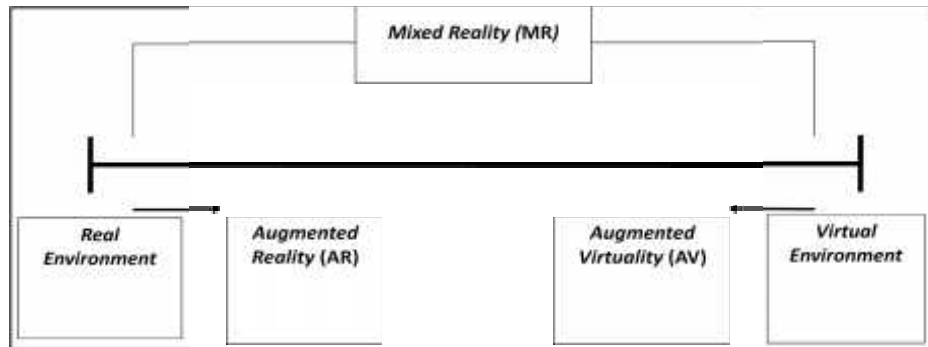
##### **2.2.1. *Pengertian Augmented Reality***

*Augmented Reality* adalah penggabungan objek-objek digital baik itu berupa dua dimensi maupun tiga dimensi dengan dunia nyata yang bersifat interaktif secara real time. Objek-objek digital yang menampilkan informasi tidak dapat diterima pengguna oleh inderanya sendiri. Informasi yang ditampilkan oleh objek-objek digital hanya sebagai alat yang membantu persepsi dan interaksi penggunaannya dengan dunia nyata. Sehingga informasi tersebut membantu penggunaannya melaksanakan kegiatan dalam dunia nyata (Milgram dan Kishino 1994).

##### **2.2.2. *Arsitektur Augmented Reality***

Pada tahun 1994, Milgram dan Kishino merumuskan kerangka kemungkinan penggabungan dan peleburan dunia nyata dan dunia maya ke dalam sebuah *virtuality continuum*. Dalam kerangka tersebut, *Augmented Reality* lebih

dekat ke sisi kiri yang menjelaskan bahwa lingkungan bersifat nyata dan benda bersifat maya. Sebaliknya augmented virtuality lebih dekat ke sisi kanan dalam kerangka tersebut, yang menjelaskan bahwa lingkungan bersifat maya dan benda bersifat nyata. Sehingga jika terjadi penggabungan antara *Augmented Reality* dengan *virtual reality* akan tercipta *mixed reality*.



Gambar 2.1. Konsep kerangka *Augmented Reality* (Milgram dan Kishino, 1994)

### 2.3. *Mobile Augmented Reality*

Rohr dan Gfeller mengenalkan konsep dari *Mobile Augmented Reality*. Dibanding menggunakan perangkat khusus untuk membangun sebuah aplikasi AR, mereka mengusulkan untuk menggunakan generasi baru dari perangkat *portable*. Peningkatan pada kemampuan komputasi, resolusi kamera pada perangkat *portable* memungkinkan pengimplementasian aplikasi AR.

Dalam proses membangun aplikasi AR, Rohr menyederhanakan tugas pengenalan tempat pada suatu gambar dengan menggunakan penanda *fiducial*. Penanda *fiducial* berbentuk persegi 2 dimensional yang terdiri dari area hitam dan putih. Selanjutnya, aplikasi mencari bentuk spesifik pada layar perangkat. Dari penanda *fiducial*, aplikasi dapat menentukan keberadaan pada layar, orientasi dan skala.

*Mobile Augmented Reality* adalah area penting dalam lingkungan penelitian yang mempunyai potensi dan kemungkinan untuk dibangun aplikasi komersial. Menggunakan hardware yang umum, sehingga murah bagi pengguna akhir.

## 2.4. Komponen *Augmented Reality*

### 1. Kamera Telepon

Kamera telepon adalah telepon genggam (ponsel) yang mempunyai kemampuan untuk menangkap foto dan merekam video. Sejak awal abad 21, kamera banyak digunakan atau diimplementasikan pada telepon genggam.

Pada umumnya kamera telepon lebih simple dibandingkan dengan kamera *digital*. Kamera telepon biasanya dilengkapi dengan *fixed focus* (lensa tanpa pengaturan focus) dan sensor kecil yang membatasi kemampuannya dalam pencahayaan yang sedikit. Tidak memiliki flash atau pembesaran optik atau buat dudukan tripod.

Keuntungan dari pengguna kamera telepon adalah biaya dan bentuknya. terutama bagi orang yang sering membawa telepon genggam, perubahan ukuran dan tambahan biaya bukanlah menjadi persoalan besar. *Smartphone* dilengkapi kamera dapat menjalankan aplikasi yang mempunyai kemampuan *geotagging* dan *image stitching* (menggabungkan beberapa foto menjadi satu gambar panorama).

### 2. GPS Navigation Device

Perangkat navigasi GPS adalah perangkat yang dapat menerima sinyal GPS (*Global Positioning System*) yang bertujuan untuk menentukan lokasi perangkat di bumi. Perangkat GPS memberikan informasi garis lintang (*latitude*) dan garis bujur (*longitude*), dan beberapa diantaranya menyertakan kalkulasi *altitude* (ketinggian), walaupun hal ini tidak selamanya akurat atau secara kontinue tersedia (terbatas pada kemungkinan sinyal terhalangi dan faktor lainnya) secara eksklusif diandalkan pada pilot pesawat terbang. Perangkat GPS telah digunakan dalam bidang militer, penerbangan, kelautan, dan produk aplikasi *consumer*.

Berdasarkan aturan terhadap pelacakan perangkat mobile (*mobile phone tracking*), termasuk E911 (*Enhanced 911*, evolusi modern dalam bidang telekomunikasi), mayoritas perangkat penerima GPS ditanam pada perangkat *mobile*, dengan variasi derajat jangkauan dan akses pengguna. Perangkat lunak navigasi komersial telah tersedia pada *smartphone* abad ke-21 serta perangkat telepon berbasis Java yang mengizinkan pelacakan perangkat *mobile* digunakan

sebagai internal atau eksternal penerima GPS (dalam kasus ini, terhubung melalui serial atau *Bluetooth*). Beberapa telepon dengan kemampuan GPS bekerja hanya menggunakan A-GPS (*assisted GPS*), dan tidak berfungsi ketika perangkat berada diluar jangkauan tower penyedia jaringan.

### 3. Accelerometer

Accelerometer adalah sebuah *transducer* yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan gravitasi bumi. Accelerometer juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, dan juga dapat digunakan untuk mengukur getaran yang terjadi dalam bumi, getaran mesin, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi.

### 4. Digital Compass

Kompas adalah alat navigasi untuk menentukan arah berupa sebuah panah penunjuk magnetis yang bebas menyelaraskan dirinya dengan medan magnet bumi secara akurat. Kompas memberikan rujukan arah tertentu, sehingga sangat membantu dalam bidang navigasi. Arah mata yang ditunjukkannya adalah utara, selatan, timur, dan barat. Apabila digunakan bersama-sama dengan jam dan sekstan, maka kompas akan lebih akurat dalam menunjukkan arah. Alat ini membantu perkembangan perdagangan maritim dengan membuat perjalanan jauh lebih aman dan efisien dibandingkan saat manusia masih berpedoman pada kedudukan bintang untuk menentukan arah.

Kompas dibedakan menjadi dua jenis, yaitu kompas analog dan kompas digital. Kompas analog adalah kompas yang biasa kita lihat dalam kehidupan sehari-hari. Misalnya saja kompas yang dipakai ketika acara pramuka. Sedangkan kompas digital merupakan kompas yang telah menggunakan proses digitalisasi. Dengan kata lain, cara kerja kompas ini menggunakan komputerisasi.

## 2.5. Geotagging

*Geotagging* adalah proses penambahan identifikasi metadata geografis terhadap berbagai media contohnya *geotagged fotografi*, atau video, *websites*, pesan SMS, *QR code*, atau *RSS feed* yang merupakan bentuk dari metada *geospatial*. Data tersebut biasanya terdiri dari koordinat garis lintang (*latitude*) dan garis bujur (*longitude*), dan mungkin juga termasuk ketinggian (*altitude*), bantalan poros (*bearing*), jarak, data akurasi, dan nama tempat.

*Geotagging* dapat membantu pengguna untuk menemukan informasi posisi keberadaan spesifik dari lingkungan yang luas. Contohnya, seseorang dapat menemukan gambar terdekat dengan memberikan informasi koordinat *latitude* dan *longitude* ke dalam mesin pencari gambar yang sesuai. *Geotagging* dapat memberikan informasi layanan yang potensial digunakan untuk menemukan lokasi berdasarkan berita, website, dan sumber lainnya. *Geotagging* dapat memberitahukan kepada pengguna lokasi dari konten dari gambar atau media lainnya atau titik pandangan (*point of view*), dan sebaliknya pada beberapa *platform* media menunjukkan media yang relevan ke lokasi tertentu.

Terkait dengan istilah *geocoding* yang mengacu pada proses menentukan geografis non kordinat berdasarkan pengidentifikasi geografi, seperti alamat jalan, dan penemuan yang diasosiasikan dengan koordinat geografi. banyak teknik yang dapat digunakan bersama dengan *geotagging* untuk menyediakan teknik pencarian alternatif.

## 2.6. Metode Pelacakan *Augmented Reality*

1. *Elektromagnetic tracking system*: mengukur medan magnet yang dihasilkan melalui arus listrik yang secara simultan melewati tiga kumparan kabel yang disusun secara tegak lurus satu dengan yang lain. Setiap kumparan kecil bersifat elektromagnet. Sensor sistem mengkalkulasikan bagaimana medan magnet terbentuk dan pengaruhnya terhadap kumparan lainnya. Pengukuran tersebut menunjukkan posisi atau orientasi dan arah dari *emitter*. Reponsibilitas dari efisiensi sistem pelacakan elektromagnet sangat baik dan tingkat latensinya cukup rendah. Satu kekurangan dari sistem ini adalah apapun yang dapat

menghasilkan medan magnet dapat mempengaruhi sinyal yang dikirim ke sensor.

2. *Acoustic tracking system*: sistem pelacakan ini menangkap dan menghasilkan gelombang suara *ultrasonic* untuk mengidentifikasi orientasi dan posisi dari target. Sistem ini mengkalkulasi waktu yang digunakan suara *ultrasonic* untuk mencapai sensor. Sensor biasanya selalu menjaga kestabilan dalam lingkungan dimana pengguna menempatkan *emitter*. Bagaimanapun, kalkulasi dari orientasi serta posisi target bergantung pada waktu yang digunakan oleh suara untuk mencapai sensor adalah dilakukan oleh sistem. Terdapat banyak kekurangan pada sistem pelacakan acoustic. Suara yang lewat sangat lambat, sehingga tingkat update posisi target juga menjadi lambat. Efisiensi sistem dapat menjadi tidak efektif dengan kecepatan suara melewati udara karena sering kali berubah, bergantung pada kelembaban, temperatur atau tekanan barometer dalam lingkungan.
3. *Optical tracking system*: perangkat ini menggunakan cahaya untuk menghitung orientasi dan posisi target. Sinyal emitter dalam perangkat optical secara khusus terdiri atas sekumpulan LED inframerah. Sensor kamera dapat menangkap cahaya inframerah yang dipancarkan. LED menyala dalam pulse secara sekuensial. Kamera merekam sinyal pulse dan mengirim informasi kepada unit pemrosesan sistem. Unit tersebut kemudian dapat menghitung kemungkinan data untuk menentukan posisi dan orientasi target. Sistem optical mempunyai tingkat upload data yang cepat, sehingga latensi dapat diminimalisir. Kekurangan sistem ini adalah penglihatan antara kamera dan LED dapat menjadi gelap, bertentangan dengan proses pelacakan. Radiasi inframerah juga dapat membuat sistem kurang efektif.
4. *Mechanical tracking system*: sistem pelacakan ini bergantung pada physical link antara target dan referensi titik tetap. Salah satu contohnya adalah sistem pelacakan mekanikal dalam lingkungan virtual reality (VR), yaitu BOOM display. BOOM display, sebuah head-mounted display (HMD), dipasang di bagian belakang dengan yang terdiri atas 2 poin artikulasi. Deteksi orientasi dan posisi dari sistem dilakukan melalui lengan. Tingkat update cukup tinggi

dengan sistem pelacakan mekanikal, tetapi sistem ini memiliki kekurangan yaitu membatasi pergerakan dari pengguna (user).

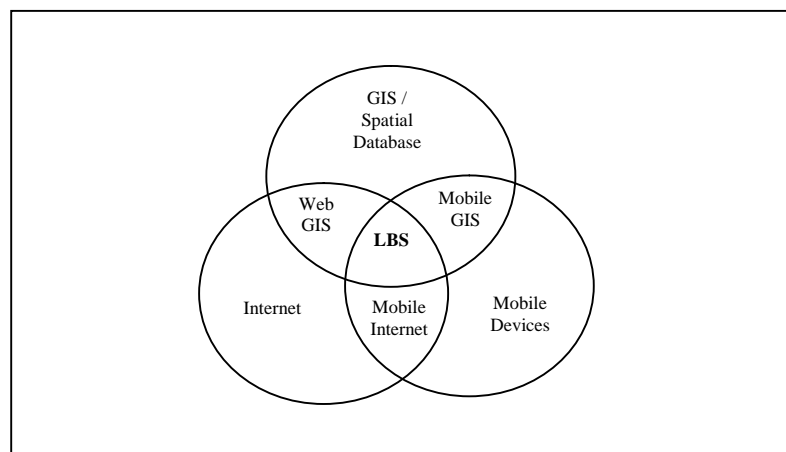
5. *Inertial navigation system*: navigasi bantuan yang menggunakan komputer, sensor gerak (accelerometer), sensor rotasi (gyroscopes) secara continue dikalkulasi melalui posisi dead reckoning (proses pengukuran posisi sekarang seseorang, dengan menggunakan posisi yang telah ditentukan sebelumnya, atau memperbaikinya, dan tingkatan posisi berdasarkan kecepatan rata-rata dari waktu-waktu), orientasi, dan kecepatan perpindahan objek tanpa membutuhkan referensi luar. Sistem ini digunakan dalam bidang transportasi seperti, kapal, pesawat, kapal selam, dan pesawat ruang angkasa.
6. *GPS Tracking*: GPS tracking memanfaatkan kombinasi teknologi GSM dan GPS untuk menentukan koordinat sebuah objek, lalu menerjemahkan dalam bentuk peta digital. Pengembangan teknik ini lebih diarahkan pada smartphone, karena teknologi GPS dan kompas yang tertanam pada smartphone tersebut. Dengan memanfaatkan fitur GPS yang berfungsi sebagai penentu lokasi pengguna pada saat itu berada sehingga lokasi terdekat yang ingin dituju dapat dilihat melalui implementasi *Augmented Reality*. Teknik ini berguna sebagai pemandu selayaknya fungsi GPS, namun dilengkapi dengan marker informasi arah yang dituju. Dalam implementasinya, teknik ini mengharuskan tersambungannya koneksi GPS dan kebutuhan paket data yang ada pada smartphone, karena data-data lokasi yang dimiliki GPS memiliki akses langsung dari satelit agar cepat mendeteksi wilayah yang telah dijadikan sebuah objek marker informasi pada *Augmented Reality*. Akses internet memiliki fungsi sebagai pemanggilan data-data berupa latitude, longitude, serta informasi yang mendukung setiap lokasi yang disimpan pada server sehingga beban ukuran aplikasi dapat diminimalisir. Teknik GPS tracking sebenarnya membutuhkan peran kompas dan akselerometer sebagai pengatur ukuran layar secara horizontal dan vertical agar marker lokasi dapat dilihat ketika kamera handset berada posisi yang sesuai dengan lokasi tersebut. Namun ketika *handset* tidak berada dalam sudut pandang lokasi tersebut maka marker tersebut tidak akan tampak.

7. *Hybrid Tracking*: sistem pelacakan yang merupakan gabungan dari dua atau lebih teknik pelacakan, hybrid tracking digunakan untuk menciptakan sistem pelacakan yang lebih baik. Teknik ini secara sinergis dapat meningkatkan kesegaran (*robustness*), kecepatan pelacakan (*tracking speed*) dan akurasi, dan mengurangi jitter dan noise. Hybrid tracking telah banyak digunakan dengan gabungan beberapa teknik pelacakan (misalnya, GPS *electronic compass* dan *sensor inertial* dan *sensor optical* [Azuma et al., 1999]. *Magnetic dan vision-based* [Auer dan Pinz, 1999]).

## 2.7. Location Based Service

Ada dua definisi yang bisa menjelaskan tentang *Location Based Service* (LBS). Definisi Pertama: LBS adalah layanan informasi yang dapat diakses menggunakan piranti *mobile* melalui jaringan Internet dan seluler serta memanfaatkan kemampuan penunjuk lokasi pada piranti *mobile* (Virrantasu, et al, 2001). Definisi Kedua: Layanan IP nirkabel yang menggunakan informasi geografis untuk memberikan layanan informasi lokasi kepada pengguna. Beberapa layanan aplikasi yang memberikan petunjuk posisi/lokasi piranti *mobile* berada.

### 2.7.1 Hubungan GIS dan LBS



Gambar 2.2. *LBS as an intersection of technologies* (Brimicombe, 2002).



Gambar 2.2 menunjukkan bahwa GIS dan LBS memiliki beberapa kesamaan. Tetapi LBS dan GIS memiliki asal yang berbeda dan kelompok pengguna yang berbeda juga seperti yang dijelaskan oleh Virrantaus et al. (2001). Mereka menjelaskan bahwa Sistem Informasi Geografis telah dikembangkan selama beberapa dekade atas dasar profesionalitas aplikasi data geografis. Sedangkan LBS baru lahir belakangan ini sebagai sebuah evolusi layanan *mobile* publik. GIS diperuntukkan untuk pengguna yang memerlukan aplikasi dengan spesialisasi tugas dan fungsi tertentu. Selanjutnya GIS memerlukan sumber daya komputasi yang besar. Sedangkan, LBS dikembangkan sebagai layanan terbatas untuk pengguna di kalangan non-profesional. Aplikasi LBS dapat beroperasi dengan pembatasan lingkungan komputasi *mobile* seperti daya komputasi yang rendah.

### **2.7.2 Komponen LBS**

Terdapat lima komponen pendukung utama dalam teknologi Layanan Berbasis Lokasi, antara lain (Steiniger, et al. 2006):

#### **1. Perangkat *Mobile***

Perangkat *Mobile* adalah komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu (*tool*) bagi pengguna untuk meminta informasi. Informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Perangkat *mobile* yang dapat digunakan bisa berupa PDA, *smartphone*, *laptop*. Selain itu, perangkat *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.

#### **2. Jaringan Komunikasi**

Komponen kedua adalah jaringan komunikasi. Komponen ini memiliki fungsi sebagai penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari perangkat *mobile*-nya untuk kemudian dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

3. Komponen *Positioning* (Penunjuk Posisi/Lokasi)

Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi *mobile* atau juga menggunakan *Global Positioning System* (GPS).

4. Penyedia layanan dan aplikasi

Penyedia layanan merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data di Yellow Pages sesuai dengan permintaan, dan masih banyak lagi yang lainnya.

5. Penyedia data dan konten

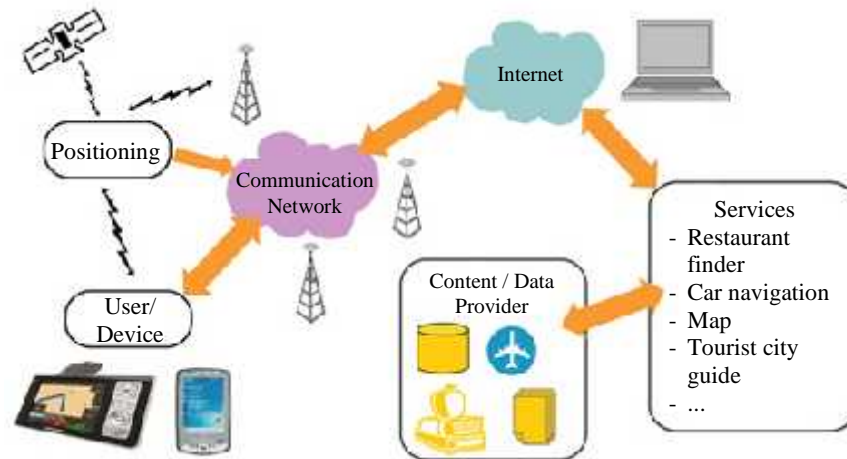
Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan-badan milik pemerintah atau juga data-data perusahaan/bisnis/industri bisa saja berasal dari Yellow Pages, maupun perusahaan penyedia data lainnya.

Secara lengkap kelima komponen pendukung LBS tersebut dapat dilihat pada gambar berikut.



Gambar 2.3. Komponen *LBS* (Steiniger, et al. 2006).

### 2.7.3 Cara Kerja LBS



Gambar 2.4. Cara Kerja *LBS* (Steiniger, et al. 2006).

Pada gambar diatas di jelaskan tentang bagaimana cara kerja dari LBS

1. Diasumsikan pengguna *mobile* sedang membuka aplikasi berbasis LBS.
2. *User* mendapatkan posisi dari satellite melalui *GPS reciever* yang dimilikinya.
3. Data Lokasi ini kemudian dikirim menggunakan jaringan komunikasi seperti GPRS, EDGE, dan 3G ke internet untuk menuju *Content Provider Service*.
4. Dari *Content Provider* akan dikirim kembali data mengenai layanan-layanan yang diminta oleh *user* seperti pencarian rumah makan, navigasi kendaraan, peta dan panduan perjalanan.

### 2.7.4 Google Maps

Google Maps merupakan layanan peta *virtual* dan *online* yang disediakan oleh Google. Layanan ini memiliki kemampuan untuk melakukan pencarian, perencanaan rute dan penambahan informasi pada peta yang disediakan. Layanan

Google Maps juga dilengkapi dengan layanan informasi bisnis, jasa, layanan umum, lokasi dan sebagainya.

Google Maps terbagi menjadi dua jenis akses peta meliputi Google Maps API dan Google Static Maps API. Google Maps API merupakan standar pemetaan yang digunakan di *desktop* dan memiliki sifat peta yang dinamis (mudah dimodifikasi), yang biasanya dimanfaatkan pada pembuatan modifikasi peta di sisi *server*. Sementara Google Static Maps API merupakan jenis pemetaan yang bersifat *static*, dimana jenis peta ini dapat dimanfaatkan untuk pengaksesan peta pada *mobile phone* (di sisi *client*).

## **2.8. Android**

Android merupakan sebuah sistem sistem operasi untuk perangkat *mobile* berbasis *linux* yang meliputi sistem operasi, *middleware*, dan aplikasi. Sedangkan Android SDK (*Software Development kit*) adalah *tools API (Application Programming Interface)* yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java (Safaat, 2011).

Android dikembangkan oleh Google dan bersamaan dengan itu dibentuklah OHA (*Open Handset Alliance*) yaitu sebuah konsorsium dari 34 perusahaan *Hardware, Software* dan perusahaan telekomunikasi yang ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

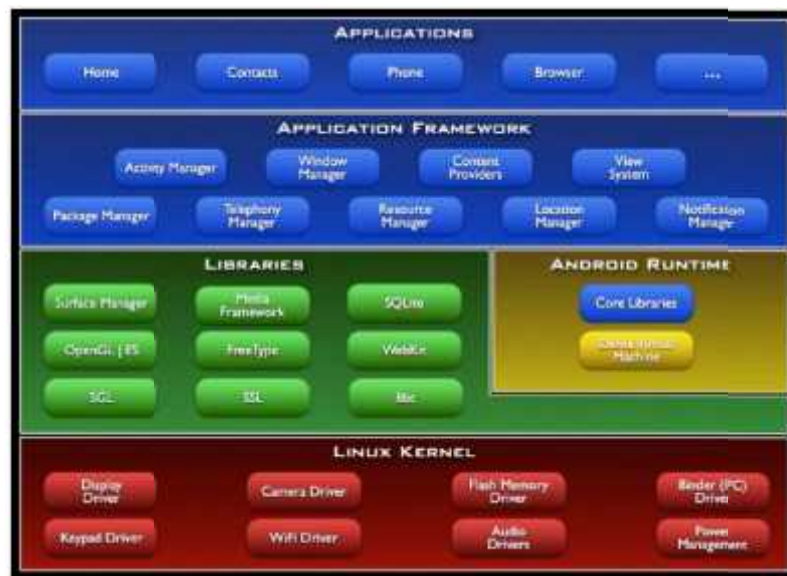
### **2.8.1. Sejarah Android**

Pada tahun 2005 Google mengakuisisi *Android Inc* yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Kemudian pada tahun itu juga memulai membangun platform *Android* secara intensif. Kemudian pada tanggal 5 November 2007 *Android* pertama kali diluncurkan, dan *smartphone* pertama yang menggunakan sistem operasi *Android* dikeluarkan oleh T-Mobile dengan sebutan G1 pada bulan September 2008. Hingga saat ini *Android* telah merilis beberapa versi *Android* untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi *Android* terdapat kode nama berdasarkan nama-nama kue. Hingga saat ini sudah terdapat beberapa

versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama Cupcake, versi 1.6 dirilis pada 15 September 2009 diberi nama Donut, versi 2.0/2.1 dirilis pada 26 Oktober 2009 diberi nama Éclair, versi 2.2 dirilis pada bulan Mei 2010 diberi nama Froyo dan versi 2.3 dirilis pada Desember 2010 yang diberi nama Gingerbread.

### 2.8.2. Arsitektur *Android*

Dalam paket sistem operasi *Android* terdiri dari beberapa unsur seperti tampak pada gambar 2.5. Secara sederhana arsitektur *Android* merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.5. Arsitektur *Android* (sumber: <http://developer.android.com>)

#### 2.8.2.1. *Linux Kernel*

*Android* bukan *linux*, akan tetapi *android* dibangun diatas *linux kernel* versi 2.6 yang kehandalannya sudah teruji. Untuk inti sistem servis *linux* yang digunakan seperti keamanan, manajemen momori, proses manajemen, *network* dan *driver* model. Seperti yang terlihat di gambar 2.5, linux kernel menyediakan *Driver* layar, *driver* kamera, flash memori, IPC (*Interprocces Communication*)

untuk mengatur aplikasi dan keamanan, *driver* keypad, *driver* WiFi, *driver* audio, dan power management. *Kernel* juga bertindak sebagai lapisan abstrak antara *hardware* dan *software* stacknya (<http://deveoper.android.com>).

### **2.8.2.2.Libraries**

*Android* menyertakan linbraries C / C++ yang digunakan oleh berbagai komponen dari sistem *Android*. Kemampuan ini disediakan kepada *developer* aplikasi melalui *framework* aplikasi *Android*. Beberapa inti libraries tercantun dibawah ini (Mulyadi, 2010):

1. *System C library*

Variasi dari implementasi BSD (*Barkeley Software Distribution*) berasal dari sistem standar *C library (libc)*, sesuai untuk perangkat *embedded* berbasis linux.

2. *Media libraries*

Untuk merekam dan memutar berbagai format *Audio* dan *Video*.

3. *Surface Manager*

Mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi.

4. *LibWebCore*

Librari untuk mesin web pada browser *Android*.

5. *SGL*

Merupakan library untuk proses mesin grafis 2D.

6. *3D libraries*

Digunakan untuk proses gambar yang membutuhkan daya 3D.

7. *FreeType*

Merupakan library untuk bitmap dan vektor font rendering

8. *SQLite*

Merupakan library mesin database yang kuat dan ringan, dan pehubung aplikasi yang tersedia.

### **2.8.2.3. *Android Runtime***

Pada *Android* tertanam paket pustaka inti yang menyediakan sebagian besar fungsi *Android*. Inilah yang membedakan *Android* dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi *Android* menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam *Android Runtime* terdapat 2 bagian utama, diantaranya (<http://deveoper.android.com>):

1. Pustaka Inti, *Android* dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin virtual Java. Pustaka inti *Android* menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus *Android*.
2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (\*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan diatas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

### **2.8.2.4. *Applications Frameworks***

*Android* merupakan “*Open Development Platform*” dimana *Android* menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan service background, mengatur alarm, menambah status pemberitahuan dan lainnya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti.

*Applications Frameworks* merupakan layer dimana para pembuat aplikasi melakukan pengembangan aplikasi yang akan dijalankan di sistem operasi



*android*. Komponen-komponen yang termasuk dalam *Applications Frameworks* adalah sebagai berikut (<http://deveoper.android.com>):

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

#### **2.8.2.5. Applications dan Widgets**

*Applications* dan *Widgets* ini adalah layer yang berhubungan dengan aplikasi saja. Biasanya aplikasi di download kemudian dilakukan instalasi dan jalankan aplikasi tersebut. Di layer ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain.

#### **2.8.3. Android SDK (Software Development Kit)**

*Android SDK* adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform Android* menggunakan bahasa pemrograman Java. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh Google. Untuk sumber SDK *Android* dapat dilihat dan diunduh langsung kesitus resmi *Android* di <http://deveoper.android.com>.

#### **2.8.4. Komponen Aplikasi**

Aplikasi *Android* ditulis dalam bahasa pemograman Java. Kode Java dikompilasi bersama dengan data file resource yang dibutuhkan oleh aplikasi, prosesnya di paket oleh *tools* yang dinamakan *atp tools* kedalam paket *Android*, sehingga menghasilkan file dengan ekstensi *apk*. Ada 4 jenis komponen pada aplikasi *Android*, yaitu (<http://deveoper.android.com>):

1. *Activities*

Suatu *Activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Pada umumnya sebuah

aplikasi *Android* memiliki banyak *activity*, tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain, kita dapat melakukan dengan satu even, misalnya klik tombol, memilih opsi atau menggunakan triggers tertentu. Secara hirarki sebuah windows *activity* dinyatakan dengan *method* `Activity setContentView()`. `ContentView` adalah objek yang berada pada *root hirarki*.

## 2. *Service*

*Service* tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara background. Komponen *service* diproses tidak terlihat, memperbaharui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang terus diproses, bahkan ketika *activity* tidak aktif

## 3. *Broadcast Receiver*

*Broadcast receiver* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. *Broadcast receiver* tidak memiliki *user interface*, tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima atau menggunakan *notification manager* untuk memberi tahu pengguna, seperti lampu latar atau getaran.

## 4. *Content Provider*

*Content provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file seperti database `SQLite`. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi dari *content provider*.

## 2.9. Web Services

*World Wide Consortium* (W3C), organisasi yang mengembangkan standar-standar dalam dunia *web*, mendefinisikan *Web Service* sebagai perangkat lunak

sistem yang dirancang untuk mendukung interaksi antara mesin dengan mesin melalui jaringan.

*Web Service* tidak dibuat untuk berinteraksi langsung dengan *user*. *Web Service* hanya menyediakan *service* atau layanan. Layanan itulah yang kemudian akan digunakan atau dipanggil oleh aplikasi lainnya sehingga dengan demikian, yang akan menjadi *interface* adalah aplikasi yang memanggilmnya bukan *Web Service* tersebut (Lucky, 2008).

*Web Service* merupakan kumpulan dari berbagai fungsi atau *method* yang terdapat pada sebuah *server* yang dipanggil oleh *client* dari jarak jauh. Penggunaan *Web Service* memungkinkan perangkat yang menggunakan sistem operasi dan aplikasi yang berbeda satu sama lain dapat saling bertukar data dan informasi dengan mudah.

Sebuah *Web Service* dapat dipanggil oleh aplikasi lain dengan menggunakan bantuan HTTP (*Hypertext Transfer Protocol*). *Web Service* juga memungkinkan untuk dipanggil dengan menggunakan protokol lain seperti SMTP (*Simple Mail Transfer Protocol*), namun yang umum digunakan HTTP.

Komponen Pendukung *Web Service* yaitu :

1. *Extensible Markup Language (XML)*

XML merupakan dasar yang penting atas terbentuknya *Web Service*. *Web Service* dapat berkomunikasi dengan aplikasi-aplikasi yang memanggilmnya dengan menggunakan XML. Karena XML berbentuk teks sehingga mudah untuk ditransportasikan menggunakan protokol HTTP. XML bersifat *platform* independen sehingga informasi di dalamnya bisa dibaca oleh aplikasi apapun pada *platform* apapun selama aplikasi tersebut mampu menerjemahkan *tag-tag* XML.

2. *Simple Object Access Protocol (SOAP)*

SOAP merupakan suatu format standar dokumen berbentuk XML yang digunakan untuk melakukan proses *request* dan *response* antara *Web Service* dengan aplikasi yang memanggilmnya. Diibaratkan XML adalah

bahasa yang digunakan oleh *Web Service* dan aplikasi, maka SOAP adalah tata bahasa yang digunakan sehingga keduanya bisa saling memahami saat sedang berbicara.

3. *Web Service Description Language (WSDL)*

WSDL adalah sebuah dokumen dalam format XML yang isinya menjelaskan informasi *detail* sebuah *Web Service*. Di dalam WSDL dijelaskan *method-method* apa saja yang tersedia dalam *Web Service*, parameter apa saja yang diperlukan untuk memanggil sebuah *method*, dan apa hasil atau tipe data yang dikembalikan oleh *method* yang dipanggil tersebut.

4. *Universal Description, Discovery and Integration (UDDI)*

UDDI merupakan suatu *directory service* yang digunakan untuk meregristasikan dan mencari *Web Service*.

## **2.10. Analisa dan Perancangan Berorientasi Objek**

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah untuk mempermudah programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata (A.Suhendar, 2002).

Perusahaan software, Rational Software, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian Unified Modelling Language (UML) sebagai bahasa standar dalam Object Oriented Analysis Design (OOAD).

### **2.10.1 Unified Modelling Language (UML)**

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan

mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti, 2006). Untuk merancang sebuah model, UML memiliki beberapa diagram antara lain : use case diagram, class diagram, statechart diagram, activity diagram, sequence diagram, collaboration diagram, component diagram, deployment diagram.

#### **2.10.1.1 Use Case Diagram**

Use case diagram merupakan sebuah gambaran fungsionalitas sebuah sistem. Sebuah use case merepresentasikan interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Dharwiyanti, 2006).

Dalam sebuah sistem use case diagram akan sangat membantu dalam hal menyusun requirement, mengkomunikasikan rancangan dengan klien dan merancang test case untuk semua fitur yang ada pada sistem.

#### **2.10.1.2 Class Diagram**

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) (Dharwiyanti, 2006).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok yaitu, nama, stereotype, atribut dan metoda

#### **2.10.1.3 Sequence Diagram**

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang

digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan (Dharwiyanti, 2006).

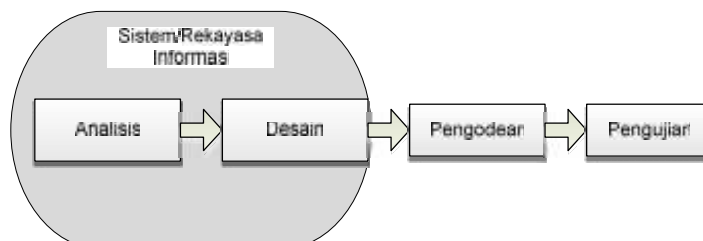
#### 2.10.1.4 Activity Diagram

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

### 2.11. Waterfall

Model air terjun (*Waterfall*) merupakan salah satu model *Software Development Life Cycle (SDLC)*. *Waterfall* sering disebut juga model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). *Waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial, atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). (Rosa & Shalahuddin, 2011)

Berikut ini merupakan gambar model *waterfall* (Rosa & Shalahuddin, 2011)



Gambar 2.6. Ilustrasi Model *Waterfall*

a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti yang dibutuhkan oleh user.

b. Desain

Desain perangkat lunak adalah proses multilangkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Pada tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program computer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian focus pada perangkat lunak secara secara dari segi loojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak.