

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Studi

Banyak penelitian yang sebelumnya dilakukan mengenai implementasi Teknologi *Location Based Service* untuk pengembangan perangkat lunak pada *platform android*. Dalam upaya pengembangan perangkat lunak berbasis android ini dengan memanfaatkan teknologi *Location Based Service* untuk pengembangan perangkat lunak pelacakan lokasi Sales PT Meliana Perkasa Sejahtera perlu dilakukan studi pustaka sebagai salah satu dari penerapan metode penelitian yang akan dilakukan. Diantaranya adalah menghindari pembuatan ulang, mengidentifikasi metode yang pernah dilakukan, mengidentifikasi kesenjangan, meneruskan penelitian sebelumnya, serta mengetahui orang lain yang spesialisasi dan area penelitiannya sama dibidang ini. Beberapa Penelitian sebelumnya yang terkait / *Literatur review* tersebut adalah sebagai berikut :

1. Penelitian ini dilakukan oleh Pankti Doshi , Pooja Jain dan Abhishek Shakwala dari NMIMS *University, Mukesh Patel School of Technology Management and Engineering , Mumbai* pada tahun 2014 yang berjudul “ *Location Based Service and Integration of Google Maps in Android* ”. Penelitian ini membahas mengenai latar belakang kenapa manusia sekarang ini lebih suka menggunakan perangkat bergerak bukan lagi perangkat *dekstop* , karena dengan menggunakan perangkat *mobile* seseorang lebih banyak mendapatkan keuntungan, salah satunya adalah teknologi *geographic location*, dengan teknologi *Location based service* yang terintegrasi *user* ( pengguna ) pengguna perangkat bergerak ini dapat mendapatkan banyak informasi yang di inginkan. Paper ini memfokuskan detail dari *Location Based Service* dan mengidentifikasi komponen kunci dari *Location Based Service* untuk menyediakan pelayanan kepada pengguna perangkat *mobile* berbasis android. Paper ini juga menjelaskan bagaimana menggunakan dan

mengimplementasikan *Google maps APIs* untuk menampilkan informasi basis lokasi pada aplikasi platform android. Dengan merujuk pada penelitian ini, penulis dapat mengambil manfaat untuk melakukan penelitian tentang Penerapan *Location Based Service* untuk pencarian Lokasi dari seseorang.

2. Penelitian ini dilakukan oleh Shaveta Bhatia dan Saba Hilal dari *Research Scholar* , *Manav Rachna International University, Faridabad, Haryana, India* pada tahun 2013 yang berjudul “ *A New Approach for Location Based Tracking* ”. Penelitian ini menjelaskan tentang cara menemukan suatu lokasi dengan akurasi yang tepat dari *user tracking*. Dalam paper ini juga dijelaskan teknik-teknik untuk mendapatkan Tracking lokasi yang di inginkan User , yaitu dengan menggunakan jaringan GSM ( *Global System for Mobile Communication* ), GPRS ( *General Packet Radio Service* ), CDMA ( *Code Divition Multiple Access* ) dan juga menggunakan GPS ( *Global Positioning System*). Paper ini juga menjelaskan perbandingan antara teknik-teknik dan pengaplikasian ke *Location Tracking*. Paper ini dapat dijadikan rujukan Penulis untuk melakukan penelitian tentang Penerapan *Location Based Service* dengan teknik pencarian menggunakan GPS untuk mengidentifikasi lokasi dari seseorang melalui sinyal GPS yang ada di perangkat Android yang di gunakannya.
3. Penelitian selanjutnya dilakukan oleh Badrul Anwar, Hendra Jaya dan Putra Indra Kusuma dari Teknik Komputer STMIK Triguna Dharma, Program Studi Sistem Komputer Jl. A.H. Nasution No. 73 F-Medan pada tahun 2014 yang berjudul “ *Implementasi Location Based Service Berbasis android Untuk Mengetahui posisi User* ”. Penelitian ini membahas tentang pencarian lokasi user dimana pencarian lokasi user menggunakan *Location Based Service*. Melalui LBS akan disematkan *mobile positioning* yang akan membantu mengetahui posisi user dari smartphone platform Android menggunakan GPS ( *Global Positioning System*). Dalam paper ini dijelaskan apa itu LBS, cara kerja LBS, prinsip dan Komponen LBS serta Analisa perancangan menggunakan LBS dalam implementasi pencarian lokasi *user* pada smartphone Android.

4. Penelitian ini dilakukan oleh Wahyu Kusuma dan Tity Septiani dari Teknik Informatika Gunadarma Jakarta pada tahun 2013 yang berjudul “ Aplikasi *Friend Tracker* Berbasis Android Smartphone Menggunakan GPS Tracking ”. Penelitian ini bermaksud untuk merancang dan membangun sebuah Aplikasi *Friend Finder* yang diharapkan dapat membantu pengguna Perangkat mobile Android untuk dapat terhubung satu sama lain untuk mengetahui posisi masing-masing dan mengetahui *best rute* antara penggunaanya. Penelitian ini memanfaatkan GPS, memanfaatkan sinyal dari *network provider* yang digunakan pada Smartphone Android untuk mengetahui lokasi user dan juga user lain yang terhubung melalui aplikasi ini. Paper ini menjelaskan cara kerja Aplikasi ini didukung oleh adanya *online server* yang bertindak sebagai tempat penyimpanan posisi user yang nantinya akan dikirim ke user lain melalui koneksi Internet. Penelitian ini dapat dijadikan sebagai referensi untuk membuat perangkat lunak *GPS Based Location Tracker* pada platform Android untuk pencarian lokasi SPG PT. Meliana Perkasa Sejahtera.
5. Penelitian berikutnya dilakukan oleh Manav Singhal dan Anupam Shukla dari ABV-Indian Institute of Information Technology and Management Gwalior, India pada tahun 2012 yang berjudul “Implementation of Location based Services in Android using GPS and Web Services”. Penelitian ini menjelaskan bahwa *location based services* dapat di implementasikan kedalam perangkat android untuk dimanfaatkan sebagai pelacakan suatu lokasi seperti : mengetahui kondisi traffic, sebagai pemandu rute saat dijalan, membantu menemukan hotel (tempat) terdekat dan pelacakan keberadaan teman kita. Pada penelitian ini diterangkan bahwa pemanfaatan *location based service* dapat di klasifikasikan menjadi 3 kategori, yaitu sebagai public safety/emergency service, customer services (navigasi, pemasaran dan penanda suatu lokasi), dan pencarian lokasi seperti pencarian Bank terdekat. Terdapat 2 metodologi untuk mengimplementasikan *location based services* yaitu dengan network provider dari ponsel android yang digunakan dan menggunakan sinyal GPS (satellites) untuk menemukan sebuah lokasi.

## 2.2 Tinjauan Pustaka

### 2.2.1 *Location Based Service*

*Location based service* merupakan serangkaian layanan yang berbasis lokasi geografis perangkat mobile yang memungkinkan pengguna untuk mencari dan menemukan orang lain, kendaraan, dan juga untuk melacak lokasi penggunanya sendiri [4]. Saat ini *Location based service* menjadi sangat populer. Teknologi ini menawarkan aplikasi yang inovatif salah satu contohnya adalah untuk membantu orang tua yang ingin menemukan dimana keberadaan anaknya dengan cara melacak perangkat pintar (*mobile*) yang digunakan anaknya [4]. Ada tiga Teknik *Location based service* dalam menentukan penentuan/pelacakan lokasi perangkat mobile yaitu [4]:

#### 1. *Cell Identification*

Teknik ini adalah teknik yang sederhana, mudah dan membutuhkan biaya pengeluaran yang efektif (*low cash*) untuk memberikan informasi lokasi berdasarkan sel telepon. Teknik ini dapat diterapkan pada jaringan GSM dan Jaringan selular WCDMA. Cara kerjanya perangkat yang menggunakan Location Based Station yang akan mengirimkan lokasi kepada pengguna perangkat mobile tersebut dengan membaca lokasi dari cell tadi (GSM dan jaringan WCDMA). Ketetapan tingkat keakuratan identifikasi menggunakan teknik ini yaitu antara 200 meter sampai 20 km. Keakuratan lokasi akan dipengaruhi oleh sinyal yang di tangkal oleh perangkat mobile yang digunakan, semakin banyak sinyal yang di tangkap maka keakuratan lokasi akan semakin akurat [4].

#### 2. *GPS (Global Positioning System)*

*GPS (Global Positioning System)* adalah metode penentuan posisi suatu objek di bumi, dalam semua kondisi cuaca [4]. GPS menggunakan sejumlah satelit di orbit bumi untuk melakukan pelacakan posisi [4]. Teknologi GPS ini akan meng-update secara cepat perubahan posisi dari seorang penggunanya [3]. Tingkat akurasi

GPS sangatlah tinggi dibandingkan dengan teknik *cell identification* tingkat akurasi teknik GPS lebih tinggi. Kelemahan teknik ini adalah waktu respon yang lambat pada saat penentuan/pelacakan lokasi user. Cara kerja teknik ini untuk mengetahui posisi adalah dengan membaca satelit-satelit yang mengorbit ke bumi [4].

### 3. *Assisted GPS (Global Positioning System)*

Teknik ini adalah penyempurnaan dari teknik GPS. Teknik ini dapat mendeteksi perangkat pengguna yang ada didalam ruangan. Assisted GPS akan menerima/mendeteksi kemudian akan memodulasi sinyal yang lemah yang diterima GPS agar tingkat akurasi lebih tinggi. Cara kerjanya sama seperti GPS yaitu dengan membaca/menerima sinyal dari satelit-satelit yang mengorbit ke bumi. Dengan *Assisted GPS* lalu lintas antarmuka dapat dioptimalkan dengan kata lain proses waktu respon akan lebih cepat di bandingkan dengan teknik sebelumnya [4].

#### 2.2.2 *Google Maps*

Android memungkinkan untuk mengintegrasikan *Google maps* dengan beberapa aplikasi dan menyediakan fungsi seperti menunjukkan lokasi suatu objek, atau menampilkan rute yang berbeda pada peta *Google*. *Google* menyediakan semua fasilitas itu menggunakan library *Google play services* yang dapat di unduh secara eksternal dari *SDK android manager*. *Google map API* adalah kunci yang diperlukan untuk mengintegrasikan dengan aplikasi yang berhubungan dengan penentuan lokasi. *Google maps* dapat diintegrasikan dengan melakukan langkah-langkah berikut [5]:

1. Menginstal dan mengkonfigurasi dengan *library Google play services* dari android SDK manager.
2. Menambahkan *Google play services* ke library aplikasi yang akan di kembangkan.
3. Membuat atau mengambil sertifikat yaitu SHA1 dari keystore yang digunakan untuk sign aplikasi yang di kembangkan.
4. Membuat sebuah projek pada *Google APIs console*

5. Mendapatkan *API key* dari *Google*
6. Menambahkan kunci *API* yang dihasilkan tadi kedalam aplikasi yang dikembangkan.

Untuk menghandle maps pada LBS , android menyediakan sejumlah objek seperti *MapView* yang akan menampilkan peta, *MapFragment* yang menyediakan *life cycle* manajemen dan layanan *Google maps* [5].

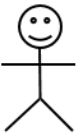
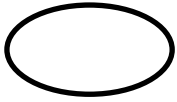
### 2.2.3 UML (*Unified Modeling System*)

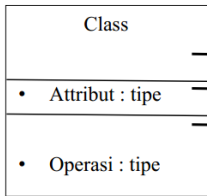

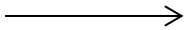

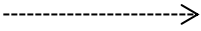

UML (*Unified Modeling System*) merupakan bahasa menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya [8]. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi objek. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson dibawah bendera *Rational Software Crop* [8]. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai prespektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan [8].

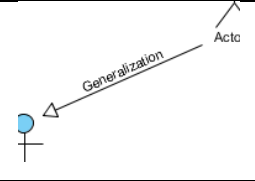
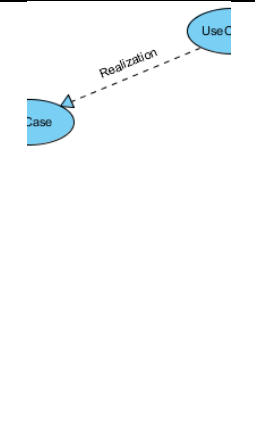
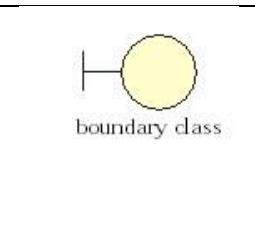
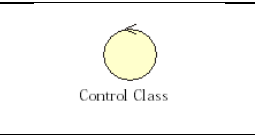
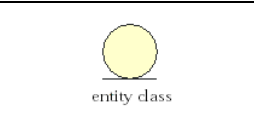
#### 2.2.3.1 Notasi Dalam UML

Adapun Notasi yang digunakan dalam pembuatan sebuah UML tampak pada tabel 2.2.3.1 dibawah ini [8] :

**Tabel 2.1** Notasi UML

<b>N O</b>	<b>NAMA</b>	<b>GAMBAR</b>	<b>FUNGSI</b>
1.	<i>Actor</i>		Menggambarkan peran dari pengguna sistem
2.	<i>Use Case</i>		Menunjukkan sekumpulan aktor dan

			<i>use case</i> dan hubungan diantara keduanya.
3.	<i>Class</i>		Mengilustrasikan sekumpulan kelas , paket dan hubungan yang merinci satu aspek tertentu dari sistem.
4.	<i>Interface</i>		Kumpulan operasi yang menspesifikasikan layanan dari kelas
5.	<i>Interaction</i>		Menunjukkan baik aliran pesan atau informasi antar objek maupun hubungan antar objek
6.	<i>Note</i>		Memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model.
7.	<i>Dependency</i>		Relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberikan pengaruh pada elemen lainnya.
8.	<i>Asociation</i>		Menggambarkan navigasi antar kelas

9.	<i>Generalization</i>		Hubungan antar elemen yang lebih umum ke yang lebih spesifik.
10.	<i>Realization</i>		Hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.
11.	<i>Boundary Class</i>		Kelas yang memodelkan interaksi antara satu atau lebih aktor dengan sistem.
12.	<i>Control Class</i>		Kelas yang mengontrol aktivitas dalam sistem
13.	<i>Entity Class</i>		Kelas yang menyimpan dan mengolah data.

### 2.2.3.2 Use Case

Diagram ini memperlihatkan himpunan *Use Case* dan Aktor-aktor (jenis khusus dari kelas). Diagram ini penting untuk mengorganisasikan dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna [8].

### 2.2.3.3 Sequential Diagram

*Sequence Diagram* merupakan diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu [8].



#### 2.2.3.4 Activity Diagram

*Activity Diagram* adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberikan tekanan pada aliran kendali antar objek [8].

#### 2.2.3.5 Class Diagram

Class diagram merupakan himpunan kelas-kelas, antarmuka, kolaborasi-kolaborasi, dan relasi-relasi. Diagram ini umumnya di temui pada pemodelan sistem berorientasi objek. Meski sifatnya statis, diagram ini sering pula memuat kelas-kelas aktif [8].

#### 2.2.3.6 Kamus Data

Kamus data berfungsi sebagai menyimpan semua objek data yang dibutuhkan dan dihasilkan oleh perangkat lain. Biasanya pembuatan kamus data dilakukan setelah pemodelan fungsional dan pemodelan status dan kelakuan selesai dibuat [8].

#### 2.2.3.7 Pengujian Black Box

Pengujian Black Box atau *behavioral testing* atau *specification-based testing*, *input/output testing* atau *Functional testing* dilakukan tanpa mendetail struktural *internal* dari sebuah sistem atau komponen sistem yang sedang di *testing*. *Black Box Testing* berfokus pada kebutuhan fungsional pada sebuah *software*, berdasarkan spesifikasi kebutuhan dari *software*. Menggunakan *Black box testing testing*, pembuat *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program [8].

Kategori *error* dapat diketahui melalui *black box testing*, antara lain :

1. Fungsi yang hilang atau tidak benar.
2. *Error* dari antarmuka
3. *Error* dari struktur data atau akses eksternal database.

4. *Error* dari kinerja atau tingkah laku program.

5. *Error* dari instalasi dan terminasi.

#### 2.2.3.8 Pengujian *White Box*

Pengujian *White box* atau *glass box testing* atau *clear box testing* adalah suatu metode pengujian yang didesain *test case* yang menggunakan struktur terkendali dari desain prosedural [8].

Metode desain *test case* ini dapat menjamin :

1. Semua *path* (jalur) yang terpisah dapat di *testing* setidaknya sekali tes.
2. Semua logika benar keputusan dapat di *testing* dengan jalur yang salah atau jalur yang benar.
3. Semua pengulangan (*looping*) dapat di *testing* terhadap batasnya dan ikatan operasionalnya.
4. Semua struktur *internal* data dapat di *testing* untuk memastikan validasinya.

#### 2.2.4 Android

Android adalah Sistem Operasi berbasis *Linux* yang dirancang untuk perangkat mobile selular layar sentuh seperti telepon pintar dan komputer tablet [2]. Android dikembangkan oleh Android Inc. dengan dukungan finansial dari *Google Inc.*, yang kemudian *Google* mengakuisisinya pada tahun 2005 [2]. Android didirikan di Palo Alto, California, pada bulan Oktober 2003 oleh Andy Rubin (pendiri Danger), Rich Miner (pendiri Wildfire Communications, Inc.), Nick Sears (mantan VP T-Mobile), dan Chris White (kepala desain dan pengembangan antarmuka WebTV) untuk mengembangkan perangkat seluler pintar yang lebih sadar akan lokasi dan preferensi penggunanya [2]. Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android lalu dialihkan bagi pasar telepon pintar untuk

menyaingi Symbian dan Windows Mobile (iPhone Apple belum dirilis pada saat itu) .[2]

Google mengakuisisi Android Inc. pada tanggal 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google. Pendiri Android Inc. seperti Rubin, Miner dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google [2]. Di Google, tim yang dipimpin oleh Rubin mulai mengembangkan platform perangkat seluler dengan menggunakan kernel Linux . Google memasarkan platform tersebut kepada produsen perangkat seluler dan operator nirkabel, dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bisa diperbarui [2].

#### 2.2.4.1 Fitur Dan Teknologi

- Dalvik VM

Dalvik VM adalah versi modifikasi dari bahasa pemrograman java yang digunakan untuk mengembangkan aplikasi berbasis android. Dalvik VM berfungsi untuk menjalankan aplikasi pada perangkat Android. Android menggunakan mesin virtual Dalvik dengan kompilasi tepat waktu untuk menjalankan 'dex-code' Dalvik (Dalvik Executable), biasanya diterjemahkan dari bytecode Java. [7]

- Application Interface & H/W Support

Pada layar perangkat Android telah dimanipulasi ( di program ) agar dapat merespon masukan dari user lewat sentuhan yang serupa dengan tindakan di dunia nyata, misal menggesekkan, mengetuk, mencubit untuk memanipulasi berbagai objek yang ada di layar perangkat android. Masukan dari user akan direspon secara cepat layaknya menyentuh permukaan air. Sistem Operasi Android mendukung berbagai hardware khusus seperti sensor herak, sensor jarak, giroskop, magnetometer, accelerometer dll. [7]

Dengan didukung oleh Google Play Store, jutaan aplikasi dapat dengan mudah di download dan digunakan sesuai kebutuhan dan keinginan pengguna [7]. Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK) [2]. SDK ini terdiri dari seperangkat perkakas pengembangan, termasuk debugger, perpustakaan perangkat lunak, emulator handset yang berbasis QEMU, dokumentasi, kode sampel, dan tutorial [2]. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) Eclipse, yang menggunakan plugin Android Development Tools (ADT) [2]. Perkakas pengembangan lain yang tersedia di antaranya adalah Native Development Kit untuk aplikasi atau ekstensi dalam C atau C++, Google App Inventor, lingkungan visual untuk pemrogram pemula, dan berbagai kerangka kerja aplikasi web seluler lintas platform [2].

- Arsitektur ( Kernel Linux )

Berdasarkan kernel linux, sebagian middleware, library, software dan API ditulis dalam bahasa C [7]. Platform perangkat keras umumnya berarsitektur ARM (Advanced RISC Machine ) pemrosesan bersifat paralel dengan didukung arsitektur MIPS atau x86 [7]. Berbeda dengan siklus pengembangan kernel Linux biasa Secara standar, Android tidak memiliki X Window System asli ataupun dukungan set lengkap dari perpustakaan GNU standar. Oleh sebab itu, sulit untuk memporting perpustakaan atau aplikasi Linux pada Android. Dukungan untuk aplikasi simpel C dan SDL bisa dilakukan dengan cara menginjeksi shim Java dan menggunakan JNI, misalnya pada port Jagged Alliance 2 untuk Android [2].

- Management Memory

Karena perangkat Android umumnya bertenaga baterai, Android dirancang untuk mengelola memori (RAM) guna menjaga konsumsi daya minimal, berbeda dengan sistem operasi desktop yang bisa terhubung pada sumber daya listrik tak terbatas. Ketika sebuah aplikasi Android tidak lagi digunakan, sistem secara otomatis akan menangguhkannya (suspend) dalam memori – secara teknis aplikasi tersebut masih "terbuka", namun dengan ditangguhkan, aplikasi tidak akan mengkonsumsi sumber daya (misalnya daya baterai atau daya pemrosesan), dan akan "diam" di latar belakang hingga aplikasi tersebut digunakan kembali . Cara ini memiliki manfaat ganda, tidak hanya meningkatkan respon perangkat Android karena aplikasi tidak perlu ditutup dan dibuka kembali dari awal setiap saat, tetapi juga memastikan bahwa aplikasi yang berjalan di latar belakang tidak menghabiskan daya secara sia-sia. [7]

Android mengelola aplikasi yang tersimpan di memori secara otomatis: ketika memori lemah, sistem akan menonaktifkan aplikasi dan proses yang tidak aktif untuk sementara waktu, aplikasi akan dinonaktifkan dalam urutan terbalik, dimulai dari yang terakhir digunakan. Proses ini tidak terlihat oleh pengguna, jadi pengguna tidak perlu mengelola memori atau menonaktifkan aplikasi secara manual. Namun, kebingungan pengguna atas pengelolaan memori pada Android telah menyebabkan munculnya beberapa aplikasi task killer pihak ketiga yang populer di Google Play. [7]

- Keamanan dan Privasi

Meskipun OS Android kebal terhadap penggunaan normal, tetapi kelemahan keamanan dapat dimanfaatkan, seperti yang dilakukan oleh komunitas open source, untuk mendapatkan akses ROOT (dapat digunakan untuk tujuan jahat) dan

memodifikasi kemampuan perangkat. Aplikasi Android berjalan di sandbox, sebuah area terisolasi yang tidak memiliki akses pada sistem, kecuali izin akses yang secara eksplisit diberikan oleh pengguna ketika memasang aplikasi. Sebelum memasang aplikasi, Play Store akan menampilkan semua izin yang diperlukan, misalnya: sebuah permainan perlu mengaktifkan getaran atau menyimpan data pada Kartu SD, tapi tidak perlu izin untuk membaca SMS atau mengakses buku telepon. Setelah meninjau izin tersebut, pengguna dapat memilih untuk menerima atau menolaknya, dan bisa memasang aplikasi hanya jika mereka menerimanya. [7]

- Network Connectivity

Sistem operasi android mendukung berbagai konektivitas mulai dari bluetooth, 2G/3G/LTE/GPRS, wifi, USB, dan juga GPS. mendukung transmisi data melalui jaringan 2G/3G/LTE/GPRS/EDGE. Koneksi internet juga dapat diakses melalui jaringan wifi dan juga dapat membagi jaringan ke perangkat lain melalui tethering. Komunikasi dengan komputer dapat dilakukan melalui USB dan bluetooth. Layanan HTTP didukung dan melalui penggunaan Google API apabila ponsel sedang mengaktifkan GPS. [7]

2.2.4.2 Versi Android Sampai Sekarang 29 Februari 2016 [2].

**Tabel 2.2** Versi Sistem Operasi Android

Versi	Nama kode	Tanggal rilis	Level API	Distribusi
6.0	Marshmallow	19 Agustus 2015	23	
5.x	<i>Lollipop</i>	15 Oktober 2014	21	
4.4.x	<i>KitKat</i>	31 Oktober 2013	19	24,5%
4.3.x	<i>Jelly Bean</i>	24 Juli 2013	18	8%
4.2.x	<i>Jelly Bean</i>	13 November 2012	17	20,7%
4.1.x	<i>Jelly Bean</i>	9 Juli 2012	16	25,1%
4.0.3– 4.0.4	<i>Ice Cream Sandwich</i>	16 Desember 2011	15	9,6%
3.2	<i>Honeycomb</i>	15 Juli 2011	13	

3.1	<i>Honeycomb</i>	10 Mei 2011	12	
2.3.3– 2.3.7	<i>Gingerbread</i>	9 Februari 2011	10	11,7%
2.3– 2.3.2	<i>Gingerbread</i>	6 Desember 2010	9	
2.2	<i>Froyo</i>	20 Mei 2010	8	0,7%
2.0–2.1	<i>Eclair</i>	26 Oktober 2009	7	
1.6	<i>Donut</i>	15 September 2009	4	
1.5	<i>Cupcake</i>	30 April 2009	3	

### 2.2.5 Agile Software Development

Kata *Agile* berarti bersifat cepat, ringan, bebas bergerak dan waspada. Kata *Agile* diambil dari kata *Agility*. *Agility* adalah dinamis, konten yang spesifik, perubahan yang agresif, dan berorientasi pada pembangunan [8]. Konsep *Agile Software Development* dicetuskan oleh Kent Beck dan 16 rekannya. Mereka tergabung dalam *Agile Alliance*, lalu mereka menandatangani “*Manifesto for Agile Software Development*” yang isinya [8]:

1. Interaksi dan personel lebih penting dari pada proses dan alat.
2. *Software* yang berfungsi lebih penting daripada dokumentasi yang lengkap.
3. Kolaborasi dengan *klien* lebih penting dari pada negosiasi kontrak.
4. Sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana.

Namun demikian, sama seperti model proses yang lain, *Agile Software Development* memiliki kelebihan dan tidak cocok untuk semua jenis proyek, produk, orang dan situasi. *Agile Software Development* memungkinkan proses model yang toleransi terhadap perubahan kebutuhan sehingga perubahan dapat cepat ditanggapi. Namun disisi lain menyebabkan produktifitas menurun [8].

Menurut *Agile Alliance* dalam *Agile Manifesto* ada 12 prinsip bagi mereka yang ingin berhasil dalam penerapan *Agile Software Development* [8]:

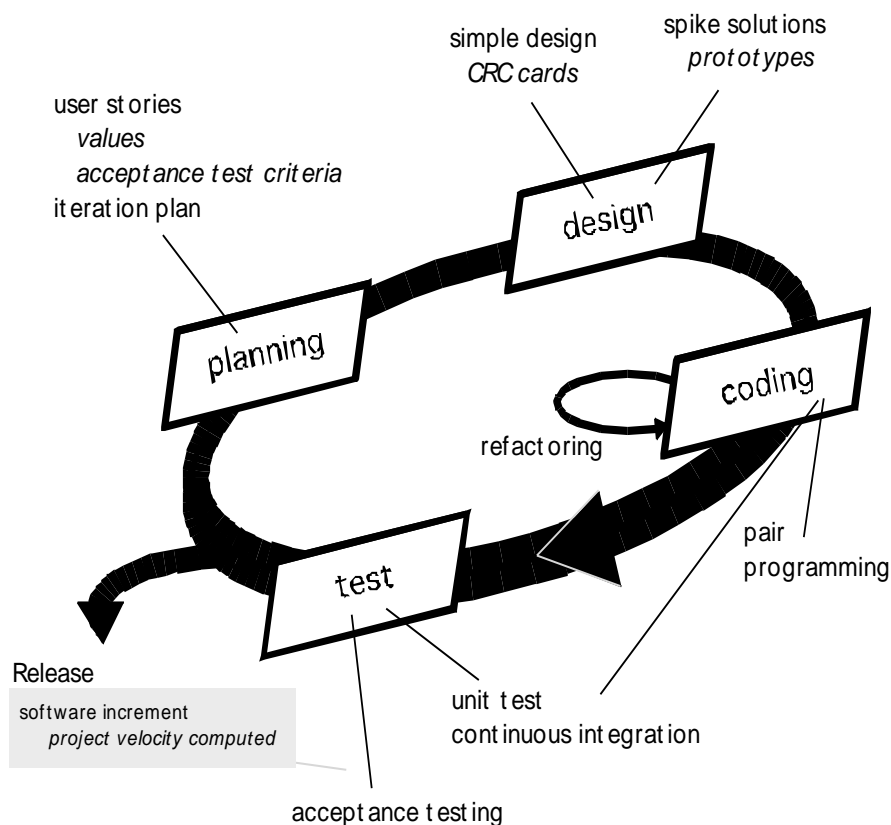
1. Kepuasan *klien* adalah prioritas utama dengan menghasilkan produk lebih awal dan terus menerus.
2. Menerima perubahan kebutuhan, sekalipun diakhir pengembangan.
3. Penyerahan hasil/*software* dalam hitungan waktu dua minggu sampai dua bulan.
4. Bagian bisnis dan pembangun kerja sama tiap hari selama proyek berlangsung.
5. Membangun proyek di lingkungan orang-orang yang bermotivasi tinggi yang bekerja dalam lingkungan yang mendukung dan yang dipercaya untuk dapat menyelesaikan proyek.
6. Komunikasi dengan berhadapan langsung adalah komunikasi yang efektif dan efisien.
7. *Software* yang berfungsi adalah ukuran utama dari kemajuan proyek.
8. Dukungan yang stabil dari sponsor, pengembang, dan pengguna diperlukan untuk menjaga perkembangan yang berkesinambungan.
9. Perhatian kepada kehebatan teknis dan desain yang bagus meningkatkan sifat *agile*.
10. Kesederhanaan adalah penting.
11. Arsitektur, kebutuhan dan desain yang bagus muncul dari tim yang mengatur dirinya sendiri. Secara periodik tim evaluasi diri dan mencari cara untuk lebih efektif dan segera melakukannya.

#### 2.2.6 *Extreme Programming Model (XP)*

Model proses ini diciptakan dan dikembangkan oleh Kent Beck. *Extreme Programming (XP)* adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel. Walaupun menggunakan kata programming, XP tidak hanya terfokus pada coding tetapi meliputi seluruh area pengembangan perangkat lunak [8].



Menurut Kent Beck, *XP* ringan, efisien, resiko rendah, mudah disesuaikan, dapat diprediksi, ilmiah dan mudah dikembangkan. Suatu model yang menekankan pada keterlibatan *user* secara langsung, pengujian dan *pay-as-you-go design* [8].



**Gambar 2.1** Skema Model Extreme Programming [8]

Keterangan Gambar [8]:

1. *Planning*, pengumpulan *user stories* dari klien yang klien tetapkan prioritasnya . Setiap *story* ditetapkan harga dan lama pembangunan, jika terlalu besar, *story* dapat dipecah menjadi beberapa *story* yang lebih kecil. Periksa dan pertimbangkan resiko .
2. *Design* (berprinsip sederhana), memanfaatkan kartu *CRC* (*Class Responsibility-Collaborator*) untuk identifikasi dan mengatur *class class* di konsep *Object Oriented* . Jika menemui kesulitan, *prototype*

dibangun (ini namanya *spike solution*) . Lakukan *refactoring*, yaitu mengembangkan desain setelah pengkodean.

3. *Coding*, menyiapkan unit *test* sebelum pengkodean dipakai sebagai fokus pemrogram untuk membuat program. *Pair programming* dilakukan untuk *real time program solving* dan *real time quality assurance* .
4. *Testing*, menggunakan *unit test* yang dipersiapkan sebelum pengkodean.

### 2.2.7 GPS Based Location Tracker

*Global Positioning System* (GPS) adalah metode penentuan posisi suatu objek di bumi, dalam semua kondisi cuaca. GPS menggunakan sejumlah satelit di orbit bumi untuk melakukan pelacakan posisi. GPS bekerja dengan menghitung jarak dari satelit penerima lokasi, minimal ada tiga satelit yang diperlukan untuk posisi dua dimensi dan empat satelit untuk posisi tiga dimensi. Satelit yang lebih banyak dapat menemukan posisi yang lebih akurat, sehingga titik persimpangan menjadi lebih kecil. Saat ini GPS semakin terintegrasi ke dalam kehidupan sehari - hari, contohnya GPS yang sudah tertanam di smartphone atau mobil. [3]

#### 2.2.7.1 Identifikasi Masalah

Aplikasi “*GPS Based Location Tracker*” adalah aplikasi berbasis layanan GPS yang akan membantu kita dalam menemukan geo-posisi yang tepat dari seseorang tergantung pada lokasi mereka saat ini / keberadaan. Geo-posisi akan ditampilkan pada Google Maps [7]. Beberapa *Point* penting tentang *GPS Based Location Tracker* :

- Semua data lokasi pengguna perangkat lunak ini akan diambil dari sebuah sensor GPS android yang di ambil dari server google maps milik sales.

- Untuk membatasi akses pengguna, akan diberlakukan proses otentikasi pada saat masuk aplikasi.
- Setiap kali perubahan geo-posisi dari pengguna aplikasi ini akan selalu “di refresh” dengan interval waktu yang sudah ditentukan sehingga database akan selalu terupdate.
- Semua perangkat akan memiliki ID dan ID itu akan digunakan untuk mencari keberadaan pengguna aplikasi ini.
- Aplikasi ini akan memiliki dukungan tambahan yaitu :
  - Street view dan Satellite View
  - Pin point pada Maps
  - Zoom in / Zoom out
  - Password

#### 2.2.7.2 Gambaran Umum Aplikasi

Dalam aplikasi ini, penulis telah menggunakan MapViews dengan didukung oleh Google API 10 keatas yang akan memungkinkan penggunaan aplikasi menjalankan pada sistem operasi android 4.0 keatas. Aplikasi ini menggunakan Server Apache dengan PHP & MySQL. Transaksi data dari atau ke database terjadi dengan bantuan script PHP. Pengambilan lokasi perangkat pengguna aplikasi akan dibantu dengan GPS.

#### 2.2.7.3 Batasan Perangkat

- Hardware
  - Semua Perangkat dengan Sistem Operasi Android
  - Versi Android : 4.0 keatas
  - Perangkat Android harus compatibel Google Maps Api 1.0
- Software
  - Server Side : Web hosting gratis

- Client End : Jaringan data Diaktifkan dan layanan lokasi juga diaktifkan.

### 2.2.8 *Android Studio*

*Android Studio official integrated development environment (IDE)* untuk pengembangan perangkat lunak pada platform android. Android Studio diperkenalkan pertama kali pada 16 Mei 2013 di konferensi Google I/O. Android studio tersedia secara gratis atau *freely available* dibawah lisensi apache 2.0 [9].

Versi android studio dimulai dari versi 0.1 pada mei 2013, kemudian muncul versi 0.8 yang dirilis pada bulan Juni 2014. Versi yang sebenarnya atau yang sudah stabil dan diperkenalkan untuk umum adalah versi 1.0 pada bulan desember 2014 [9].

#### 2.2.8.1 Fitur

Fitur yang dimiliki android studio antara lain [9] :

1. [\*Gradle\*](#)-based build support.
2. Android-specific [\*refactoring\*](#) and quick fixes.
3. [\*Lint\*](#) tools to catch performance, usability, version compatibility and other problems.
4. [\*ProGuard\*](#) integration and app-signing capabilities.
5. *Template-based wizards to create common Android designs and components.*
6. A rich [\*layout editor\*](#) that allows users to drag-and-drop UI components, option to [\*preview layouts\*](#) on multiple screen configurations.<sup>[12]</sup>
7. Support for building [\*Android Wear\*](#) apps
8. *Built-in support for Google Cloud Platform, enabling integration with Google Cloud Messaging and App Engine.*

### 2.2.9 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai perangkat salah satunya adalah telepon genggam. Java awalnya dibuat oleh James Gosling saat masih bekerja di Sun Microsystems yang sekarang merupakan bagian dari Oracle dan dirilis tahun 1995. Java banyak mengadopsi sintaksis yang terdapat pada C dan C++. Perangkat lunak berbasis java pada umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM) [10]. Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin . Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda . Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web [10].

#### 2.2.9.1 Versi Bahasa Pemrograman Java

Versi awal Java pada tahun 1996 sudah merupakan versi release sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya [10]:

- java.lang : Peruntukan kelas elemen-elemen dasar.
- java.io : Peruntukan kelas input dan output, termasuk penggunaan berkas.
- java.util : Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
- java.net : Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
- java.awt : Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).

- `java.applet`: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

#### 2.2.10 PHP ( *PHP Hypertext Preprocessor* )

PHP merupakan bahasa pemrograman berbasis “*server side*”. Bahasa pemrograman PHP banyak dipakai untuk pemrograman web yang dinamis, dan juga bisa dipakai untuk keperluan lainnya seperti pengembangan aplikasi berbasis android yang berbasis ( membutuhkan ) *server side* [11]. Kelebihan bahasa pemrograman PHP dari bahasa pemrograman lain adalah [11]:

- PHP adalah bahasa pemrograman yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- Di sisi pengembangan, bahasa pemrograman PHP lebih mudah dikembangkan karena banyak referensi yang mudah didapat dari *developer* lainnya.
- Bahasa pemrograman PHP bersifat *open source* yang dapat digunakan di berbagai sistem operasi seperti Windows, Macintosh, Linux, Unix dan saat ini juga bisa digunakan untuk pengembangan aplikasi pada sistem operasi Android.
- Disisi pemahaman *script*, PHP adalah salah satu bahasa pemrograman yang paling mudah untuk dipahami, karena terdapat banyak referensi baik dari buku maupun dari *website*.
- *Web server* yang mendukung bahasa pemrograman android dapat didapatkan dengan mudah, seperti apache, IIS, Lighttpd, hingga Xitami yang konfigurasinya relatif mudah.

### 2.2.11 The Dalvik Virtual Machine (DVM)

Salah Satu elemen kunci dari android adalah *Dalvik Virtual Machine*. Android berjala dalam DVM bukan di *Java Virtual Machine* (JVM), sebenarnya banyak persamaan dengan JVM seperti Java ME (*Java Mobile Edition*), tetapi android menggunakan *virtual machine* sendiri yang menurut saya di kostumisasi dan dirancang untuk memastikan bahwa beberapa fitur-fitur berjalan lebih efisien pada perangkat mobile [2].

*Dalvik Virtual Machine* adalah “*register bases*” sementara *Java Virtual Machine* adalah “*stack based*”, DVM didesain dan ditulis oleh Dan Bornsten dan beberapa engineers Google lainnya [2]. Jadi bisa dikatakan “*dalvik equals(java) == false*”. DVM menggunakan kernel *linux* untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan untuk menulis aplikasi C / C + sama halnya pada OS *linux* [2].

Semua *hardware* yang berbasis android dijalankan dengan menggunakan *virtual machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir mengenai implementasi perangkat keras tertentu. DVM mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa java dan di kompilasi menggunakan tools yang disediakan dalam SDK android [2].

### 2.2.12 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 di *release* oleh Google. Saat ini disediakan android SDK sebagai alat bantu dan API untuk mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman java [2]. Beberapa fitur – fitur android yang paling penting adalah [2] :

- a. Framework aplikasi yang mendukung penggantian komponen dan reusable.
- b. Mesin Virtual Dalvik dioptimalkan untuk perangkat mobile.
- c. Integrated browser berdasarkan engine open source WebKit.
- d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, Grafis 3D berdasarkan spesifikasi ES 1,0 (opsional akselerasi hardware).
- e. SQLite untuk penyimpanan data
- f. Media support yang mendukung audio, video dan gambar, GSM telephony (tergantung hardware).
- g. Kamera, GPS, kompas, dan accelerator meter (tergantung hardware).
- h. Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE eclipse.

### 2.2.13 MySQL

MySQL merupakan perangkat lunak *database management system* SQL atau sering disebut DBMS yang multi-user, multi-thread, dengan lebih dari 6 instalasi di seluruh dunia [12]. MySQL tersedia sebagai perangkat lunak yang bersifat gratis dengan lisensi GNU General Public License (GPL), tetapi juga dijual dibawah lisensi komersial untuk kasus-kasus yang dimana penggunaannya tidak cocok dengan GPL [12].

Penggunaan perangkat lunak MySQL sangat populer sekarang dikarenakan sebagai DBMS pada aplikasi web yang berbasis PHP, maka keduanya seringkali disebut sebagai *Dynamic Duo* [12].

Kelebihan MySQL dibandingkan dengan RDBMS lainnya [12]:

- Dapat diintegrasikan dengan beberapa bahasa pemrograman seperti Java, C/C++, .Net, dimana merupakan bahasa pemrograman paling dominan di kalangan developer.
- Mendukung ODBC untuk Windows sehingga dapat dipakai pada aplikasi yang berjalan di OS windows.



- Dapat dijalankan pada spesifikasi hardware yang relatif rendah karena MySQL lebih hemat resource memory ( dibandingkan dengan database lain ).
- Bisa mendeteksi kesalahan pesan pada klien dengan menggunakan lebih dari 20 bahasa meskipun bahasa indonesia belum termasuk didalamnya.

### 2.3 Kerangka Pemikiran

**Tabel 2.3** Kerangka Pemikiran

