BAB II
LANDASAN TEORI

II.1 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi snap place recognition untuk tempat wisata di Kota Bandung memanfaat teknologi camera geotag dan geofance pada anroid. Teori-teori yang terkait serta mendukung penelitian ini akan dibahas pada bab ini.

II.1.1 Kota Bandung

Kota Bandung adalah salah satu Kota terbesar di Indonesia dengan luas wilayah 16.731 hektar. Secara geografis, Kota Bandung terletak di tengah-tengah provinsi Jawa Barat, serta berada pada ketinggian kurang lebih 768 meter di atas permukaan laut. Kota Bandung memiliki luas wilayah yang secara administratif terbagi atas 30 kecamatan, 151 kelurahan. Dengan jumlah penduduk bandung tahun 2012 tercatat 2.655.160 jiwa[5].

Kota ini dikenal juga dengan julukan Kota Kembang dikarenakan pada zaman dahulu Kota ini memiliki keindahanya dengan banyak tumbuhan dan jenis bunga yang tumbuh di Kota ini, selain itu Kota Bandung disebut pula Paris Van Java atau dengan kata lain Paris nya di pulau Jawa dikarenakan keindahaan bagai Kota Paris di Prancis. Saat ini Kota Bandung masih menjadi tujuan pariwisata favorit di Indonesia yang terkenal akan wisata belanja baik pakaian ataupun makanan dan Kota ini pun terkenal dengan bangunan-bangunan bersejarah, salah satunya adalah gedung Merdeka yang menjadi tempat penyelenggaraan Konferensi Asia Afrika yang diadakan rutin setiap lima tahun sekali sehingga Kota Bandung sekarang pun dikenal dengan nama Capital City of Asia & Afrika pada tahun 2015 ini[5].

II.1.2 Traveling

Traveling dalam bahasa Indonesia adalah aktivitas melancong; berpindah dalam satu tempat ketempat lainnya dengan berbagai alasan, seperti bisnis, liburan, dan sebagainya[6].

Tetapi menurut Herajeng Gustiayu dalam bukunya yang berjudul The Backpacker's Notes, traveling adalah lebih kepada bagaimana beradaptasi dengan lingkungan baru. Didalam buku tersebut juga dibahas 3 hal dasar yang penting dari membuat perencanaan traveling dan menentukan biaya yang tepat diantaranya adalah transportasi , akomodasi dan makanan[6].

Karena tarveling menjadi sebuah gaya hidup masyarakat modern karena itulah penelitian ini diambil.

II.1.3 Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer,instruksi (*instructiom*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output* [7]. Menurut Kamus Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini[7].

Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instructionset*. Program aplikasi merupakan program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemproses kata dan *Web Browser*. Aplikasi akan menggunakan *system* operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app[7].

Secara historis, aplikasi adalah software yang dikembangkan oleh sebuah perusahaan. App adalah sofware yang dibeli perusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan

pertempuran persaingan yang baru, yang parallel dengan yang terjadi antar sistem operasi yang dimunculkan[7].

II.1.3.1 Klasifikasi Aplikasi

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

II.1.4 Sistem

Sistem adalah seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan untuk mencapai tujuan[8].

II.1.4.1 Syarat-syarat Sistem

1. Sistem harus dibentuk untuk menyelesaikan masalah.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen sistem.
4. Unsur dasar dari proses lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen

II.1.4.2 Eleman Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu: tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan. Berikut penjelasan mengenai elemen-elemen yang membentuk sebuah sistem:

1. Tujuan (*Goal*)

Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

2. Masukan (*Input*)

   Masukan(input) sistem yaitu segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud atau tampak secara fisik maupun yang tidak tampak.

   Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi, misalnya permintaan jasa pelanggan.

3. Proses

   Proses merupakan bagian yang melakukan perubahan atau ransformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapatberupa aktivitas pembedahan pasien.

4. Keluaran (*output*)

   Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistemnformasi, keluaran bisa berupasuatu informasi, saran, cetakan laporan, dansebagainyaBatas(*Boundary*)

   Batas sistem adalah pemisah antara sistem dan daerah di luar sistem atau lingkungan. Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepak bola mempunyai aturan permainan dan keterbatasan kemampuan pemain.

5. Mekanisme Pengendalian dan Umpan Balik

   Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*). Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

6. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem

II.1.4.3 Klasifikasi Sistem

Suatu sistem dapat diklasifikasikan sebagai:

1. Sistem abstrak

   Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia).
2. Sistem fisik

   Merupakan sistem yang ada secara fisik (sistem komputer, sistem akuntansi, sistem produksi)
3. Sistem alamiah

   Sistem yang terjadi melalui proses alam. (sistem matahari, sistem luar angkasa, sistem reproduksi)
4. Sistem Buatan Manusia

   Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut human-machine system contoh : sistem informasi.
5. Sistem Tertentu (*deterministic system*)

   Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh : sistem komputer)
6. Sistem Tak Tentu (*probabilistic system*)

   Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
7. Sistem tertutup (*close system*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* atau secara relatif tertutup, tidak benar-benar tertutup .

8. Sistem terbuka (*open system*)
   Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.
9. Sistem sederhana dan Sistem kompleks

II.1.5 Informasi

Informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerima dan mempunyai nilai yang nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusankeputusan yang akan datang, sedangkan menurut *Burch* dan *Strater*, informasi adalah pengumpulan atau pengolahan data untuk memberikan pengetahuan atau keterangan [9].

Jadi, secara umum informasi adalah data yang sudah diolah menjadi suatu bentuk lain yang lebih berguna yaitu pengetahuan atau keterangan yang ditujukan bagi penerima dalam pengambilan keputusan, baik masa sekarang atau yang akan datang. Informasi yang dihasilkan dari aplikasi yang dibuat adalah rute angkot rekomendasi untuk dapat dipilih oleh pengguna.

II.1.5.1 Kegunaan Informasi

Ada 4 faktor utama yang berhubungan dengan kegunaan informasi:

1. Kualitas informasi (*information quality*) Kualitas dari suatu informasi tergantung dari 4 hal, yaitu informasi harus:

a. Akurat (*accurate*) dan presisi (*precision*)
   Akurat dalam menampilkan informasi dan presisi dalam detail informasi yang diberikan.

b. Kelengkapan (*completeness*)
   Informasi yang tersedia cukup lengkap untuk setiap user dan situasi.

c. Umur (*age*) dan ketepatan waktu (*timeliness*)

Umur berarti lamanya waktu dalam meng-update informasi dan ketepatan waktu berarti menyediakan informasi secepat mungkin pada saat dibutuhkan sehingga berguna.

d. Sumber (*source*)

Orang atau organisasi yang menghasilkan informasi.

2. Aksesibilitas informasi (*information accessibility*)

a. Ketersediaan (*availability*)

Memberikan informasi kepada yang membutuhkan. Informasi dapat diakses oleh yang membutuhkan.

b. Keabsahan (*admissibility*)

Keabsahan (boleh atau tidak boleh dipakai) informasi tergantung pada hukum, peraturan atau budaya pada saat tertentu.

3. Presentasi informasi (*information presentation*)

a. Tingkatan (*level of summarization*)

Perbandingan antara data asli dengan yang ditampilkan. Manipulasi data hingga tingkatan yang sesuai, semakin sederhana semakin baik.

b. Format

Bentuk dimana informasi ditampilkan ke *user*. Manipulasi data dalam bentuk yang sesuai.

4. Keamanan informasi (*information security*)

a. Batasan akses (*access restriction*) Prosedur dan teknik mengontrol *user* yang boleh atau tidak mengakses data pada situasi tertentu. Penggunaan password atau teknik lain untuk mencegah *user* yang tidak berhak.

Enkripsi (*encryption*) Konversi data ke bentuk tertentu sehingga tidak dapat dibaca oleh *user* yang tidak berhak.

II.1.5.2 Nilai Informasi

Nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Sebagian besar informasi tidak dapat persis ditaksir keuntungannya dengan satuan nilai uang, tetapi dapat ditaksir nilai efektivitasnya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *costeffectiveness* atau *cost benefit*

II.1.6 Sistem Informasi

Sistem informasi adalah sebuah sistem informasi yang mempunyai fungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik dan berguna[13].

II.1.6.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*), dan blok kendali (*controls block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya[14].

a. Blok Masukan

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

b. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

c. Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

d. Blok Teknologi

Teknologi digunakan untuk menrima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan danmengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi ini terdiri dari tiga bagian utama yaitu: teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

e. Blok Basis

Data Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut.

f. Blok Kendali Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

Pengelompokan komponen-komponen sistem informasi berbasis computer adalah sebagai berikut:

1 Perangkat keras (*hardware*)

Hardware ini merupakan peralatan fisik yang dapat digunakan untuk mengumpulkan, memasukkan, memproses, menyimpan, dan mengeluarkan hasil pengolahan data dalam bentuk informasi.

2 Perangkat lunak (*software*)

Software adalah kumpulan dari program-program yang digunakan untuk menjalankan aplikasi tertentu pada komputer.

3 Manusia (*brainware*)

Brainware dalam sistem informasi berperan sebagai pemberi dan pengguna informasi.

4 Prosedur (*procedure*)

Prosedur adalah rangkaian aktivitas atau kegiatan yang dilakukan secara berulang-ulang dengan cara yang sama.

5 Basis data (*database*)

Database merupakan kumpulan data-data yang tersimpan di dalam media penyimpanan di suatu perusahaan (arti luas) atau di dalam komputer (arti sempit).

6 Jaringan komunikasi (*communication network*)

Jaringan telekomunikasi saat ini menghubungkan beberapa daratan dan lautan untuk memindahkan data dalam jumlah besar.

II.1.6.2 Kualitas Informasi

Dimensi kualitas bisa disebut sebagai syarat sebuah informasi dikatakan berkualitas dilihat dari beberapa sudut. Ada 3 dimensi kualitas informasi yaitu dimensi waktu informasi *(time dimension),* dimensi konten informasi (*contentdimension),* dan dimensi bentuk informasi (*form dimension*). Karakteristik dalam dimensi ini adalah pilihan analis sistem informasi. Tidak semua harus masuk tapi disesuaikan dengan kebutuhan[15].

II.1.6.2.1 Time Dimension (dimensi waktu informasi).

Informasi dikatakan berkualitas jika *Currency* alias *Up to dat*e. Dengan kata lain, Informasi yang berkualitas adalah informasi yang disampaikan dengan tepat waktu. Sebuah sistem informasi yang menyajikan informasi tidak terkini, maka tidak bisa digunakan terutama untuk mengambil keputusan. Informasi yang tersaji cepat dan aktual akan memuaskan pengguna serta mendukung pengambilan keputusan. Terlebih lagi informasi yang disajikan dalam kewaktuan nyata atau *real time*. *Timeliness* memiliki arti bahwa informasi tersedia kapan saja pengguna membutuhkan tanpa keterbatasan waktu.

II.1.6.2.2 Content Dimension (dimensi konten informasi)

Dimensi ini mencangkup ketepatan atau *Accuracy*. Jelas bahwa informasi yang tersedia akurat, bebas dari kesalahan sehingga tidak menjerumuskan user dan

berakibat salah dalam mengambil keputusan. Selain akurat, informasi yang dihasilkan haruslah sesuai dengan konten yang diinginkan oleh pengguna. Konten yang disajikan harus sesuai target pengguna juga dengan penyajian informasi yang tidak sia-sia. Data dikatakan bahwa informasi yang akurat adalah informasi yang di butuhkan oleh pengguna.

II.1.6.2.3 Form Dimension (dimensi bentuk informasi)

Dimensi ini berbicara mengenai dalam bentuk apa informasi dapat sampai kepada pengguna. Bentuk informasi dapat menggunakan jenis media yang sesuai dengan kebutuhan pengguna dalam menyampaikan informasi. Dalam penelitian ini, media penampilan informasi yang digunakan adalah *mobile phone* atau perangkat gerak telepon genggam yang menggunakan sistem operasi Android. Selain itu juga memanfaatkan teknologi internet dalam penyampaian informasinya.

II.1.7 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet.

Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008[16].

Alasan menggukan sistem android ini dari data IDC (Smartphone Market Share 2016) android diposisi pertama pengguna terbanyak dunia[2].

II.1.7.1 Android Life Cycle

Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi – fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang

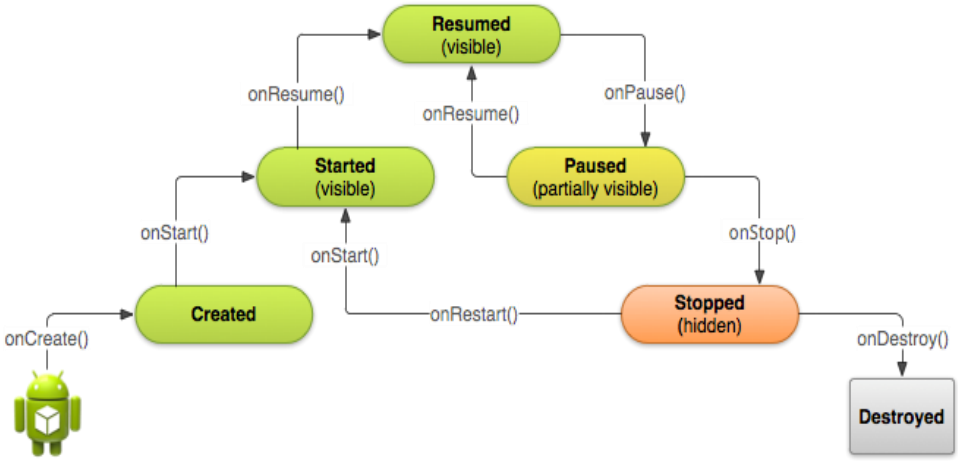
berbeda seperti ditunjukan pada klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan [17].

Tabel II. 1 Android Life Cycle

Functionality	Java Base Class	Examples
Can focused thing a user do	Activity	Edit a note, play a game
Background process	Service	Play music, update weather icon
Receive messages	BroadcastReceiver	Trigger alarm upon event
Store and retrieve data	ContentProvider	Open a phone contact

Setiap aplikasi pasti menggunkan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan *spacifed permission* sebelum digunkan seperti komponen *Service, BroadcastReceiver.ContentProvider* (Schwarz, Duston,Steel, & To, 2013).

Android memiliki peradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi main(), sistem android menjalankan kode dalam method Activity dengan menerpakan metode callback tertentu yang sesuai dengan tahanp tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi android memiliki siklus hidup yang berbeda dengan aplikasi desktop atau web, Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang lumayan tinggi seperti ketika *handling* panggilan masuk aplikasi diharuskan menghentikan proses sementara, Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna[17].



Gambar II. 1 Siklus Hidup Android (Google Inc.2014)

Terdapat beberapa *state* dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gamabar II.1 Siklus android(Google Inc.2014), akan tetapi hanya beberapa dari state tersebut yang menjadi statis diantaranya:

1. Resumed

Resumed terjadi ketika aplikasi berjalan setelah state paused. State ini akan menjalakan perintah program yang ditulis pada method onResume() [18].

2. Paused

Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karaena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam state ini [18].

3. Stopped

Dalam keadaan ini,aplikasi benar-benar tidak di tampilkan dn tidak terlihat oleh pengguna tetapi masih meninggalkan service dibackground [18].

State lain seperti Created dan Started bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan meninggalkan metode *life cyle callback* berikutnya. Artinya ,setelah sistem OnCreate() dipanggil, dengan cepat sistem akan memanggil method OnStart(), Kemudian diikuti oleh onRuseme() [19].

II.1.7.2 Fitur

Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah:

1. Multi-proses dan App Widgets

Sistem operasi android tidak melarang prosesor menjalankan lebih dari satu aplikasi dalam satu waktu. Sistem operasi android dapat mengatur aplikasi dan thread yang berjalan secara *multitasking.* Keuntungan yang didapat adalah ketika aplikasi berjalan dan berinteraksi dengan pengguna di layer depan sistem operasi, proses dari aplikasi lain dapat berjalan untuk melakukan pembaruan informasi. Sebagai contoh misalnya ketika pengguna memainkan game, proses lain dapat berjalan di belakang aplikasi seperti memeriksa harga saham dan memunculkan peringatan [20].

App Widgets adalah mini aplikasi yang dapat *embedded* dalam aplikasi seperti home screen. App widgets dapat menjalankan proses request seperti musik streaming atau mendeteksi suhu ruangan secara *background* (Google Inc, 2014).

Multi-proses dapat memberikan manfaat berupa *user experience* yang lebih banyak, namun penggunaan fitur tersebut dapat menghabiskan banyak energi baterai jika penggunaan tidak benar [20].

2. Touch Gestures dan Multi-touch

Touchscreen adalah *user interface* intuitif yang digunakan banyak *smartphone* di dunia. Dengan fitur ini interaksi dapat dibuat lebih mudah karena cukup dengan menggunakan jari tangan. *Multi-touch* adalah kemampuan yang dapat melakukan tracking lebih dari satu tangan dalam satu waktu, Fitur ini sering digunakan untuk interaksi memperbesar atau memutar objek. Selain itu pengembang dapat membuat interaksi baru dengan memanfaatkan fitur tersebut [20].

3. Hard dan Soft Keyboard

Salah satu fitur pada perangkat smartphone adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam pengoperasian android.. Pengembang aplikasi tidak perlu secara manual untuk mengintegrasikan tombol

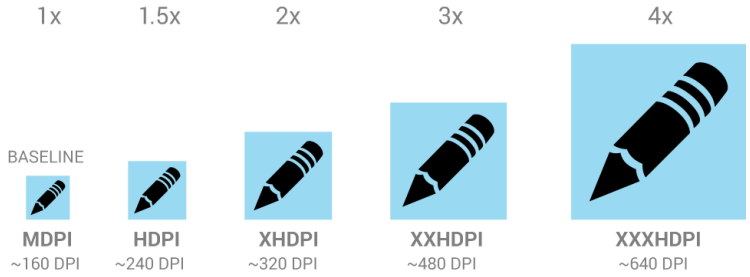
tersebut dalam aplikasi. Tombol non fisik adalah tombol yang dibuat oleh sistem operasi seperti keyboard virtual, dan tombol navigasi aplikasi [20].

II.1.7.3 Prinsip Desain

Android memiliki beberapa prinsip desain yang dapat menjadi acuan dalam membuat desain aplikasi android diantaranya adalah

1. Multiple Assets

Android mendukung jutaan smartphone, tablet dan perangkat lain dalam berbagai ukuran layar dan ukuran, untuk itu *Multiple Assets* sangat disarankan digunakan untuk mengatasi fragmentasi pada android. Seperti ilustrasi pada Gambar II.2 Klasifikasi Ukuran Ikon, android menciptakan beberapa klasifikasi ukuran icon yaitu MDPI, HDPI, XHDPI, XXHDPI dan XXXHDPI. MDPI dan HDPI dikhususkan untuk icon yang akan digunakan pada device berukuran smartphone sedangkan untuk XHDPI, XXHDPI dan XXXHDPI digunakan pada device berukuran tablet [11].



Gambar II. 2 Klasifikasi Ukuran Ikon

2. Taouch Feedback

Touch Feedback dalam android digunakan sebagai respon setiap objek yang ditekan pengguna. Hal ini bertujuan untuk memberi tahu pengguna objek mana yang berinteraksi dengan pengguna [11].



Gambar II. 3 Touch Feedback

3. Pattern Gesture

Touch gesture berguna untuk meningkatkan experience pengguna dalam menggunakan aplikasi. Terdapat beberapa gesture yang didukung oleh android diataranya adalah [11].

1. Tourch

Touch adalah gesture default sebagai event untuk item tertentu yang paling sering digunakan dalam perangkat android, hampir seluruh kontrol utama aplikasi ini menggunakan touch [11].

2. Long Press

Biasanya digunakan untuk seleksi data , dengan gesture ini dimungkinkan untuk memilih satu atau lebih item dalam sebuah tampilan dan menjalankan suatu fungsionalitas tertentu [11].

3. Swipe Or Drag

Swipe adalah menyentuh sebuah titik pada layar dan menggerakkan jari yang tetap tersentuh pada layar ke titik lain pada layar. Swipe biasanya dilakukan untuk kebutuhan berpindah layar atau tergantung dari aplikasi. Swipe dapat dilakukan dari dan ke arah mana saja [11].

4. Double Touch

Pada Smartphone dan tablet android, melakukan dua kali tapping secara berturut-turut pada satu obyek, fungsinya berbeda dengan double klik mouse komputer. Pada Android, teknik ini biasanya dipakai untuk melakukan zoom in dan zoom out atau memperbesar / memperkecil sebuah obyek gambar [11].

5. Pinch Open

Teknik lain yang biasa digunakan adalah dengan menggunakan dua jari, di mana kedua jari tersebut menyentuh dua titik pada layar yang terpisah di mana ujung dari dua jari tersebut tidak bersentuhan, kemudian kedua jari tersebut sambil tetap menyentuh layar bergerak saling mendekati. Teknik ini digunakan untuk membuka aplikasi tertentu [11].

6. Pinch Close

Teknik Pinch Close adalah kebalikan dari Pinch Open di mana spread dilakukan dengan berawal dari dua jari yang bersentuhan pada ujungnya dan ditempelkan sebuah titik yang sama pada layar, kemudian kedua jari tersebut bergerak memisahkan atau menjauhi satu sama lain Gerakan ini untuk menutup layar[11].

II.1.8 JSON

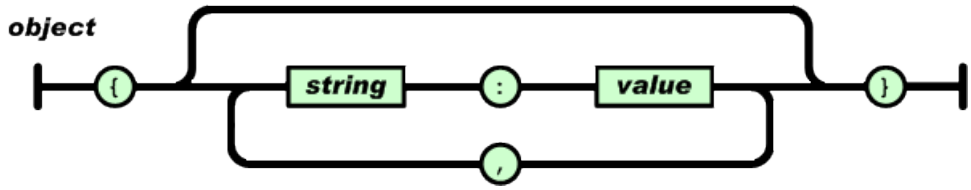
JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari dua struktur[21]:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (record), struktur (struct), kamus (dictionary), tabel hash (hash table), daftar berkunci (keyed list), atau associative array [11].
2. Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence) [11].

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut:

II.1.8.1 Objek

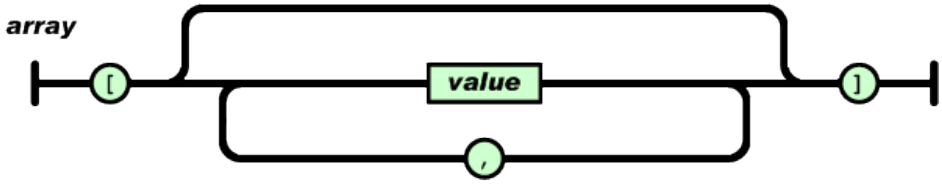
Objek adalah sepasang nama / nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal dalam bentuk JSON [11].



Gambar II. 4 Objek JSON

II.1.8.2 Larik

Larik Adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka kotak buka) dan diakhiri dengan] (kurung buka tutup). Setiap nilai dipisahkan oleh , (koma). Larik dalam JSON dapat digunakan sebagai value dari JSON object hal ini dapat berguna jika JSON menyimpan data bertingkat [21]



Gambar II. 5 Array JSON

Bentuk data JSON objek dan larik dapat saling dikombinasikan untuk mendukung struktur datang yang lebih kompleks. JSON mendukung beberapa tipe data untuk menjadi value seperti Angka, String, Bollean dan nilai NULL [21]

II.1.9 *Global Positioning System* (GPS)

GPS(*Global Positioning System*)merupakan sistem navigasi yang menggunakan satelit dalam penggunaannya. GPS terdiri dari tiga bagian yaitu sistem kontrol, satelit dan pengguna. Sistem kontrol adalah bagian yang mengontrol pergerakan satelit – satelit yang ada dan saling berinteraksi satu sama lain kemudian pengguna adalah alat navigasi yang digunakan seperti perangkat mobile yang kini sudah memiliki fitur GPS di dalamnya. GPS biasanya digunakan untuk menunjukan suatu lokasi yang berada di permukaan bumi dengan tingkat akurasi yang cukup baik, yaitu kurang dari 10 meter selama tidak ada benda padat yang dapat menghambat sinyal untuk mendapatkan lokasi pengguna[23].



Gambar II. 6 *Global Positioning System*

Cara kerja GPS secara logik ada 5 langkah:

1. Memakai perhitungan “triangulation” dari satelit.
2. Untuk perhitungan “triangulation”, GPS mengukur jarak menggunakan travel time sinyal radio.
3. Untuk mengukur travel time, GPS memerlukan memerlukan akurasi waktu yang tinggi.

4. Untuk perhitungan jarak, kita harus tahu dengan pasti posisi satelit dan ketingian pada orbitnya.
5. Terakhir harus menggoreksi delay sinyal waktu perjalanan di atmosfer sampai diterima reciever.

GPS dapat melakukan perhitungan dan menentukan posisi *user* dan menampilkan dalam *maps*. Jika sudah dapat menyimpan posisi *user* selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah tuju, rute, tujuan perjalanan serta jarak tujuan. Perhitungan waktu yang akurat sangat menentukan informasi lokasi *user*.

II.1.9.1 Fitur GPS

Selain fitur GPS pada perangkat *smartphone* ada juga fitur lain yaitu A-GPS. A-GPS merupakan sebuah singkatan dari *Assisted Global Positioning System*. Perbedaan dengan GPS biasa adalah A-GPS menggunakan bantuan operator telekomunikasi yang Anda gunakan untuk mempercepat pembacaan lokasi dimana Anda berada[23]. Kekurangan dari fitur GPS yang disematkan di smartphone adalah *Chip* atau perangkat yang bekerja menangkap sinyal GPS dari satelit tidak terlalu peka atau sensitif. Sehingga untuk menangkap sinyal informasi yang dikirimkan oleh satelit membutuhkan waktu yang lebih lama.

Di dalam fitur A-GPS, operator seluler membantu menambahkan informasi yang kurang sehingga waktu untuk menentukan lokasi Anda berada menjadi lebih cepat. Menara BTS yang mengeluarkan sinyal memiliki fitur GPS juga yang dapat menentukan lokasi dengan tepat. Namun, kekurangan dari A-GPS ini adalah dikenai biaya tambahan untuk dapat mengaksesnya. Namun fitur A-GPS ini dapat dinonaktifkan sehingga hanya akan menerima informasi dari satelit. Tapi dengan menonaktifkan fitur ini, Anda akan membutuhkan waktu yang lebih lama untuk menentukan lokasi.

II.1.9.2 Penentuan Lokasi

Sejak tahun 1978, terdapat 59 satelit GPS yang berhasil di orbitkan mengelilingi bumi, walaupun pada tahun 2010, hanya 30 satelit yang dinyatakan masih beroperasi dengan normal. Kemampuan manusia untuk menentukan lokasi dengan tepat telah dikenal lama dengan menggunakan asap sebagai sinyal untuk menandakan suatu tempat atau lokasi[22].

GPS dirancang khusus untuk sistem navigasi di bumi. Jika digambarkan dalam peta dunia, maka posisi kita dapat ditandai dengan sebuah titik koordinat lokasi tempat dimana kita berada, seperti pada gambar II.16, titik yang ditandai terdiri atas dua komponen, yaitu *latitude* dan *longitude*, yang menginformasikan keberadaan kita di bumi melalui perangkat lunak GPS. Ketika titik koordinat lokasi diketahui, maka informasi ini dapat digunakan oleh program GPS untuk mendapatkan informasi lebih banyak untuk pengguna, seperti bisnis terdekat, dan informasi kemacetan lalu lintas. Ketika titik koordinat telah ditentukan, aplikasi akan menggunakan proses *geolocation* untuk mendapatkan informasi mengenai lokasi disekitar pengguna.

II.1.9.3 Metode Penentuan Lokasi

Tentunya penentuan lokasi tidak hanya berasal dari sistem GPS. Informasi yang digunakan dan bagaimana proses yang terjadi pada perangkat dan mem-*parsing* informasi lokasi adalah berdasarkan pada jenis perangkat yang digunakan. Terdapat beberapa metode untuk menentukan lokasi pada perangkat komputasi dan tidak semua berdasarkan pada satelit GPS. Berikut adalah Gambar II.6 Penentuan Lokasi[22].



Gambar II. 7 Penentuan Lokasi

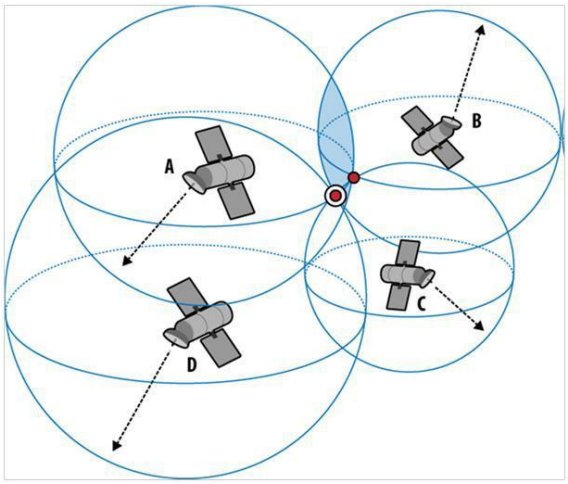
Berikut beberapa cara penentuan lokasi:

II.1.9.4.1 Global Positioning Service (GPS)

Satelit GPS secara terus-menerus mengirim informasi sehingga perangkat GPS dapat mengurainya (*parse*), contohnya: susunan GPS yang masih berfungsi dengan baik, dimana semua satelit berada dalam orbit yang tepat atau jalur transmisi satelit, dan waktu transmisi. Perangkat *receiver* akan mengkalkulasikan posisinya dengan waktu pengiriman sinyal dengan beberapa satelit dalam *array* yang terlihat (tanpa hambatan ke angkasa).

Alat penerima menentukan waktu yang dibutuhkan untuk menerima setiap pesan kemudian menghitung jarak dari setiap satelit berdasarkan informasi ini. Jarak pada setiap satelit dari alat penerima, orbit, dan perhitungan *trilateration* menginformasikan alat penerima mengenai posisi sekarang. Sementara tiga transmitter radio *triangulation* cukup untuk menentukan lokasi yang masuk akal (*reasonable location*), terdapat faktor waktu yang menjadi

perbandingan dengan satelit yang sedang orbit. Proses ini membutuhkan waktu, mungkin beberapa detik agar sinyal satelit menjangkau bumi – setiap kesalahan kecil jam pada satelit, dikalikan dengan waktu sekarang dapat menjadi kesalahan besar penentuan lokasi. Menggunakan keempat satelit mengurangi kesalahan dalam persamaan. Dalam kasus yang paling sering terjadi, sebuah alat *receiver* akan menggunakan empat satelit atau lebih untuk menghitung posisinya.Berikut adalah Gambar II.7 Posisi Satelit GPS.



Gambar II. 8 Satelite GPS

2. *IP Address*

IP (*Internet Protocol*) *address* merupakan angka unik yang diberikan pada setiap perangkat yang terhubung dengan internet sehingga dapat berkomunikasi dengan perangkat lainnya. Ketika perangkat terhubung dengan internet, perangkat dapat mengirim data kepada perangkat lain dan kemudian menerima respon kembali. Meskipun sebuah perangkat menerima alamat yang lebih atau selalu berubah-ubah, *IP address* dapat berubah statis (permanen) atau dinamis (temporar). Tanpa memperhatikan jenis alamat yang dimiliki oleh sebuah perangkat, alamat tersebut selalu akan terdiri dari empat kumpulan angka yang dipisahkan oleh titik, misalnya : 123.123.123.123.

Umumnya, sebuah *IP Address* ditentukan oleh ISP (*Internet Service Provider*) dalam bagian-bagian berdasarkan wilayah oleh lembaga *registry* lokal. Oleh karena itu, negara, wilayah, dan bahkan sebuah kota dapat dengan mudah di identifikasi dengan sebuah *IP Address*. Selanjutnya, kemajuan saat ini dalam pengumpulan data dan pengelolaannya oleh ISP, perangkat lokasi geografis mampu menentukan sebuah lokasi dengan tingkat akurasi tinggi. Yang menjadi tantangan terbesar dengan penggunaan IP Address untuk menentukan sebuah lokasi adalah bahwa terdapat ratusan institusi regional yang akan diproses untuk mendapatkan data.

3. GSM/CDMA Cell ID

Cell ID merupakan angka unik yang digunakan untuk mengidentifikasi setiap perangkat *mobile* pada jaringan seluler, umumnya seperti *IP Address* mengidentifikasikan sebuah perangkat dalam jaringan. Ada dua tipe jaringan yang paling popular digunakan yaitu *Global System for Mobile Communication* (GSM) dan *Code Division Multiple Access* (CDMA). Jenis layanan seluler yang dibutuhkan didasarkan pada daerah cakupan sinyal.GSM merupakan teknologi jaringan *mobile* yang paling tertua dan oleh karena itu peningkatan pelayanan dari teknologi GSM tersedia luas dibanding teknologi lainnya. GSM adalah teknologi 2G yang tersedia lebih dari 200 negara, dan jumlah *transfer* data pada jaringan ini lebih dari 75% dari seluruh dunia. Kemampuan jaringan GSM untuk dilakukan migrasi ke layanan jaringan 3G dan 4G (*Evolved High Speed Packet Access* (HSPA+),

Long Term Evolution (LTE),dan *Service Architecture Evolution*(SAE) juga sangat cepat, sehingga dapat melayani hampir empat juta pelanggan. Meskipun teknologi GSM tidak dapat menangani jumlah data yang sama seperti teknologi lainnya, jaringan GSM memiliki tingkat kualitas jaringan yang tinggi karena kemampuannya untuk menempatkan repeater didalam dan diluar bangunan bahkan jumlah penggunaan repeater yang semakin dikurangi. Hal ini menjadikan jaringan GSM sebagai operator dengan layanan jaringan yang baik dengan tingkat gangguan jaringan yang kecil.

CDMA merupakan teknologi terbaru setelah GSM, baik CDMA dan GSM sama-sama merupakan layanan jaringan 2G (cdmaOne(TM)), dan 3G (CDMA2000(R) dan WCDMA). Keuntungan dari teknologi CDMA adalah dapat mengijinkan banyak pengguna dalam waktu yang bersamaan dengan porsi frekuensi yang diberikan oleh *band*, dimana tidak dapat dilakukan pada jaringan GSM. Jaringan CDMA, seperti GSM juga dapat dilakukan migrasi ke teknologi LTE untuk mendukung jaringan 4G.

Tanpa memperhatikan jenis teknologi dari perangkat *mobile* yang digunakan, prinsip dasarnya adalah mempunyai *identifier unique* dalam jaringan. Menggunakan *triangulation*, *latitude* dan *longitude* Cell ID dapat diidentifikasi, dengan dilakukan *geolocated*. Penggunaan beberapa tower untuk melakukan triangulation posisi dari perangkat mobile, mempunyai tingkat akurasi tinggi. Hal

ini mengapa *geolocation* yang bergantung pada jenis teknologi, bekerja lebih baik dalam lingkungan urban (perkotaan) – dimana keberadaan tower yang satu dengan lainnya berdekatan dibanding pada area pedesaan.

4. *Wifi* dan *Bluetooth MAC Address*

Wifi dan *Bluetooth MAC address* mempunyai kinerja seperti *IP Address. Media Access Control (MAC) address* merupakan pemberian angka unik pada suatu perangkat, biasanya dilakukan oleh pabrik pembuat perangkat tersebut. Angka ini sifatnya permanen dan sebagai penanda global, namun pada perangkat modern, angka ini dapat diganti secara manual; dikenal dengan *MAC spoofing* (teknik untuk mengganti default *MAC address* dari antarmuka jaringan pada perangkat jaringan). Contoh sebuah MAC Address : 12-34-56-78-9A-BC. *MAC address* untuk router wifi adalah alamat yang mudah untuk ditemukan atau ditandai pada perangkat nirkabel (*wireless*), begitupun dengan MAC *address* untuk perangkat Bluetooth. Menggunakan alamat ini dengan cara yang sama sebagai alamat IP, *latitude* (garis lintang) dan *longitude* (garis bujur), dan lokasi fisik, dapat ditemukan.

II.1.9.4.2 *Latitude* (Garis Lintang)

Dalam geografi, garis lintang adalah garis khayal yang digunakan untuk menentukan lokasi di bumi terhadap garis khatulistiwa (utara atau selatan). Posisi lintang biasanya dinotasikan dengan simbol huruf Yunani

II.1.9.4.3 *Longitude* (Garis Bujur)

Bujur kadangkala dinotasikan oleh huruf abjad Yunani (λ), menggambarkan lokasi sebuah tempat di timur atau barat daya Bumi dari sebuah garis utara-selatan yang disebut Meridian Utama. LongitudeUtama ke $+180^0$ arah timur dan -180^0 arah barat. Tidak seperti lintang yang memiliki equator sebagai posisi awal alami, tidak ada posisi awal alami, tidak ada posisi awal untuk bujur. Oleh karena itu, sebuah dasar meridian harus dipilih. Meskipun kartografer Britania Raya telah lama menggunakan meridian Observatorium Greenwich di London, referensi lainnya digunakan di tempat yang berbeda, termasuk Ferro, Roma, Kopenhagen, Yerusalem, Saint Petersburg, Pisa, Paris, Philadelphia, dan Washington, D.C.. Pada

tahun 1884, Konferensi Meridian Internasional mengadopsi meridian Greenwich sebagai Meredian utama universal atau titik nol bujur.

II.1.10 Google Maps

Google maps adalah teknologi dan aplikasi layanan web pemetaan yang disediakan oleh Google, sifat bebas (tidak untuk penggunaan komersial), yang banyak digunakan layanan berbasis peta, mencakup website Google Maps, Google Ride Finder, Google Transit, dan peta yang digunakan oleh penyedia layanan pihak ketiga melalui Google Maps API. Google maps menawarkan peta jalan (*street maps*), sebuah perencana rute (*route planner*) untuk perjalanan dengan menggunakan mobil, pejalan kaki, sepeda atau angkutan umum dan pencari bisnis perkotaan untuk beberapa negara di seluruh dunia. Satelit Google Maps, gambar tidak diperbaharui dalam waktu nyata (realtime); mungkin beberapa minggu atau tahun lamanya[22].

Google Maps menggunakan varian yang mirip dengan *Mercator projection* (proyeksi peta silender diperkenalkan oleh *geographer* Belgian dan *cartographer* Gerardus Mercator, pada tahun 1569) sehingga tidak dapat menampilkan wilayah kutub. Produk yang berelasi dengan Google Maps adalah Google Earth, program yang berdiri sendiri (*stand-alone*) yang menawarkan fitur pandangan globe, termasuk area kutub[22].

II.1.11 Geofence

Geofence adalah parameter virtual untuk wilayah geografis di dunia nyata. Parameter virtual yang ditetapkan ini bisa menjadi statis ataupun dinamis. Salah satu contoh khas dari geofence dinamis adalah radius sekitar toko atau titik lokasi. geofence statis dapat di atur dari batas yang telah ditetapkan, seperti zona kehadiran di sekolah atau batas lingkungan. Kostum digital geofencing juga dapat digunakan, jika perangkat yang menyediakan layanan berbasis lokasi (LBS) pengguna masuk atau keluar geofencing, perangkat akan menerima pesan notifikasi. Notifikasi ini mungkin berisi informasi tentang lokasi dari perangkat. Notifikasi Geo-fence

mungkin dikirimkan ke ponsel sebagai contoh meningkatkan setiap bentuk lain dari tindakan

Kita bisa menggambarkan geo fencing sebagai LBS proaktif. Dalam banyak kasus sistem proaktif jauh lebih nyaman dari yang reaktif, di mana pengguna harus secara eksplisit meminta data untuk lokasi. Ada berbagai jenis lokasi peristiwa untuk memperbaiki posisi GPS yang dapat diuji. Misalnya, apakah pengguna di dekat dengan tempat tujuan (POI) atau ke pengguna lain.

Geofence sekarang ini menjadi salah satu wajah baru di Layanan Berbasis Lokasi. Sekarang pengembangan asli untuk LBS sendiri lebih atau kurang selesai. Kita bisa melihat beberapa standar umum dan aplikasi (berpikir tentang berbagai implementasi layanan Tempat-tempat seperti Foursquare, Twitter places, Google place, Facebook place, dll) Semua dari mereka memudahkan Anda dalam berbagi lokasi info atau mendapatkan beberapa hal yang berada di dekatnya. Dan geo-fence membuka pintu untuk beberapa personalisasi. Anda bisa mendapatkan beberapa kustom disesuaikan dengan data ditempat yang benar, terutama ketika kedua aliran data dan tempat (area) untuk menerima itu dihasilkan secara dinamis. Pada kedua, geo-fence akan terintegrasi dalam waktu terdekat dengan sensor dan aplikasi M2M

II.1.12 Android Studio

Android Studio adalah sebuah IDE untuk Android *Development* yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai *build environment*. Fitur-fitur lainnya adalah sebagai berikut[23]:

1. Menggunakan Gradle-based build system yang fleksibel.
2. Bisa mem-*build* multiple APK.
3. *Template support* untuk Google *Services* dan berbagai macam tipe perangkat.

4. Layout editor yang lebih bagus.
5. Built-in support untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
6. Import library langsung dari Maven repository.

Adapun beberapa persyaratan sistem pada windows untuk menggunakan

Android Studio adalah sebagai berikut:

1. Microsoft Windows 8/7/Vista/2003 (32 atau 64 bit)
2. RAM minimum 2 GB, Rekomendasi 4 GB RAM
3. Disk kosong 500 MB
4. Sedikitnya 16 GB untuk Android SDK, Emulator system images, and caches.
5. Java Development Kit (JDK) 7 atau yang lebih tinggi
   Resolusi minimum 1280 x 800

II.1.13 GPS Photo Tagging (Geotagging)

GPS Photo Tangging juga dikenal sebagai *geotangging,* merupakan proses penambahan infromasi posisi data GPS (*Latitude, Longtitude, Altitude)* dalam sebuah photo digital. Posel-ponsel berkamera yang memiliki GPS *receiver* intenal umumnya memiliki fitur ini. Makanisme GPS *PhotoTagging* adalah pada saat sebuah foto diambil menggunakan kamera (digintal atau ponsel) yang memiliki fitur *geotangging,* kamera atau ponsel tesebut mencatat lebih banyak infromasi/data dibandingkan dengan sebuah foto yang diambil dengan kamera biasa. Informasi tersebut termasuk waktu dan data ketika sebuah foto diambil,orintasi dari kamera (*potrait* atau *landscape)* , apakah menggunkan lampu flash dan detail kamera lainnya yang digunkan seperti Aperatur , Exposure, dan Local Lenght. Semua data di simpan pada suatu tempat yang disebut EXIF Headers[24].

EXIF (*Exchangeable Image File Format)* headers berisi petunjuk foto dengan data yang dapat dibaca oleh aplikasi manajemen foto atau sebuah situs foto tertentu. Selain itu, EXIF *Headers* juga menyediakan sebuah ruang untuk mengisi kordinat *Longtitude, Latitude,* dan *Altitude*

II.1.13.1 EXIF Headers

Data EXIF (*Exchangeable Image Format)* adalah data yang disimpan oleh kamera digital pada sebuah image yang berisi informasi mengenai kondisi dan setting kamra digital pada waktu dilakukan pmotretan. EXIF dikembangkan oleh Japanese Electronics Industry Development Association (JEIDA) sebagai upaya untuk mempermudah dan membuat standar pertukaran data antara perangkat lunak pengolah gambar/citra digital dan perangkat keras seperti kamera, serta didukung oleh hampir seluruh kamera digital.

Informasi umum yang bisa didapatkan dari EXIF sebuah foto digital adalah antara lain[24]:

1. Tanggal dan jam berapa sebuah foto dibuat.
2. Merk, tipe kamera,jenis lensa yang dipakai untuk memotret foto.
3. Resolusi dari kamera yang digunkan untuk mengambil gambar. Digunkan atau tidak fitur flash pada kamera.
4. Lokasi foto tersbut dimabil, jika kamera digunkan mendukung GPS.

II.1.13.2 EXIF Interface Android Class

Kelas Exifinterface adalah salah satu kelas dalam pemograman android yang dapat digunkan untuk membaca dan menulis informasi tag Exif pada yang berformat JPEG.

Berikut adalah Tabel II.2 Tag EXIF informasi Exif yang dapat diambil

Tabel II. 2 Tag EXIF

No	Tagging	Keterangan
1	TAG_APERATURE	Type is String
2	TAG_DATETIME	Type is String
3	TAG_DATETIME_DIGITIZED	Type is String
4	TAG_EXPOSURE_TIME	Type is String
5	TAG_FLASH	Type is int

6	TAG_FOCAL_LEGHT	Type is rational
7	TAG_GPS_ALTITUDE	The altitude (in meters) based on the refernce in TAG_GPS_ALTITUDE_REFF
8	TAG_GPS_ALTITUDE_REEF	0 if the altitude is above sea level
9	TAG_GPS_DATESTAMP	Type is String
10	TAG_GPS_LATITUDE	String
11	TAG_GPS_LATITUDE_REF	Type String
12	TAG_GPS_LONGLITUDE	String
13	TAG_GPS_LONGLITUDE_REF	Type String
14	TAG_GPS_PROCESING_METHOD	Type is String
15	TAG_GPS_TIMESTAMP	Type is String
16	TAG_IMAGE_LENGTH	Type is String
17	TAG_IMAGE_WIDTH	Type is Sring
18	TAG_ISO	Type is String
19	TAG_MAKE	Type is String
20	TAG_MODEL	Type is String
21	TAG_ORIENTATION	Type is int
22	TAG_SUBSEC_TIME	Type is int
23	TAG_SUBSEC_TIME_DIG	Type is int
24	TAG_SUBSEC_TIME_ORIG	Type is int
25	TAG_WHITE_BELANCE	Type is int

II.1.14 MySQL

MySQL adalah implementasi dari sistem menejemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, dan pemakaiannya tidak untuk kepentingan komersial. MySQL merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL

(*Structured Query Language*), adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis[25].

II.1.14.1 Sejarah MySQL

MySQL diciptakan oleh MySQL AB, sebuah perusahaan yang didirikan oleh Michael Widenius (Monty), David Axmark, dan Allan Larsson, di Swedia pada tahun 1995. Kemudian pada tahun 2008, Sun Microsystem,Inc. mengakusisi perusahaan tersebut dengan perkiraan harga $1 billion US dollar. Pada tanggal, 27 januari 2010, Sun Microsystem, Inc diakusisi oleh Oracle Corporation, setelah kedua perusahaan tersebut menandatangi persetujuan pada tanggal 20 april 2009, Sun Microsystem, Inc kemudian bergabung (*merged*) dengan Oracle USA, Inc. kemudian perusahaan tersebut berubah nama menjadi Oracle America,Inc[25].

II.1.14.2 Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Microsoft Windows, Linux, FreeBSD, Mac OS X server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-*user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. Peningkatan performa (*performance tuning*). MySQL memiliki kecepatan yang menakjubkan dalam manangani query sederhana,dengan kata lain dapat memproses lebih banyak SQL per satuanwaktu.
5. Antar muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API *(Application Programming Interface*).

6. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan(*tools*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

Struktur Tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

II.1.15 Web Service

W3C mendefinisikan web service sebagai sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML atau JSON dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan message berbasis XML atau JSON melalui protokol internet Web service adalah sebuah software aplikasi yang tidak terpengaruh oleh platform, menyediakan method-method yang dapat diakses oleh network. Web Service juga akan menggunakan XML untuk pertukaran data, khususnya pada dua entities bisnis yang berbeda. Beberapa karakteristik dari web service adalah[26]:

1. *Message-based*
2. *Standards-based*
3. *Programming language independent*
4. *Platform-neutral*

Beberapa key standard di dalam web service adalah: JSON, XML, SOAP, WSDL and UDDI (Avestro, 2007).

II.1.16 *Object Oriented Analysis Desain* (OOAD)

OOAD adalah metode analisis yang memeriksa kebutuhan (*requirements*) dari sudut pandang kelas (*class*) dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem.OOAD merupakan cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut

konsep sekitar dunia nyata. Dasar pembuatan adalah objek,yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas[27].

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem[27].

Terdapat beberapa konsep dalam OOAD, yaitu:

1. Objek (*object*)

a. Objek adalah benda secara fisik dan konseptual yang ada di sekitar kita. Sebuah objek memiliki keadaan sesaat yang disebut state.
b. State dari sebuah objek adalah kondisi dari objek atau himpunan keadaan yang menggambarkan objek tersebut. State dinyatakan dengan nilai dari atribut objeknya.
c. Atribut adalah nilai internal suatu objek yang mencerminkan karakteristik objek, kondisi sesaat, koneksi dengan objek lain dan identitas.
d. *Behaviour* (perilaku objek) mendefinisikan bagaimana sebuah objek bertindak dan memberi reaksi. Behaviour ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan oleh objek tersebut, yang dicerminkan oleh *interface*, *service*, dan method dari objek tersebut.
e. Interface adalah pintu untuk mengakses service dari objek
f. *Service* adalah fungsi yang dapat dikerjakan oleh sebuah objek
g. *Method* adalah mekanisme internal objek yang mencerminkan perilaku objek tersebut

2. Kelas (*class*)

Class adalah himpunan objek yang sejenis yaitu mempunyai sifat atau atribut, perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum. *Class* adalah abstraksi dari objek dalam dunia nyata. *Class* menetapkan spesifikasi perilaku dan atribut dari objek tersebut.

3. Pengujian Kotak Hitam (*black box*)

Sebuah objek adalah Kotak hitam. Konsep ini menjadi dasar implementasi objek. Dalam operasi OO hanya developer yang dapat memahami detail proses yang ada didalam Kotak tersebut, sedangkan *user* tidak perlu mengetahui apa yang dilakukan yang penting mereka dapat menggunakan objek untuk memproses kebutuhan mereka. Kotak hitam berisi kode dan data.

a. *Encapsulation*, yaitu proses menyembunyikan detail implementasi sebuah objek. Untuk mengakses data objek tersebut adalah melalui interface. Untuk berkomunikasi dengan objek digunakan message.
b. *Message* adalah permintaan agar objek menerima untuk membawa metode yang ditunjukkan oleh perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (*sender*)

4. Asosiasi dan Agregasi

Asosiasi adalah hubungan yang mempunyai makna antara sejumlah objek. Asosiasi digambarkan dengan sebuah garis penghubung diantara objeknya. Contohnya : Asosiasi karyawan dengan unit kerja. Setiap karyawan bekerja di satu unit kerja, sedangkan unit kerja dapat memiliki beberapa karyawan.

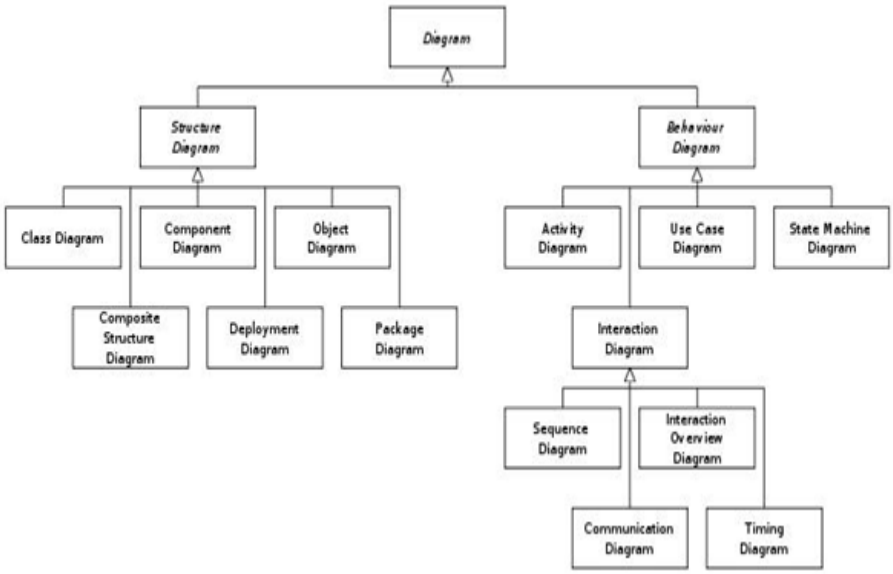
Agregasi adalah bentuk khusus sebuah asosiasi yang menggambarkan seluruh bagian pada suatu objek merupakan bagian dari objek yang lain. Contohnya : Kopling dan piston adalah bagian dari mesin, sedangkan mesin, roda, body merupakan bagian dari sebuah mobil.

II.1.16.1 UML (Unifed Modeling Language)

Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal, Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang

dibangun dengan *object-oriented* (OO).UML merupakan sintaks umum untuk membuat model logika dari suatu sistem dan digunakan untuk menggambarkan sistem agar dapat dipahami sela fase analisis dan desain. UML merupakan suatu bahasa pemodelan yang terdiri banyak model diantaranya adalah[]:

Diagram UML dikelompokan menjadi 2 yaitu Diagram Struktur atau *Statis Diagram* dan Diagram Prilaku Sistem atau *Behaviour Diagram*[28].

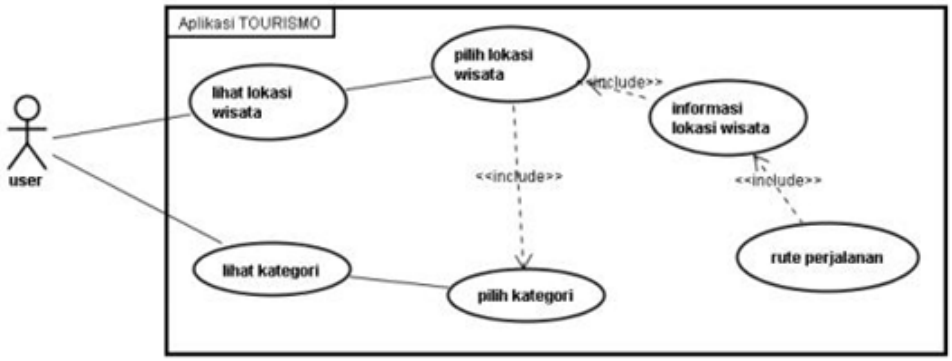


Gambar II. 9 Diagram UML

UML terdiri dari bermacam-macam diagram yang digunakan untuk pemodelan dalam pengembangan sistem mulai dari tahap analisi sampai implementasi. Pada saat melakukan desain sistem tidak harus semua diagram pada UML diimplementasikan akan tetapi perlu adanya kekonsistenan rancangan diagram yang satu dengan lainya. Pada penelitian ini diagram yang digunakan yaitu

Use case, Class, Sequence dan Activity.

1. Diagram *Use Case* (*Use Case Diagram*)



Gambar II. 10 Contoh Use Case

Use case diagram merupakan suatu pemodelan yang menitik beratkan pada fungsionalitas utama dalam suatu konteks sistem. *Use case* diagram juga menekankan tingkah laku fungsional utama dalam sistem berinteraksi dengan objek diluar sistem tersebut. Selain itu, *use case* diagram juga telah menitik beratkan jenis hubungan diantara fungsi utama. Adapun komponen-komponen dalam *use case* diagram antaranya[28].

a. Aktor

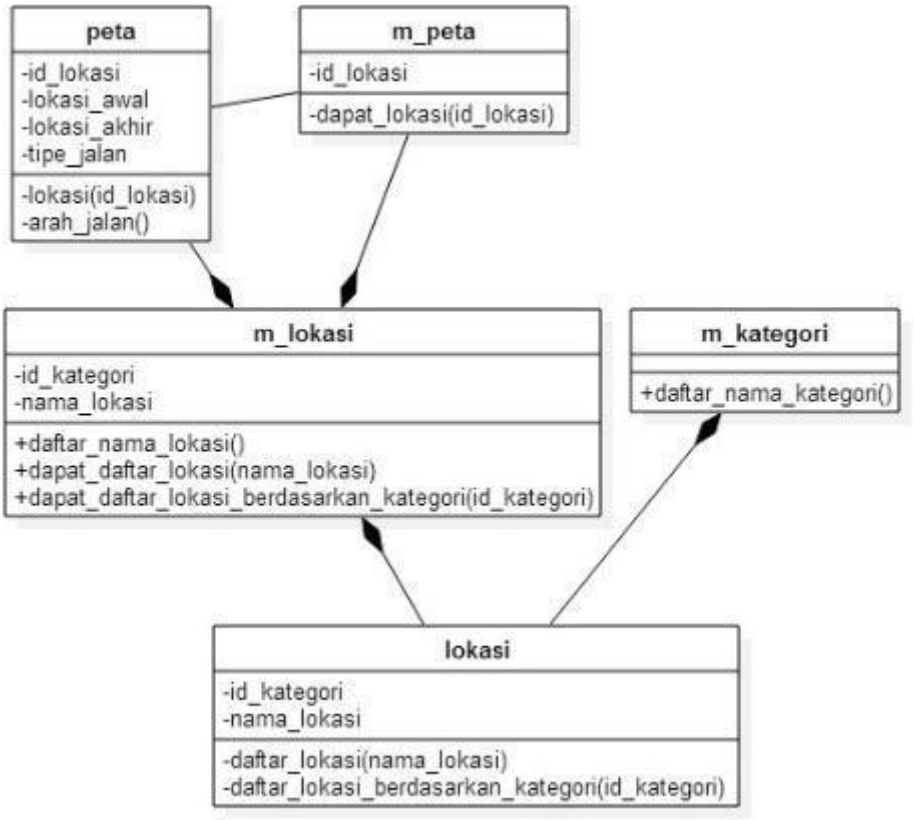
Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berada diluar sistem namun berkaitan erat dengan fungsionalitas didalamnya. Aktor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Aktor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. Aktor biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut[28].

b. *Use Case*

Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya

fungsionalitas utama yang berkaitan dengan sistem. Use case menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem[28].

2. Diagram Kelas (*Class Diagram*)

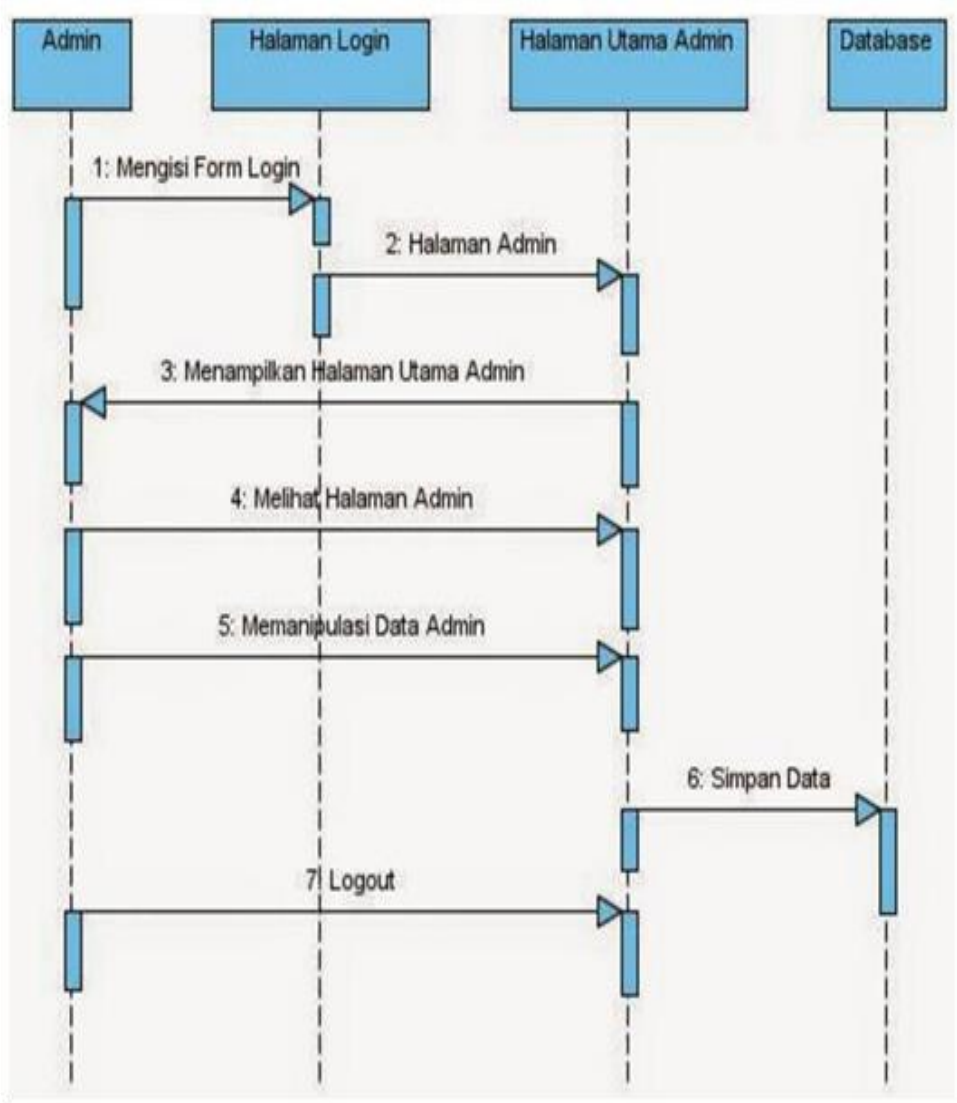


Gambar II. 11 Contoh *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Kelas pada kelas diagram terdiri dari 3 bagian utama yaitu nama kelas, isi properti dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing

komponen penyusun kelas memiliki hak akses seperti *public*, *private*, dan *protected* [28].

3. Diagram Sekuensial (*Sequence Diagram*)



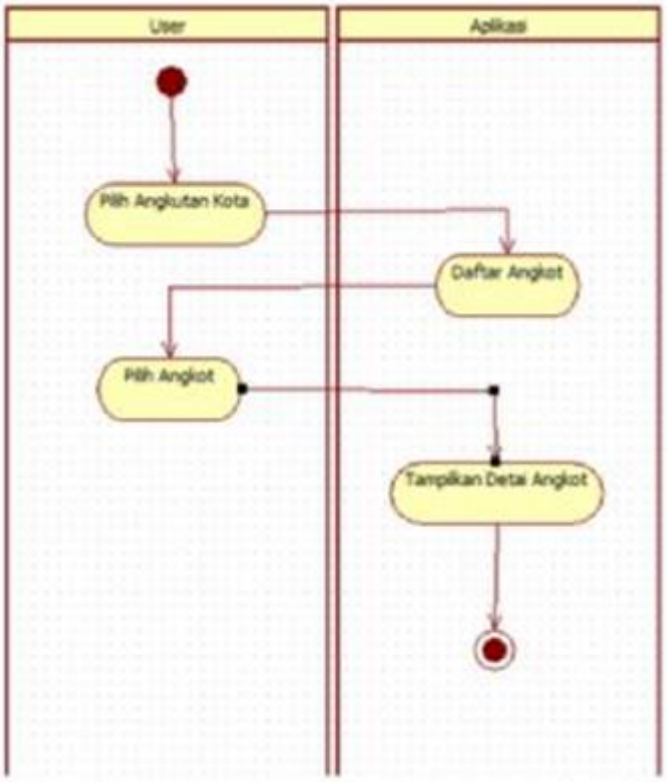
Gambar II. 12 Contoh *Sequence Diagram*

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi[28].

Tabel II. 3 Elemen-elemen Sequence Diagram

No	Nama	Penjelasan
1	Objek Lifeline	Menggambarkan batasan objek
2	Boundary	Berhubungan dengan proses *inputoutput* interface
3	Controller	Berhubungan dengan proses
4	Entity	Berhubungan dengan inpu-*output* data
5	Message Arrow	Menggambarkan alir proses, perintah atau pengiriman data
6	Aktivasi	Menggambarkan aktivitas objek
7	Aktor	Menggambarkan aktor suatu objek

4. Diagram Aktivasi (*Activity Diagram)*



Gambar II. 13 Contoh *Activity Diagram*

Diagram aktifitas adalah diagram *flowchart* yang diperluas untuk menunjukkan aliran kendali satu aktivitas ke aktivitas yang lain. Diagram aktifitas digunakan untuk memodelkan aspek dinamis sistem. Diagram aktivitas berupa operasi-operasi dan aktivitas-aktivitas di *use case*. [28].

Tabel II. 4 Elemen *Activity Diagram*

Simbol	Keterangan
	Start point
	End point
	Aktivities

Tabel II. 5 Elemen *Activity*

Simbol	Keterangan
	Join (penggabungan)
	Fork(percabangan)
Swimline	Sebuah cara mengelompokkan aktivitas berdasarkan aktor (mengelompokkan aktivitas dalam sebuah urutan yang sama)

II.1.17 Java

Java menurut defenisi sun adalah nama untuk sekumpulan teknologi untuk dapat membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Java dikembangkan pada bulan Agustus 1991, komersial, maka diganti menjadi java. Pada desember 1998, Sun memperkenalkan nama " java 2" (J2) sebagai generasi kedua dari java platform. (J2SE), Enterprise Edition (J2EE), dan Micro Edition (J2ME) (Kathy Siera,Bert Bates, 2003) [29].

Ada tiga platform java masing-masing diarahkan untuk tujuan tertentu dan untuk lingkungan komputasi yang berbeda-beda

1. Standard Edition (J2SE) : J2SE merupakan inti darii bahasa pemrograman Java. J2SE didesain untuk jalan pada komputer desktop dan komputer workstations.
2. Enterprise Edition (J2EE) : Dengan built-in mendukung untuk servlets.JSP, dan XML, edisi ini ditunjukan untuk aplikasi berbasis server.
3. Micro Edition (J2ME) : Didesain untuk piranti dengan memori terbatas, layar display terbatas dan power pemrosesan yang juga terbatas.

II.17.1 Kelebihan Java

Adapun bebrapa kelebihan dari bahasa pemrograman Java adalah sebagai berikut :

1. Sedarhana dan Ampuh

Java dirancang untuk mudah dipelajari, terutamah bagi pemrograman yang telah mengal C/C++ akan mudah sekali untuk berpindah ke Java. Pemakai dapat belajar membuat program Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. Java tidak memiliki hal-hal yang mengejutkan dan aneh. Java memberi anda kemampuan untuk menuangkan semua ide, kaerana bahasa pemrograman ini bukan merupakan scripting language (bahasa naskah) yang meghilangkan kemampuan kita untuk berinovasi, tetapi dengan cara berorientasi objek yang mudah dan jelas.

2. Aman

Java dirancang sebagai bahasa pemrograman yang handal dan aman. Aplikasi-apliksi yang dibangun dengan bahasa Java sangan handal dengan manajemen memori yang bagus. Aplikasi Java juga dikenal sangat secure, yaitu kasus-kasus seperti buffer everflow yang umumnya menjadi lubang leamanan aplikasi-aplikasi berbasis C/C++ tidak terjadi di Java, karena pengaturan securitynya bagus.

3. Berorientasi Objek

Peradigma pemrograman berorientasi objek merupakan peradigma objek. Java bukan turuanan langsung dair bahasa pemrograman berorientasi manapun, juga Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. Java tidak memiliki hal-hal yang mengejutkan dan aneh. Java memeberi anda kemampuan untuk menuangkan semua ide, karena bahasa pemrograman menyediakan makanisme peng-class-an sederhana, dengan model antar muka dinamik yang inuitif hanya jika diperlukan.

4. Kokoh

Java membatasi anda dari bebrapa hal kunci supya anda dapat menemukan kesalahan lebih cepat saat mengembangkan program. Java langsung memmeriksa prigram saat anda menulisnya,dan sekali lagi ktika program di jalankan. Karena Java adalah bahasa yang sangat ketat dalam hal tipe data dan deklarasi, banyak

kesalahan umum terjadi saat kompilasi. Hal ini akan lebih menghemat waktu jika di bandingkan dengan keharusan menjalankan program ketidakcocokan dinamis selam program berjalan. Ini adalah contoh di mana java lebih luwes dan kokoh dari bebrapa bahasa lain, tetapi dengan imbalan yang layak untuk kelebihan itu.

5. Interaktif

Java memiliki beberapa kemampuan yang memungkinkan program melalukan beberapa hal pada saat bersamaan. Tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya. Jalinan program-program Java yang mudah digunakan memumgkinkan ita untuk memikirkan pembuatan perilaku khusus, tanpa haruus mengintgrasikan prilaku trsebut dengan model pemograman global yang mengatur perulangan kejadian.

6. Netral terhadap berbagai Arsitektur

Java telah mengambil beberapa kputusan yang sulit dalam pembuatan bahasa Java dan bagaimana program dijalankan, jadi anda dapat spenuhnya percaya “tulisan sekali, jalan di mana saja,kapan saja,selamanya”.

7. Terintrepentasi dan Berkinerja Tinggi

Java dilengkapi keajaiban lintas flatform yang luar biasa dengan kompilasi ke dalam representasi langsung yang disebut kode-byte Java (Java byte-code), yang dapat diterjemahkan oleh sistem manapun yang memiliki program CPU yang tidak terlalu kuat. Walaupun Java merupakan bahsa terinterprestasi, kode-kode Java didalamnya. Java bagaimanapun dirancang dengan hati-hati sehingga mudah diterjemahkan ke dalam bahasa asli suatu mesin untuk meghasilkan kinerja yang tinggi.

Sistem program Java yang melakukan optimasi tepat waktu tersebut tidak kehilangan keuntungan dari program yang netral terhadap platfrom. “lintas platform berkinerja tinggi” bukan sekedar omong kosong. Dalam aplikasi java (*.class) merupakan Java bytecode yang berjalan di atas jvm (Java Virtual Machine), yang kemudian jvm-lah yang akan menginterpresentasikan kode-kode tersebut ke kode native atau kode mesin dari arsitektur mesin bukan jadi masalah bagi programmer tapi menjdi urusan kompiler pada bahsa pemrograman Java.

II.1.18 Skala Likert

Skala likert diciptakan oleh R.Likert untuk pengukuran sikap terhadap sesuatu. Skala likert digunakan untuk mengukur sikap, pendapat, dan presepsi sesorang atau kelompok orang tentang fenomena social. Seperti menentukan lokasi kedudukan seseorang dalam suatu kontinium sikap terhadap objek sikap, mulai dari sangat negative sampai dengan positif. Untuk setiap satu pernyataan responden memilih salah satu tanggapan berupa SS = Sangat Setuju, S = Setuju, R = Ragi, TS = Tidak Setuju, STS = Sangat Tidak Setuju. Tanggapan responden dikodekan kedalam bilangan 1 sampai dengan 5. Pemberian bilangan ini bergantung kepada bentuk pernyataan yang dijawab oleh responden[30].

Ada dua macam bentuk pernyataan dalam skala likert, pertama adalah pernyataan yang diharapkan untuk disetujui oleh responden, disebut bentuk positif. Yang kedua adalah pernyataan yang diharapkan untuk tidak disetujui oleh responden, bentuk negative. Biasanya kuesioner berisikan pernyataan positif dan negative agar responden berfikir dulu sebelum memberi jawaban.

Contoh kuesioner sikap maysarakat menggunakan skala Likert mengenai aplikasi mobile android yang akan dibangun.

A = Sangat Setuju,

B = Setuju,

R = Ragu,

D = Tidak Setuju,

E = Sangat Tidak Setuju

Pernyataan :

1. a. Perangkat lunak ini dapat mempermudah anda dalam mengetahui informasi sejarah wisata yang sedang dikunjungi?

   b. Perangkat lunak ini dapat menjadi referensi menjadi tour guide dalam perjalanan wisata anda

   Dari contoh kuesioner diatas didapat skor untuk skala Likert sebagai berikut:

Tabel II. 6 Skala Likert

Pilihan Jawaban	Pernyataan	
	Positif	Negatif
Sangat setuju	5	1
Setuju	4	2
Ragu	3	3
Tidak Setuju	2	4
Sangat Tidak Setuju	1	5

Pada bentuk positif sangat setuju memperoleh sekor tinggi dan sangat tidak setuju memperoleh sekor rendah. Pada bentuk negatif, sangat setuju memperoleh sekor terendah dan sangat tidak setuju memperoleh sekor tinggi.

II.1.19 Haversine

Metode Haversine digunkakan untuk meghitung jarak antara titik di permukaan bumi menggunakan garis lintang (longlitude) dan garis bujur (latitude) sebagai variabel inputn. Haversine dormula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang[31].

Dengan mengasumsikan bahwa bumi bentuk bulat sempurna dengan jari-jari R 6.367,45 km, dan loksi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah lon1,lat1, dan lon2, lat2 , maka perumusan Harvesine dapat dituliskan dengan persamaan sebagai berikut :

```
x = (lon2-lon1) * cos ((lat1+lat2)/2);
y= (lat2-lat1);
d= sqrt(x*x+y*y)*R
```

Keterangan :

x = Longitude (Lintang)

y = Latitude (Bujur)]

d = Jarak

R = Radius Bumi = 6371 km

1 derajat = 0.0174532925 radian

Tabel II. 7 Contoh Perhitungan Harversine Formula

Lokasi 1: lon1= 119.800801,lat1= -0.790175
Lokasi 2: lon2=119.8428 , lat2= -0.8989
Lat1 =-0.790175*0.0174532925 radian = -0.013791155 radian
Lon =119.800801 * 0.0174532925 radian = 2.090918422 radian
Lat2 =-0.8989 * 0.0174532925 radian = -0.011569 radian
Lon2 =119.8428 * 0.0174532925 radian = 2.091651 radian
x =(lon2-lat1) * cos ((lat1+lat2)/2)
=(2.091651-2.090918422) * cos ((-0.013791155+-0.01569)/2)
=0.0007329412
y =(lat2-lat1)
=(-0.01569-(-0.013791155))
=-0.001897609
d =sqrt (x*x + y*y) * R
=sqrt((0.0007329412 * 0.0007329412) + (-0.001897609 * -0.001897609))*6371
=sqrt (0.00000041381) * 6371
=12.96012927km