BAB II
LANDASAN TEORI

II.1 Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas Perangkat lunak computer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna atau *user*. Biasanya perangkat lunak sistem dapat mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media dan lain sebagainya.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi *(application suite).* Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

II.2 Kendaraan

Kendaraan bermotor adalah kendaraan yang digerakkan oleh peralatan teknik atau mesin untuk pergerakkannya, dan digunakan untuk transportasi darat. Umumnya kendaraan bermotor digunakan mesin pembakaran dalam, namun motor listrik dan mesin jenis lain (misalnya kendaraan listrik hibrida dan hibrida plug-in) juga dapat digunakan. Kendaraan bermotor memiliki roda, dan biasanya berjalan di atas jalanan. Jenis-jenis kendaraan bermotor dapat bermacam-macam:

1. Mobil

Mobil adalah kendaraan darat yang digerakkan oleh tenaga mesin, beroda empat atau lebih (selalu genap), biasanya menggunakan bahan bakar minyak (bensin atau solar) untuk menghidupkan mesinnya.

2. Motor

Sepeda motor adalah kendaraan beroda dua yang digerakkan oleh sebuah mesin. Letak kedua roda sebaris lurus dan pada kecepatan tinggi sepeda motor tetap stabil disebabkan oleh gaya giroskopik. Sedangkan pada kecepatan rendah, kestabilan atau keseimbangan sepeda motor bergantung kepada pengaturan setang oleh pengendara. Penggunaan sepeda motor di Indonesia sangat populer karena harganya yang relatif murah, terjangkau untuk sebagian besar kalangan dan penggunaan bahan bakarnya serta serta biaya operasionalnya cukup hemat.

3. Bus

Bus adalah kendaraan besar beroda, digunakan untuk membawa penumpang dalam jumlah banyak. Klasifikasi kendaraan bermotor ini bervariasi tergantung masing-masing negara. ISO 3833:1977 adalah standar untuk tipe dan definisi kendaraan darat.

Berdasarkan UU No. 14 tahun 1992 yang dimaksud dengan peralatan teknik dapat berupa motor atau peralatan lainnya yang berfungsi untuk mengubah suatu sumber daya energi tertentu menjadi tenaga gerak kendaraan bermotor yang bersangkutan. Pengertian kata berada dalam ketentuan ini adalah terpasang pada tempat sesuai dengan fungsinya. Termasuk dalam pengertian kendaraan bermotor adalah kereta gandengan atau kereta tempelan yang dirangkaikan dengan kendaraan bermotor sebagai penariknya.

Sampai tahun 2010, ada lebih dari 1 milyar kendaraan bermotor di seluruh dunia, tidak termasuk kendaraan *off-road* dan kendaraan berat. Kepemilikan

kendaraan per kapita global adalah 148 kendaraan beroperasi tiap 1000 orang. Amerika Serikat adalah negara yang memiliki jumlah kendaraan bermotor terbanyak di dunia, dengan 239,8 juta kendaraan tahun 2010. Kepemilikan kendaraan per kapita di Amerika Serikat juga tertinggi di dunia, yaitu 769 kendaraan per 1000 penduduk. Republik Rakyat Tiongkok mempunyai jumlah kendaraan terbanyak kedua di dunia, dengan jumlah 78 juta unit dan sejak 2009 juga menjadi pasar kendaraan terbesar di dunia. Pada tahun 2011, 80 juta mobil dan kendaraan komersial diproduksi di seluruh dunia, 18.4 juta unit diantaranya diproduksi di Cina. Dan akan terus bertambah setiap tahunnya.

II.3 Mobil Derek

Mobil derek adalah kendaraan yang digunakan untuk menderek kendaraan yang mengalami kecelakaan lalu-lintas sehingga tidak mampu untuk berjalan sendiri lagi atau terbalik, menderek kendaraan yang sedang mogok dan mengganggu kelancaran lalu lintas ataupun untuk menderek kendaraan yang melakukan pelanggaran parkir [3].

Ada beberapa jenis mobil derek yang biasa digunakan:

1. Boom yaitu derek yang menggunakan boom/kran yang biasanya digunakan untuk mengangkat mobil yang terperosok kedalam parit, ataupun untuk membalikkan kendaraan yang terbalik [3].
2. Kait dan Tarik, digunakan untuk menarik kendaraan dengan rantai atau tali yang dikaitkan ke mobil derek, kendaraan yang di Tarik di kemudikan oleh pengemudi [3].
3. Roda diangkat yaitu penderekan dengan mengangkat roda depan atau belakang kendaraan kemudian ditarik, hal ini biasanya dilakukan pada kendaraan yang salah parkir, dimana kendaraan tersebut rem parkirnya dijalankan sehingga roda yang terkunci diangkat [3].
4. Derek gendong, dimana kendaraan yang diderek digendong keatas kendaraan truk, biasanya digunakan untuk mengangkut mobil mewah dari show room ke

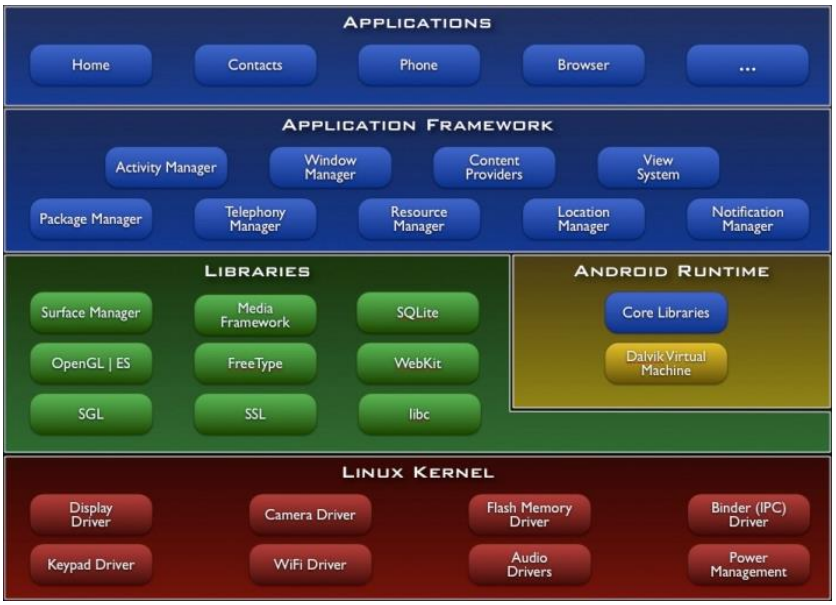
rumah pembeli atau untuk mengangkat mobil mewah yang rusak/mogok ke bengkel [3].

II.4 Android

Android adalah sistem operasi untuk telepon seluler yang berbasiskan Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam peranti penggerak [4].

II.4.1 Arsitektur Android

Diagram Arsitektur Anroid dapat dilihat pada gambar II.1:



Gambar II-1 Arsitektur Android

Berikut Merupakan garis besar pada arsitektur yang di miliki android:

1. **Application dan Widget**

Android akan mengirimkan serangkaian aplikasi initi, termasuk *e-mail*, *SMS*, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi dibuat dengan menggunakan bahasa pemrograman *Java.*

2. Applications Framework

Aristektur aplikasi ini dirancang untuk menyederhanakan penggunaan ulang komponen. Aplikasi apapun dapat mem-*publish kapabilitas*-nya dan kemudian aplikasi lain dapat menggunakannya.

Yang mendasari aplikasi-aplikasi tersebut adalah seperangkat layanan dan sistem, yaitu:

a. Satu set tampilan untuk membangun aplikasi, termasuk *list, grid, text box, button* dan bahkan *browser*.
b. *Content Providers* yang menyediakan aplikasi untuk mengakses data dari aplikasi lain (seperti Kontak), atau untuk *data sharing*.
c. *Resource Manager* menyediakan akses ke *resource non-code* seperti *grafik* dan *file layout*.
d. *Notification Manager* yang memungkinkan semua aplikasi untuk menampilkan *custom alert* pada *status bar*.
e. *Activity Manager* yang mengelola *live cycle* dari aplikasi dan menyediakan *navigasi backstack*.

3. Libraries

Android memuat seperangkat *C / C++ libraries* yang digunakan oleh berbagai komponen Android. Beberapa *libraries core*-nya adalah:

a. *System C library*, implementasi turunan *BSD* dari *system C library standar (libc)* yang disesuaikan untuk perangkat berbasis Linux.
b. *Media Libraries*, berbasis *Packet Video's OpenCORE*. Support pada library playback, rekaman video dan audio, serta file gambar statis, termasuk *MPEG4*, *MP3*, *AMR*, *JPG* dan *PNG*.
c. *Surface Manager*, mengelola akses ke *tampilan subsistem*.
d. *LibWebCore*, *modern web browser engine* yang terintegrasi pada Android dan tampilan *web browser*.
e. *SGL*, *engine* grafis *2D*.

f. *3D libraries*, implementasi berbasis *OpenGL 1.0 API*, *library* menggunakan *hardware 3D*.
g. *FreeType*, *bitmap* dan *vektor font rendering*.
h. *SQLite*, *database relasional engine* yang handal dan ringan tersedia untuk semua aplikasi.

4. Android Run Time

Setiap aplikasi Android berjalan dalam prosesnya sendiri, contohnya *Dalvik Virtual Mesin*. *Dalvik VM* mengeksekisi file dalam format *.dex* yang dioptimalkan untuk meminimalisir kapasitas memori. *VM* ini di *compile* oleh *Java Compiler* dan ditransformasikan dalam format *.dex* oleh *dx tool*.

5. Linux Kernel

Android bergatung pada Linux versi 2.6 untuk layanan sistem core seperti keamanan, manajemen memori, manajemen proses, *network stack* dan *driver model*. *Kernel* juga bertindak sebagai *layer abstraksi* antara *hardware* dan seluruh *software stack*.

II.4.2 Jenis Komponen pada Aplikasi Android

Android memiliki 4 komponen yang memiliki peran yang berbeda-beda dan berdiri sendiri meskipun terkadang salah satu komponen bergantung pada komponenen lainnya. Masing-masing komponen memiliki daur hidup sendiri-sendiri yang mengatur kapan komponen dibuat dan bagaimana komponen dihapus. Berikut ini adalah 4 tipe komponen dalam aplikasi android [4]:

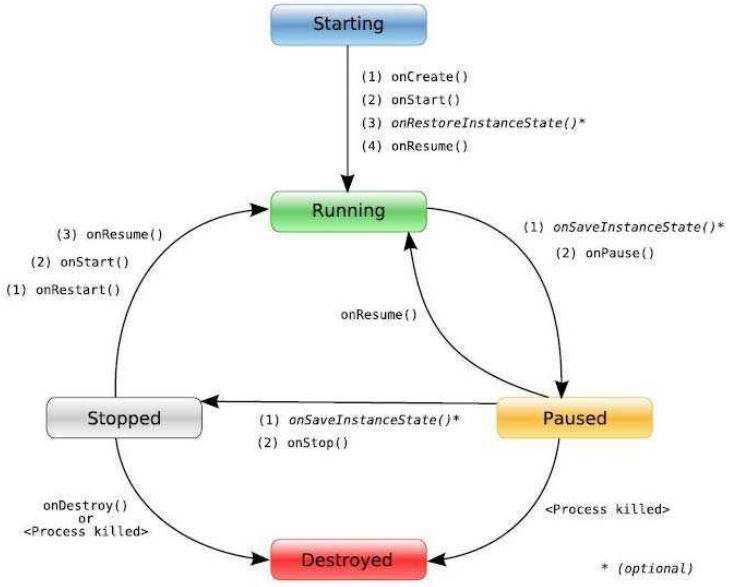
1. Activity

Aplikasi yang kita buat bisa memiliki satu activity, atau juga banyak activity. Setiap activity yang didefinisikan memiliki daur hidup tersendiri, jika saat ini

activity sedang running, mungkin selanjutnya aplikasi kita berhenti sebentar (pause) atau memang benar-benar dimusnahkan, hilang dari memori (destroy) [4].

Kita melakukan pengaturan tindakan saat activity berada di state tertentu dengan cara melakukan override terhadap method di dalam class Activity. Pada waktunya, Android akan memanggil method yang bersesuaian dengan state activity. Sebagai contoh, sebuah aplikasi email mungkin memiliki satu activity untuk menunjukan email baru, satu lagi activity untuk menulis surat, dan satu lagi activity untuk membaca email. Meskipun activity-activity bekerja bersamaan untuk membentuk sebuah aplikasi email yang bagus, setiap activity tidak bergantung kepada yang lainnya. Misalnya aplikasi lain bisa menggunakan satu dari beberapa activity tersebut jika aplikasi email memperbolehkannya [4].

Beberapa tahapan (state) yang terjadi dalam class Activity bisa dilihat gambar berikut ini :



Gambar II-2 Class Activity

Penjelasan state method pada class Activity :

1. *onCreate()*, method ini dipanggil ketika aktifitas pertama kali dimulai. Kita dapat menggunakan method ini untuk melakukan inisialiasi awal seperti menentukan antar muka (user interface), method *onCreate()* yang memiliki satu

parameter yang dapat bernilai null atau bernilai informasi dari state sebelumnya yang dijalankan oleh method *onSaveInstanceState().*

2. *onStart(),* method ini mengindikasikan aktifitas yang siap ditampilkan ke user.
3. *onResume(),* method ini dipanggil ketika aktifitas anda dinyatakan siap berinteraksi dengan user. Ini adalah aktifitas yang paling tepat untuk menyertakan animasi dan music pada aplikasi.
4. *onPause(),* method ini dijalankan ketika aktifitas akan dijalankan di background, biasanya terjadi ketika aktifitas berjalan di atas aktifitas ini. Pada state ini sebaiknya anda implementasi persistent data pada program anda, seperti ketika sebuah database sedang dilakukan proses edit.

5. *onStop(),* method ini dipanggil saat aktifitas anda tidak bisa dilihat lagi oleh user dan tidak dibutuhkan untuk sementara waktu. Jika memori dalam keadaan parah (karena banyak resource berjalan) maka method *onStop()* tidak pernah akan dipanggil (system mungkin akan mematikan prosesnya)
6. *onRestart(),* jika method ini dipanggil, berarti aktifitas anda sedang di tampilkan ulang (redisplay) oleh user dari sebelumnya distop.
7. *onDestroy(),* method ini dipanggil sebelum sebuah aktifitas di matikan (didestroy). Jika resource memory ketat method *onDestroty()* tidak pernah akan dipanggil (system mungkin akan mematikan prosesnya).
8. *onStateInstanceState()*
9. *onRestoreInstanceState()*

2. Service

Service adalah komponen yang berjalan dibalik layar. Sebuah service tidak memiliki user interface. Sebagai contoh, sebuah service bisa memainkan musik, sementara user sedang menjalankan aplikasi lain. Atau service juga bisa mengirimkan data melalui internet tanpa harus menghentikan interaksi user dengan sebuah activity. Komponen lain, misalnya activity bisa memulai service dan menjalankannya atau terikat ke service tersebut untuk berinteraksi dengan service tersebut [4].

3. Content Provider

Sebuah content provider mengatur sekumpulan data aplikasi yang terbagi (shared). Kita bisa menyimpan data di filesystem, sebuah database SQLite, di web, atau di metode penyimpanan data lainnya yang bisa diakses oleh aplikasi kita. Melalui content provider, aplikasi lain bisa memberikan query atau bahkan bisa memodifikasi, tentunya jika content provider mengijinkan aksesnya. Sebagai contoh, sistem android menyediakan content provider yang mengatur informasi kontak user. Misalnya, aplikasi apa saja yang memiliki ijin bisa memberikan query kepada sebagian data untuk membaca dan menulis informasi tentang orang tertentu. Content provider juga bisa digunakan untuk menulis dan membaca data pribadi yang tidak dibagikan (tidak di shared) [4].

4. Broadcast Receivers

Broadcast receiver adalah komponen yang merespon terhadap siaran (broadcast) pengumuman yang dikeluarkan oleh sistem. Banyak siaran broadcast yang aslinya berasal dari sistem. Misalnya broadcast yang memberitahukan bahwa layar sudah mati, battery lemah, atau gambar sudah diambil. Aplikasi juga bisa memulai broadcast, misalnya memberitahukan aplikasi lain bahwa beberapa data sudah selesai didwonload dan bisa digunakan. Meskipun broadcast receiver tidak menampilkan user interface, broadcast receiver bisa membuat notifikasi di status bar untuk memberitahukan user sedang terjadi broadcast. Secara umum, broadcast receiver hanyalah sebuah "gerbang" kepada komponen lain dan ditujukan untuk melakukan perkerjaan yang sangat minimal [4].

Tiga dari empat tipe komponen (activity, service dan broadcast receiver) diaktifkan oleh sebuah pesan tidak sinkron yang disebut ***intent***. Intent mengikat beberapa komponen-komponen indiviual menjadi satu pada saat runtime. Sebuah intent dibuat dengan objek *Intent*, yang mendefinisikan sebuah pesan untuk mengaktifkan sebuah komponen khusus atau sebuah tipe komponen (sebuah intent bisa Merupakan komponen implisit atau eksplisit). Untuk activity dan service,

sebuah intent mendefinisikan aksi apa yang harus dilakukan oleh keduanya, misalnya untuk melihat atau mengirim sesuatu [4].

Bagi broadcast receiver, intent hanyalah mendefinisikan pengumuman yang dibroadcast. Misalnya sebuah broadcast untuk mengindikasikan batery perangkat sudah lemah hanya terdiri dari sebuah action string yang bernilai "Battery lemah". Komponen tipe lain, content provider, tidak diaktifkan oleh intent. Akan tetapi, content provider diaktifkan ketika menjadi target permintaan dari sebuah content resolver. Content resolver meng-handle semua transaksi langsung dengan content provider sehingga komponen yang ingin bertransaksi dengan content provider cukup memanggil method di objek content resolver [4].

II.4.3 Versi android

Adapun versi-versi android yang sudah dirilis adalah sebagai berikut :

1. Android 1.1

Pada 9 Maret 2009, *Google* merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android 1.5 (Cupcake)

Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan Android dan *SDK (Software Development Kit)* dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke *Picasa* langsung dari telepon, *Bluetooth A2DP support*, kemampuan terhubung secara otomatis ke headset *Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. Android 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang dintegrasikan, Gestures, dan Text-to-speech engine, kemampuan dial kontak, teknologi text to change speech dan juga pada pengetikan kata yang lebih mudah.

4. Android 2.0/2.1(Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair). Perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan *Google Maps 3.1.2*, perubahan *UI* dengan *browser* baru dan dukungan *HTML5*, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan *Bluetooth 2.1*.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset* Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah *Shazam*, *Backgrounds*, dan *WeatherBug*. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh *MySpace* dan *Facebook*.

5. Android 2.2 (froyo : Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card,

kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. Android 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (*User Interface*) didesain ulang, dukungan format *video VP8* dan *WebM*, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication (NFC)*, dan dukungan jumlah kamera yang lebih dari satu.

7. Android 3.0 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk *tablet*. Honeycomb juga mendukung *multi prosesor* dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah *Motorola Xoom*.

8. Android 4.0 (IceCreamSandwich/ICS)

Android pertama yang mempunyai fitur baru membuka kunci dengan pengenalan wajah. Fitur ini belum dimiliki oleh Android versi dibawah Ice Cream Sandwich tetapi sudah disempurnakan pada versi diatasnya. Selain itu Ice Cream Sandwitch juga mempunyai penampilan Interface yang bersih dan smooth.

Ice Cream Sandwitch juga mempunyai font yang bernama Roboto. Jika digunakan font ini terlihat sangat simpel namun elegan, tampilan yang diciptakan oleh font Roboto sungguh terkesan unik.

9. Android 4.1 (JellyBean)

Jelly Bean-Android versi 4.1 yang diluncurkan pada acara Google I/O membawa fitur fitur baru yang menawan, beberapa fitur yang diperbaharui dalam sistem operasi ini antara lain, pencarian dengan menggunakan Voice Search yang lebih cepat, informasi cuaca, lalu lintas, hasil pertandingan olahraga yang cepat dan tepat, selain itu versi 4.1 ini juga mempunyai fitur keyboard virtual yang lebih baik. Permasalahan umum yang sering ditemui pengguna Android adalah baterai, namun Baterai dalam sistem Android Jelly Bean versi 4.1 ini diklaim cukup hemat.

10. Android 4.4 (Kitkat)

Android 4,4 KitKat sendiri merupakan versi terbaru dari seri sebelumnya yaitu 4,3 Jelly Bean dan sudah diperkenalkan pada umum pada bulan September 2013. Android KitKat merupakan penyermpurnaan dari versi sebelumnya.

1. Penyempurnaan yaitu mampu mengoptimasi memori dan memperbaiki respon layar sentuh agar lebih cepat dan akurat di bandingkan sebelumnya.
2. Fitur multi-tasking dimana dapat membuka 2 aplikasi secara bersamaan.
3. Pada aplikasi baru secara otomatis akan memprioritaskan nomor kontak yang sering dihubungi. Fitur pencarian berdasarkan lokasi terdekat, bisnis, kontak dan orang-orang pada aplikasi Google.
4. Pencocokan daftar nomor yang tidak ada dengan daftar nomor di peta Google secara otomatis.
5. Tampilan karakter Bahasa jepang lebih nyaman terlihat dalam layar handphone.

II.5 Tool pengembangan aplikasi android

Adapun beberapa tool di pakai dalam pembangunan perangkat lunak ini adalah sebagai berikut:

II.5.1 Android SDK (Software Development Kid)

Sesuai dengan namanya merupakan kumpulan tools-tools yang di perlukan untuk mengembangkan aplikasi Android. Seperti pada java yang dapat dijalankan pada multiplatform, SDK Android pun tersedia untuk beberapa platform seperti Windows, Mac OS dan Linux. [5]

Adapun aplikasi SDK yang didalamnya terdapat beberapa folder sebagai berikut :

1. Tools

Berisikan tools-tools yang diperlukan dalam membangun aplikasi Android seperti emulator, Android Virtual Device (AVD), SDK Manager, ddms dan banyak lagi untuk memudahkan membangun aplikasi Android. Semua tools ini dapat di update mengikuti setiap perkembangan platfom Android.

2. Platforms

Direktori ini ditujukkan untuk meletakkan versi platform-platform Android yang telah kita download dari SDK Manager. Seperti Android 1.5, 1.6, 2.0 dan seterusnya.

3. Add-ons

Direktori ini nantinya berisikan tools tambahan untuk Android SDK, yang fungsinya untuk menambahkan library yang akan digunakan untuk pengembangan aplikasi. Seperti jika ingin mengembangkan aplikasi menggunakan map dari google maka harus memerlukan platform Android yang sudah terdapat library map atau misal jika ingin mengembangkan aplikasi untuk galaxy tab atau Sony X-peria juga memerlukan library dari masing-masing platform tersebut. Semua ini dapat di download melalui SDK Manager.

Selain folder-folder diatas, akan terdapat file tambahan jika kita mendownload beberapa platfom dan tool dalam folder Android SDK tersebut. Seperti ***Docs, paltform tools, dan samples*.**

II.5.2 Android Development Tools (ADT)

Merupakan plugin Eclipse yang disediakan oleh google untuk membantu dan memudahkan Developer mengembangkan aplikasi Android. Dengan adanya ADT membuat project Android, mendesain tampilan aplikasi, menambahkan komponen-komponen yang berasal dari Android Framework, debug aplikasi menggunakan tools-tools pada SDK Android hingga membuat file installer menjadi lebih mudah berkat plugin ini. Untuk proses penginstallan ADT dapat dilakukan dengan 2 cara Pertama secara Offline dengan mendownload file ADT secara manual pada situs Android dan kedua Secara online dengan menggunakan eclipse maupun Android studio. [5]

II.6 Tinjauan Perangkat Keras/Lunak Pendukung

Perangkat pendukung yang penulis pergunakan adalah Chrome, android studio, Sublime text 3.google Maps dan GPS.

II.6.1 Chrome Browser

Chrome adalah peramban web yang cepat, sederhana, dan aman, khusus dibuat untuk web modern [6]. Keunggulan yang ada pada google chrome adalah.

1. Kecepatan

Chrome dirancang agar bekerja secepat mungkin. Dimulai dengan cepat dari desktop Anda, memuat laman web dalam sekejap, dan menjalankan aplikasi web yang rumit secepat kilat [6].

2. Kesederhanaan

Jendela peramban Chrome efektif, bersih, dan sederhana. Misalnya, Anda dapat menelusuri dan bernavigasi dari kotak yang sama dan mengatur tab sesuka Anda -- dengan cepat dan mudah [6].

3. Keamanan

Chrome dirancang untuk menjaga Anda agar lebih aman dan terjamin di web dengan perlindungan program jahat dan phishing yang sudah ada di dalamnya, pembaruan otomatis untuk memastikan bahwa Anda memiliki semua pemutakhiran keamanan terbaru, dan yang lainnya [6].

4. Privasi

Chrome memberikan Anda kontrol atas informasi pribadi sambil membantu melindungi informasi yang Anda bagikan saat online [6].

5. Pengubahsuaian

Ada berbagai macam cara untuk menyesuaikan Chrome dan menjadikannya milik Anda. Mudah untuk menyesuaikan setelan Anda dan menambahkan aplikasi, ekstensi, serta tema dari Toko Web Chrome [6].

6. Masuk login

Dengan masuk ke Chrome, maka bookmark, riwayat, dan setelan lainnya akan muncul di semua komputer Anda. Anda otomatis masuk ke semua layanan Google favorit juga [6].

II.6.2 Android Studio

Android Studio adalah lingkungan pengembangan Android baru berdasarkan IntelliJ IDEA. Ini menyediakan fitur baru dan perbaikan atas Eclipse ADT dan akan menjadi IDE Android resmi setelah siap. Di atas kemampuan yang Anda harapkan dari IntelliJ [7], Android Studio menawarkan:

1. Fleksibel Gradle berbasis sistem membangun.
2. Membangun varian dan generasi APK beberapa.
3. Diperluas dukungan template untuk Layanan Google dan berbagai jenis perangkat.
4. Layout editor kaya dengan dukungan untuk mengedit tema.
5. Lint Tool untuk menangkap kinerja, kegunaan, versi kompatibilitas, dan masalah lainnya.
6. ProGuard dan aplikasi-penandatanganan kemampuan.
7. Built-in mendukung untuk Google Cloud Platform, sehingga mudah untuk mengintegrasikan Google Cloud Messaging dan App Engine.

II.6.3 Google Map

Google Maps adalah sebuah jasa peta globe virtual gratis dan *online* disediakan oleh Google dapat ditemukan di http://maps.google.com. Teknologi tersebut menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia.

Google Maps merupakan Peta Online atau Membuka peta secara online. kini dapat dilakukan secara mudah melalui servis gratis dari Google ini. Bahkan, servis ini menyediakan API (Application Programming Interface) yang memungkinkan developer lain untuk memanfaatkan aplikasi ini di aplikasi buatannya. Tampilan GoogleMaps pun dapat dipilih, berdasarkan foto asli atau peta gambar rute saja.

II.6.4 Sublime Text 3

Teks luhur adalah editor teks yang canggih untuk kode, markup dan prosa. UI friendly, fitur luar biasa dan kinerja yang luar biasa.

II.6.5 GPS

GPS adalah singkatan dari Global Positioning System yang Merupakan sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit. Sistem yang pertama kali dikembangkan oleh departemen pertahanan amerika ini digunakan untuk kepentingan militer maupun sipil (survey dan pemetaan).

Sistem GPS, yang nama aslinya adalah NAVSTAR GPS (Navigation Satellite Timing and Ranging Global Positioning System), mempunyai tiga segmen yaitu:

1. Satellite

Bertugas untuk menerima dan menyimpan data yan ditransmisikan oleh stasiun-stasiun pengontrol, menyimpan dan menjaga informasi waktu ketelitian tinggi (ditentukan dengan jam atomic di satelit), dan memancarkan sinyal dan informasi secara kontinyu ke pesawat penerima (receiver)dari pengguna.

2. Pengontrol

Bertugas untuk mengendalikan dan mengontrol satelit dari bumi baik untuk mengecek kesehatan satelit, penentuan dan prediksi orbit dan waktu, sinkronisasi waktu antar satelit, dan mengirim data ke satelit.

3. Penerima

Bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi (posisi tiga dimensi yaitu koordinat di bumi plus ketinggian), arah jarak dan waktu yang diperlukan oleh pengguna. Ada dua macam tipe penerima yaitu tipe NAVIGASI dan tipe GEODETIC.

Satelit GPS yang mengorbit bumi, dengan orbit dan kedudukan yang tetap (koordinatnya pasti), seluruhnya berjumlah 24 buah dimana 21 buah aktip bekerja dan 3 buah sisanya adalah cadangan.

Dalam bidang survey dan pemetaan untuk wilayah terumbu karang, GPS dapat digunakan untuk menentukan posisi titik-titik lokasi penyelaman maupun transek. Posisi yang diperoleh adalah posisi yang benar terhadap sistem koodinat bumi. Dengan mengetahui posisinya yang pasti, lokasi-lokasi penyelaman maupun transek dapat diplotkan kedalam peta kerja.

II.6.6 Web Services

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:

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 [8].

II.6.7 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan

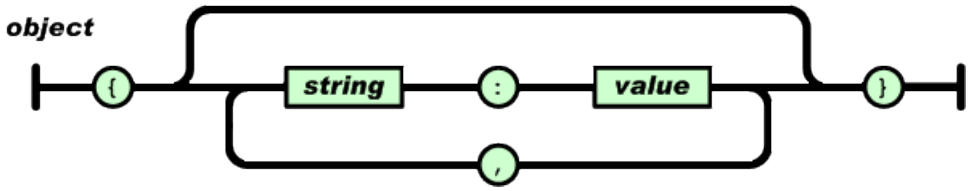
dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga 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:

a. 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 [9].
b. Daftar nilai terurutkan (an ordered list of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (array), vektor (vector), daftar (list), atau urutan (sequence) [9].

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:

1. Objek

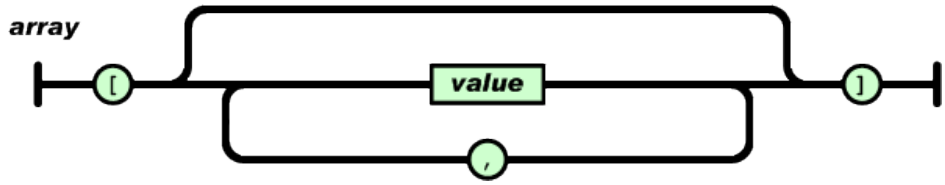
Objek adalah sepasang nama / nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma). Objek biasanya digunakan untuk menyimpan data tunggal dalam bentuk JSON [9].



Gambar II-3 Object JSON

2. Larik

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak 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 [9].

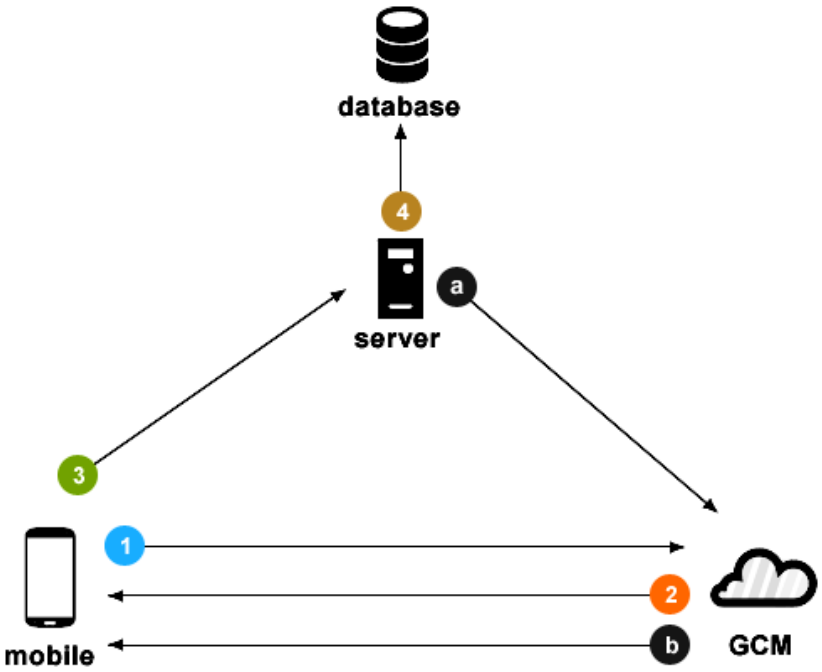


Gambar II-4 Array JSON

Bentuk data JSON objek dan larik dapat saling dikombinasikan untuk mendukung struktur data yang lebih kompleks. JSON mendukung beberapa tipe data untuk menjadi value seperti Angka, String, Bollean dan nilai NULL [9].

II.6.8 Google Cloud Messaging

Google Cloud Messaging adalah teknologi untuk membantu pengembang mengirim data dari server menuju device android secara langsung melalui server google, pemanfaatan teknologi ini biasanya digunakan untuk *push messaging.* Pengiriman pesan menggunakan teknologi ini mengharuskan adanya API key, key ini terkait langsung dengan akun google yang digunakan untuk mempublikasikan aplikasi melalui google play [5].



Gambar II-5 GCM Model *[5]*

Berdasarkan ilustrasi dari Gambar II-5 GCM Model terdapat dua server yang terlibat dalam proses *push messaging* , pada setiap server yang terlibat terdapat proses pengiriman data yang berbeda-beda diantaranya [5]:

1. Device android mengirimkan sender id dan application id menuju GCM Server untuk registrasi.
2. Setelah berhasil registrasi, GCM server akan mengirimkan registration id menuju device android.
3. Setelah device android mendapatkan registration id, d
4. evice tersebut akan mengirimkan registration id menuju server aplikasi.Server aplikasi akan menyimpan registrasi id yang dikirimkan untuk digunakan sebagai id saat menggunakan *push messaging*.

II.7 Bahasa Pengrograman yang digunakan

Adapun beberapa bahasa pemrograman yang digunakan pada pembangunan perangkat lunak ini adalah sebagai berikut:

II.7.1 PHP

PHP adalah salah satu bahasa *Server-side* yang didesain khusus untuk aplikasi web. PHP dapat disisipkan diantara bahasa HTML dan karena bahasa *Server side*, maka bahasa PHP akan dieksekusi di server, sehingga dikirimkan ke browser adalah "hasil jadi" dalam bentuk HTML, dan kode PHP anda tidak akan terlihat [10].

PHP termasuk dalam *Open Source Product*. Jadi anda dapat merubah *source code* dan mendistribusikan secara bebas. Adapun kelebihan-kelebihan dari PHP yaitu:

1. PHP mudah dibuat dan kecepatan akses tinggi.
2. PHP dapat berjalan dalam web server yang berada dan dalam system operasi yang berbeda pula. PHP dapat berjalan disistem operasi UNIX, Windows98, Windows NT dan Macintosh.
3. PHP diterbitkan secara gratis.
4. PHP juga dapat berjalan pada web server Microsoft Personal Web Server, Apache, IIS, Xitami dan sebagainya.
5. PHP adalah termasuk bahasa yang embeddeb (bisa ditempel atau diletakkan dalam tag HTML).
6. PHP termasuk *server-side* programming.

Sistem database yang didukung oleh PHP seperti Oracle, Sybase, mSQL, MySQL, Solid, Generic ODBC, Postgres SQL [10].

II.7.2 JAVA

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James

Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan Bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin [11].

Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". 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 [11].

II.7.3 MySQL

Dalam bahasa SQL data akan diatur dalam baris dan kolom pada table, karena pada database di dalamnya pasti akan anda temukan beberapa table. Ada banyak server database yang bisa digunakan, namun salah satu server database yang popular dan banyak digunakan orang adalah MySQL. Selain karena bisa didapat dengan mudah dan gratis, MySQL juga dapat berjalan pada sistem operasi apa pun. Berikut beberapa kelebihan yang dimiliki MySQL [12] [13] [14]:

1. Program database MySQL memang merupakan pasangan setia yang sering diintegrasikan dengan Apache.
2. MySQL memiliki kemampuan server multithread sehingga dapat menangani banyak permintaan (*requet*) layanan secara bersamaan.
3. Dapat berjalan pada banyak sistem operasi (Linux, Windows, SunOS, MacOS).
4. Memiliki metode enkripsi yang baik.
5. Mampu menangani data yang besar dan banyak.

6. Menggunakan autentifikasi pengguna (*user*) dan kata sandi (*password*) sehingga menjadikannya lebih aman.

II.8 Object Oriented Analysis Design

suatu pendekatan rekayasa perangkat lunak dari sebuah sistem yang terdiri dari sekelompok objek yang saling berinteraksi, dan setiap objek itu mewakili beberapa entitas.

II.8.1 Pengertian OOP

OOP (Object Oriented Programming) adalah sebuah konsep pemograman yang berbasis objek, OOP merupakan terobosanbaru dalam pemrograman setelah pemrograman prosedural, pemrograman modular dan pemrograman abstraksi data [14].

OOP memberikan kemudahan bagi programmer dalam pembuatan sebuah program, memberikan fleksibilitas yang lebih, kemudahan penggunaan program, dan dipakai luas dalam teknik piranti lunak skala besar. Selain itu dengan pemrograman OOP kode yang dibuat akan lebih mudah dikembangkan dan dirawat dalam arti seorang programer bisa menambah sebuah objek baru tanpa harus menggangu dan merubah modul yang lain. Berikut ini keuntungan dari penggunaan OOP dalam pemrograman [14] :

1. **Reusability**

(kemampuan untuk digunakan kembali). Kode yang dibuat dapat digunakan kembali di aplikasi atau programlainnya.

2. **Extensibility**

(Kemampuan untuk dapat dikembangkan). Kita dapat membuat method yang baru atau mengubah yangsudah ada sesuai dengan yang kita inginkan tanpa harus membuat kode dari awal.

3. Maintainability

(Kemampuan untuk mudah dikelola), kode yang sudah dibuat lebih mudah untuk di maintain/manage. Apabila aplikasi yang dibuat adalah aplikasi dengan skala besar dan apabila terjadi error atau kesalahan maka dengan OOP hal tersebut mudah ditangani karena pemograman yang menggunakan OOP sudah modularitas (memecahkan program yang besar menjadi bagian-bagian kecil) sehingga perbaikan cukup pada modul-modul tertentu saja tanpa harus mengubah modul yang lainnya.

II.8.2 Konsep OOP

Dalam pemrograman OOP, ada empat hal yang diperhatikan karena hal tersebut merupakan konsep vital dari OOP tersebut [14].

Berikut ini konsep utama tersebut:

1. Encapsulation/pembungkusan, merupakan penyembunyian informasi melalui private dan protected
2. Inheritance /pewarisan, adalah kemampuan untuk membuat class turunan yang mewarisi property dan method dari kelas induk.3.
3. Abtraction/abtraksi, adalah sebuah konsep yang merupakan representasi abstrak dari konsep dalam pemograman,misalnya objek mobil merupakan representasi abstrak dari mobil dalam dunia nyata.
4. Polymorphism /banyak bentuk, seorang programer bisa memakai method dengan nama yang sama pada class-class yang berbeda

II.8.3 UML

UML (*Unified Modeling Language*) meruapakan pengganti dari metode analisis berorientasi object dan design berorientasi object (OOA & OOD) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an [14].

UML merupakan gabungan dari metode Grady Booch (Booch Method), James Rumubaugh (OMT) dan Ivar Jacobson (OOSE). Tetapi UML ini akan

mencakup lebih luas daripada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (Object Managemet Group) dengan harapan UML akan menjadi bahsa standar pemodelan masa yang akan datang [14].

UML disebut sebagai pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagai besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat [14].

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain. Ada beberapa model pada UML diantaranya adalah [14]:

1. Use Case Diagram

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 [14]:

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.

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 system.

2. 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 [14].

3. 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 [14].

II.9 Pengujian *Black Box*

Metode pengujian *blackbox* fokus pada keperluan penelusuran kesalahan fungsional dari software. Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

Teknik pengujian black box terdiri dari 10 jenis diantaranya *Equivalence Partitioning, Boundary Value Analysis/Limit Testing, Comparison Testing, Sample Testing, Robustness Testing, Behavior Testing, Requirement Testing, Performance Testing, Endurance Testing, Cause-Effect Relationship Testing.* Salah satunya yang akan dibahas adalah *Equivalence partitioning*.

Equivalence partioning merupakan metode ujicoba blackbox yang membagi domain input dari program menjadi beberapa kelas data dari kasus ujicoba yang dihasilkan. Kasus uji penanganan single yang ideal menemukan sejumlah kesalahan (misalnya : kesalahan pemrosesan dari seluruh data karakter) yang merupakan syarat lain dari suatu kasus yang dieksekusi sebelum kesalahan umum diamati.

II.10 Pengujian Kuisioner

Kuesioner adalah sebuah daftar pernyataan yang harus diisi oleh orang yang akan dievaluasi (responden). Metode yang digunakan dalam kuesioner pada penelitian ini adalah skala Likert. Dalam skala likert, responden diminta untuk membaca dengan seksama setiap pernyataan yang disajikan, kemudian ia diminta untuk menilai pernyataan-pernyataan tersebut.

Derajat penilaian responden terhadap suatu pernyataan terbagi dalam 5 kategori yang tersusun secara bertingkat, mulai dari Sangat Tidak Setuju (ST), Tidak Setuju (T), Tidak Memutuskan (N), Setuju (S), dan Sangat Setuju (SS). Atau dapat pula sebaliknya. Pernyataan tiap kuesioner dibuat berdasarkan aspek-aspek yang diteliti. Bobot pemberian skor yang digunakan dapat dilihat pada Tabel II-1 Panduan Pemberian Skor .

Tabel II-1 Panduan Pemberian Skor [15]

Jenis Pernyataan	Bobot pendapat				
	SS	S	N	T	TS
Positif	5	4	3	2	1
Negatif	1	2	3	4	5

Skor yang telah dihitung pada setiap pernyataan kemudian dikalikan dengan masing-masing bobot tersebut sesuai dengan skenario kuesioner yang telah dibuat. Setelah itu total kan seluruh bobot jawaban tersebut kemudian bagi dengan total responden yang nantinya menjadi nilai rata-rata. Nilai rata-rata inilah yang diambil sebagai acuan sikap dimana jika nilai rata-rata kurang dari 3, maka dapat diartikan responden bersikap negatif dan jika nilai rata-rata lebih dari sama dengan 3, maka dapat diartikan responden bersikap positif terhadap tujuan yang ingin peneliti capai. Untuk lebih jelasnya dapat dilihat pada rumus dibawah ini.

$$\overline{X} = \frac{\sum Total}{n}$$

Dimana,

$\overline{X} \geq 3\ bersikap\ positif$

$\overline{X} < 3\ bersikap\ negatif$

Keterangan :

$\overline{X}$ = rata-rata

$\sum Total = jumlah\ seluruh\ nilai\ setelah\ dikalika\ dengan\ bobot$

n = total responden

II.11 Metode yang di gunakan

Metode yang digunakan pada sistem yang akan dibangun sebagai berikut:

II.11.1 Rumus Haversine

Rumus Haversine adalah persamaan pada navigasi, yang memberikan jarak lingkaran besar antara dua titik bumi berdasarkan garis bujur (longitude) dan garis lintang (latitude). Perhitungan dari rumus ini juga cukup akurat dimana rumus ini mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi. Secara matematis dapat dituliskan didalam persamaan sebagai berikut pada gambar II - 6 [16] :

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Gambar II-6 Rumus Haversine

Analisis perhitungan jarak ini menggunakan rumus *haversine* untuk menghitung jarak antara dua titik dengan mengabaikan kondisi geografis seperti lembah ataupun bukit berdasarkan persamaan 1 dimana sudut harus dalam satuan radian untuk menggunakan fungsi trigonometri.

Berikut sejumlah data sarana musik yang digunakan sebagai contoh untuk proses perhitungan jarak :

Tabel II-2 Data Lokasi Sarana Musik

Sarana Musik	Alamat	latitude	longitude
Sarana 1	Jl. Indramayu Raya No. 68	-6.919695	107.657364
Sarana 2	Jl. Pungkur 97	-6.927554	107.607798
Sarana 3	Jl. Soka no 4	-6.915286	107.628294
Sarana 4	Jl. Surapati No 197	-6.898055	107.633097
Sarana 5	Jl. Depok Raya No 1	-6.914517	107.655424

Berikut merupakan data lokasi pengguna :

Tabel II-3 Lokasi Awal

Lokasi Awal	Alamat	latitude	longitude
Lokasi Pengguna	Jl. Purwakarta	-6.917557	107.660445

Dari sejumlah data sarana musik yang ada akan dihitung jarak antara lokasi pengguna dengan setiap lokasi sarana untuk mendapatkan lokasi terdekat yang dalam hal ini terdekat dalam radius 2 km. Untuk mendapatkan lokasi terdekat perlu dilakukan perhitungan jarak pada setiap lokasi yang ada dengan lokasi pengguna setelah itu dilakukan pengecekan jarak dalam radius 2 Km.

Berikut contoh perhitungan lokasi pengguna (Jl. Purwakarta) dengan salah satu lokasi sarana (Jl. Indramayu) pada tabel diatas menggunakan rumus *haversine* pada persamaan 1.

Langkah awal dihitung selisih lokasi pengguna dengan lokasi sarana musik:

Δlat = (-6.919695 – (-6.917557))

= -0.000037315139 (dalam radian)

Δlon = (107.657364 - 107.660445)

= -0.000053773594 (dalam radian)

Kemudian menghitung perpotongan sumbu untuk mengetahui jarak dengan menghitung :

a = $\sin^2$(-0.000037315139 / 2) + (cos-6.917557) * (cos(-6.919695) * $\sin^2$(-0.000053773594 / 2)

= 0.00000000106051512354642 (dalam radian)

c = 2 * Atan2 (√0.000000728901324047745,

$\sqrt{(1 - 0.000000728901324047745)}$)

= 0.0000651311023681102

Hasil diatas masih dalam radian maka perlu dikalikan dengan radius permukaan bumi sehingga didapat hasil seperti berikut :

d = 6371 * 0.0000651311023681102

= 0.41 Km (pembulatan dua angka di belakang koma)

Perhitungan tersebut dilakukan terhadap lokasi pengguna dengan lokasi – lokasi sarana yang ada sehingga akan didapatkan hasil (jarak) antara lokasi pengguna dan sarana musik yang kemudian di cek ke dalam radius 2 Km untuk mengetahui lokasi mana saja yang terdekat seperti pada tabel berikut :

Tabel II-4 Kesimpulan Perhitungan Jarak

Lokasi Awal	Lokasi Tujuan	Jarak	Kesimpulan
Lokasi Pengguna	**Sarana1**	**0.41 Km**	**Didalam radius 2 Km**
Lokasi Pengguna	Sarana 2	5.92 Km	Diluar radius 2 Km
Lokasi Pengguna	Sarana 3	3.56 Km	Diluar radius 2 Km
Lokasi Pengguna	Sarana 4	3.72 Km	Diluar radius 2 Km
Lokasi Pengguna	**Sarana 5**	**0.65 Km**	**Didalam radius 2 Km**