BAB 2
LANDASAN TEORI

2.1 Fasilitas Kesehatan

Pelayanan merupakan suatu aktivitas atau serangkaian alat yang bersifat tidak kasat mata (tidak dapat diraba), yang terjadi akibat interaksi antara konsumen dengan karyawan atau hal-hal lain yang disediakan oleh perusahaan pemberi pelayanan yang dimaksudkan untuk memecahkan persoalan konsumen [5].

Pemanfaatan pelayanan kesehatan adalah pengunaan fasilitas pelayanan yang disediakan baik dalam bentuk rawat jalan, rawat inap, kunjungan rumah oleh petugas kesehatan ataupun bentuk kegiatan lain dari pemanfaatan pelayanan tersebut yang didasarkan pada ketersediaan dan kesinambungan pelayanan, penerimaan masyarakat dan kewajaran, mudah dicapai oleh masyarakat, terjangkau serta bermutu [6].

Syarat pokok pelayanan kesehatan adalah:

a. Ketersediaan dan kesinambungan

   Pelayanan yang baik adalah pelayanan kesehatan yang tersedia di masyarakat (*acceptable*) serta berkesinambungan (*sustainable*). Artinya semua jenis pelayanan kesehatan yang dibutuhkan masyarakat ditemukan serta keberadaannya dalam masyarakat adalah ada pada tiap saat dibutuhkan.

b. Kewajaran dan Penerimaan

   Pelayanan kesehatan yang baik adalah bersifat wajar *(appropriate)* dan dapat diterima *(acceptable)* oleh masyarakat. Artinya pelayanan kesehatan tersebut dapat mengatasi masalah kesehatan yang dihadapi, tidak bertentangan dengan adat istiadat, kebudayaan, keyakinan dan kepercayaan masyarakat, serta bersifat tidak wajar, bukanlah suatu keadaan pelayanan kesehatan yang baik.

c. Mudah dicapai oleh masyarakat

Pengertian dicapai yang dimaksud disini terutama dari letak sudut lokasi mudah dijangkau oleh masyarakat, sehingga distribusi sarana kesehatan menjadi sangat penting. Jangkauan fasilitas pembantu untuk menentukan permintaan yang efektif. Bila fasilitas mudah dijangkau dengan menggunakan alat transportasi yang tersedia maka fasilitas ini akan banyak dipergunakan. Tingkat pengguna di masa lalu dan kecenderungan merupakan indikator terbaik untuk perubahan jangka panjang dan pendek dari permintaan pada masa akan datang.

d. Terjangkau

Pelayanan kesehatan yang baik adalah pelayanan yang terjangkau oleh masyarakat, dimana diupayakan biaya pelayanan tersebut sesuai dengan kemampuan ekonomi masyarakat. Pelayanan kesehatan yang mahal hanya mungkin dinikmati oleh sebagian masyarakat saja.

2.2 Agumented Reality

Definisi *augmented reality* adalah sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu,dan integrasi yang baik memerlukan penjejakan yang efektif. Tujuan dari *augmented reality* sebenarnya adalah untuk menambahkan informasi ke dalam objek atau tempat yang nyata. *Augmented reality* sudah digunakan dalam beberapa bidang seperti dalam kedokteran digunakan untuk pencitraan medis, dalam dunia penerbangan membantu pilot menunjukkan dataran yang mereka lihat, dan di museum di mana artifak dapat ditandai dengan informasi mengenai sejarah dan sebagainya. realitas tertambah juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam[7].

Maka *Augmented Reality* disimpulkan sebagai teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam

sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. Tidak seperti realitas maya yang sepenuhnya menggantikan kenyataan, realitas tertambah sekedar menambahkan atau melengkapi kenyataan.

2.2.1 Perangkat keras Augmented Reality

Teknik Perangkat keras pada teknologi AR secara garis besar dibagi menjadi tiga bagian[8] yaitu:

1. Perangkat Penangkapan Video merupakan piranti masukan yang menangkap video dari lingkungan nyata untuk diolah oleh prosesor. Contoh dari perangkat penangkapan video diantaranya: kamera perekam dan *web cam.*
2. Prosesor merupakan piranti yang mengolah hasil penangkapan dari perangkat penangkapan video dengan bantuan suatu perangkat lunak AR. Pada awalnya, prosesor akan melacak dan mengidentifikasi pola dari suatu atribut fisik yang ditangkap video, lalu prosesor akan menambahkan objek virtual sesuai dengan pola yang dikenali dan kemudian meletakkannya di atas titik koordinat virtual dari atribut fisik yang ditangkap video.
3. Perangkat *Display* merupakan piranti keluaran yang menampilkan objek virtual hasil dari pengolahan prosesor. Contoh dari perangkat tampilan diantaranya: monitor komputer, LCD, TV dan Proyektor.

2.2.2 Tipe Augmented Reality

Marker AR dibagi menjadi dua tipe [9], yaitu:

a. Augmented Reality berbasis Marker

AR berbasis *Marker*, disebut juga Pelacakan berbasis *marker*, merupakan tipe AR yang mengenali *marker* dan mengidentifikasi pola dari *marker* tersebut untuk menambahkan suatu objek virtual ke lingkungan nyata. *Marker* merupakan ilustrasi persegi hitam dan putih dengan sisi hitam tebal, pola hitam di tengah persegi dan latar belakang putih. Contoh Marker dapat dilihat pada Gambar 2.1



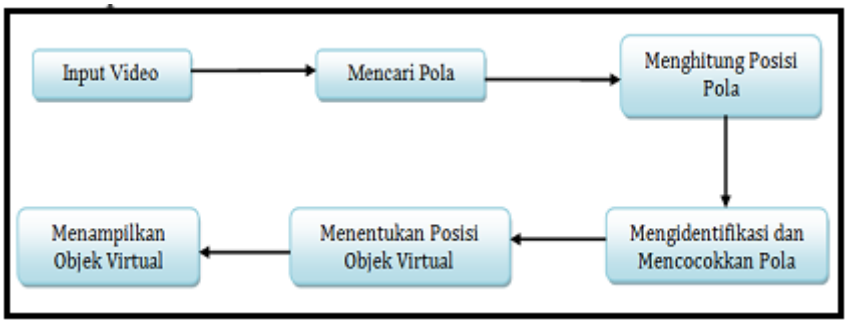
Gambar 2.1 Contoh marker [9]

b. Markerless Augmented Reality

Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode *Markerless Augmented Reality*. Dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* yang berbentuk kotak dan berwarna hitam putih untuk menampilkan elemen-elemen digital. Seperti yang saat ini dikembangkan oleh perusahaan *Augmented Reality* terbesar di dunia Total Immersion, mereka telah membuat berbagai macam teknik *Markerless Tracking* sebagai teknologi andalan mereka, seperti *Face Tracking*, *3D Object Tracking*, dan *Motion Tracking*, serta *GPS Based Tracking*.

2.2.3 Prinsip kerja Agumented Reality

Sistem *Augmented Reality* bekerja berdasarkan deteksi citra dan citra yang digunakan adalah marker. Prinsip kerjanya sebenarnya cukup sederhana. kamera yang telah dikalibrasi akan mendeteksi marker yang diberikan, kemudian setelah mengenali dan menandai pola marker, *webcam* akan melakukan perhitungan apakah marker sesuai dengan database yang dimiliki. Bila tidak, maka informasi marker tidak akan diolah, tetapi bila sesuai maka informasi marker akan digunakan untuk me-*render* dan menampilkan objek 3D atau animasi yang telah dibuat sebelumnya. Berikut diagram cara kerja *Agumented reality*



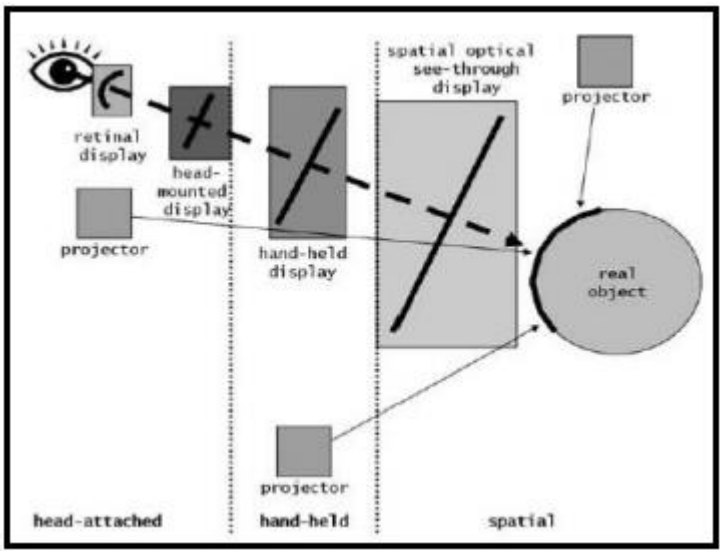
Gambar 2.2 Diagram cara kerja Augmented Reality [10]

Seperti yang ditunjukkan pada Gambar 2.2, cara kerja AR dalam menambahkan objek virtual ke lingkungan nyata adalah sebagai berikut [10]:

1. Perangkat input menangkap video dan mengirimkannya ke prosesor.
2. Perangkat lunak di dalam prosesor mengolah video dan mencari suatu pola.
3. Perangkat lunak menghitung posisi pola untuk mengetahui dimana objek virtual akan diletakkan.
4. Perangkat lunak mengidentifikasi pola dan mencocokkannya dengan informasi yang dimiliki perangkat lunak.
5. Objek virtual akan ditambahkan sesuai dengan hasil pencocokan informasi dan diletakkan pada posisi yang telah dihitung sebelumnya.
6. Objek virtual akan ditampilkan melalui perangkat tampilan.

2.2.4 Sistem Display Augmented Reality

Sistem tampilan AR merupakan sistem pembentukan objek virtual pada jalur optik diantara mata pengamat dan objek nyata dengan menggunakan seperangkat alat optik, elektronik dan komponen mekanik [11]. Sistem display AR diilustrasikan pada Gambar 2.3



Gambar 2.3 Ilustrasi sistem display AR [11]

Sistem *display* AR dibagi menjadi 3 kategori [11], yaitu:

1. *Head-Attached Display* merupakan sistem display AR dimana pengguna mengenakan perangkat keras AR di kepala.
2. *Hand-Held Display* merupakan sistem display AR dimana objek virtual terbentuk dalam jangkauan tangan pengguna.
3. *Spatial Display* merupakan sistem display AR yang memproyeksikan objek virtual ke lingkungan nyata menggunakan proyektor dijital atau tergabung dengan lingkungan nyata menggunakan panel tampilan .

2.3 Android

Menurut Safaat, Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet. Android juga menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. kemudian dalam pengembangan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia[12].

2.3.1 Arsitektur Android

Secara garis besar arsitektur android dapat dijelas dan digambarkan sebagai berikut :

a. Application and Widgets

Application and widgets adalah layer dimana berhubungan dengan aplikasi dan biasanya *download* aplikasi kemudian lakukan instalasi dan jalankan aplikasi tersebut, delayer inilah terdapat seperti aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b. Application Frameworks

Android adalah *"Open Development Platform"* yaitu android menawarkan kepada pengembang atau member kemampuan kepada pengembangan untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan tambahan seperti status *notifications* dan masih banyak lagi. Pengembang memiliki akses penuh menuju API *Framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya dengan mudah dapat menggunakan komponen yang sudah digunakan (*reuse*).

Sehingga bisa disimpulkan *Application Framework* adalah layer dimana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan di sistem android, karena pada layer inilah aplikasi dapat dirancang, seperti *content-providers* yang berupa sms dan lain sebagainya.

Komponen-komponen yang termasuk didalam application *Framework* adalah sebagai berikut :

1) *Views*
2) *Content Provider*
3) *Resource Manager*

4) *Notification Manager*
5) *Activity Manager*

c. *Libraries*

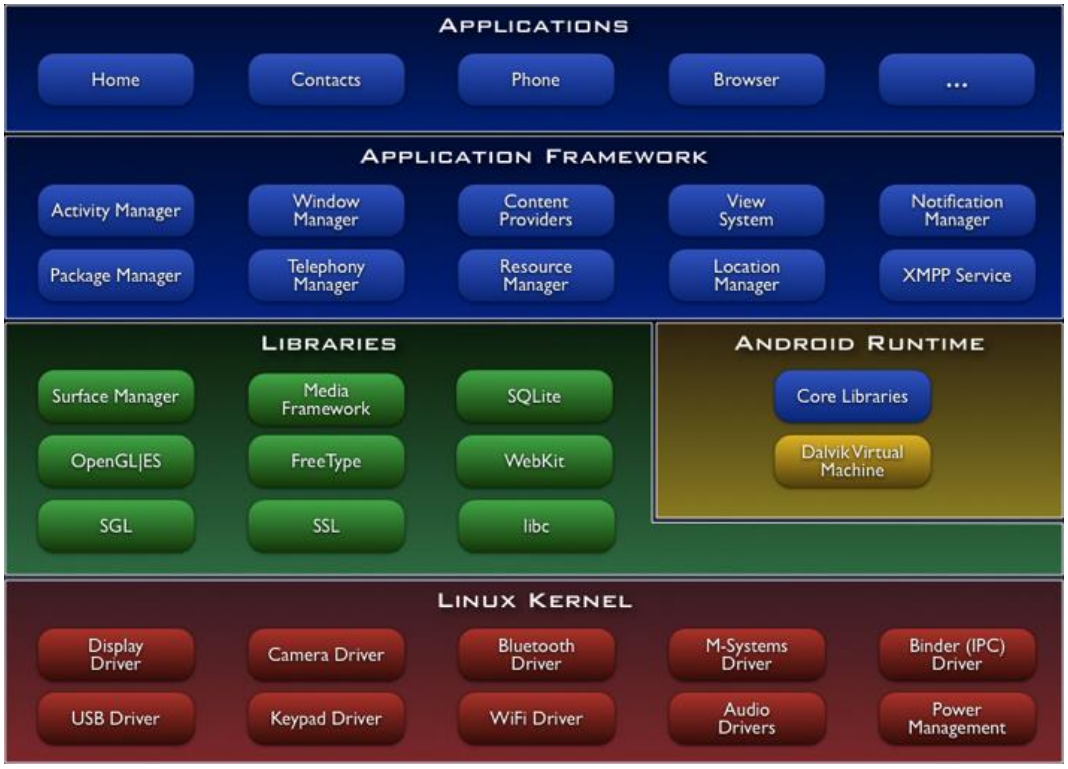
Libraries adalah layer dimana fitur-fitur android berada biasanya para pembuat aplikasi kebanyakan mengakses libraries untuk menjalankan aplikasinya berjalan diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta :

1) Libraries media untuk pemutar media audio dan video.
2) Libraries untuk manajemen tampilan.
3) Libraries *Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
4) Libraries SQLLite untuk database.
5) Libraries SSl dan WebKit terintegrasi dengan *web browser* dan *security*.
6) Libraries Live *Webcore* mencakup *modern web browser* dengan *engine embedded web view.*

d. *Android Run Time*

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android Run Time dibagi menjadi dua bagian yaitu:

1) *Core Libraries*
2) *Dalvik Virtual Machine*
3) *Linux Karnel*



Gambar 2.4 Arsitektur Android [12]

2.3.2 Versi Android

Safaat memaparkan penjelasan berbagai macam versi Android seperti di bawah ini :

1. Android versi 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 versi 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 pembaharuan untuk 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,dukungan

Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. Android versi 1.6 (Donut)

   Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan indikator baterai dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA atau EVDO, 802.1x, VPN, Gestures, dan *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to change speech* tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. Android versi 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 flashuntuk kamera 3,2 MP, digital *Zoom*, dan Bluetooth 2.1. Agar dapat bergerak cepat dalam persaingan perangkat generasi yang berikutnya, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (killer apps- aplikasi unggulan).Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih.Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

   Dengan semakin berkembangnya dan bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka dalam 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 versi 2.2 (Froyo: Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umum terhadap versi – versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engineyang 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 versi 2.3 (Gingerbread)

   Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahanperubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) yang sudah 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 versi 3.0/3.1 (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 processordan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama kali yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet tersebut bernama Eee Pad Transformer yang merupakan produk dari Asus yang masuk ke pasar Indonesia pada Mei 2011.

8. Android versi 4.0 (ICS: Ice Cream Sandwich)

   Pada tanggal 19 Oktober 2011, diperkenalkannya Android versi 4.0 yang membawa fitur Honeycomb untuk smartphone, menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, kontak jaringan sosial terpadu, perangkat tambahan fotografi, pencarian email secara offline, dan

berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

9. Android versi 4.1 (Jelly Bean)

   Android Jelly Bean yaang diluncurkan pada acara Google I/O membawa sejumlah keunggulan dan fitur baru. Adapun penambahan fitur baru diantaranya yaitu meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Google Now yang juga menjadi bagian yang diperbaruipun tak ketinggalan.Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus.

10. Android Versi 4.4 kitkat

    Diluncurkan pada 31 Oktober 2013 yang dinamai dengan izin Nastle dan Hershey.Memberikan pembaruan dibidang antarmuka dengan navigasi yang transparan pada layar depan ditambah dengan NFC Host Card sebagai emulator kartu pintar.

2.4 Pemrograman Berorientasi Objek / PBO / OOP

Pemrograman Berorientasi Objek *(Object Oriented Programming*/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam *class-class* atau *object-object*. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya [13].

Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan Object yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman sistem yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau *statement.*

Dalam konsep Pemrograman Berorientasi Objek dikenal beberapa istilah umum, yaitu:

1. Attribute

Atribut dari sebuah kelas adalah sistem global yang dimiliki sebuah kelas, Atribut dapat memiliki hak akses private, public maupun protected. Sebuah atribut yang dinyatakan sebagai private hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas lainnya tidak dapat mengakses atribut ini secara langsung. Sebuah atribut yang dinyatakan sebagai public dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya. Sebuah atribut yang dinyatakan sebagai protected tidak dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.Atribut juga biasa disebut state/ properties/ field merupakan bagian yang dimiliki oleh sebuah class yang tidak melakukan operasi, tetapi kepadanya dilakukan operasi sehingga dapat merubah nilai dari attribute tersebut.

2. Method

Method adalah fungsi atau prosedur yang dibuat oleh seorang programmer didalam suatu *Class*. Dengan kata lain, *method* pada sebuah kelas sistem sama dengan fungsi atau prosedur pada pemrograman system. Pada sebuah *method* di dalam sebuah kelas juga memiliki izin akses seperti halnya atribut pada kelas, izin akses itu antara lain private, public dan protected yang memiliki arti sama pada izin akses atribut yang telah dibahas sebelumnya. Sebuah kelas boleh memiliki lebih dari satu method dengan nama yang sama asalkan memiliki parameter masukan yang berbeda sehingga sistem atau interpreter dapat mengenali *method* mana yang dipanggil.

Di dalam sebuah kelas, terdapat juga yang disebut sebagai *method* atau atribut statis yang memiliki kata kunci statis. Maksud dari statis di sini adalah *method* yang dapat diakses secara berbagi dengan semua objek lain tanpa harus membuat objek yang memiliki *method* statis tadi (tanpa proses *new*), tapi sebuah *method* statis mempunyai keterbatasan yaitu hanya dapat mengakses atribut atau *method* lain di dalam kelas yang membungkusnya yang juga bersifat statis. *Method* statis biasanya diimplementasikan untuk *method main*.

Didalam suatu *class, behavior* disimpan dalam bentuk *Method*. *Method* merupakan serangkaian *statemen* / perintah (perintah = baris program) dalam suatu *class* yang meng*handle task* tertentu. *Method* merupakan hal-hal yang bisa dilakukan oleh *object* dari suatu *class*.

Method didefinisikan pada *class* akan tetapi dipanggil melalui *object*. *Method* memiliki peranan diantaranya :

1. Merubah nilai atribut dari suatu *object*.
2. Menerima informasi dari *object* lain.
3. Mengirim informasi ke obyek lain untuk melakukan suatu *task* (cara obyek berkomunikasi dengan obyek lain adalah dengan menggunakan *method*).

3. Class

Merupakan model yang berisi kumpulan *attribute* dan *method* dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *class* manusia memiliki *attribute* berat, tinggi, usia kemudian memiliki *method* makan, minum, tidur. *Method* dalam sebuah *class* dapat merubah *attribute* yang dimiliki oleh *class* tersebut. Sebuah *class* merupakan dasar dari modularitas dan struktur dalam pemrograman berorientasi object.

4. Object

Merupakan perwujudan dari *class*, setiap *object* akan mempunyai *attribute* dan *method* yang dimiliki oleh *class*-nya, contohnya: amir, ahmad, yani merupakan *object* dari *class* manusia. Setiap *object* dapat berinteraksi dengan *object* lainnya meskipun berasal dari *class* yang berbeda.

Object merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun *object* lain dan mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya. Pada dasarnya semua benda yang ada di dunia dapat dianggap sebagai sebuah object. Rumah, mobil, sepeda motor, meja, dan sistem merupakan contoh-contoh object yang ada di dunia nyata. *Object* adalah implementasi dari class. Secara sederhananya, dapat dikatakan terdiri dari sistem (atribut) dan *method.*

2.5 UML

Unified *Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan system [13].

2.5.1 Sejarah UML

Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi *shlaer-mellor*, metodologi *wirfs-brock*, dan sebagainya. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek.

Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group*[13].

2.5.2 Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

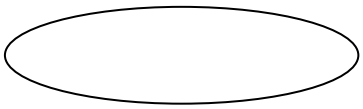
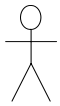
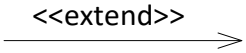
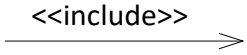
1. *Use Case* Diagram untuk memodelkan proses bisnis.
2. *Conceptual* Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence* Diagram untuk memodelkan pengiriman pesan (*message*) antar objects
4. *Collaboration* Diagram untuk memodelkan interaksi antar objek.
5. *State* Diagram untuk memodelkan perilaku objects di dalam sistem.
6. *Activity* Diagram untuk memodelkan perilaku *Use Cases* dan *objects* di dalam *system*.
7. *Class* Diagram untuk memodelkan struktur kelas.
8. *Object* Diagram untuk memodelkan struktur *object*.
9. *Component* Diagram untuk memodelkan komponen *object*.
10. *Deployment* Diagram untuk memodelkan distribusi aplikasi.

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi *object*, yaitu *use case* diagram, *sequence* diagram, *collaboration* diagram, dan *class* diagram.

2.5.3 *Use Case* Diagram

Use-case diagram merupakan pemodelan untuk kelakuan sistem informasi yang dibuat. *Use case* mendeskrisikan interaksi antara saru atau lebih actor dengan system informasi yang akan dibangun. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah system informasi dan si saja yang berhak menggunakan fungsi tersebut. Berikut adalah symbol-simbol yang ada pada diahram *use case*[13] :

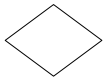
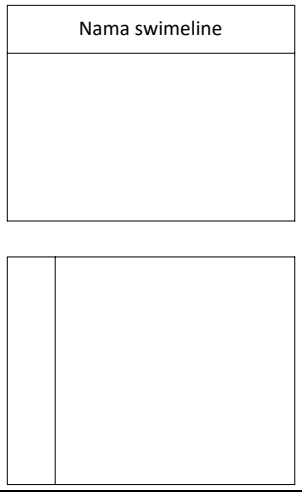
Tabel 2.1 Simbol Use Case [13]

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*
Actor / *actor* 	Orang, proses atau system lain yang berinteraksi dengan system informasi yang akan dibuat.
Asosiasi / *association*	Komunikasi antara actor dan *use case* yang berpartisipasi ada *use case* atau *use case* memiliki interaksi dengan actor
Ekstensi / *extend* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan.
Generalisasi / *generalitzation* 	Hubungan generalisasi dan spesialisasi antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainya.
Menggunakan / *include* 	Relasi *use case* tambahan ke sebuah *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya.

2.5.4 *Activity* Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran aktivitas) dari sebuah system proses bisnis atau menu yang ada pada perangkat lunak. Berikit ini adalah simbol yang terdapat pada *activity diagram* :

Tabel 2.2 Simbol Activiti Diagram [13]

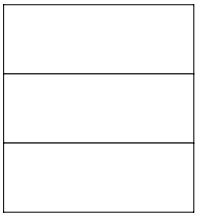
Simbol	Deskripsi
Status awal 	Status awal aktivitas system sebuah diagram aktivitas memunyai status awal
Aktivitas 	Aktivitas yang dilakukan system, aktivitas biasanya diawali dengan kata kerja
Percabangan / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / *join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir
Swimeline 	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi

2.5.5 Class Diagram

Class diagram dapat membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* banyak memperhatikan hubungan antarkelas dan penjelasan detail tiap kelas dalam pemodelan desain (dalam *logical view*) dari suatu sistem. Selama proses analisa, *class diagram* memperhatikan aturan-aturan dan tanggung jawab

entitas yang menentukan perilaku sistem. Berikut ini adalah simbol yang ada pada struktur class diagram :

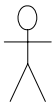
Tabel 2.3 Simbol Class Diagram [13]

Simbol	Deskripsi
Kelas 	Kelas ada struktur system
Antarmuka / *interface* 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi / *association*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / *dependency*	Kebergantungan antar kelas
Agregasi / aggregation 	Relasi antar kelas dengan makna semua bagian

2.5.6 *Sequence* Diagram

Sequence Diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use-case*. Sequence diagram memperlihatkan tahap demi tahap apa yang harus terjadi untuk menghasilkan suatu didalam *sequence* diagram.

Tabel 2.4 Simbol Sequence Diagram [13]

Simbol	Deskripsi
Aktor 	Orang, proses atau system lain yang berinteraksi dengan system
Garis hidup	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi
Pesan tipe create <<create>>	Menytakan suatu objek membuat objek yang lain
Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data masukan ke objek lain
Pesan tipe keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu

2.6 POI (Point of Interest)

POI (*Point of Interest*) atau dapat disebut titik koordinat adalah lokasi titik tertentu yang menunjukkan objek pada titik tersebut. Minimal POI memiliki titik latitude dan longitude. Latitude merupakan garis lintang, sedangkan longitude merupakan garis bujur. Selain itu ada juga altitude sebagai informasi tambahan berupa ketinggian dari objek tersebut.

2.7 GPS (Global Positioning System)

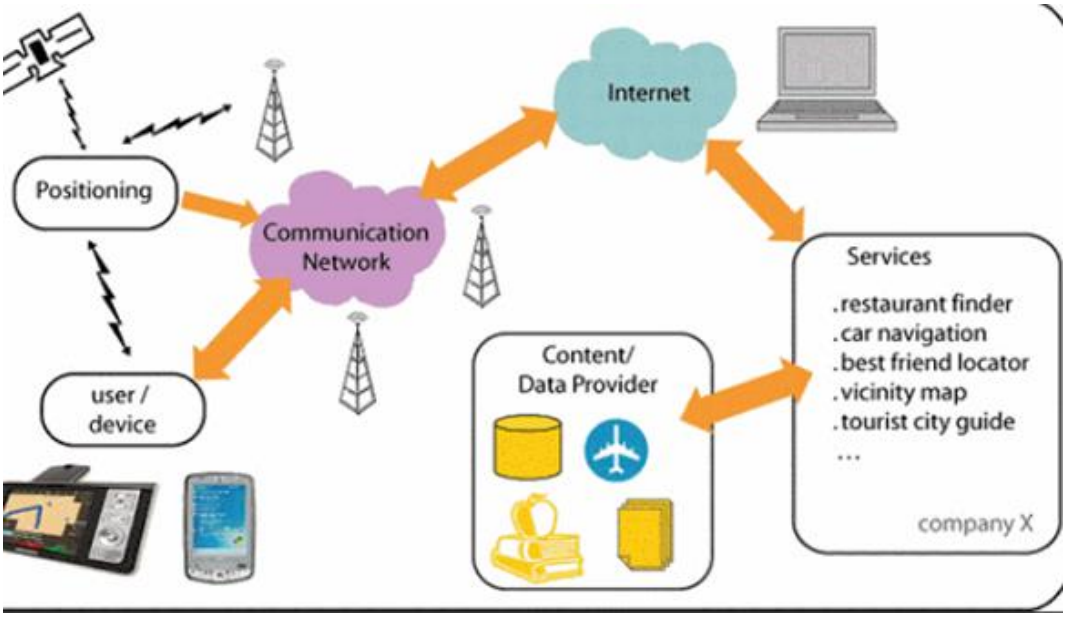
Global positioning system (GPS) atau sistem pemosisi global merupakan sistem yang digunakan untuk menentukan posisi di permukaan bumi dengan sinkronisasi sinyal satelit[14]. Dengan bantuan GPS seseorang dapat mengetahui posisi objek yang diinginkannya dengan bantuan perangkat yang memiliki sensor GPS di dalamnya. GPS bekerja ketika sejumlah satelit yang berada di orbit Bumi

memancarkan sinyalnya ke Bumi, kemudian sinyal tersebut ditangkap oleh sebuah alat penerima yang nantinya diubah menjadi informasi berupa titik lokasi dari alat penerima tersebut. Karena alat ini bergantung penuh pada satelit, maka sinyal satelit merupakan hal yang penting untuk mendapatkan informasi posisi objek yang berupa titik koordinat. Untuk itu perlu diperhatikan hal yang dapat mengganggu sinyal satelit antara lain adalah kondisi geografis, alat yang mengeluarkan gelombang elektromagnetik, gedung, dan sinyal yang memantul.

2.8 LBS (Location Based Service)

Layanan lokasi dapat didefinisikan sebagai perangkat lokasi atau posisi dengan informasi lain sehingga dapat memberikan layanan yang mengintegrasikan nilai tambah pengguna mobile[15]. LBS merupakan layanan yang mengidentifikasikan lokasi atau objek, seperti lokasi keberadaan seseorang, lokasi bank, rumah sakit atau SPBU terdekat. Dengan teknologi LBS (*Location Based Service*) yang dapat digunakan untuk mengetahui posisi berdasarkan titik geografis dari lokasi kita dan lokasi lain yang diinginkan hal tersebut dapat dilakukan. Selain itu teknologi LBS ini dapat diaplikasikan untuk GIS (*Geographical Information System*), Perbedaan LBS dengan GIS adalah LBS hanya akan mendapatkan titik geografis dari lokasi saja, sedangkan dengan GIS kita akan mendapatkan informasi keruangan secara geografis dengan bantuan LBS sebagai layanan untuk mendapatkan titik geografisnya. Untuk mengaplikasikan GIS dengan LBS paling tidak kita harus mengetahui titik koordinat lokasi yang diinginkan dan bank data (*database*) yang menyimpan titik koordinat dan informasi lain yang berguna mengenai lokasi tersebut.

Hal yang paling penting dari *Location Based Services* (LBS) dapat bekerja sesuai yang diinginkan oleh pengembang aplikasi android. Android pun memungkinkan pengembang menentukan metode pencarian lokasi yang dibutuhkan dan juga dapat mengatur kebutuhan daya, biaya dan akurasi berdasarkan spesifik yang akan untuk aplikasi yang dikembangkan.



Gambar 2.5 Prinsip kerja LBS [15]

2.8.1 Unsur Utama Location Based Services (LBS)

Location Based Service (LBS) atau layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan. Dua unsur utama dari Location Based Services (LBS) adalah: [15]

- *Location Manager (API Maps):* Menyediakan pernagkat bagi sumber atau source untuk LBS, *Application Programming Interface (API) Maps* menyediakan fasilitas untuk menampilkan atau memanipulasi peta. Paket ini berada pada “com.goole.android.maps;”.
- *Location Providers (API Location)*: Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. *PAI Location* berhubungan dengan dara GPS *(Global Positioning System)* dan data lokasi *real-time*. *API Location* berada pada paket Android yaitu dalam paket “android.location”. Lokasi, perpindahan, serta kedekatan dengan lokai tertentu dapat ditentukan melalui *Location Maneger*.

2.8.2 Komponen Utama Location Based Services (LBS)

Terdapat lima komponen pendukung utama dalam teknologi layanan berbasis lokasi [16], antara lain :

1. Piranti Mobile, adalah satu komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suar, gambar dan lain sebgainya. Piranti mobile yang dapat digunakan bisa berupa PDA, smartphone, laptop. Selain itu, piranti mobile dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya navigasi berbasis GPS.
2. Jaringan Komunikasi, komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari piranti mobilenya untuk kemudian dikrimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.
3. Komponen Positioning (Penujuk Posisi/Lokasi), setiap layanan yang di berikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang ebrfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi mobile atau juga menggunakan Global Positioning System (GPS).
4. Penyedia layanan dan aplikasi, merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menetukan posisi pengguna, menentukan rute jalan, mencari *Yellow Pages* sesuai dengan permintaan, dan masih banyak lagi yang lainnya.
5. Penyedia data dan konten, penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan-badan milik pemerintah atau juga data-data

perusahaan/bisnis/industri bisa saja berasal dari Yellow Page, maupun perusahaan penyedia data lainnya.

2.9 Google Maps Api

Google Maps API adalah sebuah layanan yang memungkinkan Google Maps agar dapat diintegrasikan ke dalam aplikasi yang memang membutuhkan fitur dari Google Maps ini . *Google Maps API* adalah sebuah library Javascript. Dengan menggunakan *Google Maps API* ini dapat memudahkan pengembang dalam membuat aplikasi yang membutuhkan teknologi peta di dalamnya dan tidak perlu memikirkan cara membuatnya secara manual karena dengan mengunakan *Google Maps API* ini sudah semua yang dibutuhkan terkait teknologi peta digital sudah cukup lengkap dan dapat dimanfaatkan untuk aplikasi yang membutuhkannya[14]. Saat ini *Google Maps API* telah mengalami perkembangan, hingga saat ini *Google Maps API* telah mencapai versi 3. Namun, *Google Maps API* versi 3 ini masih hanya bisa digunakan oleh pemrograman berbasis web dengan menggunakan Javascript. Untuk penggunaan *Google Maps API* pada perangkat mobile seperti misalnya android *Google Maps API* yang bisa digunakan saat ini telah mencapai versi 2. Perkembangan pada Google Maps Versi 2 ini adalah dengan memiliki fitur– fitur yang lengkap dari versi sebelumnya salah satunya adalah dapat menampilkan peta dengan sudut yang di inginkan dan berupa 3D *view*. Selain itu pada *Google Maps API* terdapat beberapa pilihan model peta yang ditampilkan seperti *Roadmap, Satellite, Terrain* dan *Hybrid*.

2.10 Eclipse

Dalam Perancangan aplikasi ini penulis menggunakan Eclipse Sebagai IDE untuk mempermudah proses perancangan

2.11.1 Pengertian Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform independent). Berikut ini adalah sifat dari Eclipse[19]:

a. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, HP-UX dan Mac OS X.
b. *Mulit-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
c. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi,test perangkat lunak, pengembangan web, dan lain sebagainya. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in.*

2.11.2 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat *plug-in*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform (RCP)*. Berikut ini adalah komponen yang membentuk RCP:

a. Core platform
b. OSGI
c. SWT (Standard Widget Toolkit)
d. Eclipse Workbench
e. Jface

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan plug-in baru. Eclipse beserta plug-innya diimplementasikan

dalam bahasa pemrograman Java. Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya.

2.11 PHP

Menurut dokumen resmi PHP, PHP merupakan singkatan dari *Hyper text Preprocessor*. Ia merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses di *server*. Hasilnya yang dikirim ke klien tempat pemakai menggunakan browser. Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis, Artinya ia dapat membentuk suatu tampilan berdasarkan permintaan terkini, Misalnya Anda bisa menampilkan isi *database* ke halaman *web*. Pada prinsipnya PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP *(Active ServerPage)*, *Cold Fusion*, ataupun *Perl*. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip *Perl* yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi tool yang disebut *"Personal home page"*. Paket inilah yang menjadi cikal-bakal PHP. Pada tahun 1995,Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrograman dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan kompleks sambil jalan.Pada saat ini PHP cukup popular sebagai peranti pemrograman Web, terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada

server yang berbasis UNIX, *Windows*, dan Machintosh. Pada awalnya PHP dirancang untuk diintegrasikan dengan *web server Apache*. Namun, belakangan PHP juga dapat bekerja dengan *web server* seperti PWS *(Personal Web Server)*, IIS *(Internet Information Server)*, dan *Xitami*.Untuk mencoba PHP anda tidak perlu menggunakan komputer kelas server. Hanya dengan sebuah komputer biasa, anda bisa mempelajari dan mempraktikkan PHP. Pemrograman berorientasi objek merupakan model pemrograman yang menjadi tren saat ini. PHP juga mendukung hal ini. Suatu objek dapat dibentuk melalui kelas. Dalam hal ini, kelas dapat dianggap sebagai cetakan objek. Jika menggunakan istilah teknis, kelas adalah definisi umum untuk sekelompok objek. Objek sendiri sering disebut instan kelas. Dengan menggunakan cetakan anda bisa menciptakan objek.

2.12 SDK Android

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit)* sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, android member anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone atau Smartphone. Beberapa fitur-fitur android yang paling penting adalah[12] :

a. *Framework* : Aplikasi yang mendukung pengganti komponen dan reusable.
b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat mobile.
c. *Integrated Browser* berdasarkan *engine open source WebKit.*
d. Grafis yang dioptimalkan dan didukung oleh *libraries grafis 2D*, grafis 3D berdasarkan spesifikasi Opengl ES 1.0 (*Opsional Ekselerasi hardware*).
e. SQLite untuk penyimpanan data.

f. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony (tergantung hardware).
g. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware).
h. Kamera, GPS, Kompas, dan Accelerometer (tergantung hardware).
i. Lingkungan Development yang lengkap dan termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

2.13 MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal, kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Selain itu , ia bersifat *Open Source* (Anda tidak perlu membayar untuk menggunakannya) pada berbagai platform (kecuali untuk jenis enterprise, yang bersifat komersial)[20].

MySQL termasuk jenis RDBMS (*Relational Database Management*). Itulah sebabnya, istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

2.14 Bahasa Pemroograman Java

Bahasa Java adalah sebuah bahasa pemrograman yang dapat digunakan dalam berbagai macam *platform* yang ada[18]. Hal inilah yang mendasari kesuksesan java dalam dekade ini, yaitu bahwa aplikasi Java dapat dieksekusi pada beragam sistem operasi dan lingkungan *software*, seperti halnya *Mosaic*, *Netscape Navigator*, atau *Internet Explorer*. Ketidak bergantungan ini pada platform inilah yang membuat Java ini unik.

2.15 Metode Pengujian Black-Box Testing

Pada siklus hidup pengembangan perangkat lunak (*System Development Life Cycle: SDLC*) salah satu suatu proses yang harus dilakukan adalah proses pengujian pengujian (*testing*). Pengujian perangkat lunak adalah suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah

memenuhi kebutuhan proses bisnis pengguna atau masih belum. *testing* adalah proses eksekusi suatu program untuk menemukan kesalahan sebelum digunakan oleh pengguna akhir (*end-user*)[4].

Salah satu metode pengujian perangkat lunak adalah *Black-Box Testing. Black-box Testing* merupakan sebuah metode yang digunakan untuk menemukan kesalahan dan mendemonstrasikan fungsional aplikasi saat dioperasikan, apakah input diterima dengan benar dan output yang dihasilkan telah sesuai dengan yang diharapkan.

Fokus dari pengujian mengunakan metode *Black-Box* adalah pada pengujian fungsionalitas dan output dihasilkan aplikasi. Pengujian black-box didesain untuk mengungkap kesalahan pada persyaratan fungsional dengan mengabaikan mekanisme internal atau komponen dari suatu program.

Pengujian perangkat lunak mempunyai beberapa level, untuk pengujian menggunakan metode *Black Box,* terdapat enam level yaitu *Integration, Functional, System, Acceptance, Beta,* dan *Regression.* Salah satu dari pengujian *Black-Box* yang dapat dilakukan oleh seorang penguji independen adalah *Functional testing.* Basis uji dari *functional testing* ini adalah pada spesifikasi dari komponen perangkat lunak yang akan diuji. *Functional testing* memastikan bahwa semua kebutuhan-kebutuhan telah dipenuhi dalam 35 sistem aplikasi. Dengan demikian fungsinya adalah tugas-tugas yang didesain untuk dilaksanakan 35 ystem. *Functional testing* berkonsentrasi pada hasil dari proses, bukan bagaimana prosesnya terjadi[17].