

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1. Ruang Lingkup Penelitian

Objek penelitian yang sedang diteliti adalah OXY Laundry yang berada di Jl. Raya Dipatiukur No.96, RT.02, Coblong, Bandung, Jawa Barat 40134.

##### 2.1.1. Logo OXY Laundry



**Gambar 2. 1 Logo OXY Laundry**

#### 2.2. Usaha Laundry

Istilah laundry yang umumnya di Indonesia dari dulu dikenal dengan istilah binatu adalah fasilitas di mana pakaian dicuci dan dikeringkan. Pelanggan cukup dengan menitipkan pakaiannya, kemudian dua atau tiga hari berikutnya bisa diambil dalam keadaan sudah disetrika rapi. Lebih praktis, tidak repot, murah dan pakaian pun siap pakai[10]. Selain itu bisnis *laundry* juga merupakan kategori bisnis yang berkelanjutan atau sebagai bisnis yang akan selalu dibutuhkan banyak orang. Manusia akan selalu mencuci pakaian selama kebutuhan sandang masih menjadi kebutuhan primer bagi manusia. Hal inilah yang membuat peluang bisnis *laundry* sangat menjanjikan.

### 2.3. Android

Android merupakan sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, gogle Inc. membeli Android Inc. Yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Aliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.[3]

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi Table PC. Pesatnya pertumbuhan Android itu sendiri adalah platform yang sangat lengkap baik itu sistem operasinya, Aplikasi dan Tool Pengembangan, Market aplikasi android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembang untuk melakukan pengembang sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan di bawah GNU General Public Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah “*copyleft*” lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah terms. Android didistribusikan di bawah lisensi *Apache Software* (ASL/Apache2), yang memungkinkan untuk didistribusi kedua dan seterusnya. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis android. Kebanyakan pengembang menggunakan Eclipse yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android. Eclipse adalah IDE yang paling populer untuk pengembangan Android, karena memiliki

Android plug-in yang tersedia untuk memfasilitasi pengembangan android. Selain itu Eclipse juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi Android, ini terbukti dengan adanya penambahan plugins untuk eclipse untuk membuat project android di mana *source software* langsung dari situs resminya Google. Tetapi hal diatas tidak menutup kemungkinan untuk menggunakan IDE yang lain seperti Netbeans untuk melakukan pengembangan android.

Aplikasi Android dapat dikembangkan pada sistem operasi berikut :

1. Windows XP Vista/Seven.
2. Mac OS X (Mac OS X 10.4.8 atau lebih baru).
3. Linux.

**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 API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. sebagai *platform* aplikasi-netral, Android memberi Anda kesempatan untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur-fitur Android yang paling penting adalah :

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*.
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile*.
3. *Integrated browser* berdasarkan enginen open source WebKit.
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (opsional akselerasi hardware)
5. SQLite untuk penyimpanan data.

6. Media support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung hardware).
7. Bluetooth, EDGE, 3G, dan wifi (tergantung hardware)
8. Kamera, GPS, kompas, dan accelerometer (tergantung hardware)
9. Lingkungan development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse. [3]

*Android Development Tools (ADT)* adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk eclipse akan memudahkan kita dalam membuat aplikasi project android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan Android SDK melalui eclipse. Dengan ADT juga kita dapat melakukan pembuatan package android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.[3]

Mengembangkan aplikasi android dengan menggunakan ADT di eclipse sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi android. Berikut data versi ADT untuk eclipse yang sudah dirilis :

1. ADT 12.0.0 (July 2011)
2. ADT 11.0.0 (June 2011)
3. ADT 10.0.1 (March 2011)
4. ADT 10.0.0 (February 2011)
5. ADT .0.0 (Januari 2011)
6. ADT 8.0.1 (December 2010)
7. ADT 8.0.0 (December 2010)
8. ADT 0.9.9 (September 2010)

9. ADT 0.9.8 (September 2010)

10. ADT 0.9.7 (May 2010)

11. ADT 0.9.6 (March 2010)

12. ADT 0.9.5 (December 2009)

13. ADT 0.9.4 (October 2009)

Semakin tinggi platform android yang kita gunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya munculnya platform baru diikuti oleh munculnya versi ADT yang terbaru. Untuk melakukan instalasi ADT di eclipse dapat dilakukan secara on-line maupun offline.

### **2.3.1. Sejarah Android**

Android dikembangkan oleh Android Inc. Perusahaan ini kemudian di beli oleh Google pada tahun 2005. Sistem operasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisasi Open Handset Alliance tahun 2007. Selain Google, beberapa nama-nama besar juga ikut serta dalam Open Handset Alliance, antara lain Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba, dan Intel.[3]

### **2.3.2. Versi Android**

Adapun versi android yang pernah dirilis adalah sebagai berikut :

#### 1. Android versi 1.1

Versi 1.1 dirilis pada 9 maret 2009, android ini dilengkapi dengan pembaruan estetis pada aplikasi, alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

#### 2. Android versi 1.5 (Cupcake)

Pada pertengahan bulan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit). Terdapat banyak pembaruan yakni mampu merekam dan menonton video dengan mouse kamera mengupload video ke youtube dan gambar ke picasa

langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung otomatis ke headset Bluetooth.

3. Android v1.6 Donut

Pada September 2009, google kembali merilis android versi 1.6 (Donut). Ini merupakan rilis minor dengan fitur utama berupa dukungan teknologi CDMA, gesture, text-to-speech engine, resolusi VWGA, VPN, dan fitur pencarian cepat.

4. Android v2.0-2.1 Eclair

Pada 3 Desember 2009 Google kembali meluncurkan Android dengan versi 2.0/2.1 (Eclair). Mengusung fitur baru yaitu live wallpaper, pengoptimalan hardware, peningkatan Google Maps 3.1.2, Bluetooth 2.1, dan profil bluetooth baru yaitu Object Push Profile (OPP) dan Phone Book Access Profile (PBAP).

5. Android v2.2 Frozen Yoghurt (Froyo)

Pada 20 Mei 2010 dirilis Android 2.2 (Froyo). Fitur baru yang dibawa adalah Global Positioning System (GPS), kompas, kamera, dan Accelorometer.

6. Android v2.3 Gingerbread

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Mengusung beberapa fitur baru, antara lain dukungan banyak kamera, Near field Communication (NFC), download manager service, dukungan terhadap sensor lain seperti giroskop, dan barometer.

7. Android v3.0-3.2 Honeycomb

Android Honeycomb dirancang khusus untuk tablet diluncurkan pada 22 Februari 2011. Android versi ini mendukung ukuran layar yang lebih besar. Pada Honeycomb mendukung multi prosesor dan akselerasi perangkat keras (Hardware) untuk grafis.

8. Android v4.0 Ice Cream Sandwich

Pada 19 Oktober 2011, Android versi 4.0 (Ice Cream Sandwich ) diluncurkan. Pada versi ini membawa perubahan besar pada tampilan User Interface

Android 3.0 agar cocok pada layar kecil. Fitur baru seperti Android Beam dan Wi-Fi Direct juga ditambahkan.

9. Android v4.1-4.3 Jelly Bean

Pada 27 Juni 2012, Android versi 4.1 (Jelly Bean) diluncurkan. Pada versi ini merupakan rilis minor yang membawa perbaikan fungsionalitas dan performa rendering User Interface (UI). Versi 4.2 mengusung fitur baru seperti gesture typing dan dukungan multiuser di piranti tablet. Versi 4.3 membawa perbaikan berupa dukungan Bluetooth Low Energy dan OpenGL ES 3.0. beberapa fitur baru yaitu dukungan terhadap bahasa internasional dan penulisan teks dua arah.

10. Android v4.4 Kitkat

Android versi 4.4 (Kitkat) diluncurkan pada 3 September 2013. Pada versi ini mengusung sejumlah perbaikan dan fitur baru yaitu dukungan Near Field Communication melalui host card emulation, pencetakan printer nirkabel, WebView dengan rendering engine Chromium, dan dukungan yang lebih baik bagi piranti yang menggunakan Random Access Memory (RAM) rendah.

11. Android versi 5.0 (Lollipop)

Versi terbaru saat ini dengan tampilan UI yang lebih menarik, dengan meningkatkan daya hidup baterai 30% lebih tahan lama.

12. Android versi 1.0 Marshmallow

Android 6.0 dan 6.0.1 "Marshmallow" merupakan pemutakhiran yang akan datang untuk sistem operasi telepon genggam Android, kemungkinan besar akan dirilis pada Q3 2015 ("sementara dijadwalkan untuk September"), dengan pratayang ketiga dan terakhir dirilis pada tanggal 17 Agustus 2015. Pertama diperkenalkan di Google I/O pada tanggal 28 Mei 2015, Marshmallow terutama akan berfokus pada perbaikan inkremental dan penambahan fitur lainnya. Pratayang pengembang Android "M" dirilis pada

tanggal 28 Mei 2015, untuk telepon genggam Nexus 5 dan Nexus 6, tablet Nexus 9, dan set-top box Nexus Player, di bawah nomor bentukan MPZ44Q. Pratyayang pengembang ketiga (MPA44G) dirilis pada tanggal 17 Agustus 2015 untuk Nexus 5, Nexus 6, Nexus 9 dan perangkat Nexus Player, dan diperbarui ke MPA44I yang membawa perbaikan yang berhubungan dengan profil Android for Work.

### 13. Android versi 1.0 Nougat

Android "Nougat" (kode nama N dalam pengembangan) adalah rilis 7.0 besar dari sistem operasi Android. Ini pertama kali dirilis sebagai pratinjau pengembang pada tanggal 9 Maret 2016, dengan gambar pabrik untuk perangkat Nexus saat ini, serta dengan "Program Beta Beta" baru yang memungkinkan perangkat yang didukung ditingkatkan versinya ke versi Android Nougat melalui over-the-air update. Rilis terakhir adalah pada tanggal 22 Agustus 2016. Pratinjau akhir pembuatannya dirilis pada tanggal 18 Juli 2016, dengan nomor bangunan NPD90G.

### 14. Android versi 1.0 oreo

Android Oreo adalah rilis utama ke 8 dari sistem operasi Android. Ini pertama kali dirilis sebagai preview pengembang pada tanggal 21 Maret 2017, dengan gambar pabrik untuk perangkat Nexus dan Pixel saat ini. Pratinjau pengembang terakhir dirilis pada tanggal 24 Juli 2017, dengan rilis stabil yang diharapkan pada bulan Agustus atau September 2017.

### 2.3.3. Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

#### 1. Application dan Widgets

*Appkication* dan *Widgets* ini adalah layer dimana kita berhubungan dengan aplikasi saja, di mana biasanya kita *download* aplikasi kemudian kita lakukan instalansi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua applikasi ditulis menggunakan bahasa pemrograman Java.

## 2. Applications Frameworks

Android adalah “*Open Development Platform*” yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan service background, mengatur alarm, dan menambahkan status notifications, dan sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (reuse).

Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut :

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

## 3. Libraries

Libraries ini adalah layer di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan di atas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL.

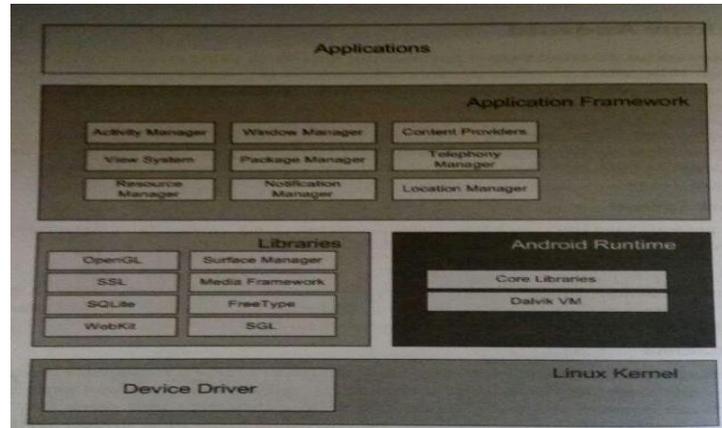
## 4. Android Run Time

Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan implementasi Linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android Run Time ada dua bagian yaitu, Core Libraries dan Dalvik Virtual Machine.

## 5. Kernel Linux

Linux kernel adalah layer dimana inti dari operating sistem dari Android itu berada. Berisi file-file system yang mengatur sistem Processing, memory,

resource, drivers, dan sistem-sistem operasi android lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6. jika anda ingin mempelajari linux kernel ini.[3]



**Gambar 2. 2 Arsitektur Android**

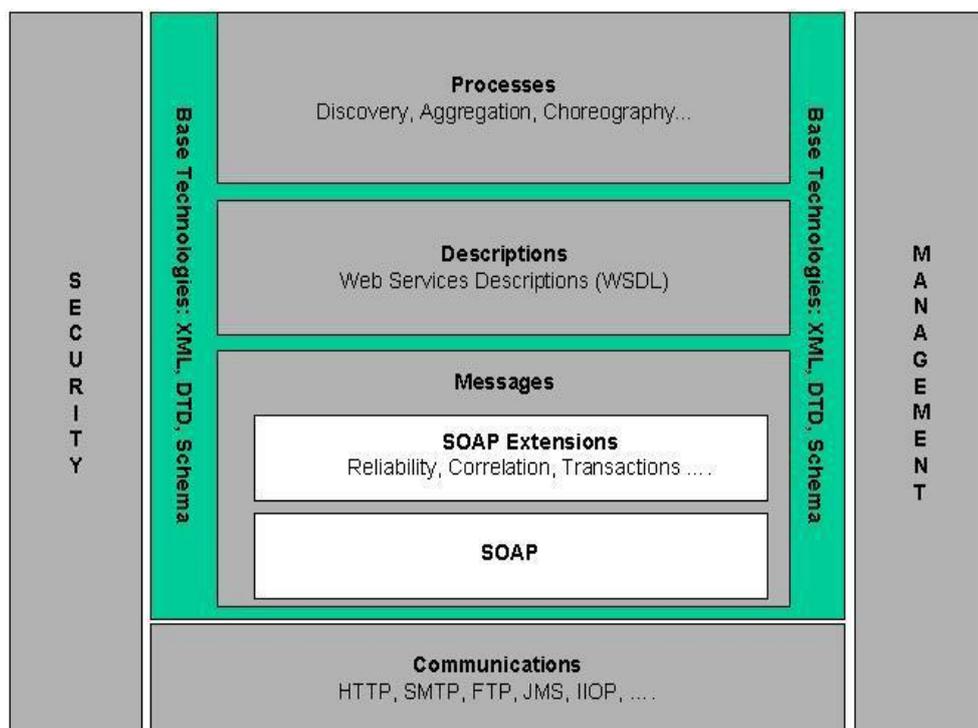
**Sumber Gambar Arsitektur Android[3]**

#### **2.4. Web Service**

Web service adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam platform dan antarmuka bahasa pemrograman yang independen. Tujuan web service adalah menjembatani komunikasi antar program, sehingga aplikasi yang satu dan aplikasi yang lain yang terdapat pada suatu jaringan yang sama atau pada jaringan berbeda dapat saling berkomunikasi asalkan menggunakan standar protokol yang ditetapkan oleh web service . Web Service digunakan pada pembangunan aplikasi ini untuk mengolah dan menjembatani komunikasi antar program dan juga basis data.[4]

##### **2.4.1. Arsitektur Web Service**

Arsitektur web service dibangun oleh beberapa layer dan teknologi yang saling berhubungan. Salah satu cara untuk menggambarkan arsitektur web service seperti pada gambar :



**Gambar 2. 3 Arsitektur Web Service**

**Sumber Gambar Arstektur Web Service[4]**

Web service disusun oleh tiga komponen standar, yaitu : [4]

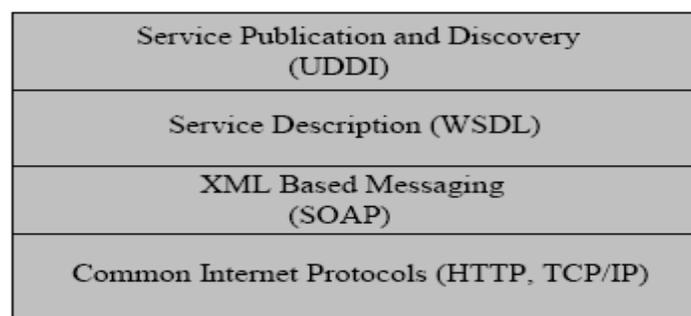
1. Simple Object Access Protocol (SOAP), yaitu protokol yang bertanggungjawab dalam pertukaran informasi dalam lingkungan jaringan terdistribusi.
2. Web Service Definition Language (WSDL), dokumen standar dituliskan dalam format XML, dan mendefinisikan kehadiran web service dalam suatu jaringan.
3. Universal Description, Discovery, and Integration (UDDI), yaitu suatu lokasi direktori yang berisikan service (layanan) dan bersifat bebas platform

(Platform Independent), dituliskan berbasis XML dan dapat diakses oleh entitas yang berada di dalam dan luar jaringan.

#### 2.4.2. Komponen Web Service

Web service memiliki 4 komponen yaitu :

1. Layer 1 : Protokol Internet standar (HTTP,TCP/IP)
2. Layer 2 : Simple Object Access Protocol (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. Layer 3 : Web Service Definition Language (WSDL), merupakan suatu bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.
4. Layer 4 :Universal Description, Discovery, and Integration (UDDI), merupakan lokasi direktori berisikan service dan bersifat bebas platform.



**Gambar 2. 4 Komponen Web Service**

**Sumber Gambar Komponen Web Service[4]**

#### 2.5. Java

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem

operasi dan bersifat open source. Java berasal dari pengembangan bahasa C++ sehingga strukturnya seperti bahasa C++. [5]

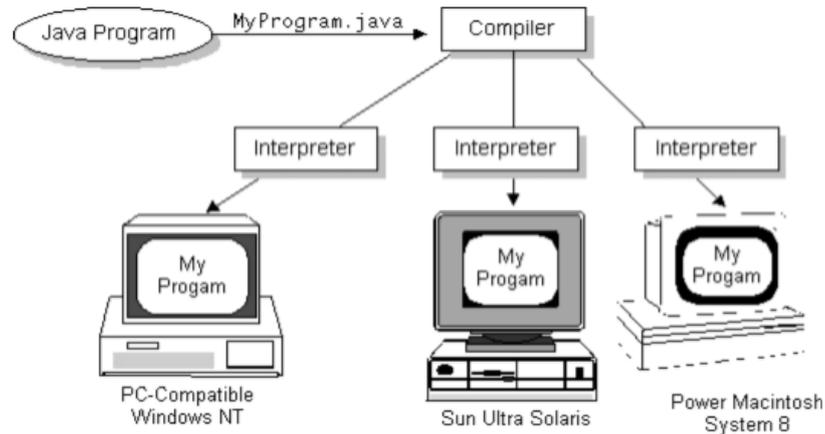
James Gosling dan Patrick Naughton adalah orang yang berhasil mengembangkan Java dalam suatu proyek dari SUN Microsystem. Awalnya James ingin memberi bahasa pemrograman tersebut dengan nama OAK yang berasal dari pohon yang terdapat pada kantor James Gosling, tetapi akhirnya diberi nama Java setelah mendapat ide sewaktu minum kopi.

Java sering digunakan oleh semua segmen, karena mampu merambah seluruh hal, dari pemrograman dekstop hingga pemrograman mobile (mobile programming) dan Java juga terkenal mudah digunakan, karena seluruh sistem operasi dapat menjalankan Java. Sistem operasi itu antara lain Window, Linux, DOS, Unix, dan lain-lainnya sehingga banyak pengguna memakainya.

Java telah berkembang dari semula diciptakan untuk pemrograman applet (program komputer yang telah dikompilasi menjadi bytecode, yaitu suatu keadaan program yang dapat dieksekusi langsung; tambahan di dalam suatu homepage selain gambar) yang berjalan di web browser menjadi bahasa pemrograman yang terkenal untuk pengembangan berbagai aplikasi komputer yang berjalan di beragam perangkat mulai dari handled devices seperti handphone, PDA (Personal Digital Assistant) sampai aplikasi khusus yang tersebar di berbagai macam komputer server.

### 2.5.1. Fase Pemrograman Java

Gambar dibawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program Java:[5]



**Gambar 2. 5 Fase Pemrograman Java**

**Sumber Gambar Fase Pemrograman Java[5]**

Langkah pertama dalam pembuatan sebuah program berbasis Java adalah menuliskan kode program pada text editor. Contoh text editor yang dapat digunakan antara lain : notepad, vi, emacs dan lain sebagainya. Kode program yang dibuat kemudian tersimpan dalam sebuah berkas berekstensi .java.

Setelah membuat dan menyimpan kode program, kompilasi file yang berisi kode program tersebut dengan menggunakan Java Compiler. Hasil dari adalah berupa berkas bytecode dengan ekstensi .class. Berkas yang mengandung bytecode tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan platform yang digunakan.

Macam-macam Text Editor / Tool yang umum digunakan untuk menuliskan source code Java, antara lain :

1. NotePad
2. JCreator ( <http://www.jcreator.com> )
3. Edit Plus ( <http://www.editplus.com> )
4. Gel ( <http://www.gexperts.com> )

5. JFrameBuilder ( <http://www.mars3000.com> )
6. NetBeans ( [www.netbeans.org](http://www.netbeans.org) )
7. Eclipse ( <http://www.eclipse.org> )
8. Borland JBuilder ( [www.borland.com](http://www.borland.com) )
9. JDeveloper ( <http://www.oracle.com/technology/jdev> )

Dari jenis text editor yang tersedia untuk bahasa java diatas, yang digunakan dalam pengembangan aplikasi Smart Laundry Berbasis Android adalah text editor Eclipse.

## 2.6. Google Maps API

Google Maps adalah sebuah jasa peta globe virtual gratis dan online yang disediakan oleh perusahaan Google, Inc dapat ditemukan di <http://maps.google.com/>. Google Map menawarkan peta yang dapat diseret dan gambar satelit serta street view untuk seluruh dunia dan juga menawarkan perencanaan rute dan pencari. Google Map Api merupakan aplikasi interface yang dapat ditampilkan pada halaman web. Google Maps sekarang ini sudah dapat digunakan dalam *smartphone* android. Untuk dapat mengakses Google Map, harus melakukan pendaftaran Api Key terlebih dahulu dengan data pendaftaran berupa nama domain web yang dibangun. Sedangkan untuk membuat Maps GPS Android yang perlu dilakukan pertama kali adalah mengunduh *library google play service* SDK yang ada pada android SDK Manager dan melakukan *import* ke dalam Eclipse. Setelah mendapatkan API Key barulah dapat membuat Aplikasi yang menggunakan fitur Google Maps ini.

## 2.7. RESTful

REST (*REpresentational State Transfer*) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai protocol untuk komunikasi data. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000.[6]

Pada arsitektur REST, REST server menyediakan *resources* (sumber daya/data) dan REST client mengakses dan menampilkan *resource* tersebut untuk

penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global ID. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

*Web service* yang berbasis arsitektur REST kemudian dikenal sebagai RESTful web services. Layanan web ini menggunakan metode HTTP untuk menerapkan konsep arsitektur REST.

### 2.7.1. Cara Kerja RESTful Web Service

Sebuah client mengirimkan sebuah data atau request melalui HTTP Request dan kemudian server merespon melalui HTTP Response. Komponen dari http request :[6]

1. Verb, HTTP method yang digunakan misalnya GET, POST, DELETE, PUT dll.
2. *Uniform Resource Identifier* (URI) untuk mengidentifikasikan lokasi resource pada server.
3. HTTP Version, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
4. Request Header, berisi metadata untuk HTTP Request. Contoh, type client/browser, format yang didukung oleh client, format dari body pesan, seting cache dll.
5. Request Body, konten dari data

## 2.8. PDF417 SDK

**PDF417 SDK** untuk Android format barcode yang dapat ditemukan pada berbagai dokumen dalam transportasi, manajemen inventaris, atau identitas & dokumen perjalanan. Untuk memindai dan mem-parsing barcode PDF417 pada dokumen identitas atau SIM. Jenis barcode yang di dukung pada PDF417 yaitu PDF417 (barcode 2D), kode QR, Kode 39, Kode 128, EAN 8, EAN 13, ITF, UPC A, UPC E, Aztec, Matriks Data.

Memilih untuk memkai PDF417 SDK ini dikarenakan BlinkID adalah alat yang hebat untuk cepat onboard atau mendaftarkan pengguna di aplikasi seluler. Ini

menggunakan kamera ponsel untuk memindai ID, SIM, paspor, izin kerja, visa, dan dokumen identitas lainnya. Jenis barcode didukung PDF417 (barcode 2D), kode QR, Kode 39, Kode 128, EAN 8, EAN 13, ITF, UPC A, UPC E, Aztec, Matriks Data PDF417 SDK berfungsi bahkan pada barcode yang rusak atau tidak lengkap, dan pada kertas yang kusut. Pemindaian berhasil dalam cahaya rendah atau miring. Kamera ponsel berkualitas rendah dan resolusi rendah didukung. PDF417 tersedia sebagai WEB API atau SDK seluler untuk iOS dan Android asli, serta untuk PhoneGap dan Xamarin. Pemindaian cepat dalam waktu nyata (100-900 ms), UI sepenuhnya dapat disesuaikan, Integrasi mudah ke dalam aplikasi apa pun, Charset: dukungan Unicode penuh.

## **2.9. OOP (Object Oriented Programming)**

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

### **1. Encapsulation (Pengkapsulan)**

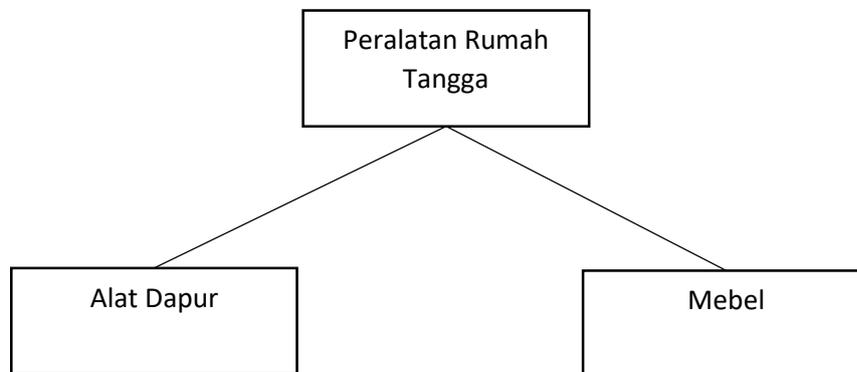
Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

Encapsulation sering disebut dengan penyembunyian informasi (information hiding). Konsep ini sebenarnya didasari pada fakta yang ada didunia

nyata bahwa tidak semua hal perlu diperlihatkan. Sebagai contoh untuk memperbesar volume suara pada TV kita hanya perlu menekan satu tombol tertentu saja. Kita tidak perlu tahu atau bahkan tidak mau tahu bagaimana proses dibelakang. Padahal kalau kita perhatikan ada serangkaian banyak proses yang harus dilakukan oleh TV tersebut, dan semua tidak diperlihatkan kepada kita.

## 2. Inheritance (Pewarisan)

Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek, demikian seterusnya. Inheritance mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di antara kelas yang mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya. Kelas Objek dapat didefinisikan atribut dan service dari kelas Objek lainnya. Inheritance



**Gambar 2. 6 Inheritance**

**Sumber Gambar OOP Inheritance [7]**

menggambarkan generalisasi sebuah kelas.

## 3. Polymorphism (Polimorfisme)

Kadangkala class yang berbeda mempunyai nama operation yang berbeda. Sebagai contoh operation 'buka' bisa dipakai untuk membuka pintu, membuka jendela, membuka buku, membuka seminar dan lain-lain. Meski kedengarannya

sama, sebenarnya apa yang dilakukan berbeda. Konsep inilah yang disebut dengan polimorphisme.

Polimorphisme merupakan konsep yang sangat handal bagi pengembang perangkat lunak untuk pemisahan secara jelas diantara sub sistem yang berbeda. Dengan demikian sebuah sistem akan bisa dimodifikasikan secara mudah karena hanya dibutuhkan interface antar class. Sebagai contoh kita bisa membuat operation hitungGaji. Operasi hitungGaji ini bisa dipakai untuk menghitung gaji semua pegawai tetap, pegawai paruh waktu maupun pegawai kontrak. Dengan konsep polimorphisme kita bisa menghitung gaji masing-masing pegawai. Sekedar ilustrasi, untuk menghitung gaji karyawan tetap jelas akan memperhitungkan jabatan dan golongan. Sedangkan untuk pegawai paruh waktu akan memperhitungkan jumlah jam kerja. Adapun untuk pegawai kontrak tidak ada tunjangan pensiun.

Alasan menggunakan OOP metode dalam penelitian ini adalah karena aplikasi yang dibuat dapat terstruktur dan lebih rapih. Dan dapat lebih mudah untuk dianalisa program yang kita buat apa lagi kalau program yang dibuat susah atau ribet. Apa lagi OOP membantu kita membuat dan membaca kode yang dibuat.

#### 4. Abstraksi

Abstraksi bertujuan untuk memfilter properties dan operation pada suatu obyek, sehingga hanya tinggal properties dan operation yang dibutuhkan saja. Seringkali masalah yang berbeda membutuhkan sejumlah informasi yang berbeda pula pada areal yang sama. Sebagai contoh pada kasus TV di depan. Ketika kita membuat program komputer untuk mengatur volume suara, perubahan channel dan pengaturan kontras, kita mungkin harus membuang attribute nomor seri karena tidak terlalu berguna. Akan tetapi ketika kita akan menelusuri transaksi penjualan TV, maka kita butuh nomor seri dari setiap TV, maka kita butuh nomor seri dari setiap TV yang terjual.

#### 5. Message Sending

Dalam sistem OO, obyek-obyek saling berkomunikasi satu sama lain dengan mengirimkan pesan. Suatu obyek mengirim sebuah pesan kepada obyek yang lain untuk menjalankan sebuah operation dan obyek yang menerima akan memberikan respon untuk menjalankan operation tersebut.

Ilustrasi tentang pengiriman pesan ini bisa dilihat pada kasus pesawat TV di depan. Jika kita ingin menonton TV, kita tinggal menekan tombol-tombol yang ada pada remote control dan kemudian TV-pun menyala. Sebenarnya fakta yang terjadi adalah obyek remote mengirim pesan kepada obyek TV untuk menjalankan operation menyala. Obyek TV menerima pesan ini dan menjalankan operation untuk menyalakan TV.

Contoh nyata yang lain bisa dilihat pada pencetakan slip gaji pada kasus ini kita akan mempunyai class karyawan bertanggung jawab untuk mengetahui berapa jumlah gaji yang diterima. Obyek berikutnya adalah cetakSlipGaji yang bertanggung jawab untuk mencetak slip gaji setiap bulannya. Untuk bisa mencetak slip gaji, obyek slipGaji harus mengetahui berapa gaji yang diperoleh karyawan. Oleh karena itu obyek cetakSlipGaji kemudian mengirimkan pesan kepada obyek karyawan untuk menanyakan berapa gaji yang harus dibayarkan.

## 6. Associaton

Association (asosiasi) adalah hubungan anatar obyek yang saling membutuhkan. Hubungan ini bisa satu arah ataupun lebih dari satu artah. Sebagai contoh saat kita menyalakan TV, bisa dikatakan kita berasosiasi dengan TV secara satu arah. Ilustrasi lain yang menunjukkan hubungan lebih dari satu arah bisa dilihat pada pertemanan si A dan si B. pertemanan disini tidak akan jalan kalau hanya berjalan satu arah. Pertemanan membutuhkan hubungan dua arah. Selain dengan sesama obyek, sebuah class bisa juga berasosiasi dengan lebih dari satu class. Seorang sopir mungkin mengendarai sebuah mobil ataupun sebuah bis.

Multiplicity adalah sebuah aspek yang penting dalam asosiasi antar obyek. Ini berkaitan dengan sejumlah obyek dalam satu class yang saling berasosiasi. Sebagai contoh sebuah kursus diajar oleh seorang instruktur. Kursus dan instruktur

adalah asosiasi satu ke satu (one-to-one). akan tetapi dalam kasus perkuliahan, seorang instruktur mungkin akan mengajar lebih dari satu kursus selama satu semester. Dalam kasus ini kursus dan instruktur berasosiasi satu ke banyak (one-to-many).

## 7. Aggregation

Aggregation (agregasi) adalah bentuk khusus dari asosiasi yang menggambarkan seluruh bagian suatu obyek merupakan bagian dari obyek yang lain. Sebagai contoh sebuah komputer dibuat dari sekumpulan komponen seperti CPU, keyboard, mouse, monitor dan lain-lain.

Salah satu bentuk agregasi yang melibatkan hubungan yang sangat erat antar obyek dan komponen obyek disebut composition. Kata kunci composition adalah komponen itu hanya bisa ada sebagai komponen pada obyek composition. Sebagai contoh baju mengandung unsur lengan, kerah, kancing dan sebagainya. Kalau kita ambil kerah saja maka tidak akan ada gunanya.

## 2.10. OOAD (Object Oriented analysis and Design)

Pendekatan rekayasa perangkat lunak dari sebuah sistem yang terdiri dari sekelompok objek yang saling berinteraksi, dan setiap objek itu mewakili beberapa entitas. Yang ditandai dengan adanya sebuah kelas, elemen data dan perilaku dari objek tersebut.

### 1. Objek

Objek didefinisikan sebagai konsep, abstraksi atau benda dengan batasan dan arti untuk suatu masalah. Semua objek mempunyai identitas yang berbeda dengan lainnya. Istilah identitas berarti bahwa objek dibedakan oleh sifat yang melekat dan bukan dengan uraian sifat yang dimilikinya. Contohnya : kembar identik, walaupun mereka nampak seperti sama, tetapi merupakan dua orang yang berbeda.

### 2. Kelas

Suatu object class menggambarkan kumpulan dari objek yang mempunyai sifat (atribut), perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum. Contoh : Orang, perusahaan , binatang, proses adalah objek.

Setiap orang mempunyai umur, IQ, dan mungkin pekerjaan. Setiap proses mempunyai pemilik, prioritas, list dari sumber daya yang dibutuhkan. Objek dan object class sering sama sebagai benda dalam deskripsi masalah.

### 3. Diagram Objek

Diagram objek melengkapi notasi grafik untuk pemodelan objek, kelas dan relasinya dengan yang lain. Diagram objek bermanfaat untuk pemodelan abstrak dan membuat perancangan program.

Kenapa memilih metode OOAD metode dalam penelitian ini karena metode OOAD mempunyai relasi obyek dengan entitas umumnya, dapat di mapping dengan baik seperti kondisi pada dunia nyata dan keterkaitan dalam sistem . Pendekatan visual, membuat metode ini mudah dimengerti oleh user maupun programmer.

#### 2.11. UML (Unified Modelling Language)

UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain.

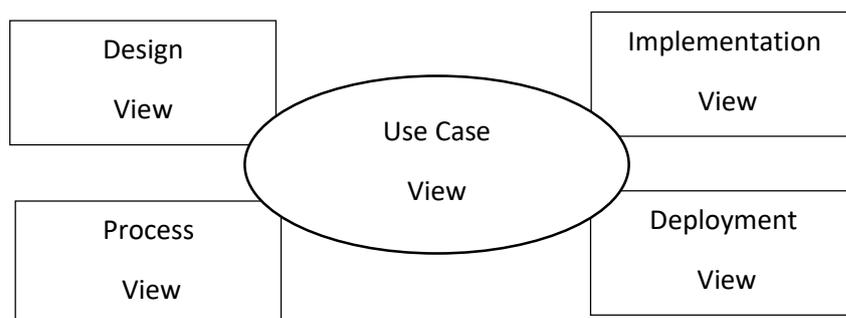
UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object modelling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design kedalam empat tahapan iteratif, yaitu : identifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh

didasarkan pada analisis terstruktur dan pemoelan entity-relationship. Tahapan utama dalam metodologi ini adalah analisis, design sistem, design obyek dan implementasi. Keunggulan metode ini adalah OOSE dari Jacobson lebih memberi penekanan pada Use Case. OOSE memiliki 3 tahapan yaitu membuat model requirement dan analisis, design dan implementasi, dan model pengujian (test model). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya.

UML juga hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam OOAD (*Object Oriented Analysis & Design*). Kontribusi untuk UML telah dihasilkan dari banyak perusahaan-perusahaan ternama diantaranya Digital Equipment Corp, Hewlett-Packard Company, I-Logic, Intelicorp, IBM, Icon Computing, Electronic Data Services Corporation, MCI Software, Taskon A/S, Unisys Platinum Technologies, Ptech, Taskon & Reich Technologies dan Softem.

UML dibangun atas model 4+1 view. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam view dimana salah satu diantaranya use case view. Use case view ini memegang peran khusus untuk mengintegrasikan content ke view yang lain.



## **Gambar 2. 7 Use Case View**

### **Sumber Gambar UML Use Case View[7]**

#### **1. Use Case Diagram**

Use case adalah deskripsi fungsi dari sebuah system dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan system disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, system yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan use case adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

Dalam pembicaraan tentang use case, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan system.

Model use case adalah bagian dari model requirement (Jacobson et all, 1992). termasuk disini adalah problem domain object model dan penjelasan tentang user interface. Use case memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user. Diagram use case menunjukkan 3 aspek aspek dari system yaitu aktor, use case dan system/sub system boundary. Aktor mewakili peran orang, system yang lain atau alat ketika berkomunikasi dengan use case. Gambar di bawah mengilustrasikan aktor, use case, dan boundary.

#### **2. Class Diagram**

Class, dalam notasi UML digambarkan dengan kotak. Nama class menggunakan huruf besar di awal kalimatnya dan diletakan di atas kotak. Bila class mempunyai nama yang terdiri dari 2 suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar.

Attribute adalah property dari sebuah class. Attribute ini melukiskan batas nilai yang mungkin ada pada obyek class. Sebuah class mungkin mempunyai nol atau lebih attribute. Secara konvensi, jika nama attribute terdiri atas satu suku kata, maka ditulis dengan huruf kecil. Akan tetapi jika nama attribute mengandung lebih dari satu suku kata maka semua suku kata digabungkan dengan suku kata pertama menggunakan huruf kecil dan awal suku kata berikutnya menggunakan huruf besar.

UML memberikan pilihan untuk memberikan informasi tambahan untuk attribute. Tipe dari setiap attribute bisa ditambahkan di sini. Tipe tipe yang mungkin ditambahkan disini diantaranya string, floating-point number, integer dan boolean. Untuk menunjukkan tipe gunakan titik dua (:) untuk memisahkan nama attribute dan tipe. Nilai default sebuah attribute bisa juga ditambahkan jika diinginkan.

### 3. State Diagram

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

### 4. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek message (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case.

Komponen utama sequence diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. Message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertical.

**Obyek/participant** diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram. Pengertian obyek hanya ada di UML 1, sedangkan di UML 2 istilah obyek diganti dengan **participant**.

Setiap participant terhubung dengan **lifeline**. Sepanjang lifeline ada kotak yang disebut **activation**. Activation mewakili sebuah eksekusi operasi dari participant. Panjang kotak ini berbanding lurus dengan durasi activation.

Sequence diagram dilakukan ketika ingin mengetahui perilaku beberapa obyek pada use case tunggal. Sequence diagram bagus dalam menunjukkan kalaborasi diantara obyek, namun tidak begitu bagus dalam memberikan definisi yang pasti tentang perilaku tersebut.

Untuk melihat perilaku suatu obyek pada banyak use case akan lebih baik jika menggunakan State Machine Diagram. Akan tetapi untuk melihat perilaku terhadap banyak use case atau banyak thread, perlu dipertimbangkan untuk menggunakan activity diagram.

## 5. Collaboration Diagram

Collaboration diagram adalah perluasan dari obyek diagram. (obyek diagram menunjukkan obyek-obyek dan hubungannya satu dengan yang lain). collaboration menunjukkan message-message obyek yang dikirimkan satu sama lain.

Untuk menunjukkan sebuah pesan, buatlah tanda panah di dekat garis asosiasi diantara 2 obyek. Arah panah menunjukkan obyek yang menerima pesan. Label di dekat panah menunjukkan pesannya apa. Tipikal message meminta kepada obyek yang menerimannya untuk menjalankan salah satu operationya. Sepasang tanda kurung digunakan untuk mengakhiri message. Jika ada parameter bisa diletakkan diantara tanda kurung.

Diantara collaboration diagram dan sequence diagram bisa saling mengisi. Dengan demikian pada collaboration diagram kita bisa tambahkan nomor urut pada label sebuah message untuk menunjukkan urutan informasi. Tiitk dua (:) perlu digunakan untuk memisahkan nomor dengan message.

## 6. Activity Diagram

Activity diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram mempunyai peran

seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa. Contoh sederhana activity diagram bisa dilihat pada gambar diatas. Gambar tersebut menjelaskan tentang aliran saat proses penerimaan order.

Dari gambar diatas terlihat bahwa pengisian order dan pengiriman invoice terjadi secara paralel. Intinya tidak jadi masalah mengenai mana yang terlebih dahulu harus diselesaikan.

## 7. Component Diagram

Component diagram adalah bagian fisik dari sebuah sistem, karena menetap di komputer, bukan di benak para analis. Komponen bisa berupa tabel, file data, file exe, dan lain-lain (dynamic link library), dokumen dan lain-lain.

Apa hubungan component dengan class? Component adalah implementasi software dari sebuah class. Class mewakili abstraksi dari serangkaian attribute dan operation. Hal terpenting yang perlu diingat tentang class dan component bisa jadi merupakan implementasi dari lebih dari sebuah class.

Jika component menetap di sebuah komputer dan bekerja sebagai bagian dari sistem, mengapa perlu repot-repot memodelkannya? Dengan memodelkan component dan relasinya maka :

1. Klien bisa melihat struktur sistem yang sudah selesai.
2. Pengembang mempunyai struktur untuk panduan kerja.
3. Dokumentator bisa memahami apa yang mereka tulis.
4. Siap untuk digunakan kembali untuk proyek lain.

Bagian terpenting dari keempat hal diatas adalah yang terakhir yaitu **reusability**.

Dalam iklim persaingan seperti sekarang ini, semakin cepat kita bisa membangun sebuah sistem akan semakin kompetitif. Dengan demikian jika kita bisa membangun component untuk sebuah sistem dan menggunakannya lagi untuk sistem yang lain jelas akan meningkatkan daya saing.

## 8. Deployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakan bagian-bagian software yang berjalan pada bagian-bagian hardware.

Bagian utama hardware/perangkat keras adalah node; yaitu nama umum untuk semua jenis sumber komputasi. Ada 2 tipe node yang mungkin. Processor adalah node yang bisa mengeksekusi sebuah component, sedangkan device tidak. Device adalah perangkat keras tipikalnya menjadi interface dengan dunia luar[7].

## 2.12. Google Cloud Platform (GCP)

Google Cloud Platform merupakan layanan *public cloud* computing dari *google* yang terdiri dari beragam layanan. Platform dari google ini menyediakan beragam layanan hosting mulai untuk komputasi, storage dan application development yang berjalan pada hardware google. Google cloud platform service dapat diakses oleh pengembang software, administrator cloud, dan professional IT lainnya menggunakan internet public atau melalui jaringan dedicated.

Layanan cloud computing yang diberikan oleh Google Cloud Platform antarlain:

1. Google Compute Engine: Layanan infrastruktur as a service (IaaS) yang menyediakan pengguna akan virtual machine yang bisa dibuat secara instan untuk workload hosting.
2. Google App Engine: Layanan platform as a service (PaaS) menawarkan pengembang software untuk mengakses Google hosting yang mudah terukur. Pengembang dapat juga menggunakan software developer kit (SDK) untuk mengembangkan produk software yang berjalan pada Google App Engine.
3. Google Cloud Storage: Platform cloud storage atau penyimpanan berbasis cloud yang didesain untuk menyimpan data yang besar, tidak terstruktur. Google juga menawarkan pilihan penyimpanan database termasuk Cloud Datastore untuk penyimpanan NoSQL non-

relational, penyimpanan Cloud SQL untuk MySQL, dan database Google Cloud Bigtable.

4. Google Container Engine: Sistem manajemen dan pengukuran untuk Docker container yang berjalan didalam Google public cloud. Google Container Engine berbasis pada engine Google Kubernetes.

Google Cloud Platform juga menyediakan layanan cloud seperti pemrosesan dan analisis data, seperti Google BigQuery untuk SQL yang digunakan untuk memproses set data multi terabyte. Sebagai tambahan, Google Cloud Dataflow merupakan layanan pemrosesan data yang khusus untuk analisis, extract transform and load (ETL), dan real-time computational projects.[8]

### **2.13. Google Cloud Messaging (GCM)**

Salah satu implementasi dari push notification yang dapat digunakan adalah Google Cloud Messaging (GCM). GCM terdiri dari tiga komponen, yaitu aplikasi yang beroperasi pada sistem operasi Android yang terhubung dengan GCM, application server yang mengirimkan notifikasi dan pesan kepada aplikasi, dan server GCM yang bertugas untuk menyampaikan notifikasi dan pesan dari application server.[9]

### **2.14. Geofencing**

Geofencing yang merupakan sebuah perimeter virtual pada wilayah geografis yang menggunakan layanan berbasis lokasi digunakan untuk pembatasan wilayah.

Geofence adalah sebuah pembatas digital yang dapat diterapkan untuk memantau pergerakan obyek pada area tertentu, sehingga ketika GPS mendeteksi sebuah obyek yang dipantau melewati batas tersebut, maka sistem akan memberikan notifikasi kepada pemantau. Sistem geofencing merupakan sistem yang dapat menganalisa dan melacak posisi obyek secara otomatis dan memberi laporan kapanpun dan dimanapun ketika obyek keluar atau masuk ke area-area geofence yang sebelumnya sudah ditentukan oleh pemantau. Area geofence tersebut merupakan area berbentuk geometri virtual yang membatasi lokasi tertentu, seperti lokasi kantor, gudang, lokasi pelanggan, dan sebagainya. Sistem geofence ini dapat

mengirim notifikasi kepada pemilik apabila kendaraan keluar atau masuk area geofence tertentu.[10]

### 2.15. Android Studio

Android Studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini di umumkan oleh *Google* pada 16 Mei 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (*Android Development Tools*). Android Studio memiliki fitur:

- a. Projek berbasis pada *Gradle Build*.
- b. *refactory* dan pembenahan bug yang cepat.
- c. *Tools* baru yang bernama “*Lint*” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *proguard* *And-signing* untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah.
- f. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang berkembang.[11]

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

GPS adalah singkatan dari Global Positioning System, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (receiver) di permukaan, dimana GPS receiver ini akan mengumpulkan informasi dari satelit GPS, seperti:

1. Waktu. GPS receiver menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi:
  - a. Latitude
  - b. Longitude

### c. Elevasi

3. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
4. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
5. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS receiver.
6. Komulasi data. GPS receiver dapat menyimpan informasi track, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya. [12]

## 2.17. MongoDB

MongoDB adalah salah satu produk database noSQL Open Source yang menggunakan struktur data JSON untuk menyimpan datanya. mongoDB adalah merupakan database noSQL yang paling populer di internet. mongoDB sering dipakai untuk aplikasi berbasis Cloud, Grid Computing, atau Big Data. Dalam konsep MongoDB tidak ada yang namanya table, kolom ataupun baris yang ada hanyalah collection (ibaratnya table), document (ibaratnya record). Data modelnya sendiri disebut BSON dengan truktur mirip dengan JSON.

## 2.18. Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan proses menelusuri dan mempelajari sebuah program dalam rangka menemukan kesalahan pada perangkat lunak sebelum diserahkan kepada pengguna. Definisi lainnya dari pengujian perangkat lunak yaitu semua kegiatan yang bertujuan untuk mengevaluasi atribut atau kemampuan dari sebuah program atau sistem dan menentukan bahwa pengujian perangkat lunak bertemu dengan hasil yang diperlukan [13].

### 2.18.1. Pengujian Black Box

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekrayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. [13] Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi–fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kerja.
5. Inisialisasi dan kesalahan terminasi.

Adapun Menurut pandangan Agustiar Budiman berpendapat bahwa “Pengujian blackbox merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak diuji apakah telah sesuai dengan yang diharapkan” maka dapat disimpulkan bahwa metode *black box* digunakan untuk menguji sistem yang menitik beratkan pada kinerja, antarmuka dan spesifikasi tanpa menguji kode program sehingga pengembang dapat melatih seluruh syarat-syarat fungsional program.

### **2.18.2. Pengujian Beta**

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana dilakukan pengujian secara langsung ke tempat dimana sistem diimplementasikan. Pengujian beta bersangkutan mengenai kepuasan pengguna dengan kandungan poin pemenuhan kebutuhan dari tujuan awal pembangunan sistem dan tampilan antarmuka dari sistem tersebut. Pengujian beta dilakukan melalui sebuah teknik pengambilan data, baik melalui wawancara atau kuesioner kepada pihak yang terlibat, yang nantinya akan menggunakan sistem. [13]







