BAB 2
TINJAUAN PUSTAKA

2.1 Oleh-oleh

Oleh-oleh atau biasa disebut cenderamata yaitu sesuatu yang dibawa oleh beberapa orang yang berpergian saat kembali ke daerah asalnya untuk kenangan yang terkait dengan benda itu. Dalam bahasa indonesia, istilah ini sering disamakan dengan kenang-kenangan, tanda mata, oleh-oleh, atau souvenir.

Oleh-oleh memang sudah terkait dengan nilai budaya dalam masyarakat pada umumnya. Dari aspek sosiologi, orang pada umumnya memiliki rasa kekeluargaan yang amat kental. Hubungan antara orang-orangpun sangat dekat. Meskipun keeratan tali kekeluargaan lebih umum ada di desa, bukan berarti masyarakat di kota besar tidak menganut nilai-nilai seperti itu. Tipe keluarga besar dan akrab merupakan tipe umum masyarakat pada umumnya. Jadi kebiasaan memberikan oleh-oleh menggambarkan kedekatan. [1]

2.2 Teknologi Mobile

Merupakan teknologi komunikasi yang bergerak, dengan kata lain teknologi yang telah tidak memakai penghubung kabel (nirkabel) dalam pemakaian teknologinya. Teknologi komunikasi mobile bisa disebut jaringan tanpa kabel yang pemakaiannya memakai penghubung sinyal. Teknologi komunikasi mobile adalah perkembangan dari teknologi komunikasi nirkabel yang semakin pesat dalam perkembangannya. [3]

2.2.1 Android

Android adalah sistem operasi berbasis linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh android, inc., dengan dukungan finansial dari google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium

dari perusahaan-perusahaan *hardware*, *software*, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. [3]

Pada bulan Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk Android, dan sekitar 50 miliar aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi bergerak. Di Google I/O 2014, Google melaporkan terdapat lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013. [4]

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia. Sifat Android yang terbuka juga telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain. [5]

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk-produk Samsung, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat Android berbeda dengan beragam versi.

Maka dari sana bisa disimpulkan bahwa android sangat cocok dalam menciptakan sebuah aplikasi yang bisa membantu masyarakat. Tentunya dengan versi minimum *ice cream sandwich*, dikarenakan versi yang paling rendah di smartphone dipasaran pada umumnya adalah versi *ice cream sandwich* dan selain itu dengan mempertimbangkan syarat minimum untuk mendukung fitur yang akan digunakan dalam aplikasi yang akan dibuat. [6]

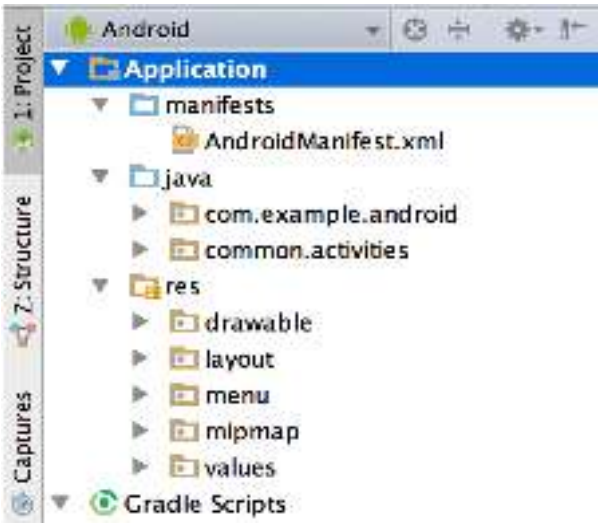
2.2.2 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat

pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya [7]:

a. Sistem versi berbasis Gradle yang fleksibel
b. Emulator yang cepat dan kaya fitur
c. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
d. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
e. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
f. Alat pengujian dan kerangka kerja yang ekstensif
g. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
h. Dukungan C++ dan NDK
i. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

1. Struktur Proyek



Gambar 2.1 File Proyek di tampilan Android [7]

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

a. Modul aplikasi android
b. Modul Pustaka
c. Modul Google App Engine

Secara default, Android Studio akan menampilkan file proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan dalam gambar 2.1. Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke file sumber utama proyek Anda.

Semua file versi terlihat di bagian atas di bawah Gradle Scripts dan masing-masing modul aplikasi berisi folder berikut:

a. Manifests: Berisi file AndroidManifest.xml.
b. Java: Berisi file kode sumber Java, termasuk kode pengujian JUnit.
c. Res: Berisi semua sumber daya bukan kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur proyek Android pada disk berbeda dari representasi rata ini. Untuk melihat struktur file sebenarnya dari proyek ini, pilih Project dari menu tarik turun Project (dalam gambar 1, struktur ditampilkan sebagai Android).

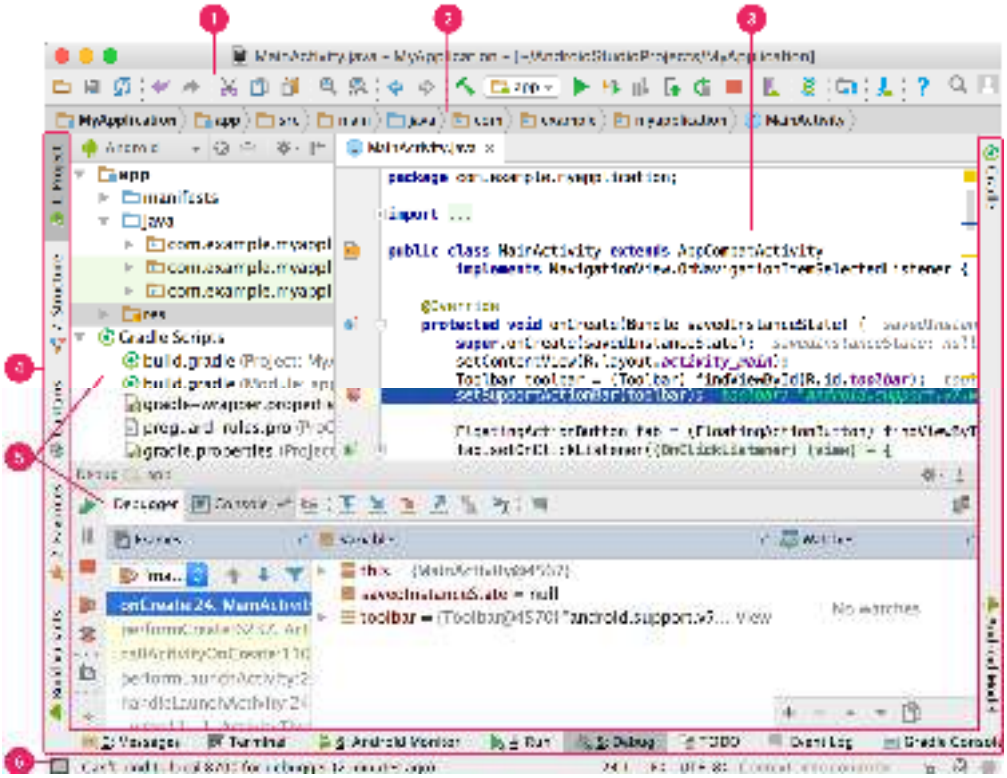
Anda juga bisa menyesuaikan tampilan file proyek untuk berfokus pada aspek tertentu dari pengembangan aplikasi Anda. Misalnya, memilih tampilan Problems dari tampilan proyek Anda akan menampilkan tautan ke file sumber yang berisi kesalahan pengkodean dan sintaks yang dikenal, misalnya tag penutup elemen XML tidak ada dalam file tata letak.



Gambar 2.2 File Proyek Dalam Tampilan Problems Arsitektur Android [7]

2. Antarmuka Pengguna

Jendela utama Android Studio terdiri dari beberapa bidang logika yang diidentifikasi dalam gambar 3.



Gambar 2.3 Jendela Utama Android Studio [7]

1. **Bilah alat** memungkinkan Anda untuk melakukan berbagai jenis tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.
2. **Bilah navigasi** membantu Anda bernavigasi di antara proyek dan membuka file untuk diedit. Bilah ini memberikan tampilan struktur yang terlihat lebih ringkas dalam jendela Project.
3. **Jendela editor** adalah tempat Anda membuat dan memodifikasi kode. Bergantung pada jenis file saat ini, editor dapat berubah. Misalnya, ketika melihat file tata letak, editor menampilkan Layout Editor.
4. **Bilah jendela** alat muncul di luar jendela IDE dan berisi tombol yang memungkinkan Anda meluaskan atau menciutkan jendela alat individual.

5. **Jendela alat** memberi Anda akses ke tugas tertentu seperti pengelolaan proyek, penelusuran, kontrol versi, dan banyak lagi. Anda bisa meluaskan dan juga menciutkannya.
6. **Bilah status** menampilkan status proyek Anda dan IDE itu sendiri, serta setiap peringatan atau pesan.

Anda bisa menata jendela utama untuk memberi Anda ruang layar yang lebih luas dengan menyembunyikan atau memindahkan bilah alat dan jendela alat. Anda juga bisa menggunakan pintasan keyboard untuk mengakses sebagian besar fitur IDE.

Anda dapat menelusuri seluruh kode sumber, basis data, tindakan, elemen antarmuka pengguna, dan seterusnya setiap saat dengan menekan tombol Shift dua kali, atau mengeklik kaca pembesar di sudut kanan atas dari jendela Android Studio. Ini akan sangat berguna misalnya saat Anda mencoba menemukan tindakan IDE tertentu yang Anda lupakan cara memicunya.

Maka dari penjelasan yang sudah dipaparkan, dalam pembuatan suatu program diperlukan sebuah aplikasi yang mendukung dan mutakhir untuk mencapai hasil yang maksimal. Dengan beberapa pertimbangan yang ada, dan juga dikarenakan aplikasi yang sepenuhnya didukung dalam pembuatan aplikasi android sekarang adalah android studio, maka *developer android* diharapkan menggunakan android studio sebagai alat pembuatan program aplikasi yang ingin dibuat, khususnya pada aplikasi pencarian oleh-oleh yang akan dibuat.

2.2.3 Android SDK (Software Development Kit)

Merupakan tools API (*Application Programming Interface*) yang dibutuhkan untuk memulai mengembangkan aplikasi di platform android memakai bahasa pemrograman java. Setiap kali Google merilis versi terbaru dari android, sebuah SDK yang sesuai juga dirilis. Untuk dapat menulis program dengan fitur terbaru, pengembang hars men-download dan menginstall SDK versi masing-masing untuk ponsel tertentu. Android memberikan kesempatan pembuatan aplikasi yang diperlukan, namun bukan merupakan aplikasi bawaan *handphone/ smartphone*. Beberapa fitur android yang paling penting antara lain [7] :

a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable.*
b. Mesin *virtual dalvik* dioptimalkan untuk perangkat mobile.
c. *Integrated browser* berdasarkan *engine open source webkit.*
d. Grafis yang dioptimalkan dan dikung oleh *libraries* grafis 2D maupun 3D berdasarkan spesifikasi OpenGL ES 1.0 (*Optional Acceleration Hardware*).
e. SQLite untuk menyimpan data (*database).*
f. Media support yang mendukung audio, video, dan gambar (tergantung perangkat keras).
g. WiFi, EDGE, 3G, dan Bluetooth (tergantung perangkat keras).
h. Accelerometer, Kompas, GPS, dan kamera (tergantung perangkat keras).
i. Lingkungan development yang lengkap dan kaya termasuk plugin untuk android studio, tool untuk debugging, kinerja memori serta profil, dan perangkat emulator.

Untuk mengembangkan sebuah aplikasi android *develover* memerlukan android SDK dalam pembuatan aplikasi yang di inginkan. Begitu pula dengan pembuatan aplikasi pencarian oleh-oleh diperlukan dukungan android SDK, agar aplikasi dapat berjalan dengan semestinya, dan gambaran aplikasi yang di inginkan dapat menjadi kenyataan.

2.3 LBS (Location Base Service)

Merupakan sebuah fitur yang bisa diakses menggunakan perangkat bergerak melalui jaringan dan dapat menunjukkan posisi secara geografis tempat perangkat bergerak tersebut. LBS bisa digunakan untuk fitur mengidentifikasi lokasi dari seseorang atau objek tertentu, seperti menemukan lokasi taman terdekat atau lokasi lainnya. [8]

LBS ini sudah banyak diterapkan oleh banyak aplikasi seperti GO-JEK, GRABBIKE, dan UberMotor. Konsep yang digunakan adalah para konsumen melakukan order dan sistem akan menampilkan *driver* yang berada dekat dengan konsumen. Selain aplikasi tersebut, banyak para developer yang membangun aplikasi *Location Based Service* seperti aplikasi pencarian oleh-oleh yang akan dibuat yang konsepnya mencari rute terdekat dalam mencari toko oleh-oleh.

Ada dua unsur utama dari LBS adalah [8] :

a. *Location Manager* (API Maps)

Menyediakan perangkat bagi sumber daya atau source untuk *Location Based Service,* API (*Aplication Programming Interface*) memfasilitasi untuk menunjukkan atau memanipulasi peta beserta layanan lainnya seperti menunjukkan *street* (jalan), satelit, serta gabungannya sesuai yang diinginkan. Paket ini berada dalam "*com.google.android.maps*"

b. *Location Providers* (API Location)

Menyediakan teknologi pencarian lokasi yang dipakai oleh perangkat/*device*. API (*Aplication Programming Interface)* ini berhubungan dengan data *Global Positioning System* (GPS) dan data lokasi secara real-time. API *location* berada pada paket "*android.location"*. Dengan menggunakan *location manager* ini, pengguna bisa menentukan lokasi pengguna saat ini, *track* gerakan/perpindahan, serta mendeteksi perpindahan dengan kedekatan dengan lokasi tertentu.

Terdapat lima komponen pendukung utama dalam teknologi layanan berbasis lokasi, antara lain [8] :



Gambar 2.4 Alur location based service [8]

1. Jaringan komunikasi, bagian ini berperan sebagai jalur penghubung yang bisa mengantarkan file-file yang dikirimkan oleh *user* dari piranti *mobile*-nya yang selanjutnya diantarkan kepada bagian yang menyediakan layanan serta selanjutnya hasil permintaan itu diantarkan kembali oleh bagian yang menyediakan layanan ke *user*.

2. Aplikasi dan penyedia layanan, adalah bagian LBS yang menyediakan macam-macam layanan yang dapat dipakai oleh *user*. Sebagai perumpaan ketika *user* menginginkan layanan supaya dapat tahu koordinat saat itu, maka penyedia layanan dan aplikasi secara langsung mengelola permintaan itu, dimulai dari perhitungan dan menentukan koordinat *user*, mencarikan data di *Yellow Pages* sesuai permintaan, menemukan rute jalan, serta lain sebagainya.
3. Piranti *Mobile*, merupakan bagian penting dari LBS. Piranti ini berperan sebagai perantara (*tool*) untuk pengguna dalam meminta informasi. Informasi yang dihasilkan diminta bisa berupa gambar, suara, teks, maupun lain sebagainya. Piranti *mobile* yang bisa dipakai bisa berupa laptop, PDA, *smartphone*. Selain itu, piranti ini bisa juga berfungsi sebagai alat navigasi pada alat transfortasi seperti halnya alat navigasi berbasis Global Positioning System.
4. Konten dan penyedia data, penyedia layanan kadang-kadang mengamankan informasi dan seluruh data yang diolahnya. Di karenakan dapat menjadi bermacam-macam informasi dan data yang diolahnya asalnya dari pihak ketiga atau pengembang yang memang punya *otoritas* untuk menyimpannya.
5. Komponen *positioning* (penunjuk lokasi/posisi), setiap layanan yang diberi oleh penyedia layanan biasanya akan berdasarkan kepada posisi *user* yang menginginkan layanan itu. Maka dari itu dibutuhkan komponen yang fungsinya sebagai pemproses atau pengolah yang akan menentukan posisi *user* layanan waktu itu. Posisi *user* itu dapat diperoleh melalui jaringan komunikasi *mobile* atau menggunakan GPS.

2.4 Google Maps

Google Maps merupakan layanan dari google yang mempermudah pengunanya untuk melakukan kemampuan pemetaan untuk aplikasi yang dibuat. Dan juga Google Maps adalah layanan pemetaan web yang dikembangkan oleh Google. Layanan ini memberikan citra satelit, peta jalan, panorama 360°, kondisi lalu lintas, dan perencanaan rute untuk bepergian dengan berjalan kaki, mobil, sepeda (versi beta), atau angkutan umum.

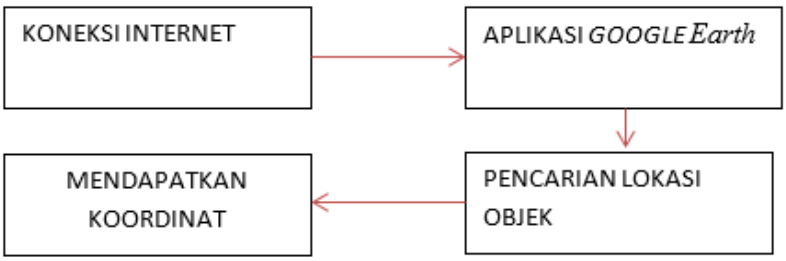
Google Maps untuk seluler dirilis pada bulan September 2008. Pada Agustus 2013, Google Maps bertekad untuk menjadi aplikasi yang paling populer di dunia untuk ponsel cerdas, dengan lebih dari 54% dari pemilik ponsel cerdas di seluruh dunia menggunakannya setidaknya sekali. [9]

Dengan adanya teknologi tersebut, maka orang-orang semakin mudah dalam mencari tempat atau lokasi. Dengan teknologi ini *develover* yang memerlukan maps merasa terbantu dalam pengembangan aplikasi yang hendak di bangun, menjadikan aplikasi semakin optimal. Khususnya pada aplikasi pencarian oleh-oleh yang hendak di bangun, akan merasakan terbantu, karena tidak perlu lagi mengukur jarak secara manual, sehingga hasil aplikasi yang telah dibuat akan terlihat optimal.

2.4.1 Pencarian Koordinat dengan Google Maps

Koordinat adalah suatu titik yang didapatkan dari hasil perpotongan dari garis latitude (lintang) dengan garis bujur (longitude) sehingga akan menunjukan lokasi pada suatu daerah. Umumnya koordinat dibedakan menjadi koordinat Geographic dan Universal Transver Mercator (UTM). Koordinat untuk sebuah konten channel yang berupa Location Based Services (LBS) merupakan hal yang sangat penting. Koordinat memiliki peranan penting untuk menampilkan posisi objek. Adapun cara untuk melakukan pencarian koordinat sebuah objek, yaitu dengan menggunakan Google Earth. [9]

Ada beberapa tahapan yang harus dilakukan untuk melakukan pencarian koordinat (latitude, longitude) dari lokasi suatu objek. Tahapan tersebut dapat dilihat pada gambar 2.5 berikut.



Gambar 2.5 Tahapan pencarian koordinat [9]

Satelit merupakan benda yang mengorbit benda lain dengan periode revolusi dan rotasi tertentu. Ada berbagai macam jenis satelit, di antaranya yaitu satelit alam dan satelit buatan. Dalam melakukan pencarian koordinat ini Penulis menggunakan satelit navigasi.

Satelit navigasi adalah satelit yang menggunakan sinyal radio yang disalurkan ke penerima di permukaan tanah untuk menentukan lokasi sebuah titik di permukaan bumi. Salah satu satelit navigasi yang sangat populer adalah Global Positioning System (GPS) milik Amerika Serikat selain itu ada juga Glonass milik Rusia. Bila pandangan antara satelit dan penerima di tanah tidak ada gangguan, maka dengan sebuah alat penerima sinyal satelit (penerima GPS) bisa diperoleh data posisi di suatu tempat dengan ketelitian beberapa meter dalam waktu nyata.

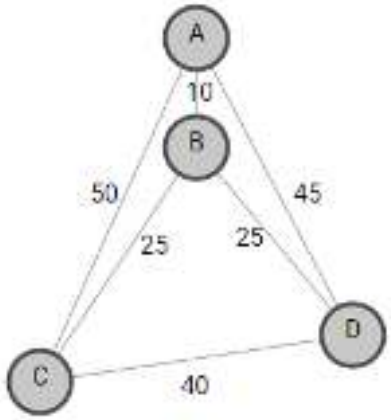
Sebuah perangkat mobile memerlukan koneksi internet untuk mendapatkan sinyal langsung ke satelit dalam pencarian data GPS pada permukaan bumi. Dibutuhkan adanya sambungan ke internet untuk dapat menggunakan aplikasi Google Maps. [9]

2.4.2 Google Maps API

Merupakan suatu fitur aplikasi yang memberikan fasilitas bagi *user* yang ingin mengintegrasikan *Google Maps* ke dalam *website* masing-masing dengan menampilkan data *point* milik sendiri. Dengan memakai *Google Maps* API, *Google Maps* bisa di-*embed* pada *web*site ekstenal. Supaya aplikasi *Google Maps* bisa keluar di *website* tertentu, dibutuhkan adanya API *key*. API (*Application Programming Interface*) *key* adalah kode unik yang digenerasikan untuk *website* tertentu, supaya *server Google Maps* dapat mengenali. API bukan hanya *method* dan *set class* atau *signature* dan fungsi yang sederhana. Melainkan tujuan utamanya untuk mengatasi *clueless* dalam menciptakan *software* yang ukurannya besar, bermula dari hal yang sederhana hingga hal yang *kompleks* dan merupakan prilaku komponen yang tidak mudah dipahami. Perubahan ini dapat dipermudah dengan bantuan API. [10]

2.4.3 Travelling Salesman Problem (TSP)

Kembali pada hari-hari ketika salesman harus merencanakan rute mereka, dari rumah ke rumah atau kota ke kota. Menemukan rute terpendek, mengunjungi satu set lokasi adalah masalah yang sangat sulit, menemukan jalur terpendek di 20 kota lebih dua kali lebih keras daripada 10 kota. Pencarian menyeluruh dari semua jalur yang mungkin akan menjamin untuk menemukan yang terpendek, namun secara komputasi sulit untuk semua kecuali rangkaian lokasi yang kecil. Untuk masalah yang lebih besar, teknik pengoptimalan diperlukan untuk mencari solusi cerdas dan mencari solusi yang hampir optimal. Secara matematis, TSP dapat digambarkan sebagai grafik, dimana lokasi adalah simpul dan ujung (atau busur) mewakili rute langsung antara simpul. Jarak masing-masing tepi adalah jarak antara simpul. Tujuannya adalah untuk menemukan jalan dengan jumlah bobot terpendek. Di bawah ini, kita melihat grafik empat simpul sederhana dan siklus terpendek yang mengunjungi setiap simpul[11]:



Gambar 2.6 Grafik empat simpul sederhana dan siklus terpendek [11]

Google juga menyediakan cara untuk menyelesaikan TSP sederhana di lokasi dunia nyata tanpa mengunduh OR-Tools. Jika Anda memiliki kunci API Google Directions, Anda dapat menyelesaikan TSP dari lokasi dunia nyata dengan API Arah , memberikan lokasi di URL dan mendapatkan tanggapan kembali sebagai JSON. Anda memerlukan API petunjuk gratis Anda sendiri untuk pengembangan, atau kunci perusahaan untuk penggunaan komersial. Sebagai

contoh, inilah URL yang bisa digunakan untuk menemukan tur singkat daerah pembuatan anggur di Australia Selatan, dimulai di Adelaide. Jika Anda ingin mencoba ini dari browser Anda, ganti API_KEY di akhir URL dengan kunci Anda.[11]

```
https://maps.googleapis.com/maps/api/directions/json?origin=Adelaide,SA
&destination=Adelaide,SA&waypoints=optimize:true|Barossa+Valley,SA|Clar
e,SA|Connawarra,SA|McLaren+Vale, SA & key = API_KEY
```

Hasilnya akan menjadi respons JSON yang panjang yang merinci solusinya, lengkap dengan petunjuk Google Maps[11]:


```
{
   "routes" : [
      {
         "bounds" : {
            "northeast" : {
               "lat" : -33.8347115,
               "lng" : 140.8547058
            },
            "southwest" : {
               "lat" : -37.3511758,
               "lng" : 138.4951576
            }
         },
         "copyrights" : "Map data ©2014 Google",
         "legs" : [
            {
               "distance" : {
                  "text" : "139 km",
                  "value" : 139119
               },
               "duration" : {
                  "text" : "1 hour 51 mins",
                  "value" : 6648
               },
               "end_address" : "Clare SA 5453, Australia",
               "end_location" : {
                  "lat" : -33.8333395,
                  "lng" : 138.6117283
               },
               "start_address" : "Adelaide SA, Australia",
               "start_location" : {
                  "lat" : -34.9285894,
                  "lng" : 138.5999429
               },
               "steps" : [
                  {
                     "distance" : {
                        "text" : "70 m",
                        "value" : 70
                     },
                     "duration" : {
                        "text" : "1 min",
                        "value" : 6
                     },
```

```
                    "end_location" : {
                       "lat" : -34.9285338,
                       "lng" : 138.6007031
                    },
                    "html_instructions"            :            "Head
\u003cb\u003eeast\u003c/b\u003e      on      \u003cb\u003eReconciliation
Plaza\u003c/b\u003e toward\u003cb\u003e Victoria Square\u003c/b\u003e",
...
```


2.5 Java

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama *Sun Microsystems*, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama "*Oak*", namun pada tahun 1995 diganti namanya menjadi "*Java*". [12]

2.5.1 Pengertian Java

Java berdiri di atas sebuah mesin penterjemah (interpreter) yang diberi nama Java Virtual Machine (JVM). JVM inilah yang akan membaca kode bit (bytecode) dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada system operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (platform independent). Itulah yang menyebabkan dalam dunia pemrograman Java dikenal adanya istilah „write once, run everywhere", yang berarti kode program hanya ditulis sekali, namun dapat 9 dijalankan di bawah kumpulan pustaka (platform) manapun, tanpa harus melakukan perubahan kode program. [12]

2.5.2 Arsitektur Java

Secara arsitektur, Java tidak berubah sedikitpun sejak awal mula bahasa tersebut dirilis. Compiler Java (yang disebut dengan javac atau Java Compiler) akan mentransformasikan kode-kode dalam bahasa Java ke dalam suatu kode bit.

Dimana bytecode adalah sekumpulan perintah hasil kompilasi yang kemudian dapat dieksekusi melalui sebuah mesin komputer abstrak, yang disebut dengan JVM (Java Virtual Machine). JVM juga sering dinamakan sebagai interpreter, karena sifatnya yang selalu menerjemahkan kode-kode yang tersimpan dalam kode bit dengan cara baris demi baris. Untuk menjalankan program Java, maka file dengan ekstensi .java harus dikompilasi menjadi file kode bit. Dimana untuk menjalankan kode bit tersebut dibutuhkan JRE (Java Runtime Environment) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan pustaka Java yang digunakan.

Dalam pembuatan aplikasi android diperlukan dasar dalam pemrograman Java. Jadi, untuk mengembangkan suatu aplikasi android developer dituntut untuk memiliki dasar dalam pemrograman java, sehingga tidak merasa kebingungan saat mengerjakan suatu aplikasi berbasis mobile. Begitupula dalam pembuatan aplikasi pencarian oleh-oleh.[13]

2.6 MySQL

MySQL merupakan software RDBMS (Relational Database Management System) atau server database yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user multi-user), dan dapat melakukan suatu proses secara sinkron atau berbarengan (multi-threaded).

Saat ini, MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industry, baik industri kecil, menengah, maupun besar.

Lisensi MySQL terbagi menjadi dua. Anda dapat menggunakan MySQL sebagai produk opn source di bawah GNU (General Public License) secara gratis atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disematkan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik. [14]

2.7 Structured Query Language (SQL)

SQL adalah pendekatan dari *Structural Query Language,* yang merupakan Bahasa atau kumpulan perintah standar yang digunakan untuk berkomunikasi dengan *database.* Perintah dalam SQL diklasifikasikan menjadi lima bagian besar, yaitu [14]:

1. *Data Definition Language* (DDL)

Data Definition Language (DDL) adalah sekumpulan perintah SQL yang berkaitan dengan pembuatan, perubahan dan penghapusan *database* prosedur/fungsi, *trigger,* dan sebagainya.

Perintah SQL yang termasuk kategori DDL adalah :

a. CREATE DATABASE, Berfungsi untuk membuat database dan objek-objek di dalam *database* berikut sintaks yang digunakan.
b. CREATE TABLE, Berfungsi untuk membuat tabel di dalam database, berikut sintaks yang digunakan.
c. CREATE INDEX , digunakan untuk membuat index berikut sintaks yang digunakan.
d. CREATE VIEW, digunakan untuk membuat view. Berikut sintaks yang digunakan.
e. DROP DATABASE,digunakan untuk menghapus basis data, berikut sintaks yang digunakan.
f. DROP TABLE, digunakan untuk menghapus tabel. Berikut sintaks yang digunakan.
g. DROP INDEX, digunakan untuk menghapus index. Berikut sintaks yang digunakan.
h. DROP VIEW, digunakan untuk menghapus view. Berikut sintaks yang digunakan.
i. ALTER TABLE, digunakan untuk mengubah struktur suatu tabel. Berikut beberapa sintaks yang digunakan.

- Merubah nama tabel
- Menambah kolom
- Mengubah kolom

- Menghapus kolom
- Mendefinisikan *primary key*
- Mendefinisikan *foreign key*
- Menghapus *primary key*
- Menghapus *Foreign key*
- Menghapus *index*

2. *Data Manipulation Language* (DML)

Data Manipulation Language (DML) adalah kumpulan perintah SQL yang berkaitan dengan data atau isi dari suatu tabel. Dengan perintah-perintah di dalam DML, kita dapat memanipulasi (menambah, mengubah, dan menghapus) data yang terdapat pada suatu tabel secara mudah.

3. *Data Control Language* (DCL)

Data Control Language (DCL) adalah salah satu komponen SQL yang berfungsi untuk mengontrol hak akses *user.* Perintah yang termasuk ke dalam DCL adalah GRANT dan REVOKE. Berikut penjelasannya.

a. GRANT, Perintah GRANT digunakan untuk memberikan hak akses (*privilege*) kepada *user* tertentu. Melalui perintah hak akses semacam ini, seseorang *user* memiliki keterbatasan dalam menggunakan *database* sehingga data akan aman dari pihak-pihak yang tidak berkepentingan. Untuk melihat daftar hak akses yang dimiliki oleh seorang *user* , gunakan perintah SHOW GRANTS. Untuk mengeksekusi perintah GRANT, anda perlu memiliki hak akses GRANT OPTION yang sebelumnya diberikan oleh administrator kepada anda.
b. REVOKE, Perintah REVOKE merupakan kebalikan dari perintah GRANT, yang berfungsi untuk mencabut salah satu atau beberapa hak akses dari *user* tertentu di dalam *database.* Sama seperti perintah GRANT, untuk menjalankan perintah REVOKE anda perlu memiliki hak akses GRANT OPTION.

4. *Transaction control Language* (TCL)

Selain pengolahan hak akses user, dalam SQL kita juga dapat mengontrol transaksi data yang telah dilakukan. Perintah-perintah yang berkaitan dengan pengontrolan transaksi data digolongkan ke dalam *Transactional Control Language* (TCL). Perintah yang termasuk ke dalam TCL adalah COMMIT dan

ROLLBACK. Berikut penjelasannya.

a. COMMIT, perintah COMMIT berfungsi untuk menyimpan perubahan-perubahan yang dilakukan terhadap *database* (melalui perintah INSERT, UPDATE, atau DELETE) secara permanent.
b. ROLLBACK, perintah ROLLBACK merupakan kebalikan dari perintah COMMIT, yang berfungsi untuk membatalkan transaksi atau perubahan-perubahan yang telah dilakukan ke dalam *database* (melalui perintanh INSERT, UPDATE, atau DELETE). Dengan melakukan pembatalan transaksi, data di dalam *database* akan kembali ke keadaan awal (keadaan sebelum dilakukan perubahan). Sehingga dengan kata lain, perubahan yang dilakukan tidak akan pernah disimpan ke dalam *database.*

2.8 Personal Home Page (PHP)

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdrof. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. Script yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi *visitor*, dan menampilkan jumlah pengunjung dari suatu *website*. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan script buatan Rasmus Lerdrof, hingga akhirnya rasmus mulai membuat sebuah *tool/script*, bernama *Personal Home Page* (PHP).

Keunggulan PHP antara lain [15]:

1. Gratis

   Apa yang membuat PHP begitu berkembang sangat pesat hingga jutaan domain menggunakan PHP, begitu populernya PHP? Jawabannya adalah karena PHP itu gratis.
2. *Cross platform*

   Artinya dapat di gunakan di berbagai sistem operasi, mulai dari linux, windows, mac os dan os yang lain.
3. Mendukung banyak *database*

PHP telah mendukung banyak *database*, ini mengapa banyak *developer web* menggunakan PHP Adabas D Adabas D, dBase dBase, Empress Empress, FilePro (*read-only*) FilePro (*read-only*) Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase mSQL, Direct MS-SQL, MySQL MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL SQLite, Solid, Sybase, Velocis, Unix dbm.

4. *On The Fly*

   PHP sudah mendukung *on the fly*, artinya dengan php anda dapat membuat *document text*, Word, Excel, PDF, menciptakan *image* dan *flash*, juga menciptakan file-file seperti zip, XML, dan banyak lagi.

2.9 Twitter API

Twitter adalah sebuah situs jejaring sosial yang sedang berkembang pesat saat ini karena pengguna dapat berinteraksi dengan pengguna lainnya dari komputer ataupun perangkat mobile mereka dari manapun dan kapanpun. Setelah diluncurkan pada Juli 2006, jumlah pengguna Twitter meningkat sangat pesat. Pada September 2010, diperkirakan jumlah pengguna Twitter yang terdaftar sekitar 160 juta pengguna.

Pada tahun April 2010, jumlah tweet yang diposting mencapai 55 juta tweet/hari, lalu kemudian pada tahun 2011, tercatat rata-rata sekitar 140 juta tweet telah dikirimkan oleh pengguna Twitter. Berbagai macam manfaat dapat diperoleh dari tweet dimulai dari event detection (deteksi kejadian, salah satunya bencana alam), prediksi pergerakan pasar saham, prediksi pemilu hingga penyebaran penyakit di suatu wilayah.

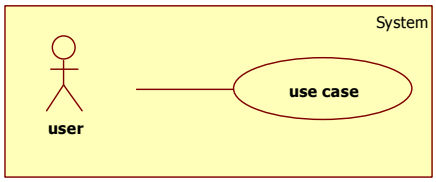
Saat ini, hanya koneksi SSL (Secure Sockets Layer) yang diizinkan di twitter sebagai titik akhir API. Berkomunikasi melalui SSL melindungi privasi pengguna dengan melindungi informasi antara pengguna dan twitter API saat dikirimkan ke publik. Secure Sockets Layer adalah protokol keamanan yang digunakan pada hampir semua transaksi aman pada internet. SSL mengubah suatu protokol transport seperti TCP menjadi sebuah saluran komunikasi aman yang cocok untuk transaksi yang sensitif seperti Paypal, Internet Banking, dan lain-lain. [16]

2.10 UML (Unified Modelling Language)

UML adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem piranti lunak yang terkait dengan objek. UML menawaran diagram-diagram untuk memodelkan suatu sistem, seperti satu set blueprint. UML adalah sekumpulan diagram yang mempresentasikan sistem dengan bahasa tekstual dan grafikal melalui simbol-simbol tertentu. Jadi UML adalah bahasa pemodelan visual yang standar untuk menggambarkan sebuah sistem. [17]

2.10.1 Use Case Diagram

Use case diagram adalah proses pemodelan fungsi-fungsi sistem dalam konteks pristiwa-pristiwa bisnis, siapa yang mengawali dan bagaimana sistem itu memproses hal tersebut. Use case diagram memperlihatkan hubungan diantara aktor dan use case. Aktor merepresentasaikan seorang user atau sub sistem lain yang akan berinteraksi dengan sistem. Sedangkan use case merupakan urutan kejadian yang menggambarkan interaksi antara user dengan sistem fungsionalitas sistem didefinisikan ke dalam use case dari sudut eksternal sistem yang berguna untuk uji kelayakan fungsionalitas. [17] Gambar berikut menampilkan usecase diagram dalam UML :

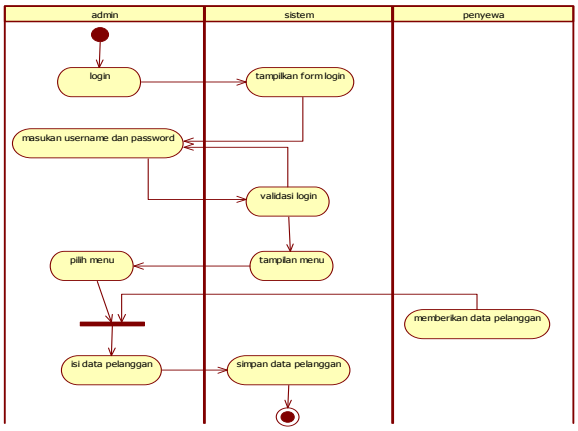


Gambar 2.7 Use case diagram [17]

Maka dalam pembuatan sebuah program, *developer* dapat memanfaatkan use case sebagai alat bantu untuk menentukan *user* atau aktor apa saja yang berinteraksi dengan use case itu sendiri. Dalam pembuatan aplikasi pencarian oleh-oleh juga dapat memanfaatkan use case dalam menentukan aktor atau user apa saja yang terlibat, sehingga dapat lebih terarah dalam pembuatan programnya.

2.10.2 Activity Diagram

Merupakan penggambaran rangkaian aliran dari aktifitas, dipakai untuk menjelaskan aktifitas yang dibentuk dalam suatu operasi, sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram *activity* ini menjelaskan alur kerja sebuah sistem. Activity diagram mirip dengan state diagram karena sejumlah aktivitas menggambarkan keadaan suatu proses dengan memperlihatkan urutan aktivitas yang dijalankan baik berupa pilihan maupun paralel. Diagram ini juga berguna untuk menganalisa sebuah use case dengan menggambarkan aksi-aksi yang diperlukan dan kapan aksi-aksi tersebut dijalankan, menjelaskan urutan algoritma yang komplek dan memodelkan sejumlah aplikasi dengan proses paralel. [17] Gambar berikut menampilkan activity diagram dalam UML:

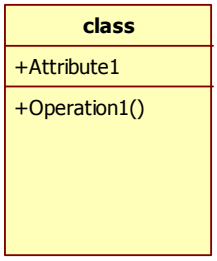


Gambar 2.8 Activity Diagram [16]

Maka dalam pembangun sebuah aplikasi, *developer* dapat memanfaatkan *activity diagram* sebagai alat untuk menggambarkan aksi-aksi yang diperlukan dan kapan aksi tersebut dijalankan. Begitupun dalam pembuatan aplikasi pencarian oleh-oleh dapat memanfaatkan *activity diagram* untuk menganalisa sebuah use case secara cepat dan tepat, sehingga dapat menjelaskan urutan algoritma yang komplek dan memodelkan dengan proses paralel.

2.10.3 Class Diagram

Merupakan penggambaran struktur statis class di dalam sistem. Class diagram merepresentasikan sesuatu yang ditangani oleh sistem. Diagram ini digunakan untuk menggambarkan tipe-tipe objek dan hubungannya dalam sebuah sistem. Class diagram memodelkan struktur class dan isinya dengan menggunakan elemen-elemen model seperti class, package dan objek. Class terdiri dari tiga bagian yaitu nama class, atribut dan operasi. Class yang didefinisikan secara global dapat diakses oleh objek diluar class tersebut. [16] Gambar berikut menampilkan class diagram dalam UML:



Gambar 2.9 Class Diagram [17]

Keterangan :

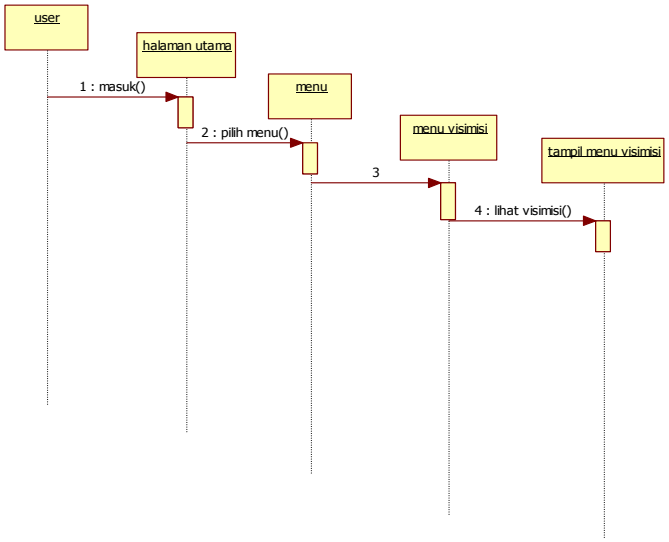
1. Nama kelas ditulis dalam huruf tebal dan diletakkan di tengah-tengah. Nama diambil dari domain permasalahan dan harus sejelas mungkin. Oleh karena itu nama kelas haruslah berupa kata benda.
2. Attribut kelas menggambarkan karakteristik dari objek. Atribut kelas yang benar adalah yang dapat mencakup informasi-informasi yang dilukiskan dan mengenali instance tertentu dari kelas. Tipe dari atribut dapat berupa primitif atribut atau tipe lainnya.
3. Operasi digunakan untuk memanipulasi atribut atau menjalankan aksi-aksi. Operasi biasanya disebut dengan fungsi, tetapi mereka terdapat didalam kelas dan dapat diaplikasikan hanya pada objek dalam kelas tersebut.

Dalam pembuatan program, developer bisa menggunakan class diagram untuk menggambarkan objek-objek yang berada dalam sistem. Pada aplikasi pencarian oleh-oleh juga dapat memanfaatkan class diagram, dikarenakan class bisa

berhubungan dengan yang lain melalui berbagai cara diantaranya *associated* (terhubung satu sama lain), *dependent* (satu class menggunakan atau bergantung class lainnya), *specialed* (satu class merupakan spesialisasi dari class lainnya), atau *package* (grup bersama sebagai satu unit).

2.10.4 Sequence Diagram

Merupakan penggambaran kolaborasi dinamis antara sejumlah objek. Diagram ini menjelaskan bagaimana objek berinteraksi dengan lainnya dengan cara mengirim dan menerima pesan. Sequence diagram memiliki dua sumbu:sumbu vertikal dan sumbu horizontal. Sumbu vertikal putus-putus merepresentasikan "lifetime" objek dan sumbu horizontal menunjukan sekumpulan objek. Sekumpulan objek diagram ini juga menyatakan interaksi khusus diantara objek yang terjadi pada beberapa tempat selama fungsi tertentu dijalankan. Komunikasi diantara objek direpresentasikan dengan garis horizontal disertai dengan nama operasinya. [17] Gambar berikut menampilkan sequence diagram dalam UML:



Gambar 2.10 Sequence Diagram [17]

Kegunaan untuk memperlihatkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek, dapat digunakan oleh developer dalam membangun aplikasi yang ingin dibuat. Dalam pembuatan aplikasi pencarian oleh-oleh juga dapat memanfaatkan *sequence diagram* sebagai alat bantu dalam

menggambarkan kolaborasi dari sejumlah objek yang ada atau interaksi yang menjelaskan bagaimana pesan mengalir dari objek ke objek lainnya yang berhubungan dengan program yang akan dibangun.

2.11 Pengujian

Kasus pengujian menspesifikasi salah satu cara untuk melakukan pengujian sistem/perangkat lunak, mencakup didalamnya menspesifikasi asupan serta hasilnya dan dalam kondisi bagaimana pengujian dilakukan.

Pada praktiknya, apa yang akan diuji dapat saja merupakan satu atau lebih spesifikasi kebutuhan sistem/perangkat lunak yang mungkin untuk dilakukan pengujian serta biayanya tidak terlalu tinggi. [18]

2.11.1 Black-box Testing

Black-box testing adalah kasus pengujian yang menspesifikasi bagaimana melakukan pengujian suatu use case atau suatu skenario yang bersifat spesifik. Beberapa kasus pengujian memverifikasi hasil dari interaksi yang terjadi diantara *actor* dengan sistem/perangkat lunak yang dikembangkan, yaitu memverifikasi apakah kondisi awal (*pre-condition*) dan kondisi akhir (*post-condition*) yang dispesifikasi oleh use case terpenuhi, serta juga memverifikasi apakah urutan aksi-aksi (baca: *flow-of-events*) yang dispesifikasi oleh use case memang diikuti.

Tujuan dari ujicoba whitebox adalah menemukan fungsi-fungsi yang kurang tepat atau hilang, menemukan kesalahan interface, menemukan kesalahan dalam struktur data atau akses penyimpanan eksternal, menemukan kesalahan performa, dan juga menemukan kesalahan inisialisasi dan terminasi. [18]

Sehingga black box testing dapat dilakukan oleh developer aplikasi untuk menguji berdasarkan detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diharapkan oleh kostumer. Pengujian ini tidak melihat dan menguji *source code* program. Kegiatan dalam pengujiannya salah satu nya mencari *bugs/ error* dari tampilan aplikasi.