

BAB 2

LANDASAN TEORI

2.1 Pengertian Buku

Buku merupakan kumpulan dari lembaran kertas yang berisi ilmu pengetahuan hasil analisis terhadap kurikulum secara tertulis. Buku disusun menggunakan bahasa sederhana menarik dan dilengkapi gambar serta daftar pustaka.

2.1.1 Jenis Buku

1. Ilmiah adalah buku yang disusun berdasarkan kaidah keilmiah.
2. Panduan adalah buku petunjuk, misalnya buku tentang beternak ayam, berkebun kelapa sawit, kiat memperoleh dan kiat menjalani beasiswa di luar negeri, dan sebagainya.
3. Biografi adalah kisah atau keterangan tentang kehidupan seseorang.
4. Dongeng, merupakan suatu kisah yang di angkat dari pemikiran fiktif dan kisah nyata, menjadi suatu alur perjalanan hidup dengan pesan moral, yang mengandung makna hidup.
5. Ensiklopedia adalah sejumlah buku yang berisi penjelasan mengenai setiap cabang ilmu pengetahuan yang tersusun menurut abjad atau menurut kategori secara singkat dan padat.
6. Komik adalah suatu bentuk seni yang menggunakan gambar-gambar tidak bergerak yang disusun sedemikian rupa sehingga membentuk jalinan cerita.
7. Novel adalah sebuah karya fiksi prosa yang tertulis dan naratif.

2.2 Android

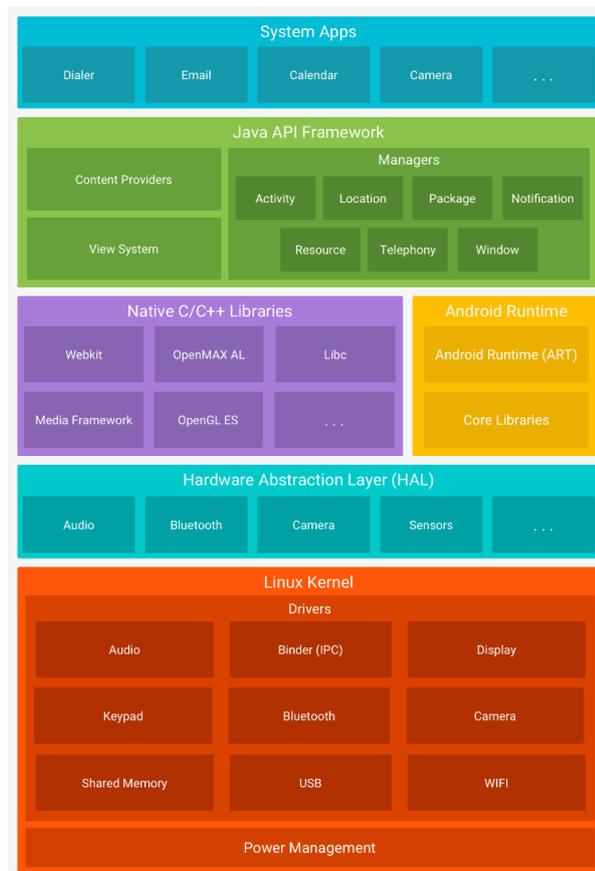
Android Studio adalah sebuah IDE untuk android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android [4].

Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur – fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment. Fitur – fitur lainnya adalah sebagai berikut :

1. Menggunakan Gradle-based build system yang fleksibel.
2. Bisa untuk build multiple APK.
3. Template support untuk Google Service dan berbagai macam tipe perangkat.
4. Layout editor yang lebih bagus.
5. Built-in support untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.

2.2.1 Arsitektur Android

Arsitektur Android dapat digambarkan seperti pada gambar 2.1 Arsitektur Android secara garis besar Arsitektur Android dapat dijelaskan sebagai berikut [4]:



Gambar 2. 1 Arsitektur Android

1. *Linux Kernel*

Fondasi platform Android adalah kernel Linux. Android Runtime (ART) bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan pengelolaan memori tingkat rendah.

2. *Hardware Abstraction Layer (HAL)*

Hardware Abstraction Layer (HAL) memberikan antarmuka standar yang mengungkapkan kemampuan perangkat keras perangkat ke perangkat kerja API Java yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing – masing menerapkan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Ketika API kerangka panggilan kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem Android memuat modul pustaka untuk komponen perangkat keras.

3. *Android Runtime*

Aplikasi menjalankan proses masing – masing dengan tahap Android Runtime (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file DEX, format bytecode yang dirancang khusus untuk Android yang dioptimalkan untuk footprint memori minimal.

4. *Pustaka C/C++ Bawaan*

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode bawaan yang memerlukan pustaka bawaan yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja API Java meningkatkan fungsi beberapa pustaka bawaan pada aplikasi.

5. *Kerangka Kerja API Java*

Keseluruhan rangkaian fitur pada Android OS tersedia untuk memulai API yang ditulis dalam bahasa Java. API membentuk elemen dasar yang harus buat aplikasi Android dengan menyederhanakan penggunaan ulang inti, komponen dan layanan sistem modular, yang mencakup berikut ini :

- a. Tampilan Sistem
- b. Pengelolaan Sumber Daya
- c. Pengelolaan Notifikasi
- d. Pengelolaan Aktivitas

2.2.2 Android Life Cycle

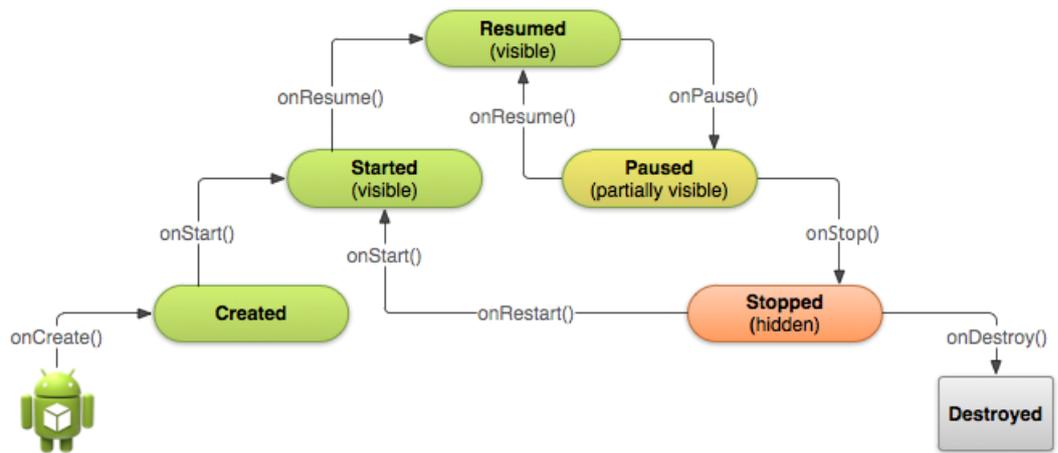
Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda seperti ditunjukkan pada, klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan [4].

Tabel 2. 1 Komponen Aplikasi Mobile

Functionality	Java Base Class	Examples
Focused thing a user can do	Activity	Edit a note, play a game
Background	Service	Play music, update weather icon
Receive messages	Broadcast Receiver	Trigger alarm upon event
Store and retrieve data	Content Provider	Open a phone contact

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan specified permission sebelum digunakan seperti pada komponen Service, BroadcastReceiver, ContentProvider.

Android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi main. Sistem android menjalankan kode dalam method Activity dengan menerapkan metode callback tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi Android memiliki siklus hidup yang berbeda dengan aplikasi desktop ataupun web. Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang cukup tinggi seperti ketika handling panggilan masuk aplikasi diharuskan menghentikan proses sementara. Penerapan siklus hidup juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna.



Gambar 2. 2 Siklus Hidup Android

Ilustrasi pada gambar 2.2 siklus hidup android akan tetapi hanya beberapa dari state tersebut yang menjadi statis seperti :

1. *Resumed*

Resumed terjadi ketika aplikasi berjalan setelah state paused. State ini akan menjalankan perintah program yang ditulis pada method `onResume()`.

2. *Paused*

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

3. *Stopped*

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihat oleh pengguna tetapi masih meninggalkan service dibackground.

State lain seperti *Created* dan *Started* bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode *lifecyclecallback* berikutnya. Artinya, setelah sistem `OnCreate()` dipanggil, dengan cepat sistem akan memanggil method `OnStart()`, kemudian diikuti oleh `onResume()`.

2.2.3 Fitur Android

Android memiliki beberapa fitur utama yang sering digunakan dalam proses pembangunan aplikasi diantaranya adalah [4] :

2.7.7 *Multi-proses dan App Widgets*

App Widgets adalah mini aplikasi yang dapat embedded dalam aplikasi seperti home screen. App widgets dapat menjalankan prosesrequest seperti musik streaming atau mendeteksi suhu ruangan secara background.

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

2.7.8 *Touch Gestures dan Multi-touch*

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

2.7.9 *Hard dan Soft Keyboard*

Salah satu fitur pada perangkat smartphone adalah tombol fisik dan non fisik, tombol fisik digunakan untuk navigasi pendukung dalam pengoperasian android. Pengembang pada aplikasi tidak perlu secara manual untuk mengintegrasikan tombol tersebut dalam aplikasi. Tombol non fisik adalah tombol yang dibuat oleh sistem operasi seperti keyboard virtual, dan tombol navigasi aplikasi.

2.2.4 Android SDK

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 [4].

Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, android member anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone atau Smartphone. Beberapa fitur-fitur android yang paling penting adalah

- a. Framework : Aplikasi yang mendukung pengganti komponen dan reusable.
- b. Dalvik Virtual Machine dioptimalkan untuk perangkat mobile.
- c. Integrated Browser berdasarkan engine open source WebKit.
- d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D dan grafis 3D berdasarkan spesifikasi opengl ES 1,0
- e. Media Support seperti audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony.
- f. Bluetooth, EDGE, 3G, dan WiFi.
- g. Kamera, GPS, Kompas, dan Accelerometer.
- h. Lingkungan Development yang lengkap dan termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin.

2.3 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa java awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa java banyak mengadopsi sintaksis

yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

Versi awal Java ditahun 1996 sudah merupakan versi rilis sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

- a. java.lang: Peruntukan kelas elemen-elemen dasa
- b. java.io: Peruntukan kelas input dan output, termasuk penggunaan berkas.
- c. java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
- d. java.net merupakan peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
- e. java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).
- f. java.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

Mekanisme penggunaan bahasa pemrograman dapat membantu menjamin keamanan juga membantu menciptakan portabilitas karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda. Saat ini java merupakan bahasa pemrograman yang populer digunakan dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. Java memiliki beberapa kelebihan dibandingkan dengan bahasa pemrograman lain, diantaranya :

- a. Multiplatform
- b. OOP (Object Oriented Programming)
- c. Library Class yang lengkap
- d. Mewarisi C/C++

2.4 API (Application Programming Interface)

API adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi. API atau Application Programming Interface juga merupakan suatu dokumentasi yang terdiri dari antar muka, fungsi, kelas, struktur untuk membangun sebuah perangkat lunak.

Dengan adanya API, maka memudahkan seorang programmer untuk membongkar suatu software untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak yang lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Suatu rutin standar yang memungkinkan developer menggunakan system function. Proses ini dikelola melalui operating system. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya untuk saling berinteraksi. Keuntungan dengan menggunakan API adalah sebagai berikut :

2.1.1 *Portabilitas*

Developer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah terinstal API tersebut.

2.1.2 Lebih mudah untuk dimengerti

API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal pengembangan. System call interface berfungsi sebagai penghubung antara API dan system call yang dimengerti oleh sistem operasi. System call interface akan menerjemahkan perintah dalam API dan kemudian akan memanggil system calls yang diperlukan. Untuk membuka suatu file tersebut user menggunakan program yang telah dibuat dengan menggunakan bantuan API, maka perintah dari user tersebut diterjemahkan dulu oleh program menjadi perintah `open()`.

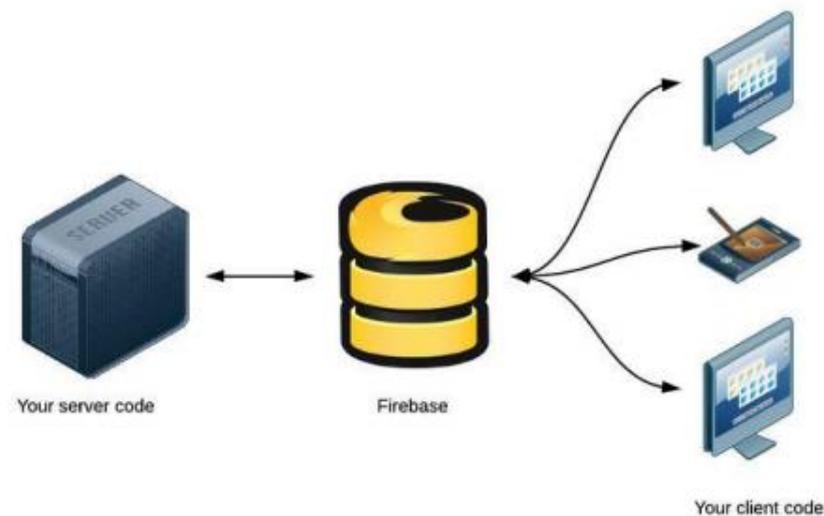
Perintah `open()` ini merupakan perintah dari API dan bukan perintah yang langsung dimengerti oleh kernel sistem operasi. Oleh karena itu, agar keinginan pengguna dapat dimengerti oleh sistem operasi, maka perintah `open()` tadi diterjemahkan ke dalam bentuk system call oleh system call interface.

Implementasi perintah `open()` tadi bisa bermacam-macam tergantung dari sistem operasi yang digunakan.

2.5 Firebase

Firebase adalah penyedia layanan realtime database dan backend sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk client yang berbeda – beda dan disimpan pada cloud-nya Firebase. Firebase memiliki banyak library yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, Ios, Javascript, Java, Objective-C dan Node.JS [5].

Database Firebase juga bersifat bisa diakses lewat REST API. REST API tersebut menggunakan protokol Server-Sent Event dengan membuat koneksi HTTP untuk menerima push notification dari server. Pengembang menggunakan REST API untuk post data yang selanjutnya Firebase client library yang sudah diterapkan pada aplikasi yang dibangun yang akan mengambil data secara realtime.



Gambar 2. 3 Sistem Kerja Firebase

Pengembang juga dapat menggunakan database ini untuk mengamankan data menggunakan server Firebase dengan rules yang ada. Untuk hosting file Firebase menyediakan hosting untuk static file dengan fasilitas CDN dan SSL.

2.6 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.

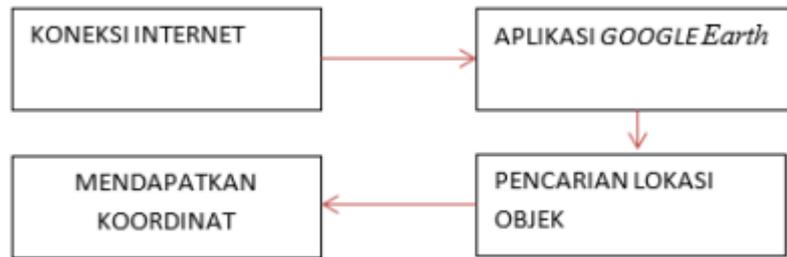
Google Maps untuk seluler dirilis pada bulan September 2008. Pada Agustus 2013, Google Maps bertekad untuk menjadi aplikasi yang paling populer di dunia untuk ponsel cerdas, dengan lebih dari 54% dari pemilik ponsel cerdas di seluruh dunia menggunakannya setidaknya sekali.

Dengan adanya teknologi tersebut, maka orang-orang semakin mudah dalam mencari tempat atau lokasi. Dengan teknologi ini developer 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.

2.6.1 Pencarian Koordinat dengan Google Maps

Koordinat adalah suatu titik yang didapatkan dari hasil perpotongan dari garis latitude (lintang) dengan garis bujur (longitude) sehingga akan menunjukkan 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 Maps.

Ada beberapa tahapan yang harus dilakukan untuk melakukan pencarian koordinat (latitude, longitude) dari lokasi suatu objek. Tahapan tersebut dapat dilihat pada gambar 2.4 berikut.



Gambar 2. 4 Tahapan pencarian koordinat

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.

2.6.2 Google Maps API

Google Maps API Merupakan suatu layanan 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 website eksternal. 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 perilaku komponen yang tidak mudah dipahami. Perubahan ini dapat dipermudah dengan bantuan API.

2.7 Unified Modeling Language

Unified Modelling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk pemodelan visual digunakan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan rancangan dari suatu sistem perangkat lunak. Pemodelan memberikan gambaran yang jelas mengenai sistem yang akan dibangun dari sisi struktural ataupun fungsional. UML dapat diterapkan pada semua model pengembangan, tingkatan siklus sistem dan berbagai macam domain aplikasi. Dalam UML terdapat konsep semantik, notasi, dan panduan statis, dinamis, ruang lingkup dan organisasional. UML bertujuan menyatukan teknik – teknik pemodelan berorientasi objek menjadi terstandarisasi [6].

2.7.1 Use Case Diagram

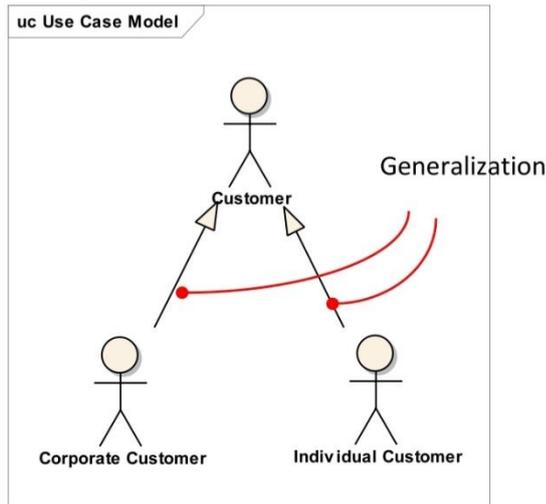
Use case diagram adalah tingkah laku sistem, subsistem, kelas, atau komponen yang tampak kepada eksternal entity (actor). Diagram use case membagi fungsionalitas sistem menjadi transaksi – transaksi yang memiliki arti bagi *actor*. Setiap potongan dari fungsi interaktif di sebut use case.

1. Use Case

Sebuah use case adalah sebuah unit eksternal dari sistem (berupa antar muka) yang akan menerima perintah dari seorang aktor berupa sebuah *event*. Use case ini terkait dengan implementasi didalamnya yang berupa urutan – urutan penyampaian pesan – pesan antar objek – objek yang berkaitan.

2. Actor

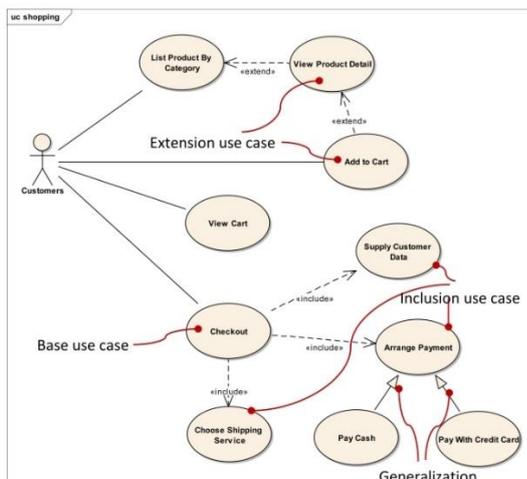
Actor adalah suatu set pesanan yang dimainkan oleh orang luar, unit kerja, atau hal – hal yang berinteraksi dengan sistem, subsistem atau kelas. Pengguna – pengguna yang berbeda mungkin terikat pada satu use case yang sama.



Gambar 2. 5 Aktor

3. Relationships

Relationships adalah sebuah hubungan antara use case dapat berpartisipasi dengan use case lain dalam beberapa jenis hubungan sebagai tambahan ketika berasosiasi dengan sebuah actor.



Gambar 2. 6 Hubungan use case

4. Skenario *Use Case*

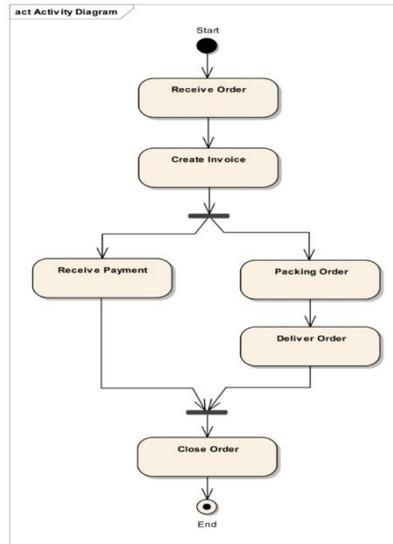
Skenario use case adalah suatu cara untuk menjelaskan apa yang sistem atau sub sistem lakukan, tetapi use case tidak menjelaskan secara spesifik bagaimana sistem tersebut melakukannya.

Use Case Name	Checkout
Requirements	A3, A4, A5
Goal	User dapat melakukan pembelian secara online via web.
Pre-conditions	User telah memilih barang
Post-conditions	System mengirim rincian faktur penjualan via email
Failed end condition	User membatalkan checkout. Kemungkinan user ingin menambah barang, atau membatalkan pembelian.
Primary Actors	User
Main Flow / Basic Path	<ol style="list-style-type: none"> 1. User memilih icon checkout. 2. System menampilkan rincian belanja. 3. User menyetujui dan melanjutkan. 4. System mengecek apakah user sudah terdaftar atau belum. Jika belum maka system akan menampilkan

Gambar 2. 7 Skenario use case

2.7.2 *Activity Diagram*

Activity Diagram adalah sebuah tahapan yang lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dimana biasanya dipakai pada businiss modeling untuk memperlihatkan urutan aktifitas proses bisnis. Dalam suatu organisasi yang berorientasi profit maupun non profit pasti memiliki proses – proses dan prosedur (*business process*). *Activity Diagram* dapat digunakan untuk menggambarkan arus kerja dan *business process* [6].



Gambar 2. 8 Activity Diagram

2.7.3 Class Diagram

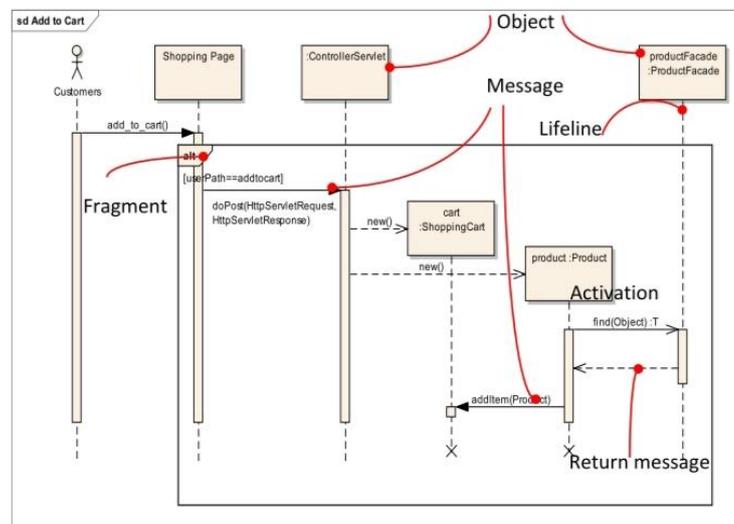
Class diagram adalah diagram yang selalu ada pada pemodelan system berorientasi objek. Class diagram menunjukkan hubungan antar class dalam system yang sedang dibangun. Sebuah kelas diagram adalah diagram yang menunjukkan suatu set kelas – kelas, antarmuka – antarmuka dan hubungan – hubungan. Kelas juga memiliki jenis-jenis hubungan seperti asosiatif, dependensi, agregasi, komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing-masing komponen penyusun kelas memiliki hak akses seperti public, private dan protected.



Gambar 2. 9 Class Diagram

2.7.4 Sequence Diagram

Sequence diagram menggambarkan tingkah laku dari suatu scenario tunggal. Sequence diagram menunjukkan objek – objek yang terlibat dalam proses dan terlibat dalam proses mengurutkan penyampaian pesan – pesan antara objek – objek. Sequence diagram menggambarkan interaksi dalam bentuk grafik dua dimensi. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, message yang digambarkan dengan garis penuh, dan waktu yang ditunjukkan dengan progress vertical.



Gambar 2. 10 Sequence Diagram