

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori menjelaskan beberapa teori-teori dan penjelasan yang berkaitan dengan aplikasi atau media yang dibangun. Landasan teori yang digunakan meliputi *push notification*, rekomendasi, aplikasi, sistem, informasi, android, GPS, LBS, API, Google Maps, OpenWeatherMap API, Java, PHP, JSON, MySQL, Web Server serta UML.

2.1.1 Push Notification

Push Notification merupakan suatu bentuk komunikasi jaringan, yaitu server akan mengirimkan pesan pemberitahuan ke *client* jika ada perubahan data, sehingga *client* tidak perlu melakukan proses *request* data tiap periode untuk mengambil data pemberitahuan. Teknologi *push notification* untuk perangkat android sangat efektif karena berjalan pada *background* proses sehingga memungkinkan aplikasi menerima pesan secara *real-time*. Seperti yang dikutip dari situs Pusher [10], pada sistem operasi android, *push notification* dapat diimplementasikan dengan memanfaatkan layanan *Firebase Cloud Messaging* (FCM) untuk mengirim pesan atau notifikasi yang disediakan oleh layanan Firebase dari Google.

2.1.2 Rekomendasi

Rekomendasi adalah saran yang sifatnya menganjurkan, membenarkan, atau menguatkan mengenai sesuatu atau seseorang. Rekomendasi sangat penting dalam meyakinkan orang lain bahwa sesuatu atau seseorang tepat dan layak. Seperti ketika seseorang akan menggunakan jasa *online shop*, biasanya mereka akan melihat rekomendasi berupa testimoni atau bisa disebut sebagai *review* dari orang-orang yang sudah bertransaksi sebelumnya. Jika banyak testimoni positif maka keyakinan orang untuk bertransaksi pada *online shop* tersebut bertambah.

2.1.3 Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Menurut Priyono [11], secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.

Pada dunia teknologi aplikasi merupakan sebuah perangkat lunak yang dibuat untuk membantu atau mempermudah pekerjaan manusia. Aplikasi juga merupakan suatu program perangkat lunak yang berjalan pada suatu sistem tertentu. Sebuah aplikasi dapat dibuat dan dikembangkan dengan tujuan untuk melakukan tugas yang bersifat umum atau juga dapat dikembangkan untuk melakukan tugas yang bersifat spesifik/khusus.

2.1.4 Sistem

2.1.4.1 Pengertian Sistem

Terdapat beberapa definisi dari sistem. Sistem dapat diartikan sebagai suatu jaringan kerja yang terdiri dari berbagai prosedur yang saling berhubungan, berkumpul dan bekerja sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Jogiyanto [12] menerangkan bahwa sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sedangkan menurut Abdul Kadir [13] sistem adalah sekelompok elemen-elemen yang saling terintegrasi dengan maksud dan tujuan yang sama untuk melaksanakan sasaran yang telah ditentukan.

2.1.4.2 Karakteristik Sistem

Suatu sistem memiliki karakteristik yang akan merepresentasikan ciri dari sebuah sistem itu sendiri. Berikut adalah karakteristik dari sebuah sistem :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi. Saling berinteraksi artinya saling bekerja sama membentuk suatu kesatuan untuk mencapai tujuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar tersebut akan mempengaruhi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Jadi, sumber daya dapat terdistribusikan antar sistem dengan baik dengan adanya penghubung sistem.

5. Masukan Sistem

Masukan sistem adalah inputan yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah inputan yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah inputan yang diproses untuk mendapatkan keluaran.

6. Keluaran Sistem

Keluaran sistem adalah hasil keluaran dari suatu proses sistem. Hasil keluaran ini diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Hasil dari olahan itu akan menghasilkan sesuatu yang berbeda dari hasil inputan sebelumnya.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengolah semua masukan. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran Sistem yang tidak mempunyai sasaran bisa disebut sebagai sistem tidak berguna. Sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan sistem tersebut.

2.1.4.3 Klasifikasi Sistem

Menurut Agus Mulyanto [14], sistem dapat di klasifikasikan berbagai sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem Abstrak dan Fisik (*Physical System*)

Dari bentuknya, sistem dapat diklasifikasikan menjadi sistem abstrak dan sistem fisik. Sistem abstrak yaitu sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem agama. Sementara sistem fisik adalah sistem yang keberadaannya dapat dilihat secara fisik, misalnya perusahaan atau komputer.

2. Sistem Alamiah dan Buatan (*Human Made System*)

Dari segi bagaimana sistem terbentuk, sistem dapat tergolong menjadi sistem alamiah dan sistem buatan. Sistem alamiah (*natural system*) yaitu sistem yang terbentuk melalui proses alami, misalnya sistem tatasurya dan pencernaan. Sementara sistem buatan manusia adalah sistem yang dirancang dan dibangun oleh manusia misalnya sistem produksi di pabrik.

3. Sistem Tertentu dan Tidak Tertentu

Dilihat dari hasil keluaran, sistem dapat dikategorikan sebagai sistem tertentu dan tidak tertentu. Sistem tertentu adalah sistem yang cara beroperasinya sudah dapat diprediksi, interaksi-interaksi didalamnya dapat dideteksi dengan pasti dan outputnya dapat diramalkan, misalnya pengolahan data. Sementara sistem tak tentu adalah sistem yang outputnya tidak dapat diprediksi dengan pasti karena mengandung unsur probabilitas.

4. Sistem Tertutup dan Terbuka (*Open System*)

Sistem tertutup yaitu sistem yang tidak berhubungan dengan dunia luar dan tidak terpengaruh dengan lingkungan luarnya (bekerja secara otomatis). Sebenarnya sistem tertutup tidak ada yang ada adalah relatif tertutup. Sementara sistem terbuka adalah sistem yang mempunyai hubungan dengan dunia luar dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan output untuk subsistem yang lain.

2.1.4.4 Kualitas Sistem

Pada penelitian yang dilakukan oleh Istianingsih dan Utami [15] menunjukkan bahwa kualitas sistem terbukti secara signifikan berpengaruh positif terhadap kepuasan pengguna. Kualitas dari sistem ditentukan dari berbagai aspek indikator. Indikator tersebut diantaranya:

1. Kemudahan Untuk Digunakan (*Ease Of Use*)

Sistem informasi dapat dikatakan sebagai sistem yang berkualitas jika dirancang untuk memenuhi kemudahan dalam penggunaan sistem informasi tersebut. Sistem yang sukar digunakan tentunya dapat menghambat pekerjaan pengguna. Hal ini tentunya berakibat menurunnya tingkat kepuasan pengguna.

2. Kecepatan Akses (*Response Time*)

Kecepatan akses merupakan salah satu indikator kualitas sistem informasi. Keterlambatan informasi yang dihasilkan dapat menyebabkan kerugian pada pengguna. Jika sistem informasi memiliki kecepatan akses yang optimal maka layak untuk dikatakan bahwa sistem informasi yang diterapkan memiliki kualitas yang baik.

3. Keandalan Sistem (*Reliability*)

Sistem yang baik tentunya harus andal. Keandalan sistem informasi adalah ketahanan sistem informasi dari kerusakan dan kesalahan. Keandalan sistem informasi ini juga dapat dilihat dari sistem informasi dalam melayani kebutuhan pengguna tanpa adanya masalah yang dapat mengganggu kenyamanan pengguna dalam menggunakan sistem tersebut.

4. Fleksibilitas Sistem (*Flexibility*)

Fleksibilitas juga merupakan salah satu indikator baik buruknya sistem. Fleksibilitas yang dimaksud adalah kemampuan sistem informasi dalam melakukan perubahan-perubahan yang terkait dengan memenuhi kebutuhan pengguna. Pengguna akan merasa lebih puas menggunakan sistem informasi jika sistem tersebut fleksibel dalam memenuhi kebutuhan pengguna.

5. Keamanan Sistem (*Security*)

Keamanan merupakan salah satu faktor terpenting dalam keberlangsungan sistem. Keamanan sistem yang buruk dapat menyebabkan kerugian yang fatal bagi pengguna. Keamanan sistem dapat dilihat melalui program yang tidak dapat diakses oleh pengguna yang tidak bertanggung jawab dan juga data program tidak dapat hilang dengan mudah walaupun terdapat kesalahan dari pengguna.

2.1.5 Informasi

2.1.5.1 Pengertian Informasi

Informasi adalah data yang diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan pada saat sekarang atau yang akan datang. Informasi juga merupakan fakta-fakta atau data yang telah diproses sedemikian rupa atau mengalami proses transformasi data sehingga berubah bentuk menjadi informasi. Informasi dibagi menjadi beberapa jenis berdasarkan fungsi, format penyajian, lokasi peristiwa dan bidang kehidupan.

2.1.5.2 Kualitas Informasi

Menurut Eko [16], Informasi merupakan data yang telah diorganisir sehingga memberikan arti dan nilai kepada penerimanya. Informasi juga dapat diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima dan membutuhkannya.

Kualitas informasi dapat berdampak terhadap individu yang diantaranya adalah kepuasan pengguna informasi tersebut. Indikator yang dapat digunakan untuk mengukur kualitas informasi adalah *relevance*, *accurate*, *timeliness*, dan *understandability* dari informasi yang dihasilkan.

2.1.6 Android

2.1.6.1 Pengertian Android

Android adalah 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. Sistem operasi ini menjadi sangat populer di Indonesia. Dengan sistem operasi android ini para pengembang dapat lebih mudah membuat aplikasi mobile yang baik dan terbaharukan.

2.1.6.2 Komponen – komponen Android



Gambar 2.1 Komponen Android

Sistem operasi android memiliki beberapa komponen utama yang disebut dengan arsitektur platform android. Komponen – komponen tersebutlah yang menyusun sistem android. Berikut adalah diagram komponen komponen utama dari platform android beserta penjelasan singkat beberapa komponennya :

1. *Linux Kernel*

Fondasi platform android adalah *linux kernel*. *Linux kernel* ini merupakan lapisan paling mendasar pada komponen android. Contohnya adalah *Android Runtime* (ART) yang bergantung pada *linux kernel* untuk fungsionalitas dasar seperti *threading* dan manajemen memori tingkat rendah.

2. *Hardware Abstraction Layer*

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem android memuat modul pustaka untuk komponen perangkat keras tersebut.

3. *Android Runtime*

Komponen selanjutnya adalah *Android Runtime* (ART). Perangkat Android yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap 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 didesain khusus untuk Android yang dioptimalkan untuk footprint memori minimal.

4. *Native C/C++ Libraries*

Komponen berikutnya yaitu *Native C/C++ Libraries*. *Native C/C++ Libraries* adalah *library* yang mendukung komponen - komponen layanan sistem android yang dibuat dengan bahasa c dan c++. Library ini senantiasa mendukung kinerja layanan android.

5. *Java API Framework*

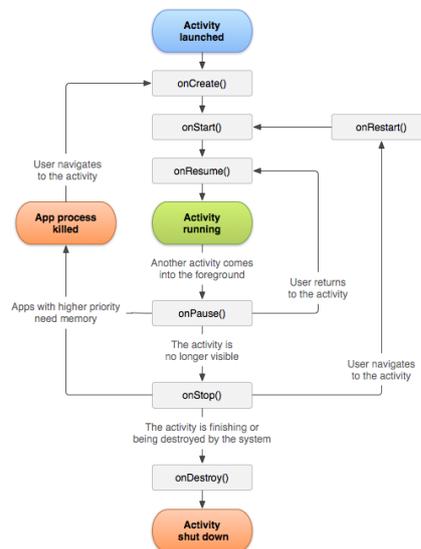
Komponen selanjutnya adalah *Java API Framework*. Keseluruhan rangkaian fitur pada Android OS tersedia untuk melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang *developer* butuhkan untuk membuat aplikasi Android dengan menyederhanakan penggunaan API .

6. System Apps

Terdapat beberapa aplikasi inti yang sudah terpasang secara *default* pada android. Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi yang disertakan bersama *platform* tidak memiliki status khusus pada aplikasi yang ingin dipasang pengguna.

2.1.6.3 Siklus Hidup Android

Di dalam sistem operasi android terdapat siklus hidup. Untuk memberikan navigasi antara siklus hidup ini, android itu sendiri telah menyediakan 6 *method* inti *callback* yaitu *onCreate()*, *onStart()*, *onResume()*, *onPause()*, *onStop()*, and *onDestroy()*. Dalam siklus hidup *method callback*, kita dapat mendeklarasikan cara perilaku *activity* saat pengguna meninggalkan dan memasuki kembali ke sebuah *activity*. Disitulah peranan *method callback* dalam sebuah aplikasi pada *platform* Android. Berikut adalah gambar siklus hidup pada *platform* Android:



Gambar 2.2 Siklus Hidup Android

Sumber Gambar : <https://developer.android.com/guide/components/activities/activity-lifecycle.html>

Keenam *method callback* tersebut memiliki peranan masing-masing dan dijelaskan secara rinci sebagai berikut :

1. *OnCreate()*

Method ini adalah *method* utama dari setiap *activity*. *Method* ini akan dipanggil pertama kali ketika menjalankan sebuah sistem. Pengembang harus mengimplementasikan metode *onCreate()* untuk menjalankan logika memulai aplikasi dasar yang hanya boleh terjadi satu kali selama hidup aktivitas. Misalnya, implementasi *onCreate()* Anda harus mendefinisikan antarmuka pengguna dan mungkin membuat *instance* beberapa variabel dalam cakupan-kelas.

2. *OnStart()*

Method ini dipanggil ketika *method onCreate()* telah dipanggil. *Method onStart()* dipanggil ketika *activity* terlihat oleh *user*. *Method* ini selesai dengan cepat dan dilanjutkan dengan *method* setelahnya yaitu *onResume()*.

3. *OnResume()*

Method ini dipanggil ketika *method onStart()* selesai dipanggil. *Method* ini adalah keadaan dimana pengguna berinteraksi dengan aplikasi. Aplikasi akan tetap dalam keadaan ini sampai terjadi suatu *statement* dari aplikasi semisal menerima panggilan telepon atau mematikan layar *smartphone*.

4. *OnPause()*

Method ini dipanggil ketika pengguna meninggalkan *activity* Pemanggilan *method* ini tidak selalu berarti berarti suatu *activity* telah dihancurkan. *Method* ini berguna untuk menghentikan sementara operasi yang sedang berjalan semisal menjeda pemutaran musik dan lain-lain.

5. *OnStop()*

Method ini dipanggil ketika *activity* tidak terlihat lagi oleh pengguna, dengan kata lain *activity* berhenti dijalankan. Hal ini dapat terjadi semisal ada aktivitas baru dijalankan meliputi seluruh layar. Sistem juga dapat menghubungi *method* ini ketika *activity* selesai berjalan, dan akan segera dihentikan.

6. *OnDestroy()*

Method terakhir adalah *method onDestroy()*. *Method* ini adalah *method callback* ketika *activity* telah selesai dijalankan dan kemudian memanggil *method*

finish(). Method ini juga dipanggil saat sistem untuk sementara menghancurkan proses yang berisi *activity* tersebut untuk menghemat ruang memori.

2.1.7 GPS

Global Positioning System atau yang biasa disingkat dengan GPS adalah alat navigasi elektronik yang menerima informasi dari 4 - 12 satelit sehingga GPS bisa memperhitungkan posisi dimana kita berada di Bumi. Satelit GPS tidak mentransmisikan informasi posisi kita, yang ditransmisikan satelit adalah posisi satelit dan jarak penerima GPS kita dari satelit. GPS terdiri dari tiga bagian yaitu sistem kontrol, satelit dan pengguna. Sistem kontrol adalah bagian yang mengontrol pergerakan satelit-satelit yang ada dan saling berinteraksi satu sama lain kemudian pengguna adalah alat navigasi yang digunakan seperti perangkat *mobile* yang kini sudah memiliki fitur GPS di dalamnya. GPS biasanya digunakan untuk menunjukkan suatu lokasi yang berada di permukaan bumi dengan tingkat akurasi yang cukup baik yaitu kurang dari 10 meter selama tidak ada benda padat yang dapat menghambat sinyal untuk mendapatkan lokasi pengguna.

2.1.8 Location Based Service

Location Based Service merupakan layanan informasi yang dapat diakses menggunakan *mobile devices*, yang dilengkapi kemampuan untuk mengetahui keberadaan lokasi dari pengguna perangkat dan kemampuan memberikan informasi mengenai layanan yang tersedia berdasarkan lokasi mereka pada saat itu. *Location Based Service* dapat didefinisikan sebagai layanan yang mengintegrasikan lokasi perangkat *mobile* atau posisi dengan informasi lain sehingga dapat memberikan nilai tambah bagi pengguna. LBS terdiri dari dua unsur utama sebagai berikut:

1. *Location Manager (API Maps)*: Menyediakan perangkat bagi sumber atau source untuk LBS API (*Application Programming Interface*). *API Maps* menyediakan fasilitas untuk menampilkan atau memanipulasi peta. Paket ini berada pada “com.google.android.maps;”.
2. *Location Providers (API Location)*: Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. *API Location* berhubungan dengan data

GPS (*Global Positioning System*) dan data lokasi *real-time*. *API Location* berada pada paket Android yaitu dalam paket “android.location”.

2.1.9 API (Application Programming Interface)

API adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Pada konteks web, API merupakan pemanggilan fungsi lewat *Hyper Text Transfer Protocol* (HTTP) dan mendapatkan respon berupa *Extensible Markup Language* (XML) atau *JavaScript Object Notation* (JSON). Pemanggilan fungsi ke suatu situs tertentu akan menghasilkan respon yang berbeda kepada pengguna untuk membangun aplikasi enterprise di dalam websitenya.

Dengan API, panggilan-panggilan yang bolak-balik antar aplikasi diatur melalui *web service*. *Web service* adalah kumpulan standar teknis dan protokol, termasuk XML (*Extensible Markup Language*), yang digunakan oleh aplikasi-aplikasi tersebut selama berkomunikasi di internet. API dan *web service* sepenuhnya bekerja di belakang layar.

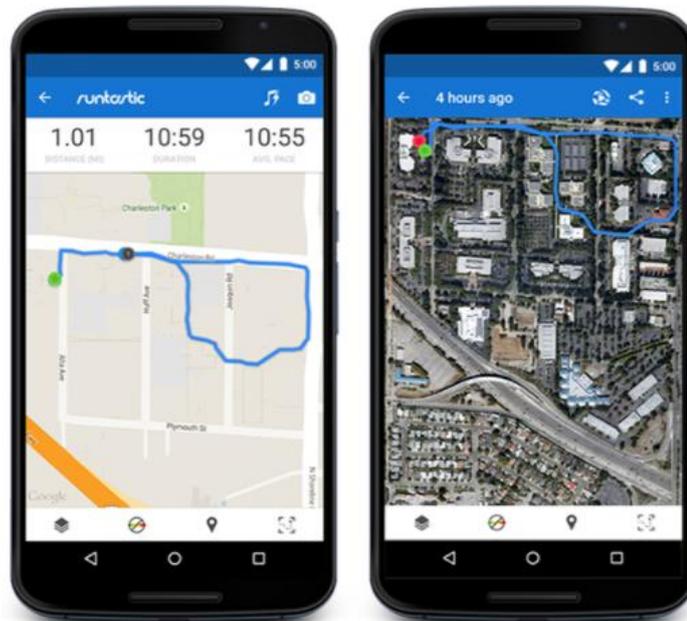
API tentunya dapat menghubungkan berbagai data dari aplikasi yang berbeda. Dengan demikian, API adalah standar komunikasi yang dibuka oleh perusahaan *software*, agar dapat dimanfaatkan oleh pengembang pihak ketiga. Hal ini tentunya dapat memudahkan pihak ketiga yang memanfaatkan layanan API dalam proses pengembangan aplikasi.

2.1.10 Google Maps Android API

2.1.10.1 Pengertian Google Maps Android API

Google Maps Android API adalah layanan untuk menampilkan peta di aplikasi android. Pengembang dapat menambahkan peta ke dalam aplikasi berdasarkan data di *Google Maps API* yang secara otomatis menangani akses ke *server Google Maps*, mengunduh data, menampilkan peta, dan merespons gerakan peta. *Google Maps API* juga menyediakan panggilan API untuk menambahkan *marker*, *poligon*, dan *overlay* ke peta dasar, serta mengubah tampilan area peta

tertentu ke pengguna. Semua objek ini memberikan informasi tambahan tentang lokasi peta, dan memungkinkan interaksi pengguna dengan peta. Seperti terlihat pada gambar berikut:



Gambar 2.3 GPS Pada Android

Sumber Gambar : <https://developers.google.com/maps/?hl=id>

2.1.10.2 Cara Mendapatkan API key Google Maps

Berikut langkah-langkah mendapatkan API key Google Maps :

1. Masuk ke halaman

https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&reusekey=true&hl=id&pli=1

2. Buat atau pilih sebuah proyek.
3. Klik *Continue* untuk mengaktifkan Google Maps Android API.
4. Pada laman *Credentials*, dapatkan kunci API.

Catatan: Jika Anda sudah memiliki kunci API dengan pembatasan Android, Anda boleh menggunakan kunci itu.

5. Dari dialog yang menampilkan kunci API, pilih *Restrict key* untuk menyetel pembatasan Android atas kunci API.

6. Di bagian *Restrictions*, pilih *Android Apps*, kemudian masukkan sidik jari SHA-1 dan nama paket aplikasi Anda. Misalnya :

```
BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:
6E:44:5D:75
```

```
com.example.android.mapexample
```

7. Klik Save.

Setelah melakukan proses diatas, kunci API akan segera didapatkan. Kunci API terbatas android yang baru akan muncul dalam daftar *API key* untuk proyek Anda. Saat menggunakan API standar, bisa menggunakan *API key* yang sama untuk aplikasi Google Maps Android API dan aplikasi *Google Places API for Android*

2.1.11 OpenWeatherMap API

OpenWeatherMap adalah layanan online yang menyediakan data mengenai cuaca, mencakup cuaca saat ini, ramalan cuaca dan histori cuaca untuk para pengembang web dan *mobile*. OpenWeatherMap menggunakan data layanan *brodacast* meteorologi, data mentah dari stasiun cuaca, data mentah dari stasionradar dan data mentah dari lembaga stasiun cuaca resmi. Semua data diproses oleh OpenWeatherMap sedemikian rupa sehingga data yang disediakan secara *real-time* dan akurat.

2.1.12 Java

Java adalah salah satu bahasa pemrograman yang paling populer. *Java* dapat digunakan untuk berbagai hal, termasuk pengembangan perangkat lunak, aplikasi *mobile*, dan pengembangan sistem yang besar. Inilah yang membuat bahasa pemrograman *Java* sangat terkenal di lingkungan pengembang perangkat lunak.

Seperti bahasa pemrograman lain, bahasa *Java* memiliki struktur sendiri, aturan sintaks, dan paradigma pemrograman. Paradigma pemrograman bahasa *Java* didasarkan pada konsep OOP. Bahasa *Java* merupakan turunan bahasa *C*,

sehingga aturan sintaks yang terlihat akan seperti bahasa C. Misalnya, blok kode yang modular dalam metode dan dibatasi oleh karakter ‘(‘ dan ‘)’’, serta variabel dideklarasikan sebelum digunakan. Secara struktural, bahasa Java diatur dengan *package*. Di Dalam *package* ada *class*, dan dalam *class* ada *method*, variabel, konstanta, dan banyak lagi. Adapun sintak dalam bahasa pemrograman Java sebagai berikut :

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

2.1.13 PHP

PHP adalah singkatan dari PHP: *Hypertext PreProcessors*. PHP adalah bahasa pemrograman *interpreter* yang paling banyak digunakan saat ini dikarenakan bersifat *open source* dan juga paling banyak didukung oleh banyak *web server*. PHP merupakan bahasa pemrograman pada sisi *server* yang memperbolehkan programmer menyisipkan perintah – perintah perangkat lunak *web server* (Apache, IIS, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke *browser* yang me-*request*-nya.

Sebagai bahasa pemrograman untuk tujuan umum, kode PHP diproses oleh aplikasi penerjemah dalam modus baris - baris perintah dan melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program pada *channel output* standar. Hal ini juga dapat berfungsi sebagai aplikasi grafis. PHP tersedia sebagai prosesor untuk *server web* yang paling modern dan sebagai penerjemah mandiri pada sebagian besar sistem operasi dan komputer *platform*.

2.1.14 JSON (Javascript Object Notation)

JSON adalah format penukaran data yang sederhana. Bagi *programmer*, format ini mudah dibaca dan ditulis. Bagi mesin, format ini mudah untuk proses *parse* dan *generate*. JSON merupakan format teks bahasa pemrograman yang

berdiri sendiri namun menggunakan konvensi standar yang biasa digunakan oleh para *programmer* bahasa pemrograman C, C++, C#, Java, JavaScript, Perl, Python, dll. Hal ini menjadikan JSON sebagai bahasa penukaran data yang ideal. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Adapun contoh dari data dengan format JSON adalah sebagai berikut:

```
{
  "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language, used to create markup languages such as DocBook.",
            "GlossSeeAlso": ["GML", "XML"]
          },
          "GlossSee": "markup"
        }
      }
    }
  }
}
```

Gambar 2.4 Contoh Data JSON

Sumber Gambar : <http://json.org/example.html>

2.1.15 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *Relational Database Management System* (RDBMS). MySQL merupakan *Relational Database Management System*

(RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*) dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*).

Berikut contoh sintak pada MySQL :

1. Membuat database

```
CREATE DATABASE bioskop;
```

2. Melihat database

```
SHOW DATABASES;
```

3. Membuat tabel

```
CREATE TABLE operator(  
    id VARCHAR (20) NOT NULL,  
    nama VARCHAR (50) NOT NULL,  
    password VARCHAR(100) NOT NULL,  
    created_at DATETIME NOT NULL,  
    updated_at TIMESTAMP,  
    PRIMARY KEY (id)  
);
```

4. Melihat tabel

```
SHOW TABLES;
```

5. Menghapus tabel

```
DROP TABLE operator;
```

6. Menghapus database

```
DROP DATABASE bioskop;
```

Dalam penggunaan *database* MySQL, setiap perintah yang diketikkan disebut *query*. Perintah MySQL dapat dikategorikan menjadi 3 sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*), dan DCL (*Data Control Language*). Berikut adalah penjelasan dari setiap kategori tersebut:

1. DDL (Data Definition Language)

Perintah dalam SQL yang pertama adalah perintah DDL. DDL sendiri merupakan kependekan dari apa yang dikenal dengan nama *Data Definition Language*. DDL dapat berarti sebuah perintah yang berhubungan dengan pendefinisian dari suatu struktur *database*. Terdapat beberapa perintah DDL pada MySQL sebagai berikut :

1. CREATE berfungsi untuk membuat *database* baru, tabel baru, *view* baru dan kolom.
2. ALTER berfungsi untuk mengubah struktur tabel. Seperti mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom maupun memberikan atribut pada kolom.
3. DROP berfungsi untuk menghapus *database* dan tabel.
4. TRUNCATE berfungsi untuk menghapus semua catatan dari tabel.
5. COMMENT berfungsi untuk menambahkan komentar pada data.
6. RENAME berfungsi untuk mengubah nama obyek.

Adapun contoh sintaks dari perintah DDL pada MySQL sebagai berikut :

```
CREATE DATABASE NILAI;
ALTER TABLE Mahasiswa ADD (NoTelp char(8));
DROP INDEX nama_index ;
TRUNCATE TABLE table_barang;
RENAME table_barang to table_barang_gudang;
```

2. DML (Data Manipulation Language)

Kategori selanjutnya adalah Data Manipulation Language (DML). DML ialah perintah yang digunakan untuk mengelola/memanipulasi data dalam *database*. Terdapat beberapa perintah DML pada MySQL sebagai berikut :

1. SELECT berfungsi untuk mengambil/menampilkan data dari *database*.
2. INSERT berfungsi untuk memasukkan data ke dalam tabel.

3. UPDATE berfungsi untuk memperbarui data dalam tabel.
4. DELETE berfungsi untuk menghapus data dari tabel.
5. CALL berfungsi untuk memanggil subprogram PL / SQL atau Java.
6. EXPLAIN PLAN berfungsi untuk menjelaskan jalur akses ke data.
7. LOCK TABLE berfungsi untuk mengunci tabel.

Adapun contoh sintaks dari perintah DML pada MySQL sebagai berikut :

```
INSERT INTO mahasiswa VALUES ("08052926",
"Frenky","70"); SELECT nama_mahasiswa FROM
mahasiswa;
DELETE FROM mahasiswa;
UPDATE mahasiswa SET nama_mahasiswa='ujang'
WHERE nim='08052926';
```

3. DCL (Data Control Language)

Kategori terakhir adalah Data Control Language (DCL). DCL ialah perintah yang digunakan untuk melakukan pengontrolan data dan server databasenya. Terdapat beberapa perintah DCL pada MySQL sebagai berikut :

1. GRANT berfungsi untuk memberikan hak akses pengguna ke database.
2. REVOKE berfungsi untuk menghilangkan hak akses yang telah diberikan dengan perintah GRANT.

Adapun contoh sintaks dari perintah DML pada MySQL sebagai berikut:

```
GRANT all privileges on * to nm_user@localhost
identified by 'nm_passwd' with grand option;

REVOKE all on nm_db.nm_tbl from nm_user@localhost
identified by 'nm_passwd';
```

2.1.16 Web Server

Web Server adalah sebuah perangkat lunak yang bertugas menerima permintaan client melalui port HTTP maupun HTTPS dan merubahnya ke dalam format HTML. Terdapat beberapa format selain HTML yaitu PHP atau ASP, tetapi format – format tersebut hanyalah berfungsi untuk menghubungkan HTML dengan *database*. Fungsi utama dari web server yaitu untuk melakukan atau akan

transfer berkas permintaan pengguna melalui komunikasi yang telah ditentukan sedemikian rupa. Halaman web yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. Pemanfaatan web server berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang didalamnya berupa teks, video, gambar, file dan banyak lagi.

Web server yang umum digunakan adalah *Apache*. Tugas utama dari Web server adalah menerjemahkan permintaan ke dalam respon yang cocok untuk keadaan pada saat itu, ketika klien membuka komunikasi dengan *Apache*, *Apache* mengirimkan permintaan untuk sumber daya. *Apache* menyediakan sumber daya yang baik atau memberikan respon alternatif untuk menjelaskan mengapa permintaan tidak dapat terpenuhi. Setiap komunikasi HTTP dimulai dengan permintaan dan berakhir dengan jawaban. *Executable Apache* mengambil nama dari protokol, dan pada sistem Unix umumnya disebut *httpd*, kependekan daemon HTTP.

2.1.16.1 Contoh Web Server

1. KSWEB: Server + PHP + MySQL

KSWEB adalah satu set untuk pengembang web untuk platform Android. Paket ini terdiri dari: web server, server FTP, bahasa pemrograman PHP, MySQL DBMS dan scheduler. KSWEB tidak perlu hak root untuk memfungsikan, kecuali, tentu saja, jika ingin menggunakan server pada port 80. KSWEB adalah aplikasi shareware. Setelah peluncuran pertama, pengguna akan memiliki 5 hari percobaan untuk menguji aplikasi. Biaya KSWEB PRO adalah \$ 3,99. Biaya KSWEB Standar adalah \$ 2,99.

2. PAW Server

PAW Server adalah Server Web untuk perangkat Android. Webserver PAW dapat menggunakan fungsi telepon dari browser web, melayani halaman web sendiri atau mengembangkan aplikasi web yang diaktifkan ponsel. Dukungan Tasker dan Locale Plug-in. Plug-in PHP tersedia. Perangkat lunak ini masih dalam versi beta.

3. Bit Web Server

Aplikasi ini adalah aplikasi server web untuk android yang merupakan alat untuk pengembang web, termasuk LIGHTTPD sebagai Web Server, PHP sebagai Server PHP, MYSQL sebagai Server MySQL, PhpMyAdmin sebagai Klien MySQL , dan MSMTMP sebagai SMTP Client. Itu semua dibungkus ke dalam satu aplikasi yang disebut Bit Web Server. Jadi mudah dijalankan di perangkat android, cukup sekali klik maka Anda dapat menjalankan aplikasi web dengan skrip PHP dan database MySQL. Aplikasi ini dirancang khusus untuk ponsel dan tablet, jadi aplikasi ini membutuhkan memori yang sangat kecil.

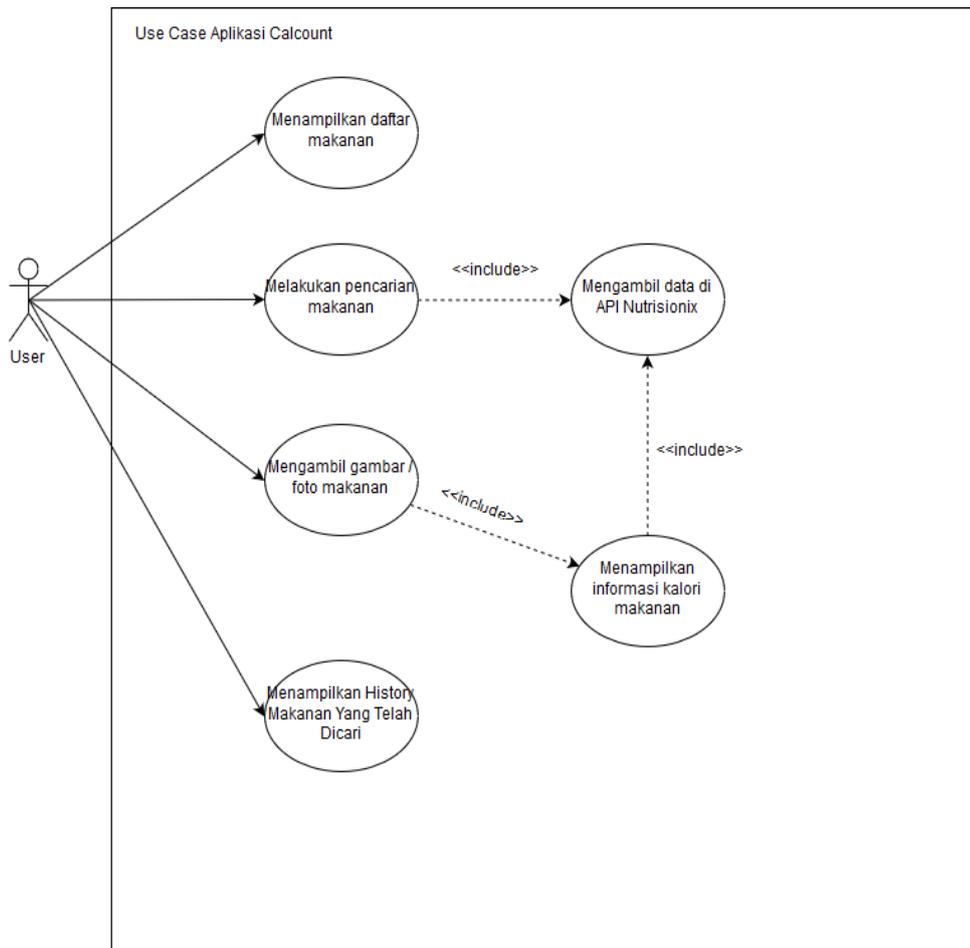
Aplikasi ini berjalan tanpa koneksi internet, membutuhkan internet hanya ketika pertama kali memeriksa lisensi, setelah aplikasi ini memiliki lisensi yang sah maka aplikasi ini dapat berjalan tanpa koneksi internet (OFFLINE). Bit Web Server dapat menjalankan berbagai CMS seperti Wordpress, Joomla, Drupal, Prestashop, dll. Selain itu CMS juga dapat menjalankan kerangka kerja PHP seperti Code Igniter, YII, CakePHP, dll.

2.1.17 UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam berbagai jenis industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

2.1.17.1 Use Case Diagram

Use case diagram merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang menggambarkan bagaimana seseorang atau aktor dalam menggunakan dan memanfaatkan sistem. Use case terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi use case, dan skenario use case. Adapun contoh gambar daripada Use Case Diagram adalah sebagai berikut:



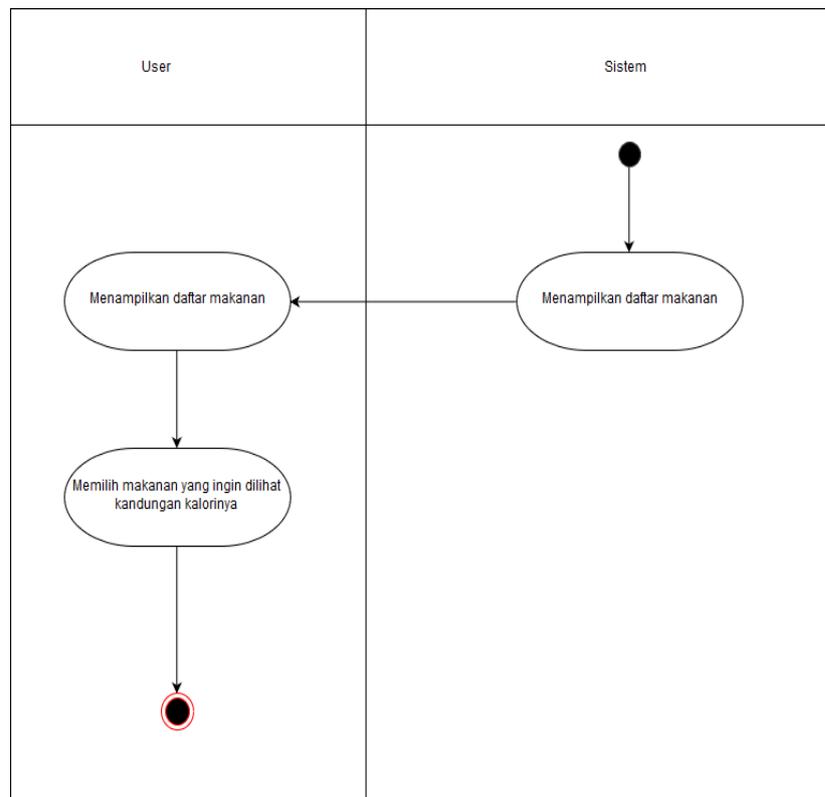
Gambar 2.5 Contoh Use Case Diagram

2.1.17.2 Activity Diagram

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah flowchart karena dapat dimodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Seringkali bermanfaat bila dibuat sebuah activity terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan.

Activity diagram juga sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai use case berinteraksi. Dapat digunakan statechart diagram untuk memodelkan perilaku dinamis satu kelas atau objek. Statechart diagram memperlihatkan urutan keadaan

sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu state atau aktivitas ke state atau aktivitas lainnya, dan aksi yang menyebabkan perubahan satu state lainnya, dan aksi yang menyebabkan perubahan satu state atau aktivitas. Diagram aktivitas paling cocok digunakan untuk memodelkan urutan aktivitas dalam suatu proses. Adapun contoh daripada Activity Diagram adalah sebagai berikut:

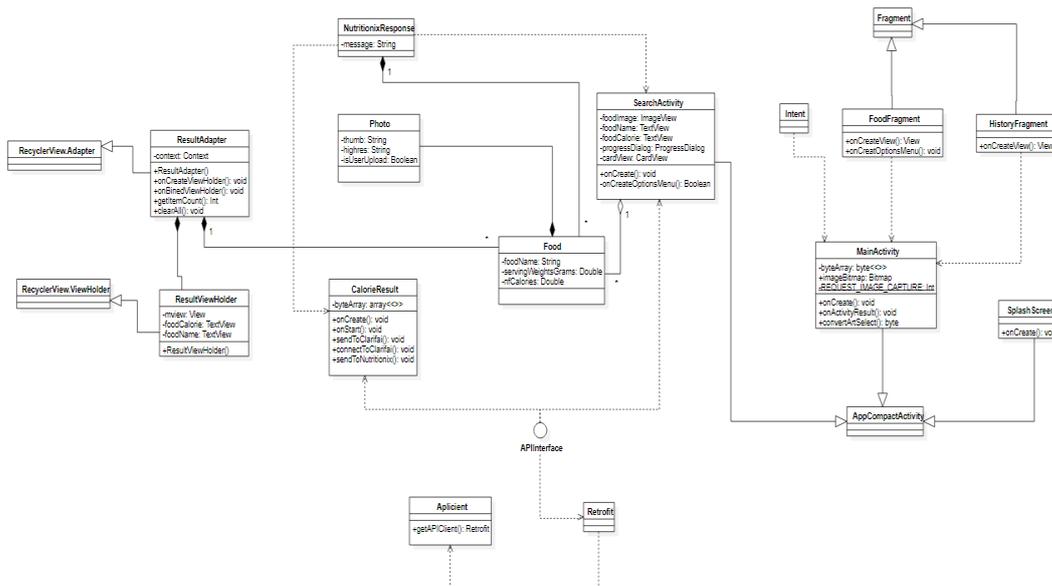


Gambar 2.6 Contoh Activity Diagram

2.1.17.3 Class Diagram

Class diagram membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, class diagram memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, class diagram

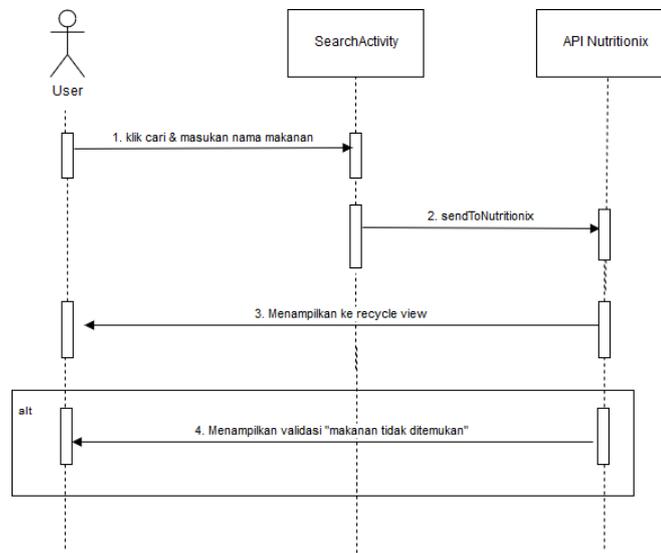
memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Adapun contoh daripada Class Diagram adalah sebagai berikut:



Gambar 2.7 Contoh Class Diagram

2.1.17.4 Sequence Diagram

Diagram sequence menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan use case. Sequence diagram memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam use case. Diagram sequence sebaiknya digunakan diawal tahap desain atau analisis karena kesederhanaannya dan mudah untuk dimengerti. Adapun contoh daripada Sequence Diagram adalah sebagai berikut:



Gambar 2.8 Contoh Sequence Diagram

