

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Makanan**

Makanan adalah segala bahan yang kita makan atau masuk ke dalam tubuh yang membentuk atau menggantikan jaringan tubuh, memberikan tenaga, atau mengatur semua proses dalam tubuh.

Setiap makanan mempunyai kandungan gizi yang berbeda. Pada umumnya terdapat beberapa kandungan pada suatu bahan makanan seperti karbohidrat, air, kalori, protein, vitamin, enzim, pigmen dan lain lain.

Kandungan yang terdapat pada sebuah makanan mempunyai manfaat yang berbeda-beda. Berikut adalah kandungan apa saja yang terdapat pada sebuah makanan beserta fungsinya:

1. Protein  
Berfungsi untuk membangun dan memelihara sel-sel dan jaringan tubuh.
2. Lemak  
Berfungsi sebagai energi cadangan dalam jaringan tubuh, bantalan organ tertentu dalam tubuh dan pelarut vitamin A, D, E, K.
3. Karbohidrat  
Berfungsi sebagai sumber energi utama, pemberi rasa manis pada sebuah makanan, penghemat protein, pengatur metabolisme lemak dan membantu pengeluaran feses.
4. Vitamin
  - a. Vitamin A  
Berfungsi untuk memproses penglihatan, metabolisme umum dan reproduksi.
  - b. Vitamin B  
Berfungsi untuk metabolisme karbohidrat untuk menghasilkan energi, dan pembentukan sel darah merah.

c. Vitamin C

Berfungsi untuk pembentukan jaringan ikat, zat perekat antar sel dan mekanisme imunitas.

d. Vitamin D

Berfungsi sebagai prohormon transport kalsium ke dalam sel.

e. Vitamin E

Berfungsi untuk reproduksi dan antioksidan.

f. Vitamin K

Berfungsi dalam proses sintesa protrombin yang penting dalam mekanisme pembekuan darah.

5. Mineral

Berfungsi sebagai konstituen tulang dan gigi, yaitu kalsium, fosfor, dan magnesium, mengatur komposisi cairan tubuh, menyusun enzim dan menyusun hormon tiroid.

## 2.2 Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran dan penggunaan. Secara istilah aplikasi merupakan suatu program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju [12]. Menurut kamus computer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan [13].

Pada saat ini, aplikasi telah berkembang menjadi beberapa *platform*. Terdapat tiga *platform* utama pada pengembangan sebuah aplikasi, yaitu aplikasi berbasis mobile, web dan desktop.

## 2.3 Android Studio

Android Studio adalah IDE (*Integrated Development Environment*) yang resmi untuk pengembangan aplikasi android yang bersifat *open source* atau gratis [13]. Android studio diluncurkan oleh Google pada tanggal 16 Mei 2013.

Pengumuman tersebut dilakukan di event Google I/O Conference untuk tahun 2013. Android studio menggantikan Eclipse sebagai IDE resmi untuk pengembangan aplikasi android.

Android Studio dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT (*Android Development Tools*) plugin. Android studio mempunyai beberapa fitur, yaitu sebagai berikut [13]:

1. Projek berbasis pada gradle build.
2. Refactory dan pembenahan bug yang cepat.
3. Tools baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
4. Mendukung proguard and app-singning untuk keamanan.
5. Mempunyai GUI aplikasi android yang lebih mudah.
6. Didukung oleh Google Cloude Platform untuk setiap aplikasi yang dikembangkan.

#### **2.4 Java Development Kit (JDK)**

*Java Development Kit* (JDK) adalah sekumpulan perangkat lunak yang dapat digunakan untuk mengembangkan perangkat lunak berbasis Java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber compiler java, buldling, debuggers, development libraries dan lain sebagainya. JRE adalah sebuah implementasi dari *Java Virtual Machine* yang benar-benar digunakan untuk menjalankan program Java [13].

#### **2.5 Web Service**

*Web Service* adalah sekumpulan aplikasi logik beserta objek-objek dan metode-metode yang dimilikinya, yang terletak disuatu server yang terhubung ke internet sehingga dapat diakses menggunakan protokol HTTP (*Hypertext Transfer Protocol*) maupun SOAP (*Simple Object Access Protocol*) [13].

Dalam penggunaannya, *web service* dapat digunakan untuk memeriksa data user yang *login* ke sebuah website ataupun untuk digunakan pada transaksi perbankan online yang rumit [13].

Tujuan dari *web service* adalah untuk memudahkan beberapa aplikasi kompenennya untuk saling terhubung dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi menggunakan standar yang tidak terikat dengan *platform* dan tidak terikat dengan bahasa pemrograman yang digunakan [13].

## 2.6 Sistem Rekomendasi

Sistem rekomendasi adalah suatu sistem yang menyarankan *informasi* yang berguna atau menduga apa yang akan dilakukan pelanggan untuk mencapai tujuannya [14]. Contoh dari sistem rekomendasi yaitu sistem rekomendasi pembelian suatu produk. Dengan adanya sistem rekomendasi tersebut dapat membuat pelanggan memilih produk dengan lebih efektif dalam menentukan produk yang diinginkan.

Rekomendasi menjadi bagian yang sangat penting saat ini, terutama pada industri *online shop*. Pelanggan akan sangat terbantu dengan adanya fitur rekomendasi ini. Hal itu dikarenakan pelanggan terbiasa membeli suatu produk berdasarkan hasil rekomendasi. Contohnya seperti rekomendasi produk berdasarkan jumlah pembeli, harga, rating, daerah dan lain-lain.

## 2.7 Android

Android adalah sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, middleware atau aplikasi [13]. Android merupakan sistem operasi *smartphone* yang paling banyak digunakan pada saat ini. Android menyediakan *platform* secara gratis atau terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Pada awalnya, Google Inc. membeli android Inc. yang merupakan pendatang baru dalam pengembangan peranti lunak untuk ponsel atau *smartphone*. Kemudian dalam pengembangan android ini, Google Inc. membentuk *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia [13].

Pada saat perilisan perdana android pada 5 November 2007, android bersama *Open Handset Alliance* menyatakan mendukung pengembangan aplikasi open source pada perangkat mobile. Google juga merilis kode-kode android dibawah linesi apache, sebuah lisensi perangkat lunak dan open *platform* perangkat lunak [13].

### 2.7.1 Google Maps 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.

Terdapat beberapa langkah-langkah yang harus dikerjakan dalam mengimplementasikan Google Maps API, berikut adalah langkah-langkahnya:

1. Masuk ke halaman google maps pada link berikut:  
*https://console.developers.google.com/flows/enableapi?apiid=maps\_android\_backend&reusekey=true&hl=id&pli=1*
2. Buat atau pilih sebuah projek.
3. Klik *Continue* untuk mengaktifkan Google Maps Android API.
4. Pada laman *Credentials*, dapatkan kunci API. Jika sudah mempunyai kunci API sebelumnya, dapat menggunakan kunci API sebelumnya.
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:

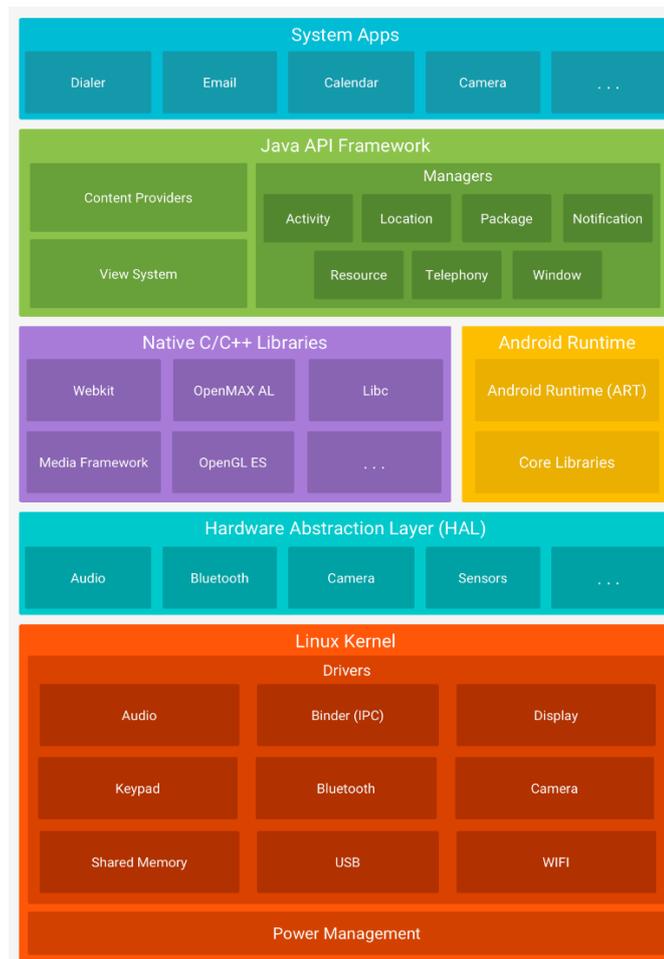
```
BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75
com.example.android.mapexample
```

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*.

### 2.7.2 Arsitektur Android

Android merupakan tumpukan perangkat lunak berbasis Linux *open source* yang dibuat untuk berbagai perangkat dan faktor bentuk. Arsitektur android dapat dilihat pada Gambar 2.1.



Sumber Gambar : [https://developer.android.com/guide/platform/images/android-stack\\_2x.png?hl=ID](https://developer.android.com/guide/platform/images/android-stack_2x.png?hl=ID)

**Gambar 2.1 Arsitektur Android**

a. Linux Kernel

Fondasi *platform* Android adalah kernel Linux. Sebagai contoh, Android *Runtime* (ART) bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan manajemen memori tingkat rendah.

Menggunakan kernel Linux memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan driver perangkat keras untuk kernel yang cukup dikenal.

b. Hardware Abstraction Layer (HAL)

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.

c. Android Runtime

Untuk perangkat 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. Buat rantai aplikasi, misalnya Jack, mengumpulkan sumber Java ke bytecode DEX, yang dapat berjalan pada *platform* Android.

Beberapa fitur utama ART mencakup:

1. Kompilasi mendahului waktu (AOT) dan tepat waktu (JIT).
2. Pengumpulan sampah (GC) yang dioptimalkan.

3. Dukungan debug yang lebih baik, mencakup *profiler* sampling terpisah, pengecualian diagnostik *mendetail* dan laporan kerusakan dan kemampuan untuk mengatur titik pantau guna memantau bidang tertentu.

Sebelum ke Android versi 5.0 (API level 21), Dalvik adalah waktu proses Android. Jika aplikasi Anda berjalan baik pada ART, semestinya berfungsi baik juga pada Dalvik, tetapi mungkin tidak sebaliknya.

Android juga menyertakan serangkaian pustaka waktu proses inti yang menyediakan sebagian besar fungsionalitas bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8, yang digunakan kerangka kerja Java API.

d. Pustaka C/C++ Asli

Banyak komponen dan layanan sistem Android inti seperti ART dan HAL dibuat dari kode asli yang memerlukan pustaka asli yang tertulis dalam C dan C++. Platform Android memungkinkan kerangka kerja Java API mengekspos fungsionalitas beberapa pustaka asli pada aplikasi. Misalnya, Anda bisa mengakses OpenGL ES melalui kerangka kerja Java OpenGL API Android guna menambahkan dukungan untuk menggambar dan memanipulasi grafik 2D dan 3D pada aplikasi Anda.

Jika Anda mengembangkan aplikasi yang memerlukan kode C atau C++, Anda bisa menggunakan Android NDK untuk mengakses beberapa pustaka *platform* asli langsung dari kode asli.

e. Kerangka Kerja Java API

Keseluruhan rangkaian fitur pada Android OS tersedia untuk Anda melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang Anda perlukan untuk membuat aplikasi Android dengan menyederhanakan penggunaan kembali inti, komponen dan layanan sistem modular, yang menyertakan berikut ini:

1. Tampilan Sistem yang kaya dan luas bisa Anda gunakan untuk membuat UI aplikasi, termasuk daftar, kisi, kotak teks, tombol, dan bahkan browser web yang dapat disematkan.
2. Pengelola Sumber Daya, memberikan akses ke sumber daya bukan kode seperti string yang dilokalkan, grafik, dan file layout.
3. Pengelola Notifikasi yang mengaktifkan semua aplikasi guna menampilkan lansiran khusus pada bilah status.
4. Pengelola Aktivitas yang mengelola daur hidup aplikasi dan memberikan back-stack navigasi yang umum.
5. Penyedia Materi yang memungkinkan aplikasi mengakses data dari aplikasi lainnya, seperti aplikasi Kontak, atau untuk berbagi data milik sendiri.

Developer memiliki akses penuh ke API kerangka kerja yang sama dengan yang digunakan oleh aplikasi sistem Android.

f. Aplikasi Sistem

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. Jadi, aplikasi pihak ketiga dapat menjadi browser web utama, pengolah pesan SMS atau bahkan keyboard utama (beberapa pengecualian berlaku, seperti aplikasi Settings sistem).

Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh developer dari aplikasi mereka sendiri. Misalnya, jika aplikasi Anda ingin mengirimkan pesan SMS, Anda tidak perlu membangun fungsionalitas tersebut sendiri—sebagai gantinya Anda bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang Anda tetapkan.

### 2.7.3 Versi Android

Versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Versi komersial pertama, Android 1.0, dirilis pada September 2008. Android dikembangkan secara berkelanjutan oleh Google dan Open Handset Alliance (OHA), yang telah merilis sejumlah pembaruan sistem operasi ini sejak dirilisnya versi awal.

Sejak April 2009, versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan penganan manis. Masing-masing versi dirilis sesuai urutan alfabet, yakni Astro (1.0), Bender (1.1), Cupcake (1.5), Donut (1.6), Eclair (2.0 – 2.1), Froyo (2.2 – 2.2.3), Gingerbread (2.3 – 2.3.7), Honeycomb (3.0 – 3.2.6), Ice Cream Sandwich (4.0 – 4.0.4), Jelly Bean (4.1 – 4.3), KitKat (4.4), Lollipop (5.0 – 5.1), Marshmallow (6.0), hingga yang terbaru adalah Nougat (7.0 – 7.1), Android O (8.0 – 8.1) dan Android Pie (9.0).

Berikut adalah versi-versi android berdasarkan waktu rilisnya:

1. Android 1.0 Astro (Alpha)

Android 1.0, versi komersial pertama Android, dirilis pada 23 September 2008. Perangkat Android pertama yang tersedia secara komersial adalah HTC Dream.

2. Android 1.1 Bender (Beta)

Pada 9 Februari 2009, pemutakhiran Android 1.1 dirilis, awalnya hanya untuk HTC Dream. Android 1.1 juga dikenal dengan "Petit Four", meskipun nama ini tidak digunakan secara resmi. Versi ini memperbaiki beberapa bug, mengubah API Android, dan menambahkan beberapa fitur.

3. Android 1.5 Cupcake

Pada 27 April 2009, Android 1.5 dirilis, menggunakan kernel Linux 2.6.27. Versi ini adalah rilis pertama yang secara resmi menggunakan nama kode berdasarkan nama-nama makanan pencuci mulut ("Cupcake"), nama yang kemudian digunakan untuk semua versi rilis selanjutnya. Pembaruan pada versi ini termasuk beberapa fitur baru dan perubahan UI.

4. Android 1.6 Donut

Pada 15 September 2009, SDK Android 1.6 – dinamai Donut – dirilis, berdasarkan kernel Linux 2.6.29. versi ini hadir untuk menutupi bug pada versi sebelumnya, sekaligus untuk penambahan beberapa fitur seperti misalnya dukungan untuk perangkat dengan ukuran layar yang lebih besar.

#### 5. Android 2.0 – 2.1 Éclair

Pada 26 Oktober 2009, SDK Android 2.0 – dinamai Eclair – dirilis, berbasis kernel Linux 2.6.29. Versi 2.0.1 rilis tanggal 3 Desember 2009 dan versi 2.1 rilis tanggal 12 Januari 2010. Beberapa penambahan fitur pada versi ini seperti Bluetooth, flash pada kamera, digital zoom pada kamera, multi touch, live wallpaper dan lainnya.

#### 6. Android 2.2 – 2.2.3 Froyo

Pada 20 Mei 2010, SDK Android 2.2 (Froyo, singkatan untuk frozen yogurt) dirilis, yang berbasis kernel Linux 2.6.32.

#### 7. Android 2.3 – 2.3.7 Gingerbread

Pada tanggal 6 Desember 2010, SDK Android 2.3 (Gingerbread) dirilis, berbasis kernel Linux 2.6.35. Versi 2.3.1 yang rilis Desember 2010 dan versi 2.3.2 yang rilis Januari 2011 berisi perbaikan bug pada Google Nexus S. Versi 2.3.3 sampai versi 2.3.7 rilis antara bulan Februari 2011 sampai bulan September 2011.

Dirilis pada tanggal 6 Desember 2010 bersamaan dengan dihidirkannya Nexus S yang merupakan perangkat smartphone seri Nexus yang diproduksi oleh Samsung. Versi OS ini juga mengawali kesuksesan Android di jagad smartphone meski masih kalah populer dengan BlackBerry OS. Beberapa vendor mulai serius untuk menggarap perangkat dengan OS Android. Pada saat itu, Samsung dengan Galaxy series nya berperan besar dalam kesuksesan Android. Promosi yang luar biasa gencarnya membuat orang awam mulai mengenal sistem operasi Android. Bahkan saat itu sebagian besar orang beranggapan bahwa OS Android adalah milik Samsung karena kuatnya branding yang dilakukan oleh Samsung. Ini juga menjadi awal mula kedigdayaan Samsung di jagad smartphone.

#### 8. Android 3.0 – 3.2 Honeycomb

Pada 22 Februari 2011, SDK Android 3.0 (Honeycomb) – pembaruan pertama Android yang ditujukan hanya untuk komputer tablet – dirilis, berdasarkan kernel Linux 2.6.36. Perangkat pertama yang menggunakan versi ini adalah tablet Motorola Xoom, yang dirilis pada 24 Februari 2011. Versi 3.1 rilis tanggal 10 Mei 2011. Versi 3.2 rilis tanggal 15 Juli 2011. Google TV generasi pertama dan kedua menggunakan Honeycomb.

9. Android 4.0 – 4.0.4 Ice Cream Sandwich

SDK Android 4.0.1 (Ice Cream Sandwich), berdasarkan kernel Linux 3.0.1, dirilis pada 19 Oktober 2011. Petinggi Google, Gabe Cohen, menyatakan bahwa Android 4.0 "secara teoretis kompatibel" dengan perangkat Android 2.3x yang diproduksi pada saat itu. Kode sumber untuk Android 4.0 tersedia pada tanggal 14 November 2011.

10. Android 4.1 – 4.3 Jelly Bean

Google mengumumkan Android 4.1 (Jelly Bean) dalam konferensi Google I/O pada tanggal 27 Juni 2012. Berdasarkan kernel Linux 3.0.31, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna (UI). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasukantisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus. Android 4.1 Jelly Bean dirilis untuk Android Open Source Project pada tanggal 9 Juli 2012. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

Google berencana merilis Jelly Bean 4.2 pada sebuah acara di New York City pada 29 Oktober 2012, tapi dibatalkan karena Badai Sandy. Jelly Bean 4.2 didasarkan pada kernel Linux 3.4.0, dan pertama kali digunakan pada Nexus 4 LG dan Nexus 10 Samsung, yang dirilis pada 13 November 2012.

Google merilis Jelly Bean 4.3 pada 24 Juli 2013 di San Francisco. Kebanyakan perangkat Nexus menerima pembaruan dengan segera. Nexus 7 generasi kedua adalah perangkat pertama yang menggunakan sistem operasi ini. Sebuah pembaruan minor dirilis pada tanggal 22 Agustus 2013.

#### 11. Android 4.4 Kitkat

Google mengumumkan Android 4.4 KitKat (dinamai dengan izin dari Nestlé dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013. Sebelumnya, rilis berikutnya setelah Jelly Bean diperkirakan akan diberi nomor 5.0 dan dinamai 'Key Lime Pie'.

#### 12. Android 5.0 – 5.1 Lollipop

Android 5.0 pertama kali diperkenalkan di bawah codename "Android L" pada 25 Juni 2014 selama presentasi keynote pada konferensi pengembang Google I / O. Di samping Lollipop, presentasi difokuskan pada sejumlah *platform* Android yang berorientasi dan teknologi baru, termasuk Android TV, pada *platform* Android Auto, dapat dipakai pada *platform* komputasi Android Wear, dan *platform* pelacakan kesehatan Google Fit.

Bagian dari presentasi didedikasikan untuk bahasa desain *cross-platform* baru yang disebut sebagai "material design". Memperluas pada "kartu" motif pertama kali terlihat di Google Now, adalah desain dengan peningkatan penggunaan layout berbasis grid, animasi dan transisi responsif, padding, dan efek kedalaman seperti pencahayaan dan bayangan.

1. Desain antarmuka (tampilan) yang dinamakan "Material Design".
2. 64-bit ART compiler
3. Project volta, yang berguna untuk meningkatkan daya hidup baterai 30% lebih tahan lama.
4. 'factory reset protection'. Fitur ini berguna ketika smartphone hilang, ia tidak bisa direset ulang tanpa memasukkan id google dan kata sandi (password).

#### 13. Android 6.0 Marshmallow

Android Marshmallow memperkenalkan model izin yang didesain ulang: sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. Sebuah sistem opt-in sekarang digunakan, di mana pengguna akan diminta untuk memberikan atau menolak izin individu (seperti kemampuan untuk mengakses kamera atau mikrofon) untuk aplikasi ketika mereka

dibutuhkan. Aplikasi mengingat hibah izin mereka, dan mereka dapat disesuaikan oleh pengguna setiap saat. Model izin baru akan digunakan hanya oleh aplikasi yang dikompilasi untuk Marshmallow menggunakan kit pengembangan perangkat lunak (SDK) tersebut, sementara semua aplikasi lainnya akan terus menggunakan model izin sebelumnya.

Marshmallow juga memiliki skema manajemen daya baru bernama Doze yang mengurangi tingkat aktivitas aplikasi latar belakang saat perangkat menentukan bahwa itu tidak sedang aktif ditangani oleh pengguna, yang, menurut Google, menggandakan pemakaian baterai perangkat. Hal ini juga memperkenalkan pilihan untuk mengatur ulang semua pengaturan jaringan, tersedia untuk pertama kalinya pada Android, yang membersihkan pengaturan terkait jaringan untuk WI-FI, *Bluetooth* dan koneksi seluler.

Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan otentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan otentikasi berbasis sidik jari dalam aplikasi lain. Android Marshmallow mendukung USB Type-C, termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui USB. Marshmallow juga memperkenalkan "pranala yang diverifikasi" yang dapat dikonfigurasi untuk membuka langsung dalam aplikasi tertentu mereka tanpa petunjuk pengguna lanjut.

Versi API Android yang disediakan oleh Marshmallow adalah 23. Alat pengembang Android Marshmallow tersedia di Pengelola SDK di bawah tingkat API "MNC".

#### 14. Android 7.0 – 7.1 Nougat

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

tanggal 22 Agustus 2016. Pratinjau akhir pembuatannya dirilis pada tanggal 18 Juli 2016, dengan nomor bangunan NPD90G.

Pada tanggal 19 Oktober 2016, Google merilis Android 7.1.1 sebagai pratinjau pengembang untuk Nexus 5X, Nexus 6P dan Pixel C. Pratinjau kedua mulai tersedia pada 22 November 2016, sebelum versi final diluncurkan ke publik pada bulan Desember. 5, 2016

#### 15. Android 8.0 – 8.1 Oreo

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

#### 16. Android 9.0 Pie

Terakhir ada Android 9.0 Pie yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi Android ini memberi banyak ubahan, terutama untuk HP dengan desain baru.

Misal Android 9.0 Pie memberikan navigasi berupa gesture yang menggantikan tombol fisik Home, Back dan Recent Apps.

Fitur lainnya yang cukup berguna adalah sistem notifikasi, pengatur kecerahan hingga sistem screenshoot terbaru yang lebih memudahkan kamu loh.

## 2.8 Java

Java merupakan bahasa pemrograman yang berorientasi objek yang diciptakan oleh Sun Microsystem pada tahun 1995 [15]. Java merupakan salah satu bahasa pemrograman yang banyak digunakan saat ini. Java dapat membuat seluruh bentuk aplikasi, baik dalam bentuk web, desktop, maupun mobile.

Java merupakan bahasa pemrograman *open source* yang artinya terbuka untuk umum sehingga para developer dapat menggunakan bahasa pemrograman secara gratis dalam membuat sebuah produk berupa aplikasi, baik dalam bentuk mobile, desktop, maupun web.

Java mempunyai beberapa karakteristik, yaitu sebagai berikut [15]:

1. Sederhana
2. Berorientasi objek
3. Dapat didistribusikan dengan mudah
4. Diinterpretasi oleh interpreter
5. Aman
6. Portable
7. Multithreading
8. Dinamis
9. Netral
10. Robust

## 2.9 Node.js

Node.js adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi berbasis web. Node.js ditulis dalam bahasa *Javascript* dengan menggunakan basis *event* dan *asynchronous I/O*. Node.js tidak seperti *javascript* pada umumnya yang dijalankan pada peramban melainkan berjalan pada sisi server. Node.js terdiri dari *V8 Javascript Engine* buatan Google dan beberapa modul bawaan yang terintegrasi [16].

Modul-modul yang digunakan dalam implementasi klien SIP ini antara lain *Sip.js* sebagai implementasi protokol SIP pada *node.js*, *Websocket-Node* yang merupakan implementasi *Websocket* pada *Node.js* dan *Express* yang merupakan kerangka kerja HTTP pada *Node.js* [16].

Berikut adalah contoh *source code* pada *Node.js* :

```
//Membuat object dengan property readonly
var facebook = {};
Object.defineProperty(facebook, 'app', { value: 50, writable:
false });
facebook.app = 103; //Dikenali sebagai kesalahan (error)
```

```
// Membuat object dengan property readonly
var twitter = { get app() { return 51; } };
twitter.app = 130; //Dikenali sebagai kesalahan (error)
// Membuat object dengan property pada object yang tidak dikenali
var google = {};
Object.preventExtensions(google);
google.newProp = 'app'; // Dikenali sebagai kesalahan (error)
```

## 2.10 MySQL

MySQL adalah suatu perangkat lunak *database* relasi atau *Relational Database Management System (RDBMS)* yang didistribusikan gratis di bawah lisensi GPL (*General Public License*) [17]. MySQL merupakan *database* yang bersifat gratis digunakan oleh siapa saja, akan tetapi tidak diperbolehkan untuk dijadikan produk turunan yang dijadikan closed source atau komersial. MySQL merupakan jenis basis data yang paling populer saat ini.

Berikut adalah contoh sintak pada MySQL:

### 1. Membuat *Database*

```
CREATE DATABASE peminjaman_buku;
```

### 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 TABLE;
```

### 5. Menghapus Tabel

```
DROP TABLE operator;
```

### 6. Menghapus *Database*

```
DROP DATABASE peminjaman_buku;
```

## 2.11 JSON

JSON (*Javascript Object Notation*) adalah struktur data yang universal, dalam artian dapat digunakan dalam berbagai bahasa pemrograman. JSON merupakan *format* pertukaran data yang ringan untuk memasukkan data ke dalam sebuah variabel, sangat mudah dimengerti dan diimplementasikan, serta mudah juga untuk komputer melakukan parsing [18].

JSON merupakan bagian dari bahasa pemrograman javascript. JSON merupakan *format* teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga C, termasuk C, C++, C#, Java, Javascript, Perl, Python dan sebagainya [18].

JSON dibangun dalam dua struktur, yaitu [18]:

1. Beberapa pasangan yang terdiri dari nama dan nilai. Dalam beberapa bahasa pemrograman biasa disebut dengan istilah *object*, *record*, *struct*, *table hash*, *key list* atau *associative array*.
2. Nilai-nilai yang tersusun secara *ordered list*. Dalam bahasa pemrograman biasa disebut sebagai *array*, *vector*, *list* atau daftar.

Berikut adalah contoh script json:

```
{
  "videos": [
    {
      "id": "71CDEYXw3mM",
      "snippet": {
        "channelId": "UC_x5XG1OV2P6uZZ5FSM9Ttw",
        "title": "Google I/O 101: Q&A On Using Google APIs",
        "categoryId": "28"
      },
      "statistics": {
        "viewCount": "3057",
        "likeCount": "25",
        "dislikeCount": "0",
        "favoriteCount": "17",
```

```

    "commentCount": "12"
  }
}
]
}

```

## 2.12 *Application Programming Interface (API)*

*Application Programming Interface (API)* adalah seperangkat definisi subrutin, protokol, dan alat untuk membangun perangkat lunak aplikasi. Secara umum, ini adalah seperangkat metode komunikasi yang jelas antara berbagai komponen perangkat lunak. API yang bagus memudahkan pengembangan program komputer dengan menyediakan semua blok bangunan, yang kemudian disatukan oleh pemrogram. API mungkin untuk sistem berbasis web, sistem operasi, sistem basis data, perangkat keras komputer atau perpustakaan perangkat lunak. Spesifikasi API dapat mengambil banyak bentuk, namun sering kali mencakup spesifikasi untuk rutinitas, struktur data, kelas objek, variabel atau panggilan jarak jauh. POSIX, Microsoft Windows API, C ++ Standard Template Library dan Java API adalah contoh dari berbagai bentuk API. Dokumentasi untuk API biasanya disediakan untuk memudahkan penggunaan.

## 2.13 **Youtube Data API**

Youtube Data API adalah kumpulan fungsi-fungsi yang disediakan oleh pengembang Youtube sendiri sehingga memungkinkan konten-konten video yang terdapat pada Youtube dan fungsionalitasnya dapat diintegrasikan ke situs web, aplikasi perangkat lunak, atau perangkat lainnya [19].

Berikut adalah contoh *request* dan *response* Youtube Data API:

### a. Request

```

https://www.googleapis.com/youtube/v3/videos?id=7lCDEYXw3mM&key=YOUR_API_KEY&part=snippet,contentDetails,statistics,status

```

### b. Response

```

{
  "kind": "youtube#videoListResponse",

```

```

"etag":
"\UCBpFjp2h75_b92t44sqraUcyu0/sDAIsG9NGKfr6v5AlPZKSEZdtqA\"
,
"videos": [
  {
    "id": "71CDEYXw3mM",
    "kind": "youtube#video",
    "etag":
"\UCBpFjp2h75_b92t44sqraUcyu0/iYynQR8AtacsFUwWmrVaw4Smb_Q\"
,
    "snippet": {
      "publishedAt": "2012-06-20T22:45:24.000Z",
      "channelId": "UC_x5XG1OV2P6uZZ5FSM9Ttw",
      "title": "Google I/O 101: Q&A On Using Google APIs",
      "description": "Antonio Fuentes speaks to us and takes
questions on working with Google APIs and OAuth 2.0.",
      "thumbnails": {
        "default": {
          "url": "https://i.ytimg.com/vi/71CDEYXw3mM/default.jpg"
        },
        "medium": {
          "url":
"https://i.ytimg.com/vi/71CDEYXw3mM/mqdefault.jpg"
        },
        "high": {
          "url":
"https://i.ytimg.com/vi/71CDEYXw3mM/hqdefault.jpg"
        }
      },
      "categoryId": "28"
    },
    "contentDetails": {
      "duration": "PT15M51S",
      "aspectRatio": "RATIO_16_9"
    },
    "statistics": {
      "viewCount": "3057",
      "likeCount": "25",

```

```

        "dislikeCount": "0",
        "favoriteCount": "17",
        "commentCount": "12"
    },
    "status": {
        "uploadStatus": "STATUS_PROCESSED",
        "privacyStatus": "PRIVACY_PUBLIC"
    }
}
]
}

```

#### 2.14 Location Based Service (LBS)

*Location Based Service* (LBS) merupakan salah satu bagian dari implementasi mobile gis yang lebih cenderung memberikan fungsi terapan sehari-hari seperti menampilkan direktori kota, navigasi kendaraan, pencarian alamat, serta jejaring sosial [20].

LBS mempunyai dua unsur utama, yaitu [20]:

1. *Location Manager (API Maps)*

*Location Manager* menyediakan *tools/source* untuk LBS, API Maps menyediakan fasilitas untuk menampilkan, memanipulasi peta beserta *feature* lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya.

2. *Location Providers (API Location)*

*Location Providers* menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. API *Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real time*. Dengan adanya API *Location* kita dapat menentukan lokasi kita saat ini beserta rute menuju tempat tujuan kita.

#### 2.15 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan merekomendasikan sistem perangkat lunak. Unified Modelling Language (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. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET [21].

### 2.15.1 Diagram UML

Setiap diagram yang kompleks dapat dipandang dari sudut yang berbeda-beda, hal ini dapat membuat kita mendapatkan pemahaman secara menyeluruh. Dalam upaya untuk mendapatkan pemahaman secara menyeluruh, UML menyediakan sembilan jenis diagram yang dapat dikelompokkan sifat statis atau dinamis. Berikut adalah sembilan jenis diagram tersebut [21] :

1. Diagram Kelas

Diagram kelas bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi serta relasi.

2. Diagram Objek

Diagram objek bersifat statis. Diagram ini memperlihatkan objek-objek serta relasi antar objek. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas.

3. *Use Case Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. *Sequence Diagram*

Diagram ini bersifat dinamis. Diagram *sequence* merupakan diagram interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu.

5. *Collaboration Diagram*

Diagram ini bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek – objek yang menerima serta mengirim pesan (message).

6. *Statechart Diagram*

Diagram ini bersifat dinamis. Diagram ini memperlihatkan state – state pada sistem, memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem – sistem yang reaktif.

7. *Activity Diagram*

Diagram ini bersifat dinamis. Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. *Component Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan organisasi serta kebergantungan pada komponen – komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelaskelas, antarmuka – antarmuka serta kolaborasi – kolaborasi.

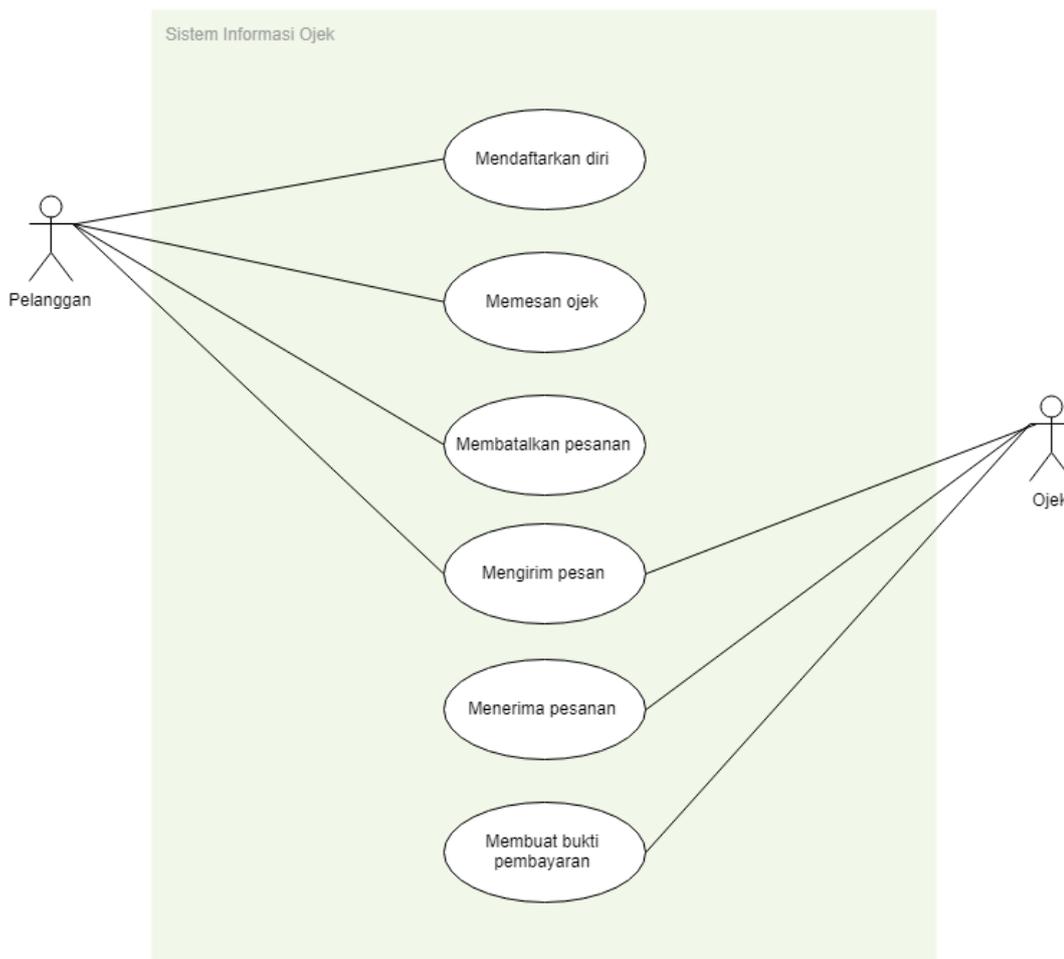
9. *Deployment Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (saat run time). Dengan ini memuat simpul – simpul (node) beserta komponen – komponen yang ada di dalamnya. Deployment diagram berhubungan erat dengan diagram kompoen dimana deployment diagram memuat satu atau lebih komponen – komponen. Diagram ini sangat berguna saat aplikasi berlaku sebagai aplikasi yang dijalankan pada banyak mesin (distributed computing).

Kesembilan diagram yang sudah disebutkan di atas tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semua dibuat sesuai dengan kebutuhan.

### 2.16 Use Case Diagram

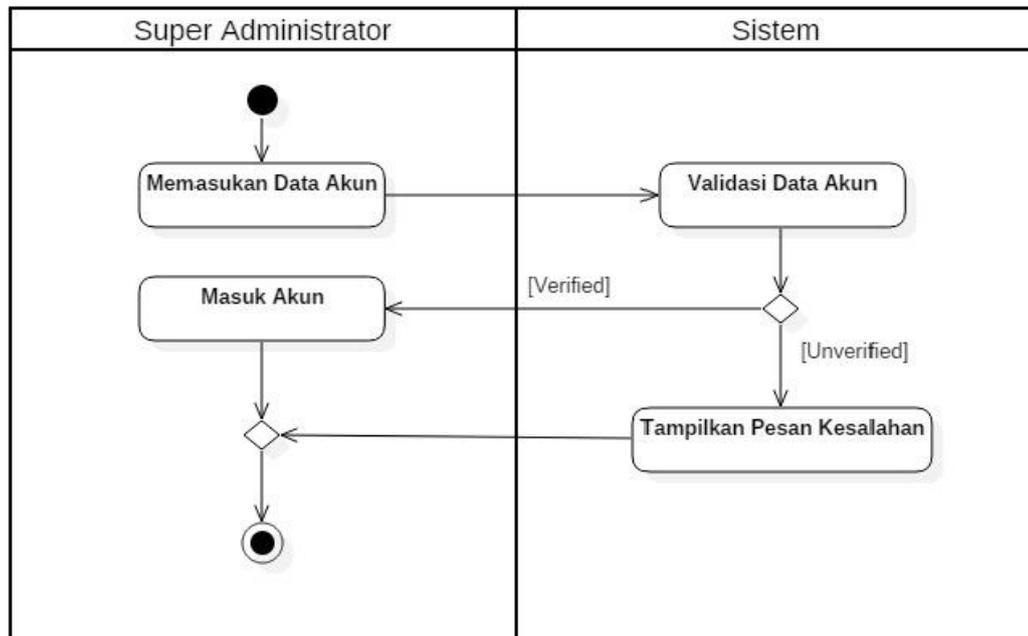
*Use case diagram* merupakan pemodelan untuk kelakuan atau tingkat laku suatu sistem informasi yang nantinya akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada dalam sistem informasi dan siapa saja yang berhak menggunakan aplikasi tersebut [22]. Contoh *use case diagram* dapat dilihat pada Gambar 2.2.



**Gambar 2.2 Use Case Diagram**

### 2.17 Activity Diagram

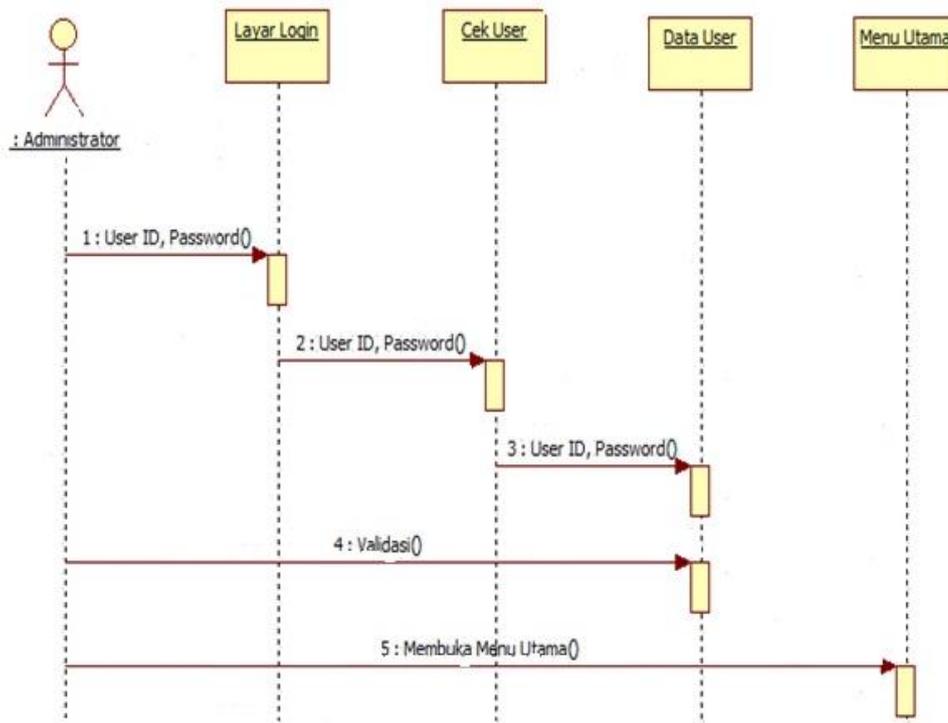
*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis [22]. Contoh *activity diagram* dapat dilihat pada Gambar 2.3.



**Gambar 2.3 Activity Diagram**

### 2.18 Sequence Diagram

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek [22]. Contoh *sequence diagram* dapat dilihat pada Gambar 2.4.



**Gambar 2.4** *Sequence Diagram*

### 2.19 Kalori

Secara umum, kalori adalah jumlah energi yang didapatkan dari suatu makanan dan minuman, atau energi yang kita bakar melalui aktivitas sehari-hari. Sederhananya, kalori adalah energi yang dibutuhkan oleh tubuh agar dapat beraktivitas dan menjalankan fungsinya dengan baik. Menurut kamus medis Medilexion, kalori adalah satuan unit panas atau energi. [23].

Sebagai contoh, satu buah apel yang kita konsumsi mengandung sekitar 80 kalori dan berjalan kaki selama 10 menit membutuhkan sekitar 30 kalori. Kalori yang dibakar oleh tubuh bergantung dengan tinggi badan, berat badan, usia, jenis kelamin dan tingkat aktivitas fisik seseorang.

Terdapat 2 jenis kalori, yaitu kalori kecil (kal) dan kalori besar (Kkal). 1 kalori besar (1 Kkal) sama dengan 1.000 kalori kecil (kal).

Kebutuhan kalori pada seseorang berbeda-beda. Terdapat suatu metode yang dapat menghitung kebutuhan kalori seseorang berdasarkan tinggi badan, berat badan, usia, jenis kelamin dan tingkat aktivitas fisiknya. Metode tersebut dinamakan metode Harris Benedict.

### 2.19.1 Metode Harris Benedict

Komponen utama yang menentukan kebutuhan energi seseorang adalah Angka Metabolisme Basal (AMB) dan aktivitas fisik. Angka metabolisme basal adalah kebutuhan energi minimal yang dibutuhkan tubuh untuk menjalankan proses tubuh [24]. Angka metabolisme basal dinyatakan dalam satuan kilokalori (kcal). Angka metabolisme basal pada penelitian ini menggunakan rumus Harris Benedict yang ditunjukkan pada persamaan berikut:

AMB untuk laki-laki:

$$66 + (13,7 \times BB) + (5 \times TB) - (6,8 \times U) \quad (1)$$

AMB untuk perempuan:

$$655 + (9,6 \times BB) + (1,8 \times TB) - (4,7 \times U) \quad (2)$$

Catatan:

BB = Berat Badan

TB = Tinggi Badan

U = Usia

Setelah nilai AMB diketahui, langkah selanjutnya adalah menghitung aktivitas fisik. Kebutuhan energi berdasarkan aktivitas dapat dilihat pada Tabel 2.1 [24].

**Tabel 2.1 Kebutuhan Energi Berdasarkan Aktivitas**

Aktivitas	Gender	
	Laki-laki	Perempuan
Sangat ringan (<25% bergerak)	1,30	1,30
Ringan (75% duduk atau berdiri, 25% bergerak)	1,65	1,55
Sedang (40% duduk atau berdiri, 60% aktivitas pekerjaan)	1,76	1,70
Berat (25% duduk atau berdiri, 75% aktivitas pekerjaan)	2,10	2,00

Setelah mendapatkan nilai aktivitas fisik yang sesuai dengan jenis aktivitasnya, kalikan nilai tersebut dengan AMB yang ditunjukkan pada persamaan berikut:

$$\text{Kebutuhan energi} = \text{nilai aktivitas fisik} \times \text{AMB} \quad (3)$$

Nilai kebutuhan energi untuk AMB diperhitungkan menurut berat badan normal atau ideal dengan menggunakan Indeks Masa Tubuh (IMT) seperti pada Persamaan berikut:

$$\text{IMT} = \text{Berat Badan (kg)} / \text{Tinggi Badan (m)} \quad (4)$$

Nilai IMT ini digunakan sebagai ambang batas kondisi tubuh. Kategori batas ambang IMT dapat dilihat pada Tabel 2.2.

**Tabel 2.2 Kategori Batas Ambang IMT**

Kondisi Tubuh	Kategori	Batas Ambang
Kurus	Kekurangan berat badan tingkat berat	< 17,0
	Kekurangan berat badan tingkat ringan	17,0 – 18,5
Normal		18,5 – 25,0
Gemuk	Kelebihan berat badan tingkat ringan	25,0 – 27,0
	Kelebihan berat badan tingkat berat	>= 27,0

Bila nilai IMT kurang dari normal maka berat badan kurang dari ideal untuk itu kebutuhan energinya ditambah sebanyak 500 kilokalori. Jika nilai IMT lebih dari normal maka berat badan lebih dari ideal sehingga kebutuhan energinya dikurangi sebanyak 500 kilokalori.

## 2.20 Zomato API

Zomato API adalah API yang menyediakan data-data dari sebuah restoran. Zomato API memberikan Anda akses ke informasi paling baru dan paling lengkap untuk lebih dari 1,5 juta restoran di lebih dari 10.000 kota di dunia.

Fitur yang terdapat pada API Zomato yaitu antara lain:

1. Cari restoran berdasarkan nama, jenis makanan, atau lokasi.
2. Tunjukkan informasi mendetail termasuk nilai, lokasi dan jenis masakan.
3. Gunakan Zomato Foodie Index untuk menemukan area-area dengan restoran terbaik.

Cara menggunakan API Zomato sangat lah mudah. Diharuskan mendapatkan API Key terlebih dahulu sebelum memulai *request*. Contoh *request*-nya dapat dilihat di bawah ini :

```
curl -X GET --header "Accept: application/json" --header "user-
key: 2eb407b7d87b1b6605cbbea95773c110"
"https://developers.zomato.com/api/v2.1/search"
```

Respon yang akan didapatkan nantinya yaitu berupa kode Json atau XML.

Kita bisa memilih antara keduanya. Berikut adalah contoh responsnya:

```
{
  "results_found": "53",
  "results_start": "11",
  "results_shown": "10",
  "restaurants": [
    {
      "id": "16774318",
      "name": "Otto Enoteca & Pizzeria",
      "url": "https://www.zomato.com/new-york-city/otto-enoteca-
pizzeria-greenwich-village",
      "location": {
        "address": "1 5th Avenue, New York, NY 10003",
        "locality": "Greenwich Village",
        "city": "New York City",
        "latitude": "40.732013",
        "longitude": "-73.996155",
        "zipcode": "10003",
        "country_id": "216"
      }
    }
  ]
}
```

```

    },
    "average_cost_for_two": "60",
    "price_range": "2",
    "currency": "$",
    "thumb":
    "https://b.zmtcdn.com/data/pictures/chains/8/16774318/a54deb9e4dbb79dd7c8091b30c642077_featured_thumb.png",
    "featured_image":
    "https://d.zmtcdn.com/data/pictures/chains/8/16774318/a54deb9e4dbb79dd7c8091b30c642077_featured_v2.png",
    "photos_url": "https://www.zomato.com/new-york-city/otto-enoteca-pizzeria-greenwich-village/photos#tabtop",
    "menu_url": "https://www.zomato.com/new-york-city/otto-enoteca-pizzeria-greenwich-village/menu#tabtop",
    "events_url": "https://www.zomato.com/new-york-city/otto-enoteca-pizzeria-greenwich-village/events#tabtop",
    "user_rating": {
        "aggregate_rating": "3.7",
        "rating_text": "Very Good",
        "rating_color": "5BA829",
        "votes": "1046"
    },
    },
    "has_online_delivery": "0",
    "is_delivering_now": "0",
    "has_table_booking": "0",
    "deeplink": "zomato://r/16774318",
    "cuisines": "Cafe",
    "all_reviews_count": "15",
    "photo_count": "18",
    "phone_numbers": "(212) 228-2930",
    "photos": [
        {
            "id": "u_MjA5MjY1OTk5OT",
            "url":
            "https://b.zmtcdn.com/data/reviews_photos/c15/9eb13ceaf6e90129c276ce6ff980bc15_1435111695_640_640_thumb.JPG",
            "thumb_url":
            "https://b.zmtcdn.com/data/reviews_photos/c15/9eb13ceaf6e90129c276ce6ff980bc15_1435111695_200_thumb.JPG",
            "user": {
                "name": "John Doe",
                "zomato_handle": "John",
                "foodie_level": "Super Foodie",
                "foodie_level_num": "9",
                "foodie_color": "f58552",
                "profile_url": "https://www.zomato.com/john",
                "profile_deeplink": "zoma.to/u/1170245",
                "profile_image": "string"
            },
        },
        "res_id": "16782899",
        "caption": "#awesome",
        "timestamp": "1435111770",
        "friendly_time": "3 months ago",
        "width": "640",
        "height": "640",
        "comments_count": "0",

```

```

        "likes_count": "0"
      }
    ],
    "all_reviews": [
      {
        "rating": "5",
        "review_text": "The best latte I've ever had. It tasted
a little sweet",
        "id": "24127336",
        "rating_color": "305D02",
        "review_time_friendly": "2 months ago",
        "rating_text": "Insane!",
        "timestamp": "1435507367",
        "likes": "0",
        "user": {
          "name": "John Doe",
          "zomato_handle": "John",
          "foodie_level": "Super Foodie",
          "foodie_level_num": "9",
          "foodie_color": "f58552",
          "profile_url": "https://www.zomato.com/john",
          "profile_deeplink": "zoma.to/u/1170245",
          "profile_image": "string"
        },
        "comments_count": "0"
      }
    ]
  }
}

```

## 2.21 Smartband

*Smartband* merupakan perangkat *wearable* yang banyak diminati belakangan ini. *Smartband* sangat cocok digunakan untuk orang yang suka berolahraga karena akan mencatat berbagai aktivitas sehari-hari. Pengertian umumnya, *smartband* merupakan sebuah gelang pintar yang mempunyai banyak fitur, terutama fitur untuk mencatat kegiatan seseorang sehari-harinya.

Terdapat banyak sekali fitur yang ditawarkan oleh *smartband*. Fitur pada *smartband* bergantung dengan brand *smartband*. Berikut adalah fitur-fitur yang terdapat pada sebuah *smartband* pada umumnya:

1. Jam alarm
2. Pengingat panggilan dan pesan masuk
3. Waktu
4. Tanggal

5. Perekaman jarak
6. Pengukuran detak jantung
7. Penghitungan kalori yang dibakar
8. Perekaman langkah kaki
9. Pengelolaan tidur
10. Panggilan *bluetooth*, dan lain-lain.

### 2.21.1 Xiaomi Mi Band 2

Xiaomi Mi Band 2 merupakan gelang pintar keluaran Xiaomi yang diluncurkan pada tahun 2016 lalu. Xiaomi Mi Band 2 memiliki banyak fitur, diantaranya adalah menghitung langkah kaki, jarak yang ditempuh, penghitungan kalori yang dibakar, pengukuran detak jantung, pengelolaan tidur, pemberitahuan telepon dan pesan masuk, dan lain-lain.

Xiaomi Mi Band 2 dapat dihubungkan dengan aplikasi android menggunakan *bluetooth*. Salah satu cara untuk menghubungkannya yaitu dengan menggunakan Mi Band 2 JS library. Library ini berjalan menggunakan bahasa Node.js. Berikut adalah cara penggunaannya:

```
'use strict';

function delay(ms) {
  return new Promise(resolve => setTimeout(resolve, ms))
}
async function test_all(miband, log) {
  let info = {
    time:      await miband.getTime(),
    battery:   await miband.getBatteryInfo(),
    hw_ver:    await miband.getHwRevision(),
    sw_ver:    await miband.getSwRevision(),
    serial:    await miband.getSerial(),
  }

  log(`HW ver: ${info.hw_ver} SW ver: ${info.sw_ver}`);
  info.serial && log(`Serial: ${info.serial}`);
  log(`Battery: ${info.battery.level}%`);
  log(`Time: ${info.time.toLocaleString()}`);

  let ped = await miband.getPedometerStats()
  log('Pedometer:', JSON.stringify(ped))

  log('Notifications demo...')
  await miband.showNotification('message');
  await delay(3000);
  await miband.showNotification('phone');
```

```
await delay(5000);
await miband.showNotification('off');

log('Tap MiBand button, quick!')
miband.on('button', () => log('Tap detected'))
try {
  await miband.waitButton(10000)
} catch (e) {
  log('OK, nevermind ;)')
}

log('Heart Rate Monitor (single-shot)')
log('Result:', await miband.hrmRead())

log('Heart Rate Monitor (continuous for 30 sec)...')
miband.on('heart_rate', (rate) => {
  log('Heart Rate:', rate)
})
await miband.hrmStart();
await delay(30000);
await miband.hrmStop();

//log('RAW data (no decoding)...')
//miband.rawStart();
//await delay(30000);
//miband.rawStop();

log('Finished.')
}

module.exports = test_all;
```

