BAB 2
TINJAUAN PUSTAKA

2.1.Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan digunakan oleh berbagai macam perangkat mobile. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk telepon seluler. Kemudian untuk mengembangkan Android,dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan hardware,software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola,Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache.

Di dunia ini terdapat dua jenis distributor sistem operasi *Android*. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD) [4].

2.1.1.Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. Bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Versi android terbaru

yaitu Android versi 3.0. juga sudah bergabung dengan beberapa smart mobile seperti Nokia, Sony Ericsson, dan lainnya [4].

2.1.2.Versi Android

Adapun versi android diantaranya [4]:

2.1.2.1.Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencariansuara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2.1.2.2.Android versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler denganmenggunakan Android dan SDK (Software Development Kit) dengan versi 1.5(Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapafitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke youtube dan gambar ke picasa langsung dari telepon, dukungan bluetooth A2DP, kemampuan terhubung secara otomatis ke headset bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

2.1.2.3. Android versi 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan *Text-to-speech engine*; kemampuan dial kontak; teknologi text to change speech.

2.1.2.4. Android versi 2.0/2.1 (*Eclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

2.1.2.5. Android versi 2.2 (*Froyo: Frozen Yoghurt*)

Pada Mei 2010, Android versi 2.2 Rev 1 diluncurkan. Android inilah yang sangat banyak beredar di pasaran, salah satunya adalah dipakai di Samsung FX tab yang sudah ada di pasaran. Fitur yang tersedia di android versi ini sudah kompleks diantaranya adalah:

1) Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2) *DalvikVirtual Machine* dioptimalkan untuk perangkat mobile.
3) Grafik: grafik di 2D dan grafis 3D berdasarkan libraries OpenGL.
4) SQLite: untuk penyimpanan data.
5) Mendukun media: audio, video dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG,GIF).
6) GSM, Bluetooth, EDGE, 3G, dan WiFi (hardware independent.
7) Kamera, Global Positioning System (GPS), Kompas, dan accelerometer (tergantung hardware).

2.1.2.6. Android versi 2.3 (*Gingerbread*)

Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan seperti berikut:

1) SIP-based VoIP
2) *Near Field Communication* (NFC)
3) *Gyroscope dan sensor*
4) *Multiple cameras support*
5) *Mixable audio effect*

6) *Download manager*

2.1.2.7. Android versi 3.0/3.1 (*Honeycomb*)

Dirilis Februari 2011 sebagai android versi 3.0 revisi 1 serta android versi 3.0 revisi 2 telah dirilis pada juli 2011

2.1.2.8. Android versi 4.0 (*ICS: Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

2.1.2.9. Android versi 4.1 (*Jelly Bean*)

Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat.

2.1.2.9. Android versi 4.4 (*Kitkat*)

OS android versi v4.4 sampai dengan v4.4.4 Dirilis pertama pada tanggal 31 bulan Oktober tahun 2013 di namakan dengan Android kitkat .os android kitkat memiliki tampilan 100% lebih dinamis dan berbeda total dengan android jelly bean,android kitkat di optimasi pada sisi konsumsi baterai dan kinerja os lebih cepat ketika di jalankan pada perangkat memiliki spesifikasi lebih rendah,seperti kita tahu jika andorid jelly bean memiliki kelebihan pada sisi konsumsi baterai yang lebih tinggi dan ketika di jalankan di perangkat yang memiliki versi rendah os ini tidak maksimal[5].

2.1.2.10. Android versi 5.0 (*Lollipop*)

OS android versi v5.0 dirilis pertama pada tahun 2014 dikenal dengan nama "*Android Lollipop*" android versi v5.0 memiliki peningkatan sistem keamanan dan tampilan lebih dinamis ,OS versi terbaru dari android ini sepertinya bisa di gunakan berbagai perangkat elektronik.

Android versi v5.0 memiliki beberapa perubahan di banding dengan android kitkat sebelumnya,android Lollipop memiliki perubahan besar pada sisi tampilan desain menu,tampilan desain menu dan warna. Pada OS android versi v5.0 ini jauh lebih hidup dan tajam di banding android kitkat,karena OS ini sudah di lakukan perubahan pada material desain responsive UI desain sehingga semua menu terkesan sangat sensitif dan alami.

Konsumsi baterai pada android lollipop lebih hemat , jika di android kitkat memiliki perubahan penghematan baterai di banding jelly bean hingga 30 %, Sedangkan android lollipop memiliki perubahan penghematan baterai hingga 50% dibandingkan dengan android jelly bean atau 20% jauh lebih irit di banding dengan android kitkat[5].

2.2.Android Studio

Android Studio adalah sebuah IDE untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangkan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android[6].

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 mem-build multiple APK .
3) Template support untuk Google Services 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.
6) Import library langsung dari Maven repository

2.3.Java

Sejarah pemrograman java dimulai ketika pada tahun 1990, Jamis Gosling, pemrogram yang bekerja pada sun Microsystems mendapat tugas menciptakan program untuk mengendalikan perangkat elektronika rumah tangga. Java adalah penyederhanaan dari C++. Sun sendiri mendeskripsikan java sebagai bahasa pemrograman yang sederhana, berorientasi objek, bertipe statik, tekompilasi, multi-threaded, gerbage collected, netral arsitektur, robust, aman, dan dapat diperluas[7].

1. Sederhana

Java tidak mendukung pointer matematis, casting tipe implisit, struktur atau unions, operator overloading, template, header filer, maupun pewarisan majemuk.

2. Berorientasi objek

Seperti halnya C++, java menggunakan kelas-kelas untuk mengelola kode ke dalam modul-modul logika. Pada saat runtime, program menciptakan objek dari kelas-keals yang ada.

3. Bertipe statik

Semua objek yang digunkan dalam peogram harus dideklarasikan sebelum digunkan.

4. Tekompilasi

Sebelum dapat menjalankan program yang ditulis dalam bahasa pemrograman java, program harus dikompilasi oleh kompilator java. Hasil

kompilasinya berupa file kode byte, yang mirip dengan file kode mesin yang dapat dieksekusi di bawah sistem operasi apa pun asal memiliki interpreter java.

5. Multi-threaded

Program java dapat berisi eksekusi multi threaded yang memungkinkan program mengulas citra pada layar pada satu threas, sementara tetap menerima masukan dari keyboard pada thread utamnya.

6. Gerbage collected (Pembuang sampah)

Program java nenyingkirkan sendiri “sampah-sampah” yang tak berguna, artinya program tidak perlu menghapus objek-objek yang dialokasikannya di memori.

7. Bebas arsitektur

Pada dasarnya, java tidak dirancang untuk prosesor ataupun sistem operasi tertentu. Java dirancang untuk bekerja pada berbagai arsitektur prosesor dan berbagai sistem operasi.

8. Tangguh (robust)

Karena interpreter java memeriksa seluruh akses sistem yang diakukan program, maka program java tidak akan membuat sistem menjadi crash. Bila terjadi masalah serius, program java membuat perkecualian (exception). Exception ini dapat ditangani dan dikelola oleh program tanpa beresiko memacetkan sistem.

9. Aman

Sistem java tidak hanya memverifikasi seluruh akses ke memori tetapi juga menjamin bahwa tidak ada virus yang membonceng pada applet yang sedang berjalan.

10. Dapat diperluas

Program java mendukung metode native yakni fungsi-fungsi yangditulis dalam bahasa lain, biasanya C++. Dukungan terhadap metode native ini memungkinkan pemrogram menulis fungsi yang bisa dieksekusi secara lebih cepat daripada fungsi yang sama yang ditulis dalam java.

2.3.1 Java Script Object Notation(JSON)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini di buat berdasarkan bagian dari Bahasa Pemprograman Java Script, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat – sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran - data. JSON terbuat dari dua struktur[8].

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).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini

2.5.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. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomastis[9].

Adapun kelebihan MySQL diantaranya :

1) Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2) Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3) Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4) 'Performance tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5) Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6) Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).

7) Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8) Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9) Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10) Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11) Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
12) Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13) Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.6.Unified Modelling Language (UML)

Pemodelan (modeling) adalah proses merancang piranti lunak sebelum melakukan pengkodean (coding). Model piranti lunak dapat dianalogikan seperti pembuatan blueprint pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin komplek sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik.

Dengan menggunakan model, diharapkan pengembangan perangkat lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk faktor-faktor seperti scalability, robustness, security, dan sebagainya.

Kesuksesan suatu pemodelan perangkat lunak ditentukan oleh tiga unsur,yang kemudian terkenal dengan sebuan segitiga sukses (the triangle for success).Ketiga unsur tersebut adalah metode pemodelan (notation), proses (process) dantool yang digunakan.

Memahami notasi pemodelan tanpa mengetahui cara pemakaian yangsebenarnya (proses) akan membuat proyek gagal. Dan pemahaman terhadapmetode pemodelan dan proses disempurnakan dengan penggunaan tool yang tepat[10].

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untukmenggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki maknatertentu, dan UML syntax mendefinisikan bagaimana bentuk bentuk tersebut dapatdikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah adasebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT(Object Modeling Technique), dan Ivar Jacobson OOSE (Object-OrientedSoftware Engineering)[10].

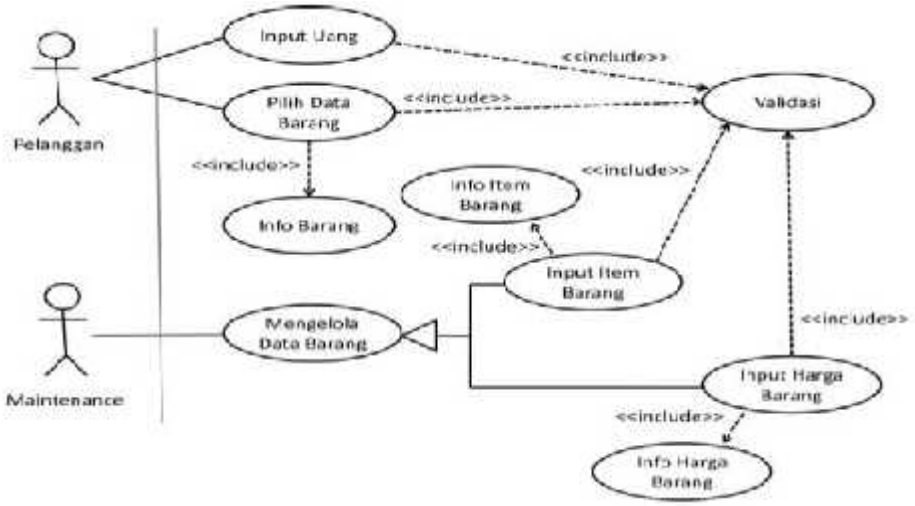
2.6.1.Use Case Diagram

Diagram use case menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Diagram *use case* dapat sangat membantu bila kita sedang menyusun kebutuhan sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem.

Sebuah use case dapat melakukan include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang melakukan include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common.

Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain[10].



Gambar 2.1 Contoh Use Case Diagram

2.6.2.*Activity* Diagram

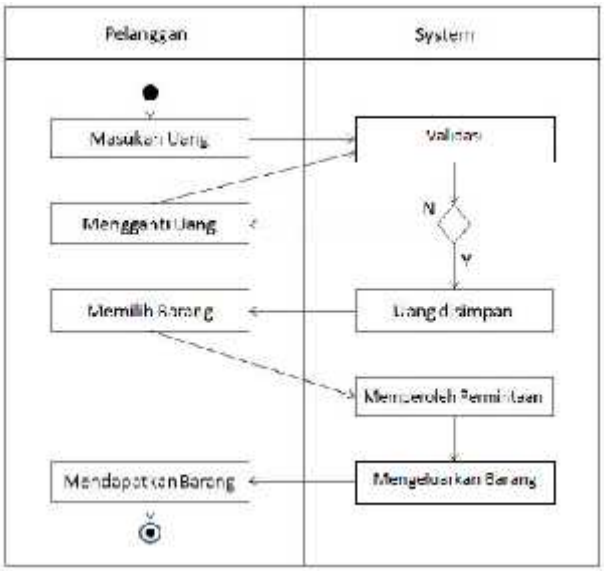
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state

sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar sub sistem) secara eksak, tetapi lebih menggambarkan proses - proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan prosesproses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity* diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu[10].



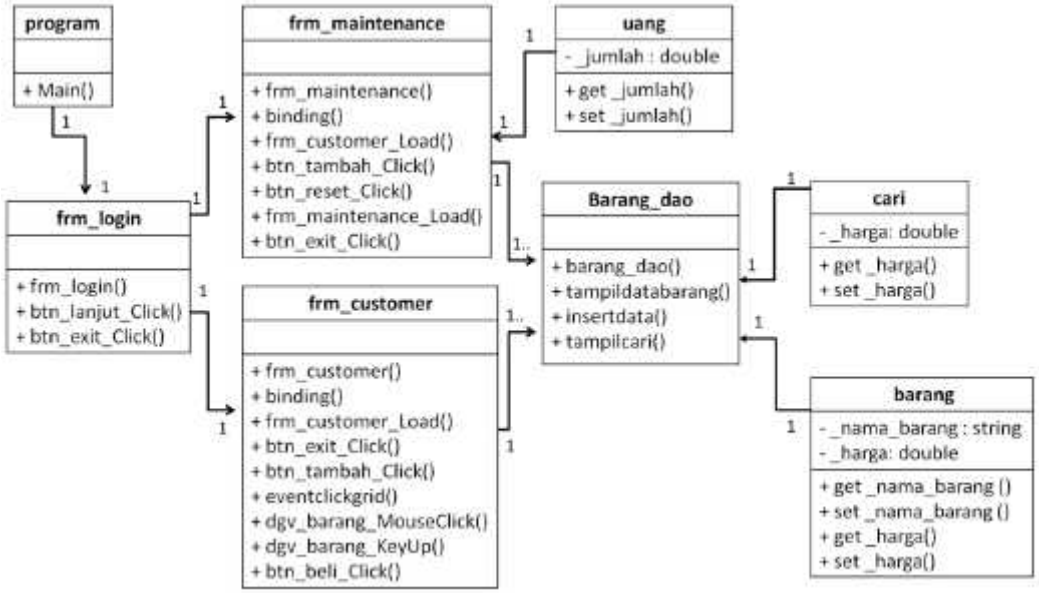
Gambar 2.2 Contoh Activity Diagram

2.6.3.*Class* Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain[10].

Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time.



Gambar 2.3 Contoh Class Diagram

2.6.4.*Sequence* Diagram

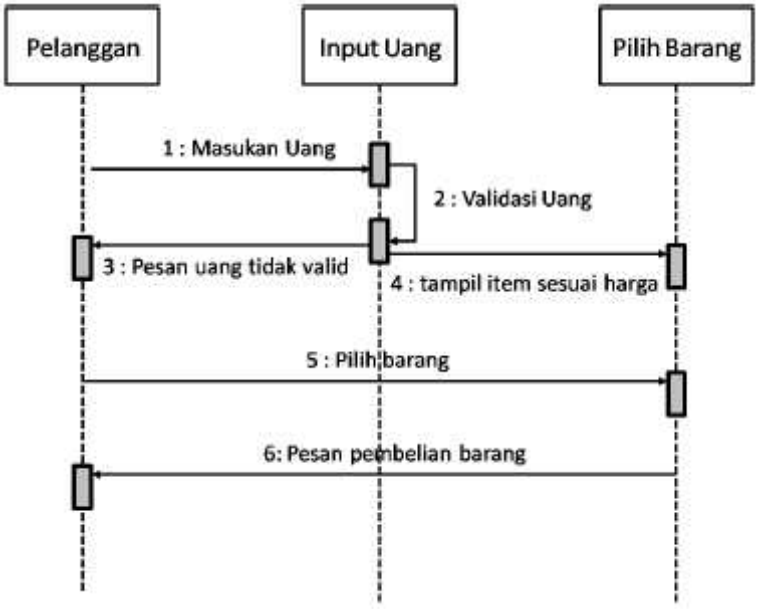
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang

digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*[10].

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity[10].



Gambar 2.4 Contoh Sequence Diagram

2.7.GPS (*Global Positioning System*)

GPS adalah sistem navigasi yang menggunakan satelit yang didesain agar dapat menyediakan posisi secara instan, kecepatan dan informasi waktu di hampir semua tempat di muka bumi, setiap saat dan dalam kondisi cuaca apapun.

Sedangkan alat untuk menerima sinyal satelit yang dapat digunakan oleh pengguna secara umum dinamakan GPS Tracker atau GPS Tracking, dengan menggunakan alat ini maka dimungkinkan user dapat melacak posisi kendaraan, armada ataupun motor dalam keadaan Real-Time[11].

2.7.1.Cara Kerja GPS

Bagian yang paling penting dalam sistem navigasi GPS adalah beberapa satelit yang berada di orbit bumi atau yang sering kita sebut di ruang angkasa. Satelit GPS saat ini berjumlah 24 unit yang semuanya dapat memancarkan sinyal ke bumi yang lalu dapat ditangkap oleh alat penerima sinyal tersebut atau GPS Tracker.

Selain satelit terdapat 2 sistem lain yang saling berhubungan, sehingga jadilah 3 bagian penting dalam sistem GPS. Ketiga bagian tersebut terdiri dari: GPS *Control Segment* (Bagian Kontrol), GPS Space Segment (bagian angkasa), dan *GPS User Segment* (bagian pengguna).

2.7.1.1 GPS Control Segment

Control segment GPS terdiri dari lima stasiun yang berada di pangkalan Falcon Air Force, Colorado Springs, Ascension Island, Hawaii, Diego Garcia dan Kwajalein. Kelima stasiun ini adalah mata dan telinga bagi GPS. Sinyal-sinyal dari satelit diterima oleh bagian kontrol, kemudian dikoreksi, dan dikirimkan kembali ke satelit. Data koreksi lokasi yang tepat dari satelit ini disebut data ephemeris, yang kemudian nantinya dikirimkan ke alat navigasi yang kita miliki.

2.7.1.2 GPS Space Segment

Space Segment adalah terdiri dari sebuah jaringan satelit yang tediri dari beberapa satelit yang berada pada orbit lingkaran yang terdekat dengan tinggi nominal sekitar 20.183 km di atas permukaan bumi. Sinyal yang dipancarkan oleh seluruh satelit tersebut dapat menembus awan, plastik dan kaca, namun tidak bisa menembus benda padat seperti tembok dan rapatnya pepohonan.

Terdapat 2 jenis gelombang yang hingga saat ini digunakan sebagai alat navigasi berbasis satelit. Masing-masingnya adalah gelombang L1 dan L2, dimana L1 berjalan pada frequensi 1575.42 MHz yang bisa digunakan oleh masyarakat umum, dan L2 berjalan pada frequensi 1227.6 Mhz dimana jenis ini hanya untuk kebutuhan militer saja.

2.7.1.3 GPS User Segment

User segment terdiri dari antenna dan prosesor *receiver* yang menyediakan positioning, kecepatan dan ketepatan waktu ke pengguna. Bagian ini menerima data dari satelit-satelit melalui sinyal radio yang dikirimkan setelah mengalami koreksi oleh stasiun pengendali (*GPS Control Segment*).

2.8.Kuesioner (Skala *Likert*)

Skala *likert* adalah metode pengukuran sikap yang banyak digunakan karena kesederhanaannya. Langkang-langkah menggunakan skala *likert* adalah [12]:

1. Tentukan sikap terhadap permasalahan apa yang akan diukur
2. Tentukan dimensi yang ada dalam menyusun sikap tersebut. Menurut likert dimendi sikap adalah :
    a. *Cognitive domain* (tahu/tidak tahu)
    b. *Affective domain* (perasaan terhadap sesuatu)
    c. *Conative domain* (tendensi untuk bertingkah laku)
3. Susun pertanyaan atau item yang merupakan alat pengukur dimensi yang menyusun sikap dan akan diujur sesuai indikator. Item yang disusun harus terdiri dari item positif dan negatif. Item positif bila pertanyaan memberi isyarat mendukung permasalahan yang sedang diteliti. Item negatif bila

pertanyaan bertentangan dengan permasalahan yang sedang diteliti. Apabila item positif angka terbesar diberikan untuk aternatif jawaban sangat setuju, sedangkan bila item negatif angka terbesar diberikan untuk alternatif jawaban sangat tidak setuju. Skor yang diberikan pada jawaban dijumlahkan.

Tabel 2.1 Panduan Pemberian Skor

Jenis Pertanyaan	**Alternatif Jawaban**				
	SS	**S**	**TP**	**TS**	**STS**
Positif	5	4	3	2	1
Negatif	1	2	3	4	5

Setelah dijumlahkan, dicari nilai maksimum, minimum, median, kuartil I, dan kuartil III baik untuk setiap responden ataupun seluruh responden. Berdarakan hal tersebut diberikan ketentuan, yaitu jika skor [6]:

1. ≥ kuartil III, dianggap sikap yang sangat positif.
2. ≥ Median, dianggap sikap yang positif.
3. < Median s/d Kuartil I, dianggap sikap yang negatif.
4. < Kuartil III, dianggap sikap yang sangat negatif.

2.9.Firebase *Cloud Mesagging*

Firebase *Cloud Messaging*(FCM), adalah layanan yang diberikan oleh Firebase untuk menggantikan Google Cloud Messaging (GCM). Pihak Google menyarankan untuk aplikasi yang masih menggunakan GCM untuk segera migrasi ke FCM. Firebase *Notification* adalah service gratis yang disediakan oleh Google untuk mengirimkan *notification* ke *mobile application*[13].

Fitur-fitur yang diberikan oleh GCM sebenarnya tidak terlalu jauh berbeda dengan GCM. Dengan FCM kita bisa memberikan push notification dan membuat komunikasi dua arah antara device. Teknologi yang digunakan terbagi menjadi dua:

1. XMPP (Extensible Messaging and Presence Protocol)
2. HTTP (Hypertext Transfer Protocol).

2.10.Fuzzy Multi-Attribute Decision Making (FMADM)

Fuzzy Multiple Attribute Decision Making (FMADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengankriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiapatribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksialternatif yang sudah diberikan. Pada dasarnya, ada 3 pendekatan untuk mencarinilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif danpendekatan integrasi antara subyektif dan obyektif. Masing-masing pendekatanmemiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobotditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehinggabeberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebasSedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematissehingga mengabaikan subyektifitas dari pengambil keputusan.

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah FMADM salah satunya adalah Simple Additive Weighting Method (SAW). Metode SAW sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada[14].

2.10.1.Simple Additive Weighting (SAW)

Metode SAW sering juga dikenal istilah metode penjumlahan terbobot.Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari ratingkinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkanproses normalisasi matriks keputusan (X) ke suatu skala yang dapatdiperbandingkan dengan semua rating alternatif yang ada. Secara garis besar,SAW digunakan untuk melakukan pembobotan pada data tempat kuliner

dan wisata sehingga mempunyasi hasil akhir rekomendasi tempat wisata dan kulinerdi kota Bandung.

2.10.2.Langkah Penyelesaian Simple Additive Weighting (SAW)

Langkah Penyelesaian SAW sebagai berikut :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalampengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria(Ci), kemudianmelakukan normalisasi matriks berdasarkan persamaan yang disesuaikandengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehinggadiperoleh matriks ternormalisasi R.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dariperkalian matriks ternormalisasi R dengan vektor bobot sehinggadiperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai)sebagaisolusi.

Formula untuk melakukan normalisasi tersebut adalah :

Rij = $\frac{Xij}{Max\ Xij}$ Jika J adalah atribut keuntungan (benefit)

Rij = $\frac{Min\ Xij}{Xij}$ Jika J adalah atribut biaya (cost)

Dimana :

rij = rating kinerja ternormalisasi

Max Xij = nilai maksimum dari setiap baris dan kolom

Min Xij = nilai minimum dari setiap baris dan kolom

Xij = baris dan kolom dari matriks

Dengan rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj;

I=1,2,…m dan j = 1,2,…,n.

Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai :

Dimana :

Vi = Nilai akhir dari alternatif

wj = Bobot yang telah ditentukan

rij = Normalisasi matriks

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.

2.11.Metode TOPSIS

TOPSIS (*Technique For Order Preference by Similarity to Ideal Solution*) adalah salah satu metode pengambilan keputusan multikriteria yangpertama kali diperkenalkan oleh Yoon dan Hwang tahun 1981. TOPSIS didasarkanpada konsep dimana alternatif yang terpilih atau terbaik tidak hanya mempunyaijarak terdekat (terpendek) dari solusi ideal positif, namun juga memiliki jarakterjauh (terpanjang) dari solusi ideal negatif dari sudut pandang geometris denganmenggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatualternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlahdari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusinegatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut[15].

TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif danjarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadapsolusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunanprioritas alternatif bisa dicapai. Metode ini banyak digunakan pada beberapa modelMADM untuk menyelesaikan masalah pengambilan keputusan secara praktis. Halini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien,dan memiliki kemampuan mengukur kinerja relatif dari alternatif-alternatifkeputusan dalam bentuk matematis yang sederhana[15].

2.11.1 Langkah-langkah Metode TOPSIS

Prosedur perhitungan metode TOPSIS dalam pendukung keputusan adalah sebagaiberikut :

1. Membangun sebuah matriks keputusanMatriks keputusan X mengacu terhadap m alternatif yang akan dievaluasiberdasarkan n kriteria. Matriks keputusan X dapat dilihat sebagai berikut:

$$X = \begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_m \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & . & . & x_n \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & x_{13} & . & . & x_{1n} \\ x_{21} & x_{22} & x_{23} & . & . & x_{2n} \\ x_{31} & x_{32} & x_{33} & . & . & x_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & . & . & x_{mn} \end{bmatrix} \end{array}$$

Keterangan :

a_i = alternatif-alternatif yang mungkin

x_j = atribut dimana performansi alternatif diukur

x_{ij} = performansi alternatif ai dengan acuan atribut xj

2. Membangun matriks keputusan yang ternormalisasi. Persamaan yangdigunakan untuk mentransformasikan setiap elemen xij adalah

$$r_i = \frac{x_i}{\sqrt{\sum_{i=1}^{m} x_i^2}}$$

Dengan i = 1, 2, 3, ..., m; dan j = 1, 2, 3, ..., n;

Keterangan :

rij = elemen dari matriks keputusan yang ternormalisasi R

xij = elemen dari matriks keputusan X

3. Membuat matriks keputusan yang ternormalisasi terbobot. Dengan bobot wj =(w_1, w_2, w_3,w_4 …. w_n), dimana $w_j$adalah bobot dari kriteria ke-j dan$\sum_{j=1}^{n} wj$=1 maka normalisasi bobot matriks Y adalah :

$$Y_{ij} = w_j r_{ij}$$

Dengan i= 1, 2, 3,..., m; dan j = 1, 2, 3,..., n;

Keterangan:

yij = elemen dari matriks keputusan yang ternormalisasi terbobot Y

wj = bobot kriteria ke-j

rij = elemen dari matriks keputusan yang ternormalisasi R

4. Menentukan matriks solusi ideal dan matriks solusi ideal negatif. Solusi idealpositif dinotasikan A^+, sedangkan solusi ideal negatif dinotasikanBerikut ini adalah persamaan dari A^+dan A^-.

a. $A^+ = \{(\max y_i \mid j \in J), (\min y_i \mid j \in J'), i = 1,2,3 \dots m\}$
$= (y_1^+, y_2^+, y_3^+, \dots, y_n^+)$

b. $A^- = \{(\min y_i \mid j \in J), (\max y_i \mid j \in J'), i = 1,2,3 \dots m\}$
$= (y_1^-, y_2^-, \dots, y_n^-)$

Keterangan:

yij = matriks ternormalisasi terbobot [i][j]

A^+ = elemen matriks solusi ideal positif.

A^-= elemen matriks solusi ideal negatif.

5. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matrik ideal negatif.

   a. Jarak antara alternatif $A_i$dengan solusi ideal positif dirumuskan sebagai:

$$D_i^+ = \sqrt{\sum_{j=1}^{m}(y_i^+ - y_i)^2}$$

Keterangan :

D_i^+ = jarak alternatif Ai dengan solusi ideal positif

y^+ = solusi ideal positif

y_i = matriks normalisasi terbobot

b. Jarak antara alternatif A_i dengan solusi ideal negatif dirumuskan sebagai:

$$D_i^- = \sqrt{\sum_{j=1}^{m}(y_i - y_i^-)^2}$$

Keterangan :

D_i^- = jarak alternatif Ai dengan solusi ideal negatif

y^- = solusi ideal negatif

y_i = matriks normalisasi terbobot

6. Menghitung kedekatan terhadap solusi ideal positif. Kedekatan setiap alternatif terhadap solusi ideal dihitung berdasarkan rumus:

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

Keterangan :

V_i= kedekatan tiap alternatif terhadap solusi ideal

D_i^+= jarak alternatif Ai dengan solusi ideal positif

D_i^-= jarak alternatif Ai dengan solusi ideal negatif

7. Alternatif yang memiliki nilai kedekatan terhadap solusi ideal positif (V_i) tertinggi merupakan alternatif terbaik berdasarkan perhitungan TOPSIS.

2.12.Google Maps

Google Maps adalah layanan aplikasi peta online yang disediakan oleh Google secara gratis. Layanan peta Google Maps ini dapat diakses melalui situs resminya yaitu http://maps.Google.com. Layanan yang dibuat secara interaktif oleh Google Maps menawarkan peta dan gambar satelit untuk seluruh dunia dimana di dalamnya peta dapat digeser sesuai keinginan penggunam, mengubah level zoom, serta mengubah tampilan jenis peta. Fasilitas lain yang disediakan oleh Google Maps antara lain adalah pencarian lokasi dengan memasukkan kata kunci seperti nama, tempat, kota atau jalan, dan fasilitas lain seperti perhitungan rute perjalanan dari satu tempat ke tempat lain.

Google Maps dibuat dengan menggunakan kombinasi dari gambar peta, database, serta obyek-obyek interaktif yang dibuat dengan bahasa pemrograman HTML, Javascript, dan AJAX, serta beberapa bahasa pemrograman lainnya. Gambaran yang muncul pada peta merupakan hasil komunikasi dengan database pada web server Google untuk menampilkan gabungan dari potongan-potongan gambar yang diminta. Dimana, keseluruhan citra yang ada di integrasikan ke dalam database pada Google server yang dapat dipanggil sesuai kebutuhan[16].

2.13.Google Maps Distance Matrix API

Google Maps *Distance Matrix* API adalah layanan yang menyediakan waktu jarak perjalanan untuk matriks tempat asal dan tujuan. Google Maps *Distance Matrix* API mengambil informasi berdasarkan rute yang disarankan antara titik awal dan titik akhir, sesuai perhitungan Google Maps API, dan terdiri dari baris-baris berisi nilai *duration* dan *distance* untuk setiap pasang. Layanan Google Maps *Distance Matrix* API tidak mengembalikan informasi rute secara detail. Informasi rute akan dapat diperoleh secara detail dengan meneruskan satu tempat asal dan tujuan yang diinginkan ke Google Maps *Directions* API.

Permintaan Google Maps *Distance Matrix* API mengambil bentuk seperti berikut [17] :

https://maps.googleapis.com/maps/api/distancematrix/***outputFormat***?***parameters***

dalam hal ini, format output dapat berupa dari salah satu nilai JSON atau XML. Parameter yang diperlukan adalah sebagai berikut [17]:

a. *Origins*, merupakan titik awal untuk menghitung jarak dan waktu perjalanan.
b. *Destination*, merupakan satu atau beberapa lokasi untuk digunakan sebagai titik akhir untuk menghitung jarak dan waktu perjalanan.
c. *Key*, kunci API aplikasi yang mengidentifikasikan bahwa aplikasi yang dibuat untuk keperluan manajemen kuota.