BAB 2
LANDASAN TEORI

2.1. Sistem Informasi

Sistem informasi menurut Kertahadi (2007) adalah alat untuk menyajikan informasi sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk memberikan informasi dalam perencanaan, memulai, pengorganisasian, operasional sebuah perusahaan yang melayani sinergi organisasi dalam proses mengendalikan pengambilan keputusan. [1]

2.2. Objek Wisata

Pengertian obyek dan daya tarik wisata menurut Marpaung (2002:78) adalah suatu bentukan dari aktifitas dan fasilitas yang berhubungan, yang dapat menarik minat wisatawan atau pengunjung untuk datang ke suatu daerah atau tempat tertentu. Obyek dan daya tarik wisata sangat erat hubungannya dengan travel motivation dan travel fashion, karena wisatawan ingin mengunjungi serta mendapatkan suatu pengalaman tertentu dalam kunjungannya.

Menurut UU RI No 9 Tahun 1990 tentang Kepariwisataan, dinyatakan bahwa obyek dan daya tarik wisata adalah segala sesuatu yang menjadi sasaran wisata baik itu pembangunan obyek dan daya tarik wisata, yang dilakukan dengan cara mengusahakan, mengelola dan membuat obyek-obyek baru sebagai obyek dan daya tarik wisata.

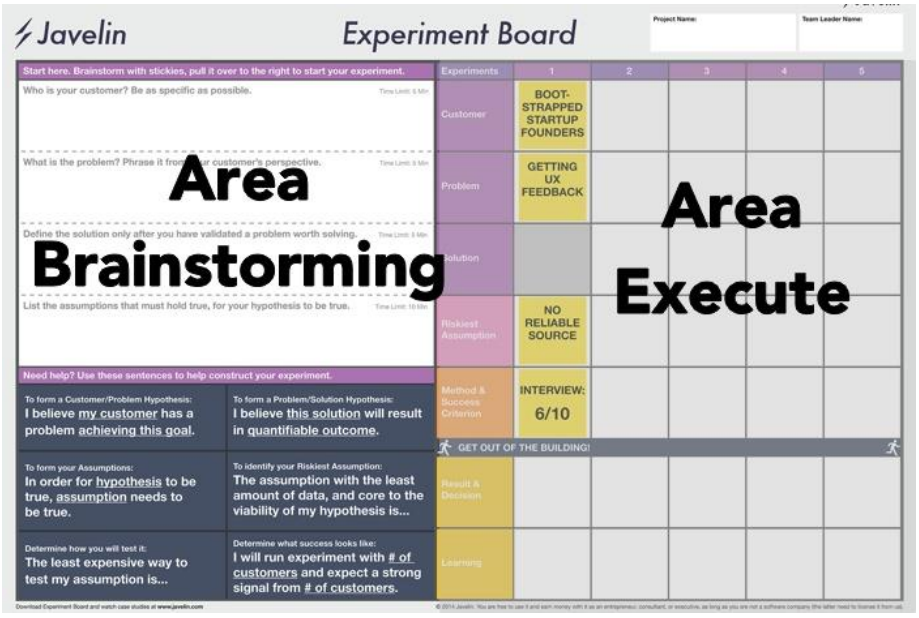
Pengertian objek wisata secara umum menurut Peraturan Pemerintah Republik Indonesia No. 24/1979, tentang penyerahan sebagian urusan pemerintah dalam bidang kepariwisataan pada Daerah Tingkat I adalah sebagai berikut: [2]

1. Objek Wisata adalah perwujudan dari pada ciptaan manusia, tata hidup, seni budaya, serta sejarah bangsa dan temoat atu keadaan alam yang mempunyai daya tarik wisata bagi wisatawan untuk dikunjungi.

2. Atraksi Wisata adalah semua yang diciptakan manusia berupa penyajian kebudayaan seperti tari-tarian, kesenian rakyat, upacara adat, dan lain-lain.

2.3 Javelin Board

Javelin Board atau Papan Validasi merupakan salah satu cara untuk validasi ide *startup* menguraikan menggunakan *Experiment Board*. Dimana validasi ide merupakan salah satu tahap krusial yang menentukan. Dari analisis menggunakan *Javelin Board* kita dapat mengetahui titik kritis, identifikasi masalah, asumsi, melakukan metode pengujian dan juga penentuan kriteria sukses. *Board* ini terbagi menjadi 2 bagian, yaitu bagian kiri atau disebut dengan area *Brainstorming*. Dan bagian kanan yang disebut dengan *Area Execute*.



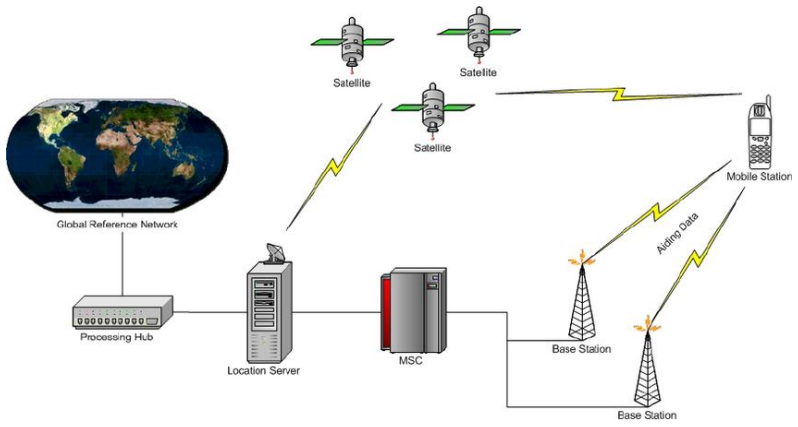
Gambar 2. 1 Contoh *Javelin Board*

Langkah pertama yang harus di ketahui adalah, menentukan calon konsumen untuk produk yang dibuat. Kumpulkan beberapa calon potensial, kemudian pilih salah satu yang akan menjadi fokusnya. Untuk tahap selanjutnya di javelin Board sudah terdapat instruksi secara singkat supaya mempermudah kita dalam pengisian. [3]

2.4 Assisted Global Positioning System (A-GPS)

Assisted-Global Positioning System (A-GPS) merupakan pengembangan dari sistem GPS biasa sebagai penentu posisi. (A-GPS) akan membantu perangkat menemukan satelit. Saat menggunakan (A-GPS), perangkat akan menerima informasi satelit dari server data melalui jaringan seluler. Apabila perangkat tidak menerima data tersebut, perangkat akan mencoba mendeteksi satelit lain. Dengan bantuan tersebut, perangkat akan mendeteksi satelit yang tepat, di saat satelit berada di sisi yang sama pada planet dengan perangkat.

Dengan fitur (A-GPS), operator yang digunakan membantu menambahkan informasi yang kurang, sehingga waktu untuk menentukan lokasi mejadi lebih cepat dan tepat daripada GPS standalone. Sebagai contoh, positioning dilakukan lebih cepat dalam keadaan cold-start, yaitu pada saat koneksi GPS telah dimatikan dalam waktu yang lama, atau apabila pengguna melakukan perjalanan ke Negara yang berbeda. Selain itu A-GPS mengurangi waktu yang di perlukan perangkat untuk menemukan posisi saat itu, disebut sebagai '*Time To First Fix*' (TTFF), untuk sebagian geografis di seluruh dunia. [4]



Gambar 2. 2 Contoh *Assisted Global Positioning System*

2.5 Global Positioning System (GPS)

GPS atau *Global Positioning System*, merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya berada secara global

di permukaan bumi yang berbasiskan satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. Dimanapun posisi saat ini, maka GPS bisa membantu menunjukan arah, selama masih terlihat langit. Layanan GPS ini tersedia gratis, bahkan tidak perlu mengeluarkan biaya apapun kecuali membeli GPS *reciever*-nya.

Sistem GPS, yang nama aslinya adalah NAVSTAR GPS *(Navigation Satelite Timing and Ranging Global Positioning System),* mempunyai tiga segment yaitu: satelit, pengontrol, dan penerima / pengguna. Satelit GPS yang mengorbitkan bumi, dengan orbit dan kedudukan yang tetap (kordinat pasti), seluruhnya berjumlah 24 buah dimana 21 buah aktif bekerja daan 3 buah sisanya adalah cadangan. [6]

1. Satelit bertugas untuk menerima dan menyimpan data yang di transmisikan oleh stasiun-stasiun pengontro, menyimpan dan menjaga informasi waktu berketelitian tinggi ditentukan dengan jam atomic di satelit dan memancarkan sinyal untuk informasi secara kontinyu ke pesawat penerima receiver dari pengguna.
2. Pengontrol bertugas untuk mengendalikan dan mengontrol satelit dari bumi baik untuk mengecek kesehatan satelit, penentuan dan prediksi orbit dan waktu, sinkronisasi waktu antar satelit, dan mengirim data ke satelit.
3. Penerima bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi tiga dimensi yaitu kordinat di bumi beserta ketinggianya, arah, jarak dan waktu yang di perlukan penggunanya. Ada dua macam tipe penerima, yaitu tipe NAVIGASIdan tipe GEODETIC, yang termasuk tipe NAVIGASI antara lain: *Trimble Ensign, Trimble Pathfinder, Garmin, Sony* dan sebagainya. Sedangkan tipe GEODETIC antara lain: Topcon, *Leica, Astech, Trimble* seri 4000 dan sebagainya.

2.6 Android

Android merupakan sistem operasi yang dikembangkan untuk perangkat mobile yang berbasis linux seperti telepon pintar dan komputer tablet. Android

menyediakan sistem operasi yang terbuka bagi para pengembang untuk menciptakan aplikasi Android diluncurkan untuk umum pada musim gugur di tahun 2008. Android sangat berkembang pesat di industri karena dua aspek utama yaitu bersifat opensource dan model arsitekturya. Sebagai sebuah proyek yang bersifat opensource, memungkinkan Android untuk sepenuhnya dipahami dan dianalisis mengenai fitur, penyelesaian pada *bug program* hingga *hardware.*

Komponen aplikasi android adalah blok – blok penting untuk mebangun aplikasi Android. Setiap komponen merupakan bagian berbeda yang digunakan sistem dalam aplikasi. Tidak semua komponen merupakan titik masuk sebenarnya bagi pengguna dan sebagian komponen saling bergantung, namun masing-masing komponen tersedia sebagai kesatuan sendiri dan memainkan perannya masing-masing merupakan blok pembangun unik yang mendefinisikan penggunaan aplikasi secara keseluruhan. [7]

Ada beberapa macam tipe komponen aplikasi. Setiap tipe memiliki kegunaan tersendiri dan *lifecycle* sendiri yang mendefinisikan bagaimana komponen dibuat *(create)* dan dimusnahkan *(destroy).*

1. *Activities*

   mewakili layar utama dengan antarmuka pengguna, *activities* melakukan tindakan pada layar. Sebagai contoh, aplikasi email mungkin salah satu kegiatan yang menunjukkan daftar email baru, kegiatan lain untuk menulis email dan kegiatan lain untuk membaca email. Jika sebuah aplikasi memiliki lebih dari satu aktivity, salah satu dari mereka harus ditandai sebagai aktivity yang disajikan ketika aplikasi diluncurkan.

2. *Services*

   Service merupakan komponen yang berjalan di latar belakang untuk melakukan operasi berjalan lama. Sebagai contoh, sebuah *services* mungkin memainkan musik di latar belakang sementara pengguna dalam aplikasi yang berbeda, atau mungkin mengambil data melalui jaringan tanpa menghalangi interaksi pengguna dengan *aktivity*.

3. *Content Providers*

Content providers digunakan untuk mengelola dan berbagi database. Data dapat disimpan dalam *file system*, dalam database SQLite, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *content provider* memungkinkan antar aplikasi untuk saling berbagi data.

4. *Broadcast receivers*

   Broadcast Receiver adalah sebuah komponen Android yang berfungsi untuk melakukan broadcast pesan dari aplikasi lain atau dari sistem itu sendiri. Pesan yang di *broadcast* disebut *event* atau *intent*. Aplikasi melakukan broadcast untuk memberi tahu aplikasi lainnya bahwa beberapa data telah di unduh ke perangkat yang tersedia, lalu penerima *broadcast* akan menerima pesan untuk melakukan transfer informasi.

5. *Intent*

   Intent merupakan suatu pesan yang digunakan untuk mengaktifkan tiga komponen dasar pada aplikasi Android yaitu ***Activity***, ***Service***, dan ***Broadcast Receiver***. Aktifasi pada komponen-komponen tersebut bisa terjadi pada aplikasi yang sama atau berbeda, seperti *menjalankan Activity*, *inisiasi Service*, atau *pengiriman pesan kepada Broadcast Receiver*.

6. *Layout*

   Layout adalah yakni untuk mengatur tata letak suatu komponen pada pemrograman android, diperlukan sebuah file XML. File XML tersebut digunakan untuk mengatur antarmuka aplikasi android seperti komponen teks, tombol, dan sebagainya. Untuk meletakkan komponen-komponen tersebut diperlukan sebuah cara agar komponen yang dibuat sesuai dengan keinginan dan rapi.

2.6.1 Sejarah Android

Android berperan besar dalam perkembangan teknologi selama lebih dari enam tahun terakhir ini. Selama ini, banyak sekali orang telah dimanjakan oleh berbagai kemudahan yang diberikan oleh OS Android. Setelah Android muncul dan diakuisisi oleh Google, perusahaan ini kian fokus untuk berperang dalam kategori

smartphone dan terus memberikan *update* terhadap sistem operasi yang diusungnya tersebut [7].

Sebelum Android tersedia secara komersial dan diimplementasikan ke dalam sebuah perangkat, sebenarnya OS Android cukup lama mengalami perkembangan. Android merilis beberapa versi *Software Developer Kit* (SDK) atau hanya ditujukan untuk developer saja. Android versi SDK yang pertama kali muncul adalah Android 0.5 Milestone 3. OS Android versi ini muncul pada tahun 2007 dan tampil hampir mirip dengan BlackBerry. Pada bulan Agustus 2008, akhirnya Google mulai serius melakukan transformasi. Google merilis OS Android 0.9 beta yang hadir dengan desain lebih *modern*, *home screen* yang tampil dengan beberapa *widget* dan secara keseluruhan berhasil menghilangkan *image* BlackBery. Berikut adalah urutan perkembangan pada sistem Android [8]:

1. Android *Beta*

   Pertama kali dirilis pada 5 November 2007, kemudian pada 12 November 2007 Software Development Kit (SDK) dirilis oleh Google.

2. Android 1.0 *(Astro)*

   Pertama kali dirilis pada 23 Spetember 2008. Sebenarnya Android versi pertama ini akan dinamai dengan nama *Astro* tapi karena alasan hak cipta dan trademark nama *Astro* tidak jadi disematkan pada versi pertama dari OS Android ini. HTC Dream adalah ponsel pertama yang menggunakan OS ini.

3. Android 1.1 *Bender*

   Pertama kali dirilis pada 9 Februari 2009. Awalnya versi OS Android ini dirilis untuk perangkat T-Mobile G1 saja. Versi ini merupakan update untuk memperbaiki beberapa bugs, mengganti API dan menambahkan beberapa fitur.

4. Android 1.5 *Cupcake*

   Pertama kali dirilis pada 30 April 2009. Mulai versi Android ini penamaan menggunakan nama makan pencuci mulut *(dessert)* mulai digunakan. OS ini berbasiskan pada kernel Linux 2.6.27 dan menambahkan beberapa update serta UI baru dari versi Android sebelumnya. Mulai terdapat “widget” yang dapat dibesar

kecilkan. Kemudian ditambah kemampuan untuk meng-upload video dan gambar ke Youtube dan Picasa.

5. Android 1.6 *Donut*

Dirilis pertama kali pada 15 September 2009. Terdapat peningkatan pada fitur pencarian dan UI yang lebih user friendly. Pada versi ini juga sudah mendukung teknologi CDMA/EVDO, 802.1x, VPNs. Kemudian support layar dengan resolusi WVGA. Berikut penampakan Android v1.6 Donut.

6. Android 2.0/2.1 *Éclair*

Dirilis pertama kali pada 9 Desember 2009. Terjadi penambahan fitur untuk 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. Beberapa versi updatenya antara Android v.2.0 kemudian v2.0.2 dan terakhir v.2.1.

7. Android 2.2 *Froyo (Froze Yoghurt)*

Dirilis pertamakali pada 20 Mei 2010 pada smartphone *Google Nexus One*. Pada versi ini sudah support terhadap *Adobe Flash Player* 10.1. Peningkatan pada kecepatan membuka dan menutup aplikasi, serta penggunaan SD Card sebagai tempat penyimpanan aplikasi. Ketika Android *Froyo* hadir mulai muncul banyak diskusi yang membahas mengenai persaingan antara Android dengan iOS yang akan semakin ketat di masa yang akan datang. Beberapa versi update yang dirilis antara lain Android v.2.2.1 hingga v.2.2.3.

8. Android 2.3 *Gingerbread*

Pertama kali diperkenalkan pada 6 Desember 2010. Terjadi banyak peningkatan pada versi Android yang satu ini dibandingkan dengan versi sebelumnya. Dirancang untuk memaksimalakan kemampuan aplikasi dan game. Serta mulai digunakannya *Near Field Communication* (NFC). Perbaikan terhadap dukungan layar resolusi WXGA dan diatasnya. Beberapa versi update yang dirilis antara lain v.2.3.3 hingga v.2.3.7. Sampai saat ini Android *Gingerbread* merupakan versi Android yang memiliki pengguna terbanyak dibandingkan dengan seri Android lainnya, yaitu mencapai 65% dari seluruh versi Android yang dirilis.

9. Android 3.0/3.1 *Honeycomb*

Pertama kali diperkenalkan pada 22 Februari 2011 dan Motorola Xoom adalah yang pertama kali menggunakannya. Android versi ini merupakan OS yang didesain khusus untuk pengoptimalan pengunaan pada tablet PC.

10. Android 4.0 ICS *(Ice Cream Sandwich)*

Sampai tulisan ini ditulis ICS merupakan versi Android yang paling anyar. Pertama kali dirilis pada 19 Oktober 2011. *Smartphone* yang pertama kali mengunakan OS Android ini adalah *Samsung Galaxy Nexus*. Secara teori semua perangkat seluler yang menggunakan versi Android sebelumnya *Gingerbread* dapat di-update ke Android *Ice Cream Sandwich*.

11. Android versi 4.1/4.3 *(Jelly Bean)*

Android Jelly Bean yaang 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. Tak ketinggalan Google Now juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga.

12. Android 4.4 *Kitkat*

KitKat pertama kali muncul pada Oktober 2013 bersama perangkat LG Nexus 5. OS Android KitKat mengalami pembaharuan UI seperti navigasi dan status bar transparan dan dapat bekerja dengan optimal pada perangkat yang memiliki spesifikasi rendah.

13. Android 5.0 *Lolipop*

Android Lollipop adalah versi stabil terbaru dengan versi antara 5.0 dan 5.1. Diresmikan pada 25 Juni 2014 saat Google I / O, dan tersedia secara resmi melalui *over-the-air* (OTA) update pada tanggal 12 November 2014, untuk memilih perangkat yang menjalankan distribusi Android dilayani oleh Google (seperti perangkat *Nexus* dan *Google Play edition*).

14. Android 6.0 *Marshmallow*

Android 6.0 atau disebut dengan Marshmallow adalah versi paling terbaru dari sistem operasi mobile Android. Pertama kali diperkenalkan Mei 2015 di bawah

codename Android M, secara resmi dirilis pada bulan Oktober 2015. Fokus utama dari Android Marshmallow adalah pada peningkatan pengalaman pengguna secara keseluruhan untuk OS sebelumnya *Lollipop*, memperkenalkan arsitektur permissions baru, API baru contextual assistants (sebuah fitur yang dikenal dengan "*Google Now On Tap*" -sebuah kemampuan baru dari aplikasi Google Penelusuran), sebuah sistem manajemen daya baru yang mengurangi aktivitas di *Background* saat perangkat tidak sedang digunakan, dukungan asli untuk pengenalan sidik jari dan konektor USB tipe-C, kemampuan untuk migrasi data ke kartu *microSD* dan menggunakannya sebagai penyimpanan utama, serta perubahan internal lainnya. [9]

2.7 Google Design Sprint

Google design sprint adalah suatu metodologi yang inovatif dan dikembangkan oleh para ahli UX Google Ventures. Hal ini memungkinkan anda untuk mempercepat dan menyederhanakan proses desain produk digital. Bersama-sama dalam lima atau enam hari tim akan mengikuti jalan startups terbaik. Dalam hal ini maka akan didapat ide-ide asing dan mendapatkan berbagai solusi dan desain antarmuka yang memungkinkan kita dapat membangun prototype aplikasi. Setelah itu memvalidasi konsep dan menguji pada pengguna ril. Dengan prototype, maka anda dapat meyakinkan investor anda, mengurangi risiko, menghindari pengembangan fitur yang tidak diperlukan dan memaksimalkan laba atas investasi [10]

2.8 Perangkat Lunak Pendukung

Berikut adalah perangkat lunak pendukung dalam menunjang pembangunan aplikasi wisata Kota Jayapura yang akan di bangun.

2.8.1 Android Studio

Android Studio adalah sebuah Integrated Development Environment (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. [11]



Gambar 2. 3 Logo Android Studio

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. Sistem Pembuatan berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat penguji dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatbilitas ver si dan masalah-masalah lain
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine.

2.8.2 PHP

Hypertext Prepocessor (PHP) bahasa pemograman yang digunakan secara luas untuk penanganan, pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Beberapa kelebihan menggunakan PHP adalah sebagai berikut:

1. PHP lebih mudah untuk dikembangkan karena banyak sekalis forum atau milis yang secara khusus membahas tentang bahasa pemrograman ini.
2. PHP lebih mudah untuk difahami, pada sebagian sintak-sintak programnya menggunakan bahasa inggris.
3. PHP adalah basa pemrograman yang bersifat open source dan multi platform. Ini berari PHP bisa dijalankan di berbagai sistem operasi seperti Windows, Unix, Linux, Macintosh dll.
4. Web server yang mendukung PHP mudah ditemukan. [12]

2.8.3 MySQL

MySQL adalah perangkat lunak *database server* atau sebut saja *Database Smart*. Database ini semakin lama semakin populer. Dengan menggunakan database ini, data semakin aman dan berdaya guna. Database ini juga banyak dipakai pada web database sehingga data semakin terintegrasi antara database dekstop dengan database web. Untuk menggunakan database MySQL harus menginstalasinya dahulu ke komputer. Sebagai server database dengan konsep database modern, MySQL memiliki keistimewaan. keistimewaan dimiliki MySQL sebagai berikut:

1. *Portability*

Database MySQL berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, HP-Unix, dan lain-lain.

2. *Open Source*

MySQL merupakan database *open source* (gratis), di bawah lisensi GPL sehingga dapat memperoleh dan menggunakannya secara cuma-cuma tanpa membayar sepersen pun.

3. *Multiuser*

MySQL merupakan database yang dapat digunakan untuk menangani beberapa user dalam waktu bersamaan tanpa mengalami masalah. Dan memungkinkan sebuah database *server* MySQL dapat diakses *client* secara bersamaan pula.

4. *Performace Tuning*

MySQL mempunyai kecepatan yang cukup baik dalam menangani query-query sederhana, serta mampu memproses lebih banyak SQL per satuan waktu.

5. *Column Type*

Database MySQL didukung dengan tipe data yang sangat kompleks, seperti*signed/unsigned integer, float, double, char, varchar, text, blob, data, time, datetime, timestamp, year, set* serta *enum.*

6. *Command dan Functions*

MySQL server memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam *query.*

7. Security

Sistem Security pada MySQL mempunyai beberapa lapisan sekuritas seperti tingkatan *subnetmask, hostname*, dan izin akses user dengan sistem perizinan yang mendetil serta password terenkripsi. [13]

2.8.4 Java Development Kit (JDK)

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

2.8.5 Software Development Kit (SDK)

Sebuah Software Development Kit (SDK atau devkit) tipikal merupakan satu set perkakas pengembangan software yang digunakan untuk mengembangkan atau membuat aplikasi untuk paket software tertentu, software framework, hardware platform, sistem komputer, konsol video game, sistem operasi atau platform sejenis lainnya. Ia mencakup mulai dari pemrograman sederhana seperti sebuah application programming interface (API), sampai dengan pemrograman yang lebih rumit dengan hardware yang canggih atau pada sistem embedded termasuk perangkat mobile. [14]

2.8.6 Java Script Object Nonation (JSON)

JSON singkatan dari *JavaScript Object Notation* adalah suatu format ringkas pertukaran data komputer. Formatnya berbasis teks dan terbaca-manusia serta digunakan untuk merepresentasikan struktur data sederhana dan larik asosiatif (disebut objek).

JSON merupakan bagian dari bahasa pemrograman JavaScript (Standard ECMA-262 3rd Edition – December 1999). 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. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa data-interchange yang ideal. JSON dibangun dalam dua struktur [15]:

1. Beberapa pasangan dari nama/nilai. Dalam beberapa bahasa perograman biasa disebut dengan istilah object, record, struct, tabel hash, key list atau associative array.
2. Nilai-nilai yang terusun secara ordered list. Biasa disebut dengan array, vector, list atau daftar dalam bahasa pemrograman.

Format data JSON mempunyai aturan sebagai berikut:

1. Object adalah satu set nama/nilai yang tidak terurut (An object is an unordered set of name/value pairs). Penulisan object dimulai dengan tanda { (left brace) dan diakhiri dengan tanda } (right brace). Setiap nama diikuti oleh tanda : (colon) dan pasangan nama/nilai dipisahkan dengan tanda , (comma).

2. Array adalah sekumpulan nilai yang teratur (*An array is an ordered collection of values*). Penulisan sebuah array dimulai dengan tanda [(*left bracket*) dan diakhiri dengan tanda] (*right bracket*). Nilai dipisahkan menggunakan tanda , (*comma*).
3. Nilai bisa berupa string dalam tanda kutip, atau number (angka), TRUE atau FALSE atau NULL, sebuah object atau sebuah array. Struktur ini dapat ditulis menggunakan metode bersarang.
4. String adalah rangkaian atau urutan karakter unicode yang berada dalam tanda kutip, bisa juga hanya berisi karakter kosong, menggunakan tanda \ (*backslash*) untuk escape. Karakter di representasikan sebagai string tunggal. Tipe data string pada JSON sangat mirip dengan definisi pada bahasa C atau Java.
5. Number sangat mirip dengan definisi pada bahasa C atau Java, hanya saja tipe bilangan oktal dan heksadesimal tidak digunakan.

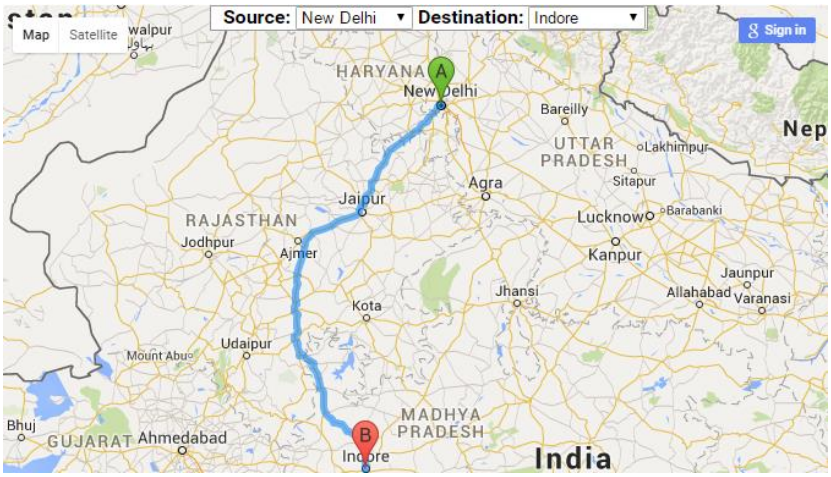
2.8.7 Google Direction API

Google Directions API adalah layanan dari google yang memudahkan kita (developer) untuk mencari rute dan navigasi dari satu tempat ke tempat tertentu. Kita tinggal memasukkan latitude dan langitude posisi berangkat dan juga latitude longitude posisi tujuan.

Keunggulan dari API ini adalah dia mudah digunakan, kita hanya tinggal melakukan HTTP Request untuk memanggil Google Directions API. Selain mudah, dia juga menyediakan banyak moda transportasi, setiap moda transportasi bisa saja memiliki rute tersendiri dan waktu tempuh tersendiri. Misalkan, orang berjalan bisa melawan arus di jalan raya sedangkan mobil tidak, mobil harus memutar jika tempat yang seharusnya didatangi terlewati sedangkan jalan hanya satu arah. Adapun fasilitas yang bisa digunakan secara gratis tetapi mempunyai batasan sebagai berikut [16]:

1. 2500 *reques*t dalam 24 jam
2. Mode transit dihitung 4 *request*
3. Bisa menggunakan 8 *waypoint* dari setiap *request*

Jika aplikasi yang dibangun besar dan 2500 tidaklah cukup, bisa menggunakan google maps API for *business.*

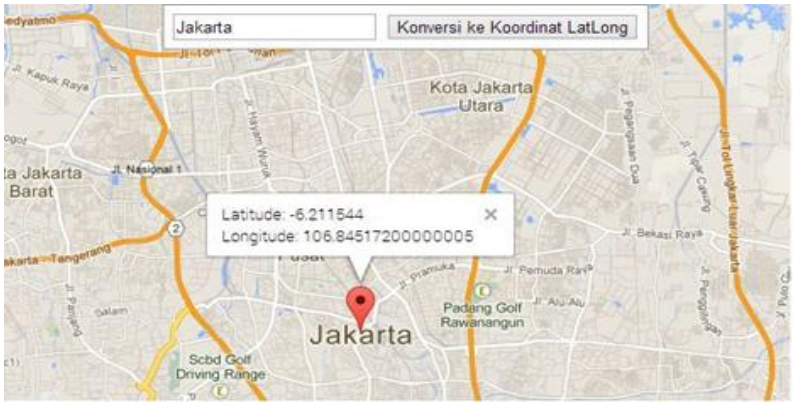


Gambar 2. 4 *contoh Google Direction API*

2.8.8 Google Geocoding API

Geocoding adalah proses konversi alamat (seperti "1600 Amphitheatre Parkway, Mountain View, CA") menjadi koordinat geografis (seperti garis lintang 37,423021 dan garis bujur -122,083739), yang bisa Anda gunakan untuk menempatkan marker pada peta, atau memosisikan peta.

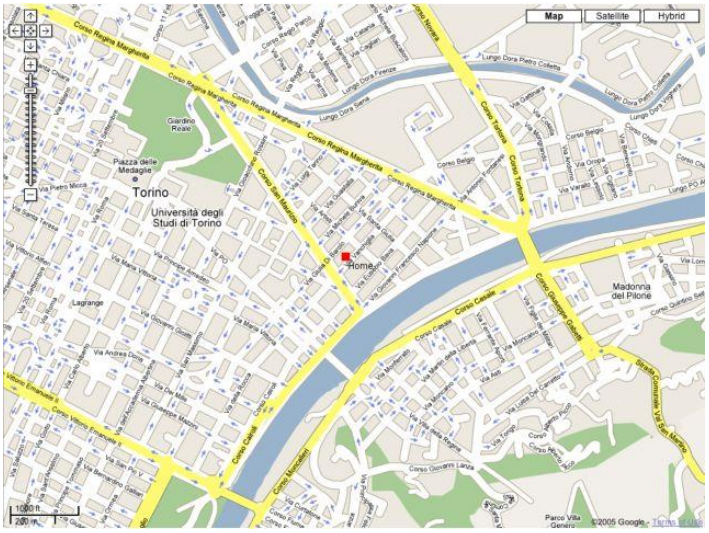
Geocoding terbalik adalah proses mengonversi koordinat geografis menjadi alamat yang bisa dibaca orang. Layanan geocoding terbalik pada Google Maps Geocoding API juga memungkinkan Anda menemukan alamat untuk ID tempat yang diberikan [17].



Gambar 2. 5 contoh *Google Geocoding API*

2.8.9 Google Maps API

Google Maps Application Programming Interface (API) merupakan suatu fitur aplikasi yang dikeluarkan oleh google untuk memfasilitasi pengguna yang ingin mengintegrasikan *Google Maps* ke dalam *website* masing-masing dengan menampilkan *data point* milik sendiri. Dengan menggunakan *Google Maps* API, *Google Maps* dapat di-*embed* pada *website* eksternal. Agar aplikasi *Google Maps* dapat muncul di *website* tertentu, diperlukan adanya API *key*. API *key* merupakan kode unik yang digenerasikan oleh google untuk suatu *website* tertentu, agar *server Google Maps* dapat mengenali [16].



Gambar 2. 6 *Google Maps*

2.9 Pemrograman Berorientasi Objek

Object Oriented Programing (OOP) merupakan paradigma pemrograman yang popular saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programing yang berarti pula Pemrograman Beorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970.

Pemrograman Berorientasi Objek (*Object Oriented Programming*/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat

menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya.

Pemrograman adalah mengharuskan pemahaman. Bahasa pemograman merupakan akas untuk perancangandan pemograman. Hanya bila pemograman dapat memahami gagasan-gagasan keberadaan fasilitas-fasilitas Dwibahasa pemograman memungkinkan pemrogram menjadi mahir menggunakan bahasa pemrograman. Bahasa berorientasi objek adalah akas yang dapat dipergunakanoleh pemrogram secara bagus atau jelek [18].

Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan Object yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat benyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya [19].

2.9.1 Konsep Dasar Pemrograman Berorientasi Objek

Berikut ini adalah konsep-konsep dalam pemrograman berorientasi objek [20]:

1. *Class*

Kelas (*Class*) merupakan penggambaran satu set objek yang memiliki atribut yang sama. Kelas mirip dengan tipe data ada pemrograman non objek, akan tetapi lebih komprehensif karena terdapat struktur sekaligus karakteristiknya. Kelas baru dapat dibentuk lebih spesifik dari kelas ada umumnya.kelas merupakan jantung dalam pemrograman berorientasi objek.

2. *Object*

Objek merupakan teknik dalam menyelesaikan masalah yang kerap muncul dalam pengembangan perangkat lunak. Teknik ini merupakan teknik yang efektif dalam menemukan cara yang tepat dalam membangun sistem dan menjadi metode yang paling banyak dipakai oleh para pengembang perangkat lunak. Orientasi objek merupakan teknik pemodelan sistem riil yang berbasis objek.

3. *Abstraction*

Kemampuan sebuah program untuk melewati aspek informasi yang diolah adalah kemampuan untuk fokus pada inti permasalahan. Setiap objek dalam sistem melayani berbagai model dari pelaku abstrak yang dapat melakukan kerja, laporan dan perubahan serta berkomunikasi dengan objek lain dalam sistem, tanpa harus menampakkan kelebihan diterapkan.

4. *Enkapsulasi*

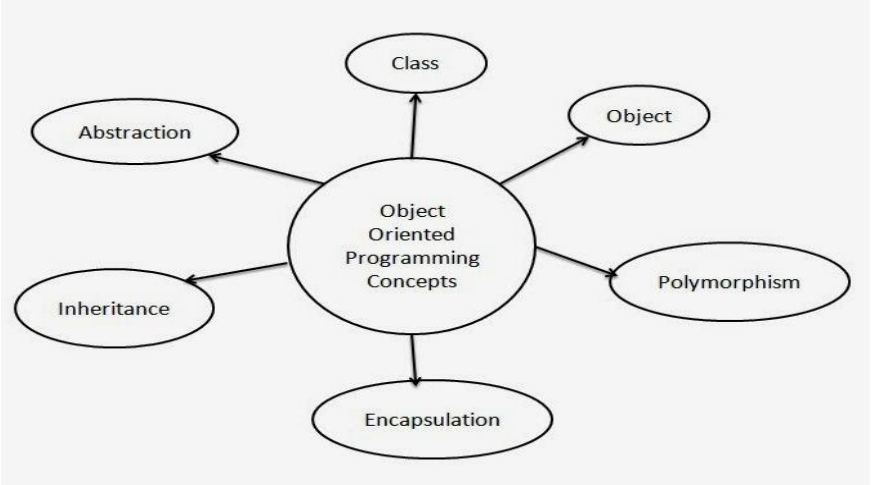
Enkapsulasi adalah proses memastikan pengguna sebuah objek tidak dapat menggantikan keadaan dari sebuah objek dengan cara yang tidak sesuai prosedur. Artinya, hanya metode yang terdapat dalam objek tersebut yang diberi izin untuk mengakses keadaan yang diinginkan. Setiap objek mengakses interface yang menyabutkan bagaimana objek lainnya dapat berintegrasi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. *Polimorfisme*

Polimorfisme merupakan suatu fungsionalitas yang diimplikasikan dengan berbagai cara yang berbeda. Pada program berorientasi objek, pembuat program dapat memiliki berbagai implementasi untuk sebagian fungsi tertentu.

6. *Inheritas*

Konsep *inheritas* mempunyai fungsi mengatur polimorfise dan enkapsulasi dengan mengizinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Objek-objek ini dapat membagi dan memperluas perilaku mereka tanpa mengimplementasikan perilaku tersebut.



Gambar 2. 7 Konsep Dasar Pemrograman Berorientasi Objek

2.10 Unifield Model Language (UML)

Unified Modelling Language (UML) yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software. Saat ini UML sudah menjadi bahasa standar dalam penulisan blue print software.

UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corps*. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai prespetktif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. [21] Tujuan dan fungsi dari penggunaan UML di antaranya:

1. Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemerograman maupun proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.

4. Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak *(software) saja*.
6. Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

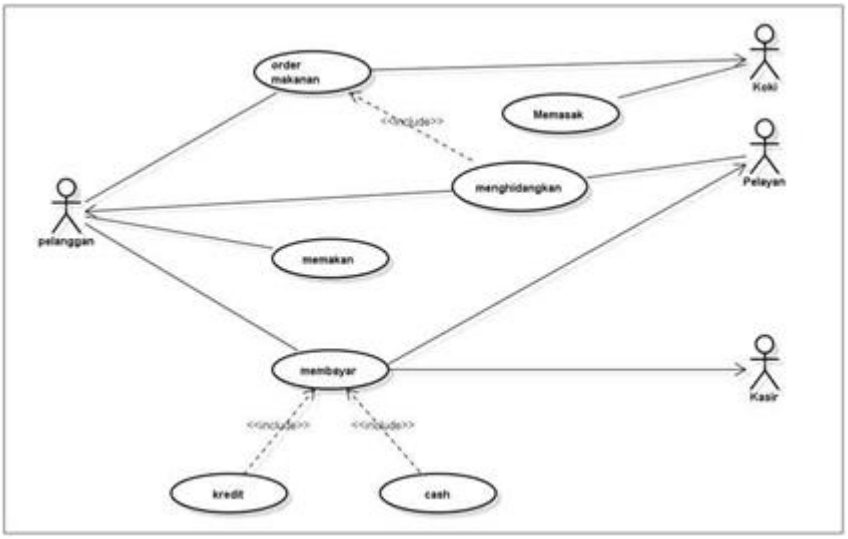
2.10.1 Jenis-Jenis Diagram UML

Unified Modelling Language (UML) adalah bahasa grafis untuk mendokumentasi, menspesifikasikan dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstrak, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha dari banyak pihak, di dukung oleh kakas-kakas yang diintegrasikan lewat XML. Standar UML dikelola Loe OMG *(Object Management Group* [22].

Berikut ini adalah diagram pada uml yang biasa digunakan dalam pembangunan perangkat lunak

1. Use case diagram

Use case diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, use case diagram juga dapat mendeskripsikan tipe interaksi antara pemakai sistem dengan sistemnya.



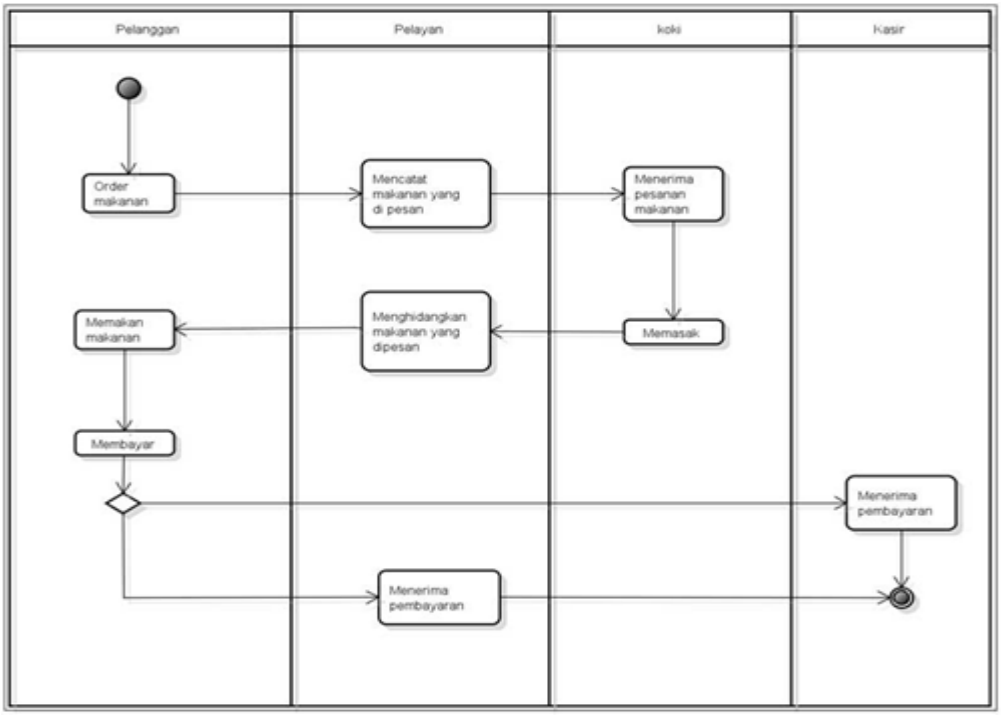
Gambar 2. 8 Contoh *Use Case Diagram*

2. Activity Diagram

Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.

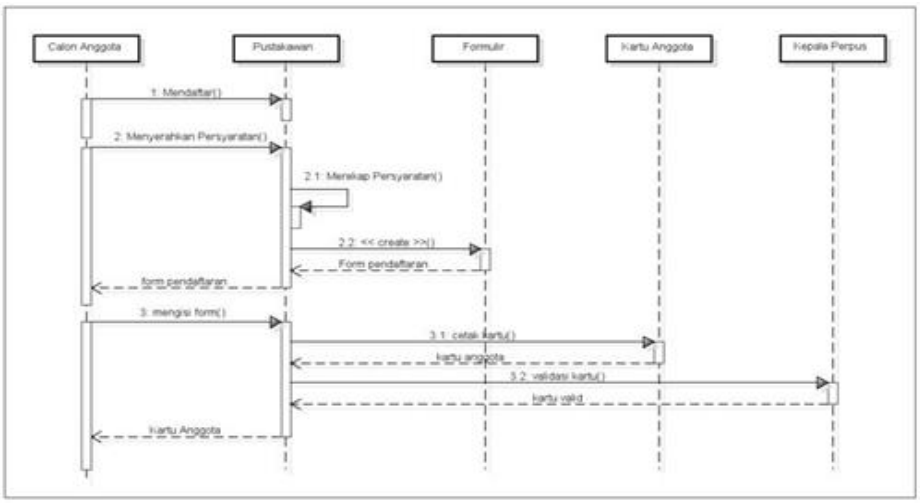
Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.



Gambar 2. 9 Contoh *Activity Diagram*

3. Sequence Diagram

Sequence diagram yaitu salah satu jenis diagram pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*.

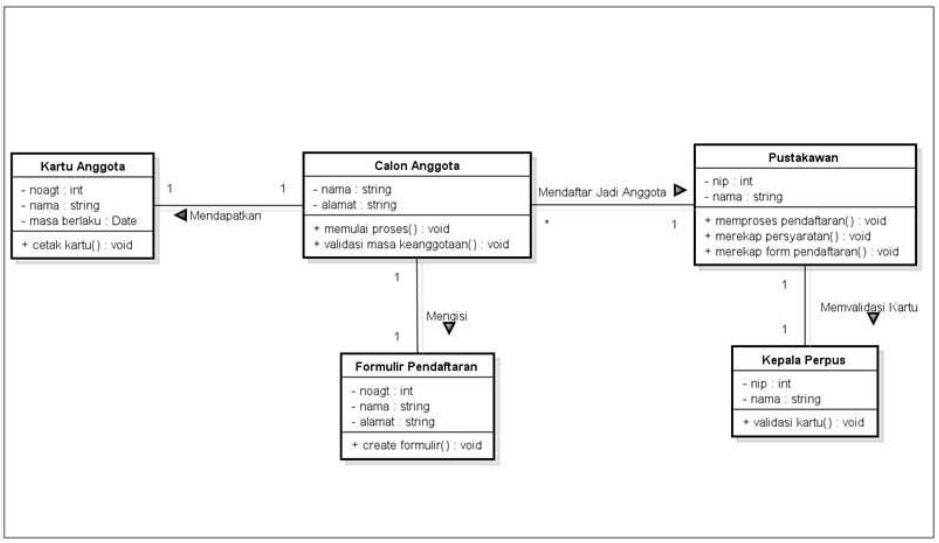


Gambar 2. 10 Contoh *Sequence Diagram*

4. Class Diagram

Class diagram yaitu salah satu jenis diagram pada UML yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut. Sususan struktur kelas ynag baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

a) Kelas Main yaitu kelas yang memiliki fungsi awal di eksekusi ketika sistem dijalankan
b) Kelas Interface kelas yang mendefinisikan dan mengatur tampilan ke pemakai, biasanya disebut kelas *boundaries.*
c) Kelas yang di ambil dari pendefinisian *use case* merupakan kelas yang menangani fungsi-fungsi yang harus diambil dari pendefinisian *use case*
d) Kelas Entitas merupakan kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.



Gambar 2. 11 Contoh *Class Diagram*

2.10.2 Ruang Lingkup UML

Secara khusus, Unified Modeling Language (UML) menspesifikasikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan serta implementasi dalam sistem yang sangat bernuansa perangkat lunak (software intensive system). Dalam hal ini, *Unified Modeling Language* (UML) bukanlah merupakan bahasa pemprograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemprograman, sehingga adalah mungkin melakukan pemetaan (*mapping*) langsung dari model-model yang dibuat dengan *Unified Modeling Language* (UML) dengan bahasa-bahasa pemprograman berorientasi obyek, seperti Java, Borland Delphi, Visual Basic, C++, dan lain-lain. Pemetaan (*mapping*) *Unified Modeling Language* (UML) bersifat dua arah yaitu [19]:

1. Generasi kode bahasa pemprograman tertentu dari *Unified Modeling Language* (UML) forward engineering.
2. Generasi kode belum sesuai dengan kebutuhan dan harapan pengguna, pengembang dapat melakukan langkah balik bersifat *iterative* dari implementasi ke *Unified Modeling Language* (UML) hingga didapat sistem/peranti lunak yang sesuai dengan harapan pengguna dan pengembang. [21].

2.11 Pengujian *Black Box*

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari sofware. Karena itu ujicoba *blackbox* memungkinkan pengembang sofware untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional satu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainya, selain menggunakan *whitebox*.

Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

1. Fungsi-funsi yang salah atau hilang
2. Kesalahan interface

3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminal

Tipe-tipe dari *blackbox testing*:

1. *Equivalence class partitioning*
2. *Sampel testing*
3. *Robustness testing*
4. *Behavior testing*
5. *Requirement testing*

2.12 Pengujian Skala Likert

Pengertian atau defenisi Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Skala ini merupakan suatu skala psikometrik yang biasa diaplikasikan dalam angket dan paling sering digunakan untuk riset yang berupa survei, termasuk dalam penelitian survei deskriptif.

Penggagas dan pencipta skala likert adalah Rensis Likert asal Amerika Serikat yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan Skala Likert, variabel yang akan diukur dijabarkan menjadi indikator variabel. [23]

Skala Likert merupakan metode skala bipolar yang mengukur baik tanggapan positif ataupun negatif terhadap satu pernyataan. Empat skala pilihan juga kadang digunakan untuk kuesioner skala Likert yang memaksa orang memilih satu kutub karena pilihan “netral” tak tersedia. Ada beberapa langkah prosedur yang harus dilakukan peneliti, antara lain: [24]

1. Peneliti mengumpulkan item-item yang cukup banyak, memiliki relevansi dengan masalah yang sedang diteliti dan terdiri dari item-item yang cukup jelas disukai dan tidak disukai.

2. Kemudian item-item itu dicoba kepada sekelompok responden yang cukup representatif dari populasi yang ingin diteliti.
3. Responden di atas diminta untuk mengecek tiap item, apakah ia menyenangi (+) atau tidak menyukainya (-). Respons tersebut dikumpulkan dan jawaban yang memberikan indikasi menyenangi diberi skor tertinggi. Tidak ada masalah untuk memberikan angka 5 untuk yang tertinggi dan skor 1 untuk yang terendah atau sebaliknya. Yang penting adalah konsistensi dari arah sikap yang diperlihatkan. Demikian juga apakah jawaban "setuju" atau "tidak setuju" disebut yang disenangi, tergantung dari isi pertanyaan dan isi dari item-item yang disusun. Biasanya di sediakan lima pilihan skala dengan format seperti ini:

- Pertanyaan Positif (+)
  Skor 1. Sangat (tidak setuju/buruk/kurang sekali)
  Skor 2. Tidak (setuju/baik/) atau kurang
  Skor 3. Netral / Cukup
  Skor 4. (Setuju/Baik/suka)
  Skor 5. Sangat (setuju/Baik/Suka)
- Pertanyaan Negatif (-)
  Skor 1. Sangat (setuju/Baik/Suka)
  Skor 2. (Setuju/Baik/suka)
  Skor 3. Netral / Cukup
  Skor 4. Tidak (setuju/baik/) atau kurang
  Skor 5. Sangat (tidak setuju/buruk/kurang sekali)

4. Total skor dari masing-masing individu adalah penjumlahan dari skor asing-masing dari individu tersebut.
5. Respons di analisis untuk mengetahui item-item mana yang sangat nyata batasan antara skor tinggi dan skor rendah dalam skala total.