

BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Madrasah Ibtidaiyah Al-Firdaus

Tinjauan sekolah menjelaskan tentang profil tempat penelitian yang terdiri dari Logo Madrasah Ibtidaiyah Al-Firdaus, Sejarah Madrasah Ibtidaiyah Al-Firdaus, struktur organisasi dan deskripsi yang ada di Madrasah Ibtidaiyah Al-Firdaus Bandung.

2.1.1 Logo Madrasah Ibtidaiyah Al-Firdaus



Gambar 2.1 Logo Madrasah Ibtidaiyah Al-Firdaus

2.1.2 Sejarah Madrasah Ibtidaiyah Al-Firdaus

Madrasah Ibtidaiyah Al-Firdaus Kota Bandung, merupakan instansi pendidikan yang didirikan oleh Drs. Yusron Fauzi selaku ketua Yayasan Al-Insyirah Firdaus Al-Haq guna turut serta dalam program Pemerintah di bidang pendidikan, dalam rangka mencerdaskan kehidupan Bangsa dengan membentuk siswa yang Beriman, Bertaqwa Kepada Allah Subhanahu Wa Ta'ala serta berakhlak mulia dan mempunyai rasa tanggung jawab terhadap masyarakat sesuai dengan nilai Islam. Sebagai instansi pendidikan, Madrasah Ibtidaiyah Kota Bandung tentunya memberikan pengajaran ilmu agama dan pengajaran umum yang dapat bermanfaat bagi siswa dan sebagai bekal untuk menempuh jenjang pendidikan yang lebih tinggi ataupun sebagai bekal kehidupannya dimasa yang akan datang.

Madrasah Ibtidaiyah Al-Firdaus didirikan pada tanggal 15 Mei 2012 bertempat di Komplek Gempol Asri Jl. GA XII No. 4 RT. 09 RW. 010 Kel. Gempolsari, Kec. Bandung Kulon Kota Bandung 40215. Awal mula bangunan sekolah sebagai tempat kegiatan pembelajaran merupakan sebuah rumah dengan dua lantai, di lantai pertama dipakai untuk kegiatan pembelajaran Raudhatul Athfal (RA) Al-Firdaus, sedangkan lantai kedua dipakai untuk kegiatan pembelajaran Madrasah Ibtidaiyah Al-Firdaus. Pada Tahun 2015 terjadi renovasi bangunan disebabkan bertambahnya jumlah siswa dan membutuhkan ruangan lebih, sehingga untuk kegiatan pembelajaran Raudhatul Athfal Al-Firdaus pindah bangunan ke Jl. GA XII No.2.

Tahun 2016 proses pembangunan telah selesai dan seluruh ruangan dapat digunakan untuk kegiatan pembelajaran, sampai saat ini Madrasah Ibtidaiyah Al-Firdaus mengelola sekolah dengan jumlah Guru 12 orang dan sebanyak 113 orang siswa.

Meskipun baru beberapa tahun berdiri, Madrasah Ibtidaiyah Al-Firdaus telah menunjukkan prestasi dari setiap siswa/siswi baik dari kurikulum mata pelajaran umum yang mengacu kepada KEMENDIKNAS ataupun dari mata pelajaran Agama Islam yang mengacu kepada KEMENAG.

2.1.3 Visi dan Misi Madrasah Ibtidaiyah Al-Firdaus

2.1.3.1 Visi

Memberikan bimbingan kepada generasi islam agar menjadi manusia yang taqwa, cerdas, terampil, kreatif, inovatif, dan berakhlaqul karimah serta dapat mengikuti perkembangan zaman melalui pendidikan umum dan pendidikan Agama.

2.1.3.2 Misi

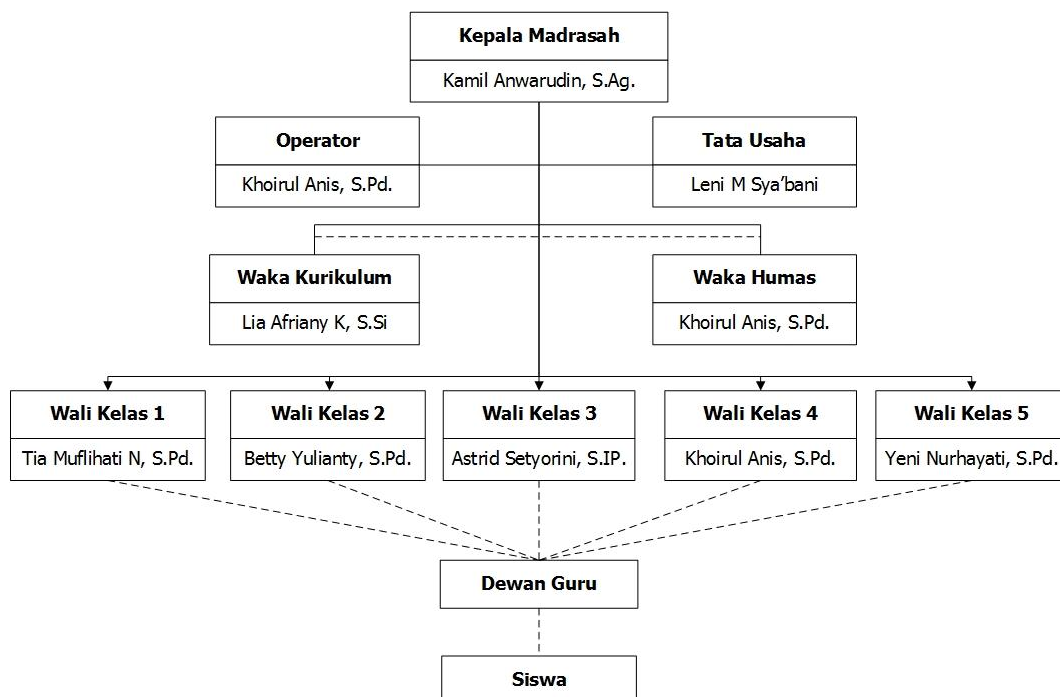
Didalam pelaksanaan kegiatan, Madrasah Ibtidaiyah Al-Firdaus Bandung mengemban misi :

1. Mengembangkan kompetensi siswa dalam aspek kognitif, Afektif, dan psikomotor dengan melaksanakan proses belajar mengajar yang aktif, kreatif dan menyenangkan.
2. Mengaplikasikan nilai-nilai Islam dalam kehidupan sehari-hari.

3. Menjadikan Sekolah Dasar Islam Al-Firdaus sebagai sekolah unggulan yang menitik beratkan pada pengembangan Intelegensi Emosional dan Spiritual Quotion.

2.1.4 Struktur Organisaasi Madrasah Ibtidaiyah Al-Firdaus

Dalam sebuah organisasi agar semua kegiatan berjalan dengan baik dan dapat mencapai tujuan, perlu adanya suatu struktur organisasi dan pembagian kerja (*job description*) yang jelas. Struktur organisasi yang baik harus menggambarkan dengan jelas wewenang dan tanggung jawab serta fungsi-fungsi dari setiap bagian yang ada dalam instansi, yang mana dalam hal ini merupakan salah satu syarat terciptanya suatu pengendalian internal yang memadai. Adapun struktur organisasi Madrasah Ibtidaiyah Al-Firdaus dapat dilihat seperti pada Gambar 2-1.



Gambar 2.2 Struktur Organisasi

2.2 Jemputan

Jemputan berasal dari kata jemput yang artinya diundang atau disambut, sedangkan jemputan itu sendiri adalah kendaraan (mobil) yang dipakai untuk menjemput [11].

2.3 Monitoring

Monitoring adalah proses menjaga atau pengawasan terhadap keberadaan dan besarnya perubahan keadaan dan arus data dalam sebuah sistem. *Monitoring* bertujuan untuk mengidentifikasi kesalahan dan kemajuan dalam menentukan keputusan selanjutnya. Teknik yang digunakan dalam *monitoring* informasi sistem memotong bidang pengolahan real-time, statistik, dan analisis data. Satu set Komponen perangkat lunak yang digunakan untuk pengumpulan data, pengolahan, dan presentasi disebut sistem *monitoring*[12].

2.4 Landasan Teori

Landasan teori yang digunakan dalam penyusunan laporan tugas akhir dan pembangunan aplikasi *monitoring* jempukan siswa di Madrasah Ibtidaiyah Al-Firdaus Bandung adalah sebagai berikut :

2.4.1 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. Membeli Android Inc. Yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 peranti keras, peranti lunak, dan telekomunikasi, termasuk Google Inc.

Android merupakan generasi baru *platform mobile*, *platformi* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkan. Sistem operasi yang mendasari Android dilisensikan di bawah GNU, General Public Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah “*copyleft*” lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah term. Adnroid didistribusikan di bawah Lisensi *Apache software* (ASL/Apache), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk meningkatkan platform tanpa harus memberikan perbaikan mereka ke masyarakat open source. Sebaliknya pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka di bawah lisensi apapun

yang mereka inginkan. Pengembang aplikasi android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan.

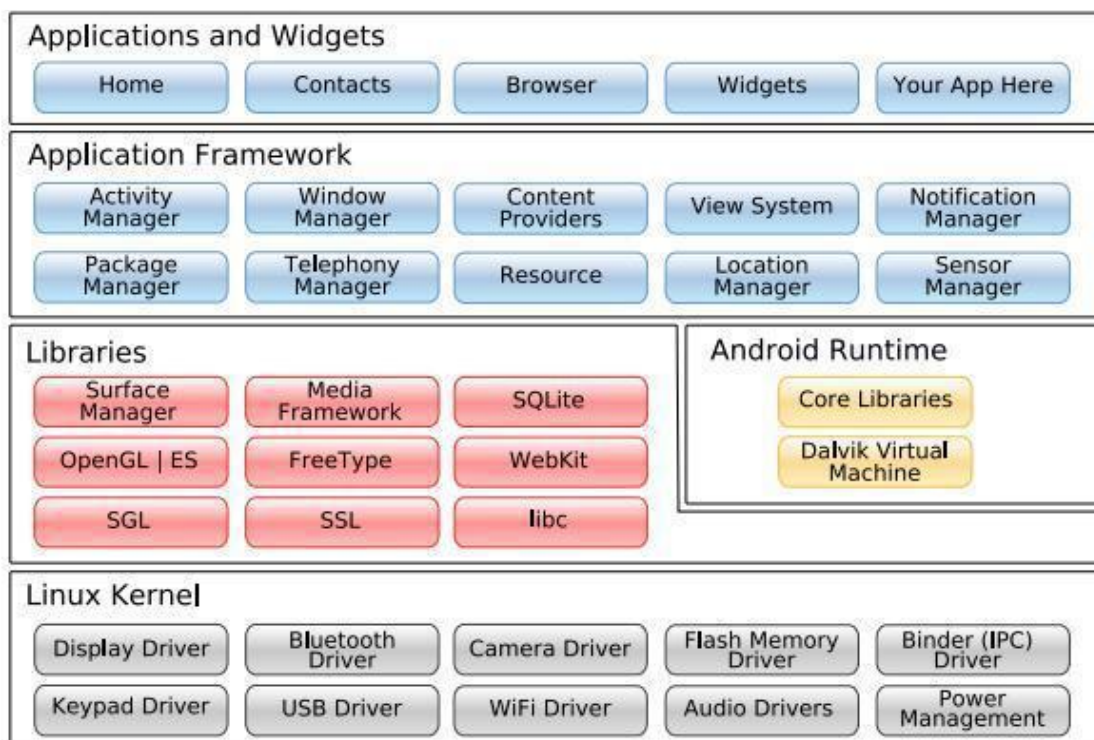
Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman java. [2]

2.4.1.1 Arsitektur Android

Arsitektur Android dapat digambarkan seperti pada gambar 1 dan secara garis besar Arsitektur Android dapat dijelaskan sebagai berikut:

a. Application dan Widgets

Application dan Widgets ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Hampir semua aplikasi ditulis menggunakan bahasa pemrograman Java.



Gambar 2.3 Arsitektur Android

Sumber : Buku N. Safaat, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Pada Android. Bandung: Informatika, 2011.

b. Application Framework

Android adalah “Open Development Platform” yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resource, menjalankan service background, mengatur alarm, dan menambah status notifications, dan sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (reuse). Sehingga bisa kita simpulkan Application Frameworks ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content providers yang berupa sms dan panggilan telepon. Komponen-komponen yang termasuk di dalam Application Frameworks adalah sebagai berikut :

1. Views
2. Content Provider
3. Resource Manager
4. Notification Manager
5. Activity Manager [2]

c. Libraries

Libraries ini adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan di atas Kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc SSL, serta :

1. Libraries media untuk pemutaran media audio dan video
2. Libraries untuk manajemen tampilan

3. Libraries Graphics mencakup SGL dan OpenGL untuk grafis 2D dan 3D
 4. Libraries SQLite untuk dukungan database
 5. Libraries SSL dan WebKit terintegrasi dengan web browser dan security
 6. Libraries LiveWebcore mencakup modern web browser dengan engine embedded webview
 7. Libraries 3D yang mencakup implementasi OpenGL ES1.0 API's.
- [2]

d. Android Run Time

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android Run Time dibagi menjadi dua bagian yaitu:

1. Core Libraries: Aplikasi Android dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan Virtual Machine Java, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh Core Libraries.
 2. Dalvik Virtual Machine: Virtual mesin pada register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat Linux Kernel untuk melakukan threading dan manajemen tingkat rendah.
- [2]

e. Linux Kernel

Linux Kernel adalah layer dimana inti dari system operasi Android itu berada. Berisi file-file system yang mengatur sistem processing, memory, resource, drivers, dan sistem-sistem operasi Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel release 2.6. [2]

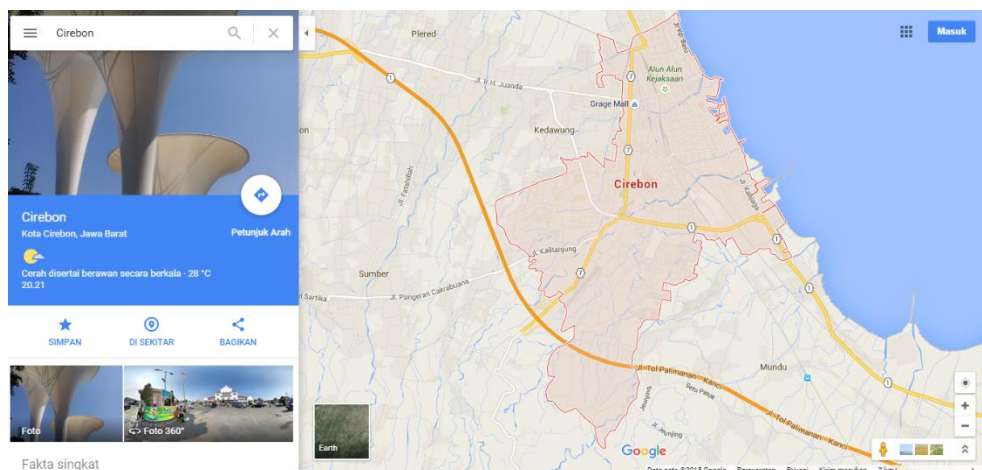
2.4.2 Google Maps

GoogleMaps adalah peta online atau membuka peta secara online, dapat dilakukan secara mudah melalui layanan gratis dari Google. Bahkan layanan ini

menyediakan API (Application Programming Interface) yang memungkinkan developer lain untuk memanfaatkan aplikasi ini di aplikasi buatannya. Tampilan GoogleMaps pun dapat dipilih, berdasarkan foto asli atau peta gambar rute saja.

GoogleMaps adalah layanan gratis yang diberikan oleh Google dan sangat populer. GoogleMaps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah seperti pada gambar 2. Dengan kata lain, GoogleMaps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur GoogleMaps dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan GoogleMaps API. GoogleMaps API adalah suatu library yang berbentuk JavaScript.

Cara membuat GoogleMaps untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta JavaScript, serta koneksi Internet yang sangat stabil. Dengan menggunakan GoogleMaps API, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia.



Gambar 2.4 Google Maps

Sumber : Website www.google.co.id/maps

Pada GoogleMaps API terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

- a. ROADMAP, untuk menampilkan peta biasa 2 dimensi.
- b. SATELLITE, untuk menampilkan foto satelit.
- c. TERRAIN, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
- d. HYBRID, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota). [3]

2.4.3 GPS (Global Positioning System)

GPS adalah singkatan dari Global Positioning System, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (receiver) di permukaan, dimana GPS receiver ini akan mengumpulkan informasi dari satelit GPS, seperti :

- a. Waktu. GPS receiver menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
- b. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi:
 - 1. Latitude
 - 2. Longitude
 - 3. Elevasi
- c. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
- d. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
- e. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS receiver.
- f. Komulasi data. GPS receiver dapat menyimpan informasi track, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya. [3]

2.4.3.1 Akurasi GPS

Posisi yang ditunjukkan oleh suatu GPS mempunyai faktor kesalahan atau juga disebut tingkat akurasi. Sebagai contoh suatu alat GPS menunjukkan titik koordinat

dengan tingkat akurasi 5 meter, itu berarti posisi pengguna bisa berada dalam range radius 5 meter dari titik yang ditunjukkan tersebut. Mengapa tingkat akurasi yang terlihat bisa berubah-ubah? Kadang terlihat 10 meter, 15 meter, atau 5 meter. Ada beberapa hal yang mempengaruhi tingkat akurasi tersebut, antara lain:

- a. Kesalahan Ephemeris. Terjadi jika satelit tidak dapat mentransmisikan posisinya di orbit dengan tepat.
- b. Keadaan Ionosphere. Ionosphere berada pada jarak sekitar 43-50 mil di atas permukaan bumi. Satelit yang melewati ionosphere akan menjadi lambat dikarenakan adanya plasma (gas dengan tingkat kepadatan rendah). Walaupun GPS receiver berusaha untuk mengkoreksi/memperbaiki faktor keterlambatan yang terjadi tetap saja aktivitas tertentu dari plasma bisa menyebabkan kesalahan perhitungan.
- c. Keadaan Troposphere. Troposphere adalah bagian terendah dari atmosfer sampai dengan ketinggian sekitar 11 mil dari permukaan tanah. Variasi pada temperatur, tekanan, dan kelembaban bisa menyebabkan perbedaan kecepatan penerimaan gelombang radio.
- d. Kesalahan Waktu. Karena penempatan jam atom pada setiap GPS receiver tidak berjalan sebagaimana mestinya. Kesalahan waktu dari GPS receiver yang tidak presisi dapat menimbulkan ketidakakurasian.
- e. Kesalahan Multipath. Terjadi karena sinyal satelit membentur permukaan keras (seperti bangunan atau tebing) sebelum mencapai GPS receiver. Hal tersebut bisa menyebabkan terjadinya delay sehingga perhitungan jarak menjadi tidak akurat.
- f. Buruknya Sinyal Satelit. Keadaan langit yang terhalang akan menyebabkan GPS sulit menerima data satelit. Sebuah sinyal satelit yang pada hari tertentu diterima dengan sangat bagus belum tentu pada hari lain bisa diterima dengan kualitas yang sama walaupun user berdiri pada tempat yang sama. Hal tersebut dikarenakan posisi dari satelit yang terus bergerak atau bisa juga disebabkan faktor penghalang lain seperti pohon, gedung bertingkat, dan sebagainya. [3]

2.4.3.2 Android dan GPS

Teknologi berkembang dengan pesat. Dulu ponsel hanya sekedar digunakan untuk menelpon dan SMS saja. Sekarang ponsel sudah menjelma menjadi kotak kecil ajaib yang serba bisa. Salah satunya adalah ponsel dengan sistem operasi Android. Dengan standarisasi fitur dan hardware yang dimiliki, menjadikan ponsel Android ponsel canggih yang disukai banyak orang. Tidak lagi canggih karena adanya fitur MMS, radio, atau internet berkecepatan tinggi tapi juga karena ditanamkannya fitur teknologi satelit di dalamnya. Ya, perangkat GPS receiver yang dulu besar dan eksklusif, sekarang sudah bisa dimiliki dengan “hanya” membeli sebuah ponsel.

Dengan ponsel berteknologi satelit (GPS), banyak hal yang bisa dilakukan. Ingin melihat di mana posisi user sekarang dalam sebuah peta? Mengambil foto/video yang sudah dilengkapi dengan data koordinat? Ingin tahu jalur olahraga bersepeda yang sudah pernah user lalui? Atau ingin tahu dimana posisi ponsel yang hilang dicuri orang? Semua itu bisa dilakukan dengan ponsel Android yang memiliki fitur GPS. Tidak hanya itu, user juga dapat pergi ke tempat wisata tertentu dengan dipandu gambar dan suara dari sebuah ponsel! Bahkan lebih jauh lagi, GPS dapat digunakan untuk membantu memberikan peringatan dini terhadap terjadinya bencana alam.

Sekarang ini banyak sekali pengembang - pengembang aplikasi untuk sistem operasi Android termasuk aplikasi-aplikasi GPS. Yang menyenangkan aplikasi-aplikasi tersebut jenisnya beragam dan jumlahnya pun banyak. [3]

2.4.3.3 Latitude dan Longitude

Latitude dan longitude adalah sebuah koordinat yang ada di bumi dan digunakan untuk menentukan kedudukan kita di atas bumi.

1. Latitude

Latitude adalah garis yang melintang dari kutub utara dan kutub selatan. Titik 0 adalah sudut ekuator, tanda + menunjukkan arah ke atas menuju kutub utara, sedangkan tanda minus di koordinat Latitude menuju ke kutub selatan. Titik yang dipakai dari 0 ke 90 derajat arah kutub utara, dan 0 ke -90 derajat ke kutub selatan.

2. Longitude

Longitude adalah garis lintang. Angka dari sudut bundar bumi horisontal. Titik diawali dari 0 ke 180 derajat, dan - ke arah sebaliknya. Titik 0 dimulai dari garis negara Inggris. Mengarah ke Indonesia akan menjadi angka positif. Kebalikannya koordinat Longitude minus adalah arah kebalikan.

2.4.4 JSON

JSON (*Java Script Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman 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 [4].

JSON terbuat dari dua struktur:

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

2.4.5 Bahasa Pemrograman PHP

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja.

PHP adalah bahasa pemrograman yang memungkinkan para *web developer* untuk membuat aplikasi *web* yang dinamis dengan cepat. PHP merupakan singkatan dari *Hypertext Preprocessor*. PHP dapat diintegrasikan ke dalam *web server*, atau dapat berperan sebagai program CGI yang terpisah. Karakteristik yang paling unggul dan paling kuat dalam PHP adalah lapisan integrasi *database* (*database integration layer*). *Database* yang didukung PHP adalah : Oracle, Adabas-D,

Sybase, FilePro, mSQL, Velocis, MySQL, Informix, Solid, dBase, ODBC, Unix dbm, dan PostgreSQL.2 [5].

Kelebihan PHP dari bahasa pemrograman lain:

- a. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. *Web Server* yang mendukung PHP dapat mudah ditemukan dari mulai IIS sampai dengan apache.
- c. Dalam sisi pengembangan lebih mudah, karena banyaknya developer yang siap membantu dalam pengembangan.
- d. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah system .

2.4.6 CodeIgniter

Menurut Blanco & Upton *CodeIgniter* adalah *powerful open source* PHP framework yang mudah dikuasai, dibangun untuk PHP *programmers* yang membutuhkan *toolkit* sederhana dan baik untuk membuat *full-featured web applications*. *CodeIgniter* adalah MVC framework yang di design untuk mempermudah penggunaanya [6].

2.4.7 JavaScript

JavaScript pada awalnya diciptakan untuk menangani validasi dari sebuah formulir dalam *website* yang sebelumnya ditugaskan pada bahasa pemograman *server* seperti Perl. Namun dengan berkembangnya zaman JavaScript menjadi pemain penting dalam perkembangan teknologi *website*.

JavaScript adalah bahasa *scripting* yang dibuat untuk berinteraksi dengan halaman *website* dan dibuat dengan 3 bagian :

1. ECMAScript, yang didefinisikan dalam ECMA-262 dan menjadi bagian utama dalam menjalankan fungsionalitas.
2. *Document Object Model* (DOM), yang menyediakan *method* dan *interface* untuk bekerja dengan isi dari halaman *website*.
3. *Browser Object Model* (BOM), yang menyediakan *method* dan *interface* untuk berinteraksi dengan *browser*.

2.4.8 MySQL

MySQL termasuk dalam kategori database management system, yaitu suatu database yang terstruktur dalam pengolahan dan penampilan datanya. MySQL merupakan database yang bersifat client server, dimana data diletakkan di server yang bisa diakses melalui komputer client [7].

Berikut ini adalah beberapa kelebihan MySQL :

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, dibawahh 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. *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. 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.

2.4.9 Android Studio

Android Studio adalah IDE resmi untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Di atas IntelliJ yang kuat *code editor* dan pengembang alat, Android Studio menawarkan lebih banyak fitur yang meningkatkan produktivitas Anda ketika membangun aplikasi Android, Seperti [8]:

1. Sebuah *fleksibel* pada *Gradle* membangun sistem
2. Membangun varian dan beberapa generasi file APK
3. Kode template untuk membantu anda membangun fitur aplikasi umum
4. Sebuah *layout editor* kaya dengan dukungan untuk *drag* dan *drop tema editing*
5. Alat *Lint* untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya.
6. Kode menyusut dengan *ProGuard* dan sumber daya menyusut dengan *Gradle*
7. *Built-in* dukungan untuk Google *Cloud Platform*, sehingga mudah untuk mengintegrasikan Google *Cloud Messaging* dan *App Engine*.

2.4.10 XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi merupakan kompilasi dari beberapa program seperti Apache HTTP Server, MySQL dan PHP dan Perl, program ini tersedia dalam GNU (*General Public License*).

2.4.11 Konsep Pemodelan UML 2.0

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek [8].

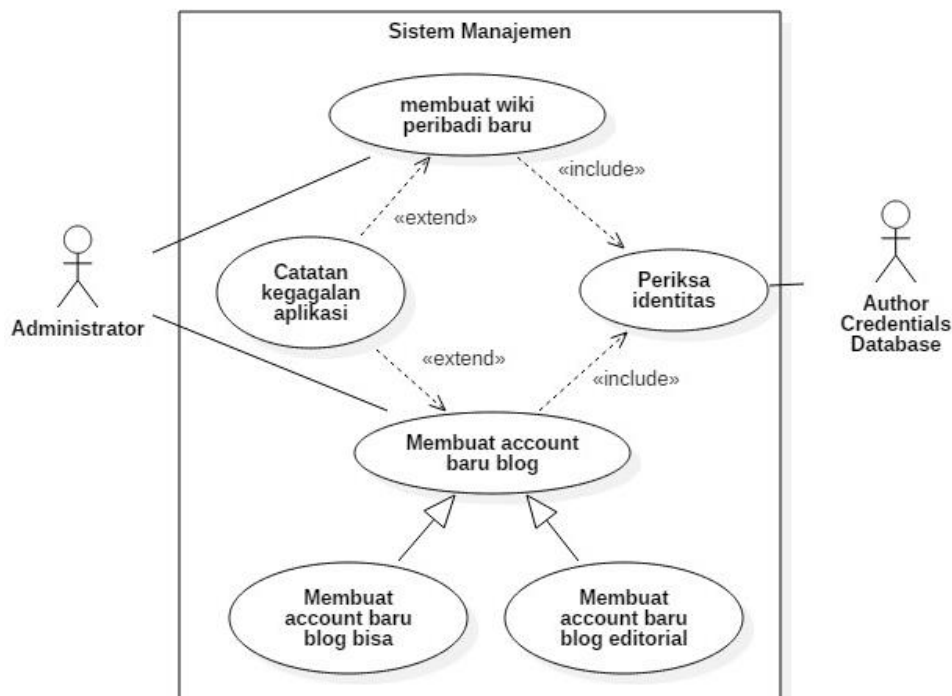
Untuk dapat mengerti konsep pemodelan UML, dibutuhkan pemahaman dari beberapa diagram yang akan digunakan dalam pemodelan dari sistem yang akan dibangun. Ada dua diagram yang harus difahami yang diantaranya adalah *structural diagram* dan *behavioral diagram*. *Structural diagram* digunakan untuk mendeskripsikan relasi antar kelas yang diantaranya terdapat *class diagram*, *object diagram*, *component diagram*, *deployment diagram*, sedangkan *behavioral diagram* digunakan untuk mendeskripsikan interaksi antara *actor* dan sebuah *use case* yang diantaranya terdapat *use case diagram*, *sequence diagram*, *collaboration diagram*, *statechart diagram* dan *activity diagram*.

Pada pembangunan Web *Visual Map Tracking* ini ada beberapa diagram yang digunakan yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*.

2.4.12 Use Case Diagram

Use case merupakan salah satu pemodelan kebutuhan dari sistem, menjelaskan bagaimana sistem digunakan dan merupakan titik awal dari pemodelan *UML*. *Use case diagram* yaitu diagram yang mendeskripsikan apa yang sistem lakukan tanpa mendeskripsikan bagaimana sistem menyelesaikannya, *use case diagram* dibuat berdasarkan interaksi dan relasi dari individual *use case* dan berisi *actor*, *event*, dan *use case*.

Use Case memiliki garis yang berguna untuk menjelaskan hubungan dari beberapa *use case* dan terdapat *use case scenario* yang berguna untuk menjelaskan alur dari setiap *use case*.



Gambar 2.5 Contoh Use Case Diagram

2.4.12.1 Use Case Relationship

Sebuah *use case* menggambarkan cara sistem berperilaku untuk memenuhi kebutuhan, untuk mengetahui hubungan antara satu *use case* dengan *use case* yang lainnya dibutuhkan suatu *flow* atau aliran yang digambarkan oleh suatu garis. Terdapat beberapa keterhubungan yang diantaranya adalah :

1. *Include*

Keterhubungan secara *include* antar use case menunjukkan bahwa *use case* asal secara eksplisit memasukan perilaku dari *use case* lain yang ditunjuk oleh *use case* tersebut. *Included use case* tidak pernah berdiri sendiri, tetapi hanya merupakan bagian dari beberapa *use case* yang lebih besar yang diikutinya. Keterhubungan *use case* secara include pada dasarnya merupakan sebuah contoh dari pendelegasian sekumpulan dari tanggung jawab sebuah sistem diambil dan ditangkap di dalam satu tempat (*included use case*), kemudian bagian lainnya dari sistem (*use case* lainnya) memasukan kumpulan tanggung jawab yang baru setiap saat mereka memerlukan fungsi – fungsi *use case* tersebut.

2. *Ekstend*

Keterhubungan *use case* secara *ekstend* menunjukkan bahwa *usecase* yang ditunjuk merupakan perluasan perilaku dari *use case* asal. *Use Case* asal dapat berdiri sendiri, tetapi untuk kondisi tertentu perilaku *use case* tersebut dapat diperluas oleh perilaku dari *use case* lain. Hubungan perluasan digunakan untuk memodelkan bagian dari *use case* yang dapat dilihat oleh user sebagai perilaku yang dapat dipilih dari sistem. Hubungan perluasan juga dapat digunakan untuk memodelkan sub aliran yang terpisah pisah yang hanya dapat dijalankan dalam kondisi tertentu.

2.4.13 Activity Diagram

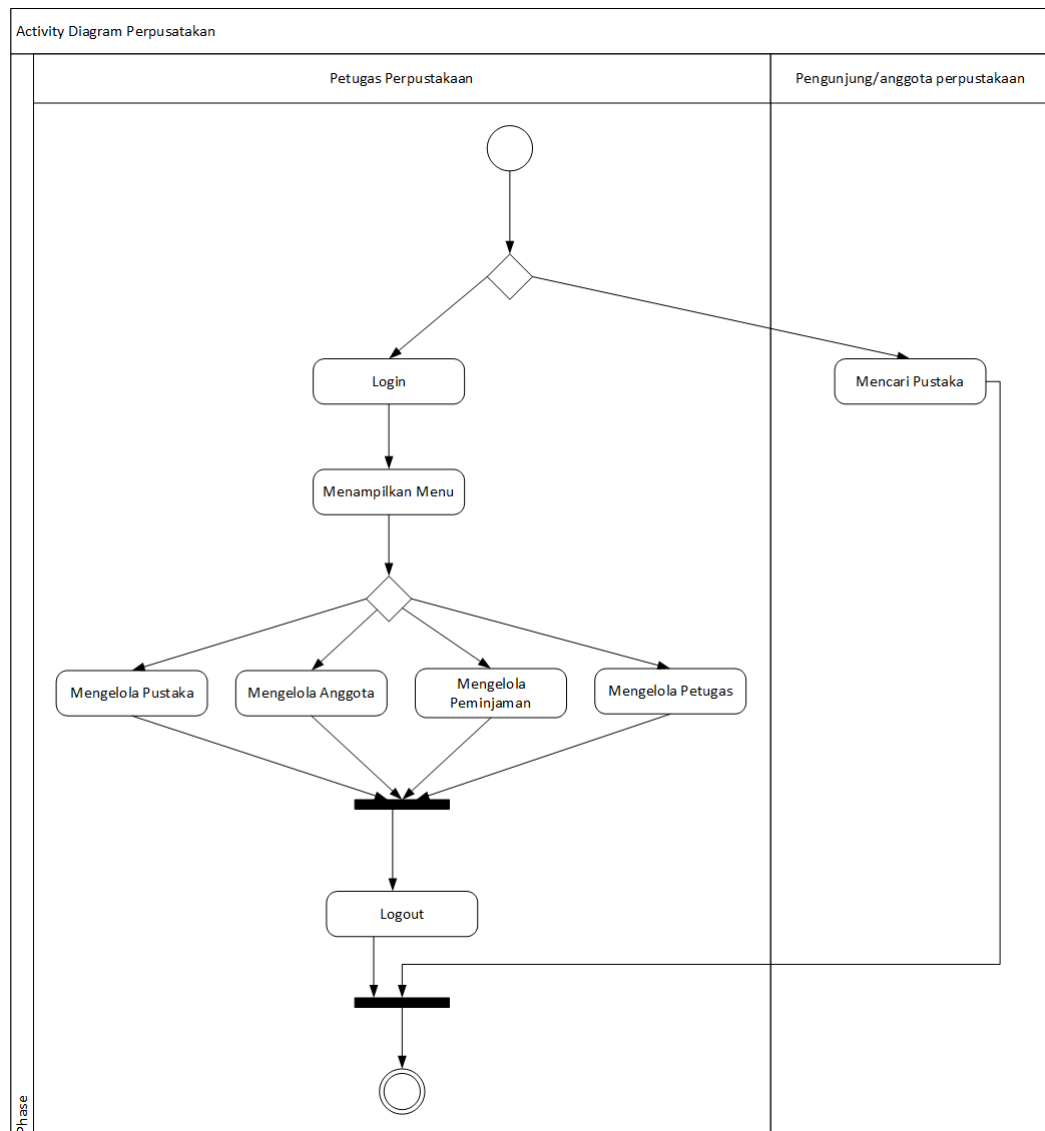
Activity diagram menjelaskan bagaimana sistem yang dibangun mencapai tujuannya, dalam pembangunan sistem ini *activity diagram* digunakan untuk menjelaskan setiap *use case* yang dibangun, acuan pembuatan dari *activity diagram* dengan melihat *step* dan *action* dari *use case scenario*, ada beberapa komponen yang harus diperhatikan pada *activity diagram* diantaranya *activity dan action, dan decisions dan merges*.

1. Activity dan Action

Activity merupakan suatu aktivitas dari suatu proses pada sistem sedangkan *action* merupakan langkah yang aktif untuk menyelesaikan dari suatu proses dan menghubungkan satu *activity* dengan *activity* lainnya.

2. Decisions dan Merges

Decisions merupakan suatu keputusan yang bisa digunakan ketika sistem akan mengeksekusi urutan atau proses yang berbeda dari suatu tindakan dan tergantung pada suatu kondisi, pada *activity diagram decision* digambarkan dalam bentuk diamond. *Merge* berguna untuk menyatukan kembali *flow* dari hasil proses *decision, merge* digambarkan dalam bentuk diamond tetapi *flow* yang datang boleh lebih dari satu.



Gambar 2.6 Contoh Activity Diagram

2.4.14 Class Diagram

Class diagram adalah diagram yang menunjukkan *class – class* yang ada dari sebuah sistem dan ada hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem, sehingga *class diagram* merupakan tulang punggung atau kekuatan dasar dari hamper setiap metode berorientasi objek termasuk UML. Ada beberapa komponen *class diagram* yang harus diperhatikan, diantaranya hak akses pada *class*, hubungan setiap *class*, dan *stereotype* pada *class*.

Hak akses pada *class* digunakan untuk membatasi penggunaan suatu method pada *class*. Ada beberapa hak akses pada *class diagram* yang diantaranya adalah sebagai berikut:

1. *Private*, variabel atau method yang hanya dapat diakses oleh class itu sendiri.
2. *Protected*, variabel atau method dapat digunakan oleh class turunan dari *class parent*.
3. *Public*, variabel atau method dapat digunakan oleh seluruh class.

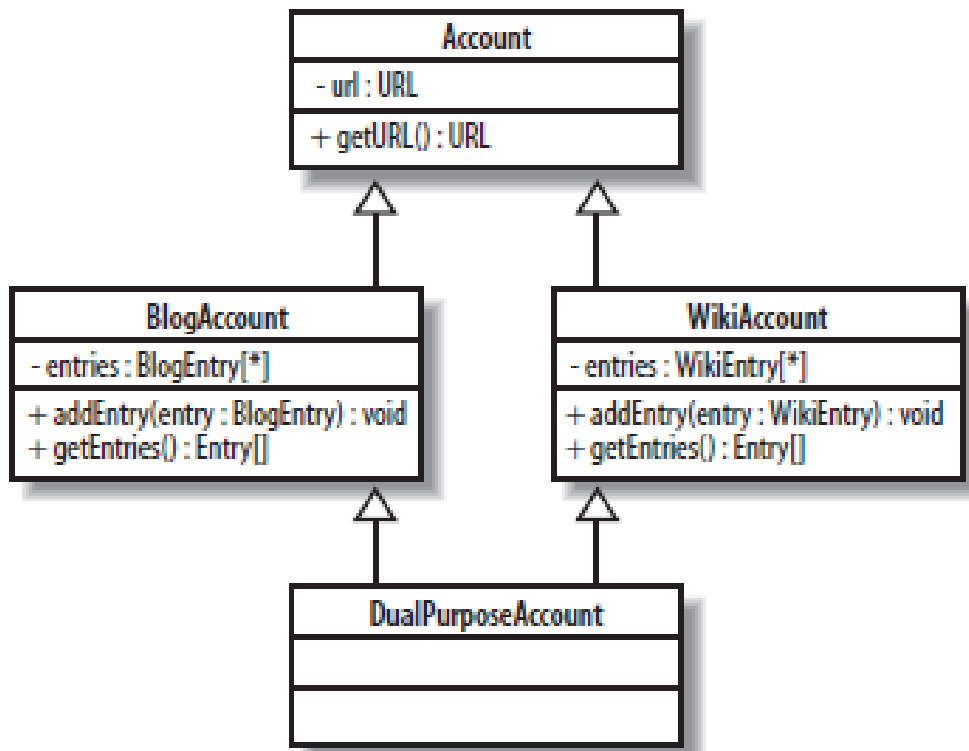
Hubungan pada setiap *class* digambarkan dalam sebuah *flow* atau garis yang memiliki makna yang berbeda dari setiap jenis hubungan, terdapat beberapa garis penghubung pada *class diagram*, yang diantaranya adalah sebagai berikut:

1. *Association*, sebuah asosiasi merupakan sebuah hubungan paling umum antara 2 class dan dilambangkan oleh sebuah garis yang solid.
2. *Composition*, sebuah *class* tidak dapat berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka relasi tersebut memiliki relasi *composition* terhadap *class* tempat memiliki ketergantungan. *Composition* digambarkan oleh garis dengan ujung diamond berwarna solid.
3. *Dependency*, digunakan jika sebuah *class* menggunakan *class* yang lain, pengguna *dependency* digunakan untuk menunjukkan operasi pada suatu class yang menggunakan *class* yang lain. Sebuah *dependency* dilambangkan sebagai sebuah panah bergaris putus – putus.

Stereotype pada *class* digunakan untuk mengkategorikan *class – class*. Ada beberapa *stereotype* pada *class* yang harus diperhatikan, diantaranya adalah sebagai berikut:

1. *Boundary Class*, merupakan *class* yang dapat menjelaskan hubungan antara aktor dan *use case*, tetapi tidak harus selalu membuat *class* untuk setiap pasangan aktor dan *use case* karena ada kemungkinan satu *boundary class* dapat mewakili dari beberapa proses *use case*.
2. *Entity Class*, merupakan *class* yang menangani informasi yang disimpan dalam penyimpanan tetap. *Entity class* dapat dibuat dengan cara memperhatikan

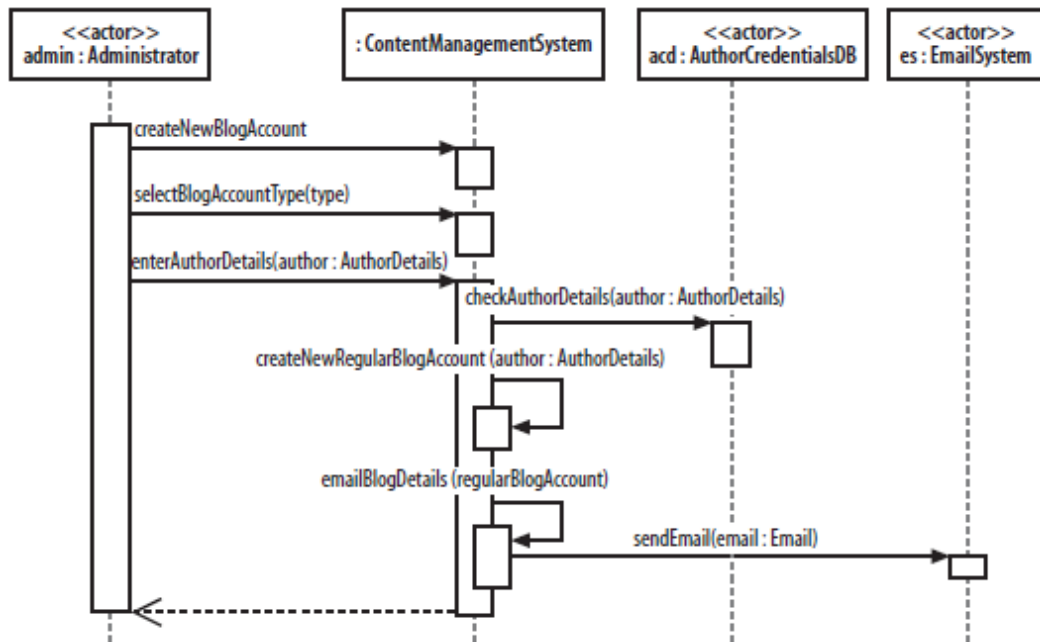
struktur basis data karena atribut pada *entity class* bisa saja merupakan field yang ada pada basis data.



Gambar 2.7 Contoh Class Diagram

2.4.15 Sequence Diagram

Sequence Diagram menjelaskan interaksi antar method pada *class diagram*, interaksi tersebut harus berdasarkan *use case diagram* yang telah di defenisikan sebelumnya, sehingga interaksi penggunaan setiap method pada *class diagram* untuk mencapai tujuan dari *use case* dapat dijelaskan secara detail.



Gambar 2.8 Contoh Sequence Diagram