BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan PT.4848 Irawan Sarpingi

Tinjauan perusahaan menjelaskan tentang profil tempat penelitian yang terdiri dari sejarah perusahaan, struktur organisasi dan deskripsi yang ada di PT.4848 Irawan Sarpingi.

2.1.1 Sejarah PT.4848 Irawan Sarpingi

Perusahaan jasa angkutan 4848 Irawan Sarpingi didirikan pada tanggal 4 Agustus 1959 oleh H. Irawan Sarpingi, seorang putra Jawa Barat kelahiran Tasikmalaya, 13 November. Perusahaan jasa angkuran 4848 Irawan Sarpingi bergerak pada bidang usaha angkutan darat, lebih tepatnya pada angkutan jalan raya. Dalam pelaksanaan usahanya, perusahaan 4848 Irawan Sarpingi berorientasi pda usaha menjual jasa kepada masyarakat umum. Pelayanan jasa yang ditawarkan adalah angkuran penumpang (travel) dalam dan luar kota serta pengantaran paket dan barang (angkutan barang), Nama perusahaan yang berupa deretan angka 4 dan 8 memiliki arti sendiri, yaitu diambil dari awal berdirinya perusahaan tanggal 4 bulan 8 (Agustus).

Dengan bermodalkan sebuah kendaraan serta tekad yang kuat H.Irawan Sarpingi memulai usahanya dengan mengambil tempat sebagai kantor di jalan Trunojoyo Nomor 20 Bandung. Pada waktu itu perusahaan 4848 Irawan Sarpingi hanya memiliki seorang supir yaitu Bapak Hamidin dan seorang karyawan bernama Aen Karnen. Sebagai perusahaan yang masih baru, maka hal-hal baru ditawarkan oleh perusahaan 4848 Irawan Sarpingi. Salah satu pelayanan yang banyak disambut baik oleh masyarkat adalah penyanan antar-jemput penumpang dan barang langsung dari tempat penumpang atau pengirim barang ke alamat tujuan yang dinginkan oleh penumpang atau pengirim barang

Perusahaan jasa angkutan 4848 Irawan Sarpingi resmi menjadi badan usaha berdasarkan akta Notaris Koswara, SH Nomor 31 Tanggal 8 April 1871. Dengan akta pendirian perusahaan tersebut, nama perusahaan menjadi PT.4848 Irawan Sarpingi. Pada tanggal 20 Januari 1981 terjadi perubahan nomor akta pendirian

perusahaan menjadi Nomor 50 dengan Notaris yang sama Tanggal 23 April 1981, PT 4848 Irawan Sarpingi memperoleh izin usaha dari Dinas Pariwisata Pos dan Telekomunikasi dengan Nomor 351/DIP/BIN/PP.1981. Tanggal 23 Febuari 1988, PT Irawan Sarpingi memperoleh izin usaha jasa titipan barang sebagai pengembangan usahanya dari Direktorat Jendral Pos dan Tekekomunikasi No.087/SIUJT/DIRJEN/1988. Kemudian pada tanggal 15 Mei 1997dilakukan penyesuaian izin usaha oleh PT.4848 Irawan Sarpingi berdasarkan keputusan DIrektorat Jendaral Pos dan Telekomunikasi Nomor 108/SIPJT/DIRJEN/1997 yang ditetapkan di Jakarta.

Atas Permintaan masyarakat, PT4848 Irawan Sarpingi membuka cabang dan agen di berbagai kota di pulau Jawa, Bali dan Sumatra. Hal tersebut dimaksudkan untuk melayani kebutuhan masyarakat yang semakin meningkat terhadap jasa angkutan. Sebagai kantor pusat, PT4848 Irawan Sarpingi memperoleh penghargaan "Adhi Karya Pariwisata dan Perintis Pembangunan Pariwisata Jawa Barat" dari Departemen Pariwisata Pos dan Telekomunikasi yang diserahkan oleh Gubernur Jawa Barat.

2.1.2 Visi dan Misi PT.4848 Irawan Sarpingi

2.1.2.1 Visi

Pada awal berdirinya, PT.4848 Irawan Sarpingi hanya menjalankan usahanya sebatas pekerjaan biasa. Dengan berkembangnya kegiatan usaha tersebut PT.4848 Irawan Sarpingi menyadari bahwa pentingnya sarana angkutan semakin meningkat, sehingga perlu diperhatikan pengadaan sarana angkutan tersebut untuk melayani kebutuhan masyarakat. Inilah yang menjadi visi bagi PT.4848 Irawan Sarpingi.

2.1.2.2 Misi

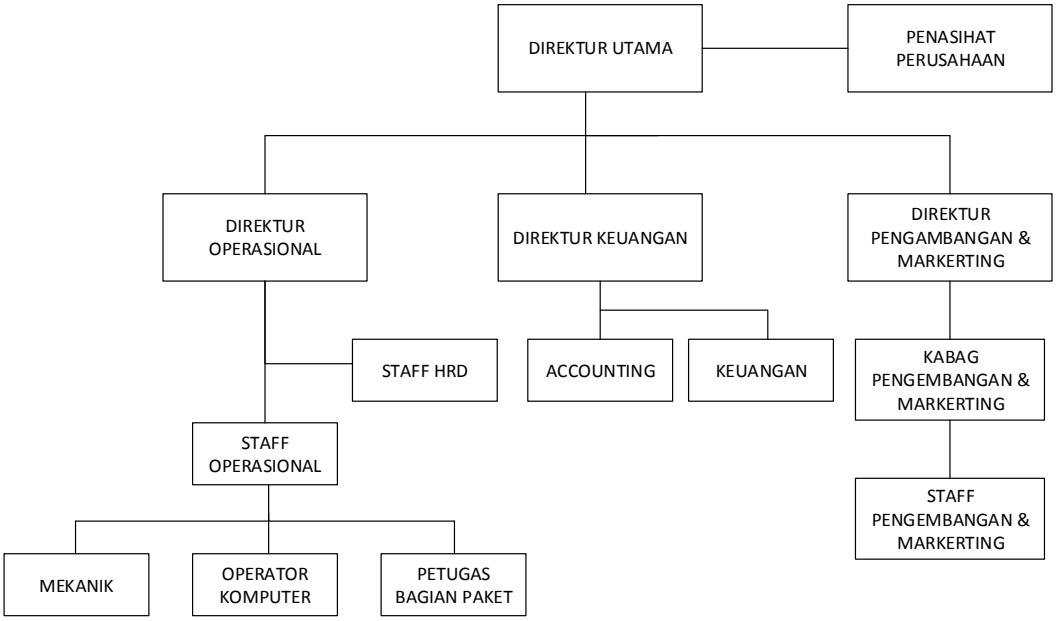
Didalam pelaksanaan kegiatan usahanya, PT.4848 IRAWAN SARPINGI mengemban misi :

1. Menyediakan dan memenuhi permintaan dan melayani masyarakat luas yang memerlukan jasa angkutan barang dan penumpang.
2. Membantu program pemerintah dalam menciptakan lapangan kerja, baik secara langsung maupun tidak langsung.

3. Membantu program pemerintah dibidang angkutan penumpang dan barang

2.1.3 Struktur Organisaasi PT.4848 Irawan Sarpingi

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 perusahaan, yang mana dalam hal ini merupakan salah satu syarat terciptanya suatu pengendalian internal yang memadai. Adapun struktur organisasi PT.4848 Irawan Sarpingi dapat dilihat seperti pada Gambar 2-1.



Gambar 2-1 Struktur Organisasi

2.1.4 *Job description* PT.4848 Irawan Sarpingi

Untuk melengkapi struktur organisasi suatu perusahaan, diperlukan uraian tugas yang akan menjelaskan tentang wewenang dan tanggung jawab dari masing-masing fungsi dalam perusahaan. Uraian tugas pada PT.4848 Irawan Sarpingi adalah sebagai berikut :

1. Penasihat Perushaan
    a) Memberikan pengarahan dan nasihat, gagasan dan pengembangan serta saran kepada direksi perusahaan.

b) Mengawasi jalannya perusahaan agar tidak menyimpang dari peraturan perusahaan yang berlaku.
c) Menjaga stabilitas dan semangat kekeluargaan yang berlandaskan peraturan perusahaan, serta menjaga citra dan wibawa perusahaan.

2. Direktur Utama
   a) Bertanggung jawab penuh untuk memastikan perusahaan beroperasi dengan efektif, aman secara ekonimi sangat layak serta menguntungkat dalam usaha penyediaan lanyanan jasa transportasi bagi masyarakat di Bandung dan sekitarnya , dengan menghadirkan solusi lanyanan jasa transportasi dengan kualitas lanyanan standar nasional.
   b) Memberikan pengarahan dan memimpin tim management dalam kegiatan operasi perusahaan, memastikan kepemimpinan dan motivasi seluruh karyawan dapat dioptimumkan sehingga kehormonisan suasana kerja dapat terlaksana dengan baik.
   c) Melaksanakan usaha karyawan yang efesien dan ekonomis dengan mengkoordinasikan secara efektif seluruh kegiatan secara baik dan mempertahankan image perusahaan.
   d) Memastikan seluruh asset dan operasi perusahaan berjalan dengan lancar dan memperhatikan keselamatan dan kesehatan kerja sesuai dengan standar nasional.
   e) Menyetujui anggaran belanja perusahaan dan memastikan pengawasan yang berkesinambungan dari anggaran tersebut dengan melakukan usaha penghematan biaya dengan melakukan pengukuran indikantor keberhasilan.
   f) Secara berkala mengadakan penilaian terhadap manfaat dan efesiensi dan sistem dan prosedur kerja yang berkaitan erat dengan perusahaan dan melakukan penyempurnaan.

3. Direktur Operasional
   a) Bertanggung jawab langsung atas pelaksanaan tugas di wilayah kerjanya kepada direktur utama.

b) Menyusun perencanaan, melakukan pembinaan dan pengawasan terhadap kegiatan bawahan dilingkungan perusahaan.
c) Melakukan kegiatan-kegiatan komersil dengan jalan membina hubungan baik dengan para pengguna jasa perusahaan.
d) Melakukan pengawasan dan penilaian secara menyeluruh terhadap bawahaanya serta memelihara suasana kerja yang baik dalam organisasi perusahaan.
e) Melakukan inisiasi dan pengarahan untuk membangun formulasi yang relevan, baik untuk jangka panjang maupun jangka pendek.
f) Menerima instruksi langsung dari direktur utama untuk dilaksanakan sendiri atau diteruskan kepada bawahan.
g) Menjamin penyelenggaraan suatu jasa penyerahan baik pada konsumen secara efisien di dalam wilayah.
h) Menjalani koordinasi dengan kepala koordinator kantor cabang, dalam kegiatan oprasional perusahaan.

4. Direktur Keuangan
    a) Bertanggung jawab langsung atas pelaksanaan, tugas di lingkungan kerja kepada direktur utama.
    b) Mempelajari rencana operasi perusahaan dalam rangka menyusun, melaksanakan, dan mengendalikan rencana anggaran pendapatan dan belanja.
    c) Mengawasi pelaksanaan pembayaraan gaji dan utang piutang serta menyelesaikan masalah perbankan.
    d) Mengendalikan pemakaian keuangan sesuai dengan sistem prosedur yang berlaku.
    e) Menyusun laporan bulanan, triwulan, dan tahunan dalam bentuk rugi/laba perusahaan.
5. Bagian Accounting
    a) Menerima bukti transaksi dari kantor pusat, kantor cabang dan agen.
    b) Membuat catatan adminitrasi atas kegiatan yang terjadi.
    c) Membuat juarnal kerja.

d) Membuat laporan keuangan perusahaan.

6. Bagian Keuangan
    a) Menginput setoran dari semua kantor cabang dan semua agen.
    b) Membuat laporan kas harian kantor pusat dan kantor cabang.
    c) Mengeluarkan cek.
    d) Membuat jadwal pembayaran.
    e) Membuat laporan pengajuan pengeluaran kas harian dan mingguan.
7. Kabag Pengembangan dan Markerting
    a) Menganalisa program kerja oprasional yang sudah teralisasi maupun belum terealisasi.
    b) Menganalisa dan mengembangkan strategi marketing untuk meningkatkan jumlah pelanggan dan area sesuai dengan target yang ditentukan .
    c) Melakukan evaluasi data yang dihasilkan untuk memastikan tercapainya target kerja.
    d) Berkoordinasi dengan bagian operasional dan keuangan.
8. Staff Operasional
    a) Merekap pendapatan paket percabang, kantor cabang bandung dan Jakarta.
    b) Merekap pendapatan jurusan timur.
    c) Memeriksa laporan seluruh resi dan manifest paket dari Bandung ke Jakarta.
    d) Memeriksa laporan seluruh resi dan manifest dari Jakarta ke Bandung.
    e) Membuat grafik laporan paket.
9. Staff HRD
    a) Merencanakan dan mengembangkan kebijakan dan sistem pengelolaan SDM, serta mengkoordinasikan dan mengontrol pelaksanaan fungsi manajemen SDM di seluruh perusahaan agar dapat menunjang dan meningkatkan kinerja SDM dalam mencapai target perusahaan.

b) Menyusun strategi dan kebijakan pengelolaan SDM perusahaan berdasarkan strategi jangka panjang dan jangka pendek yang telah ditetapkan sesuai dengan peraturan pemerintah yang berlaku agar diperoleh SDM dengan kinerja kapabilitas dan kompetensi yang sesuai dengan yang diinginkan perusahaan.
c) Menyusun rencana kerja dan anggaran bagiannya sesuai dengan strategi kebijakan dan sistem SDM yang telah ditetapkan untuk memastikan tercapainya sasaran bagian SDM
d) Mengkooridnasikan dan mengontrol pelaksanaan fungsi SDM di seluruh perusahaan untuk memastikan semuanya sesuai dengan strategi, kebijakan, sitem dan rencana kerja yang telah disusun.
e) Merencanakan kebutuhan tenaga kerja yang dibutuhkan sesuai dengan permintaan dan kualifikasi yang diinginkan dalam jangka waktu yang telah disepakati
f) Mengelola dan mengontrol aktifitas administrasi kantor, kepersonaliaan dan sistem operasi SDM untuk memastikan tersedianya dukungan yang optimal bagi kelancaran operasional perusahaan.

10. Staff Pengembangan dan Marketing
    a) Membuat laporan analisa yang sudah terealisasi maupun belum terealisasi.
    b) Membuat laporan evaluasi data hasil yang telah mencapai target kerja.
    c) Membuat rencana kerja.
11. Mekanik
    a) Menerima komplain dan kerusakan kendaraan
    b) Mendeteksi dan menganalisa sumber kerusakan kendaraan.
    c) Melakukan perbaikan kerusakan kendaraan sesuai target waktu.
12. Operator Komputer
    a) Membuka e-mail ager terus online.
    b) Selalu membuka yahoo messenger untuk keperluan komunikasi.

c) Menginput data resi yang telah diterima.
d) Membuat manifest berdasarkan resi paket yang diterima untuk mobil 4848 eksekutif, manifest dibuat perkantor dan pada worksheet yang berbeda.
e) Manifest yang telah sesesai dibuat dikirim ke kantor pusat melalui e-mail.
f) Membuat laporan keberangkatan mobil 4848 eksekutif.

13. Petugas Bagian Paket
a) Menerima dan melayani pelanggan yang akan mengirim paket.
b) Bertanggung jawab penuh pada paket yang telah diserahkan terimakan dari pelanggan baik dalam penyimpanan ataupun pemuatan paket ke mobil pengangkutan.
c) Menerima, dan menyorti paket yang datang dari Jakarta/Bandung berdasarkan wilayah pengiriman paket serta menunjuk kurir pengantar paket
d) Membuat laporan harian pengiriman paket oleh kurir termasuk menghitung presentase untuk kurir berdasarkan jumlah resi pengiriman yang diserahkan kurir.
e) Membuat laporan harian penerimaan paket secara manual meskipun sudah ada program komputer.
f) Menjaga kesinambungan tugas dengan petugas pada shift sebelum dan berikutnya.

2.2 Landasan Teori

Landasan teori yang digunakan dalam penyusunan laporan tugas akhir dan pembangunan Web *Visual Map Trcaking* Pada Jasa Angkutan Barang di PT.4848 Irwan Sarpingi adalah sebagai berikut :

2.1.1 Android

Android merupakan sistem operasi mobile yang tumbuh di tengah sistem operasi lainnya yang berkembang dewasa ini. Sistem operasi lainnya seperti Windows Mobile, i-Phone OS, Symbian dan masih banyak lagi juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat hardware yang ada. Akan

tetapi, sistem operasi yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

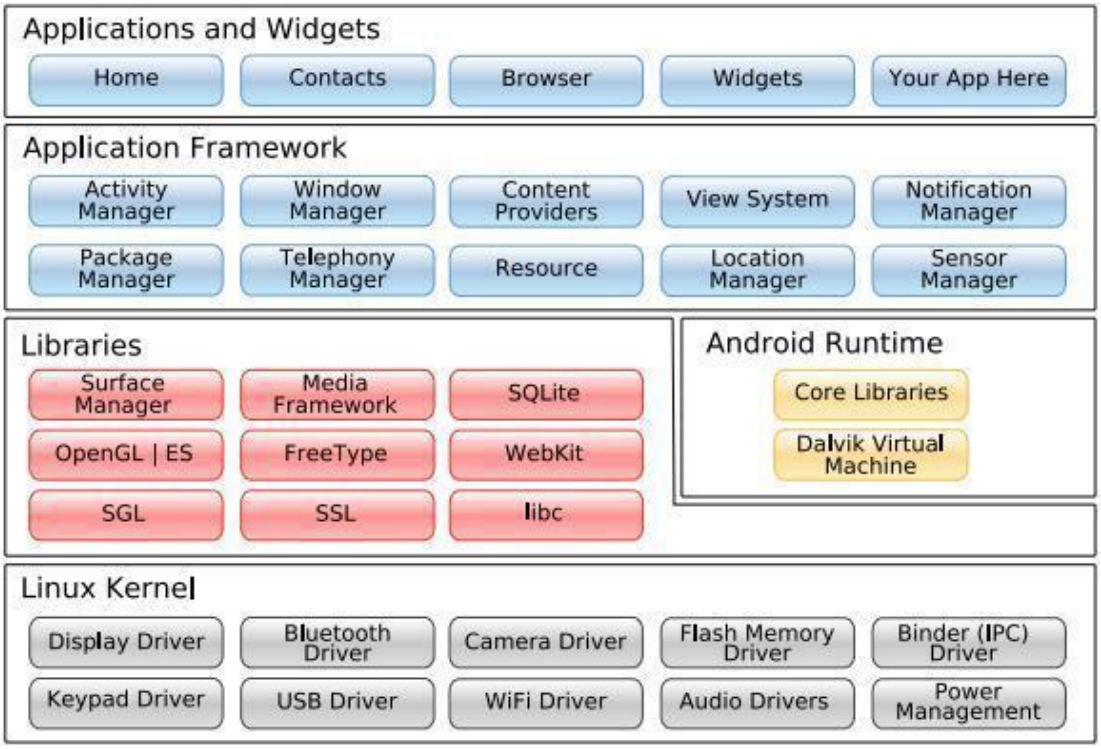
Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. Application Programming Interface (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga. [1]

2.1.1.1 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 2-2 Arsitektur Android

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

b. Aplication 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 [1]

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. [1]

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 yangberfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh Core Libraries.

2. Dalvik Virtual Machine: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat Linux Kernel untuk melakukan threading dan manajemen tingkat rendah. [1]

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. [1]

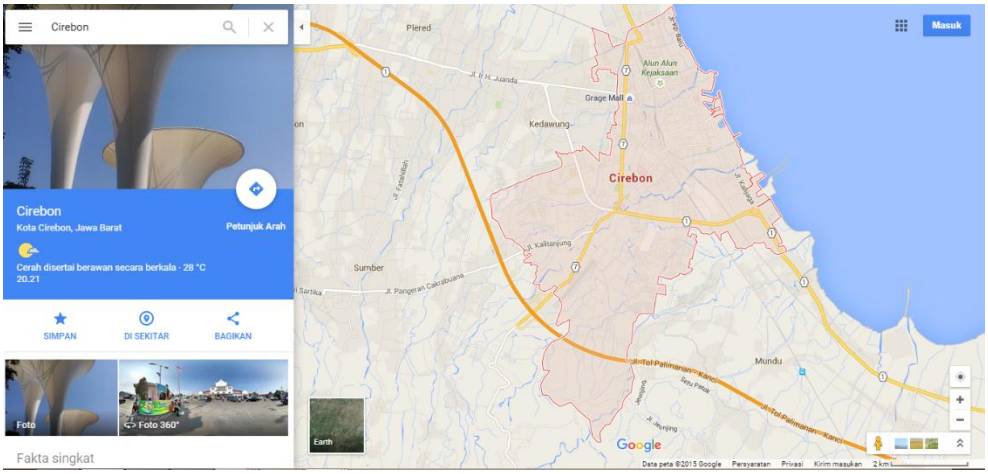
2.1.2 Google Maps

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 popular. 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 mambuat peta suatu lokasi, bahkan dunia.



Gambar 2-3 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 diatasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota). [2]

2.1.3 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. [2]

2.1.3.1 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. [2]

2.1.3.2 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. [2]

2.1.3.3 Latitude dan Longitude

Latitude dan longitude adalah sebuah koordinat yang ada di bumi dan digunakan untuk menentukan kedudukan kita di atas bumi. [3]

1. Latitude

Latitude adalah garis yang melintang dari kutub utara dan kutub selatan. Titik 0 adalah sudut ekuator, tanda + menunjukan 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. [3]

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. [3]

2.1.4 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 Pemprograman 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*).

2.1.5 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 .

2.1.6 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 penggunanya [6].

2.1.7 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*.

2.1.8 MySQL

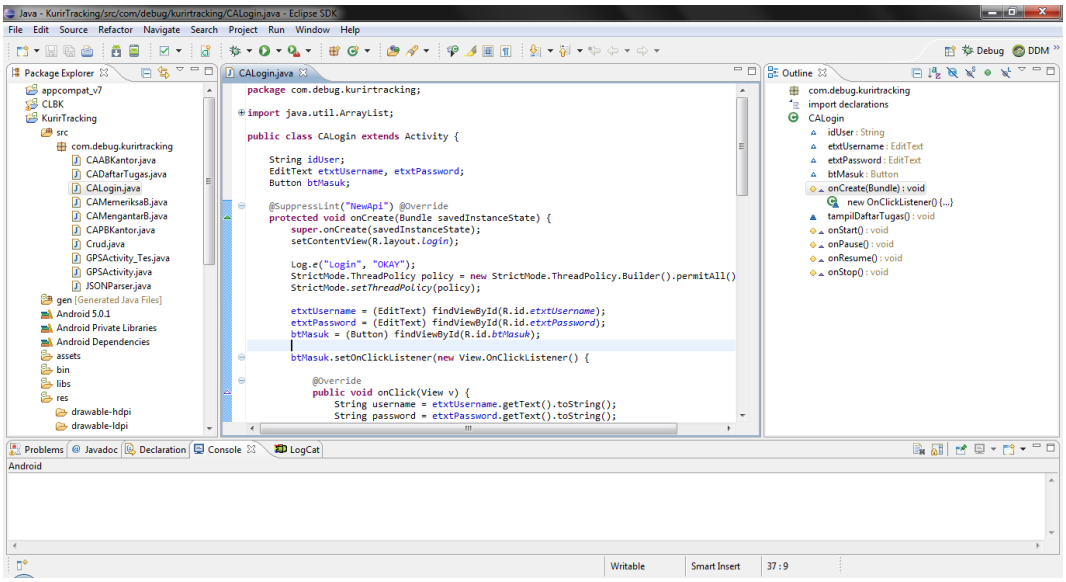
MySQL termasuk dalam kategori database management system, yaitu suatu database yang terstruktur dalam pengolahan dan penampilan datanya. MySQL merupakan database yang besifat client server, dimana data diletakan di server yang bisa diakses melalui komputer client [7].

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.

2.1.9 Eclipse

Eclipse adalah sebuah project Java open source yang diprakarsai oleh IBM, di mana sebelumnya merupakan hasil kerja OTI, sebuah perusahaan yang diakusisi oleh IBM karena keahliannya membuat aplikasi yang dikenal dengan *VisualAge*. Eclipse dapat dikatakan adalah *VisualAge* versi baru. Perbedaannya *VisualAge* bersifat komersial, sedangkan Eclipse bersifat *open source* dengan lisensinya CPL (*Common Public License*). Selain itu, *Eclipse* juga memposisikan sebagai IDE multilanguage, jadi selain bahasa Java, juga mendukung bahasa C++, *Cobol*, *Fortran*. Dapat dikatakan Eclipse ini menjadi sebuah *single environment* untuk pengembangan aplikasi yang berjalan di banyak platform [8].



Gambar 2-4 Tampilan Aplikasi *Eclipse*

2.1.10 XAMPP

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

2.1.11 Konsep Pemodelan UML 2.0

Unifield 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 [9].

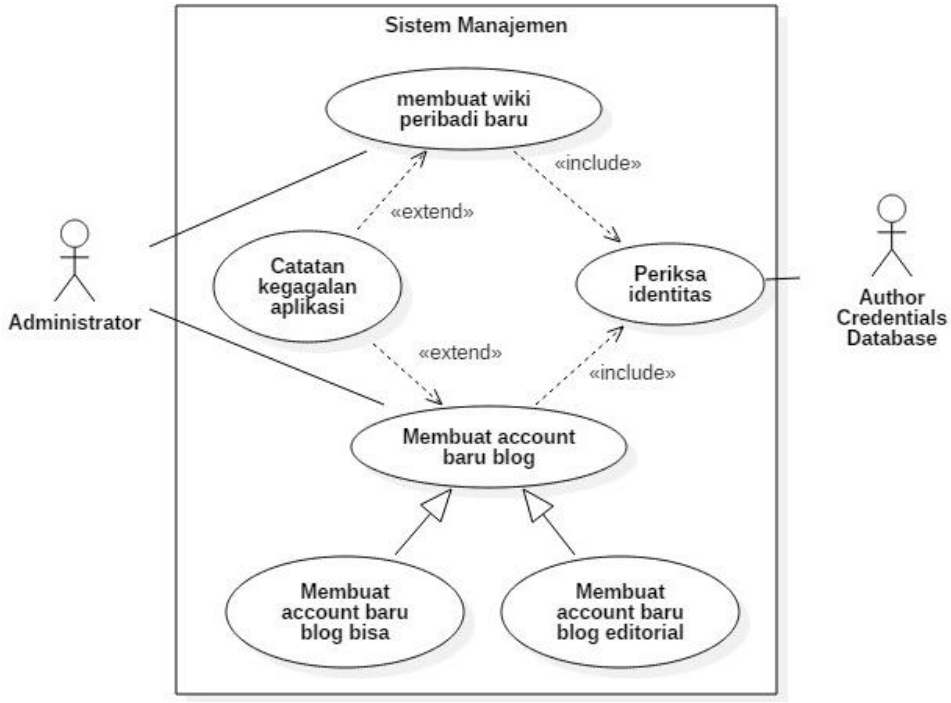
Untuk dapat mengerti konsep pemodelan UML, dibutuhkan pemahaman dari beberapa diagram yang akan digunakan dalam pemodelan dari sistem yang akan dibangun. Ada dua diagram yang harus difahami yang diantaranya adalah *structural diagram* dan *behavioral diagram. Structural diagram* digunakan untuk mendeskripsikan 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.*

2.1.12 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 sitem lakukan tanpa mendeskripsikan bagaimana sistem menyelesaikannya, *use case* diagram dibuat berdasarkan interaksi dan relasi dari individual *use case* dan berisi *actor, even*t, 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

Sumber : Russ Miles Kim Hamilton, Buku Learning UML 2.0, O'Reilly Media, 2006

2.1.12.1 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 menunjukan bahwa *use case* asal secara eksplisit memasukan perilaku dari *use case* lain yang ditunjuk oleh *uce 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* menunjukan 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.

2.1.12.2 Use Case Scenario

Use case scenario merupakan hasil instansi dan penjelasan dari setiap *use case*. Ada beberapa komponen yang perlu dipahami dalam *use case scenario,* yang diantaranya adalah:

1. *Name,* memberikan penjelasan singkat tentang nama *use case.*
2. *Actors,* daftar aktor yang dapat mengakses *use case.*
3. *Goals,* menjelaskan apa yang aktor coba untuk dapatkan dari *use case.*
4. *Precondition,* kondisi sistem sebelum *use case* dijalankan.
5. *Summary,* memberikan penjelasan singkat tentang deskripsi informal sesuai use case.
6. *Related use case,* daftar *use case* yang berhubungan dengan *use case* tersebut.
7. *Steps,* menjelaskan setiap langkah yang dijalankan pada *use case* tersebut.
8. *Post condition,* kondisi sistem setelah *use case* dijalankan.

Use case name	Membuat account baru blog editorial	
Related Requirements	Requirement A.1.	
Goal In Context	Penulis baru atau permnitaan account baru blog editorial oleh administrator	
Preconditions	Penulis memiliki kesesuaian bukti identitas	
Succesfull End Condition	Account baru editorial dibuat untuk penulis	
Failed End Condition	Aplikasi gagal membuar Account baru editorial	
Primary Actors	Administrator	
Secondary Actors	None	
Trigger	Administrator meminta CMS untuk membuar akun editorial baru yang akan memungkinkan untuk mengedit entri dalam satu set blog	
Base Use Cases	Membuat account baru blog editorial	
Main Flow	Step	Action
	1	Meminta administrator sistem membuat account baru blog
	2	Administrator memilih jenis editorial account
	3	Administrator memasukan data detail penulis
	4	Administrator memilih blog yang memiliki hak akses editorial account
	5 Include :: Periksa identitas	Memeriksa detail data penulis
	6	Editorial account baru dibuat
	7	Ringkasan rincian rekening editorial baru diemail ke penulis
Extensions	Step	Branching Action
	5.1	Penulis tidek diperbolehkan untuk mengedit blog yang ditunjukan
	5.2	Aplikasi menolah account editorial

Gambar 2-6 Contoh Use Case Scenario

Sumber : Russ Miles Kim Hamilton, Buku Learning UML 2.0, O'Reilly Media, 2006

2.1.12.3 Activity Diagram

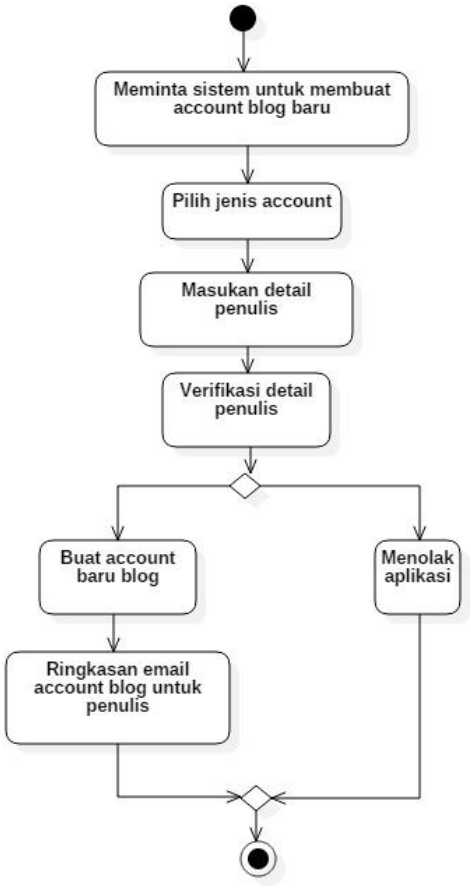
Activity diagram menjelaskan bagaimana sistem yang dibangun mencapai tujuannya, dalam pembangunan sistem ini *activity diagram* dgunakan untuk menjelaskan setiap *use case* yang dibangun, acuan pembuatan dari *activity diagram* dengan melihat *step* dan *action* dari *use case scenario*, ada beberapa 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 *acvtivy* 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 2-7 Contoh Activity Diagram

Sumber : Russ Miles Kim Hamilton, Buku Learning UML 2.0, O'Reilly Media, 2006

2.1.12.4 Class Diagram

Class diagram adalah diagram yang menunjukan *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:

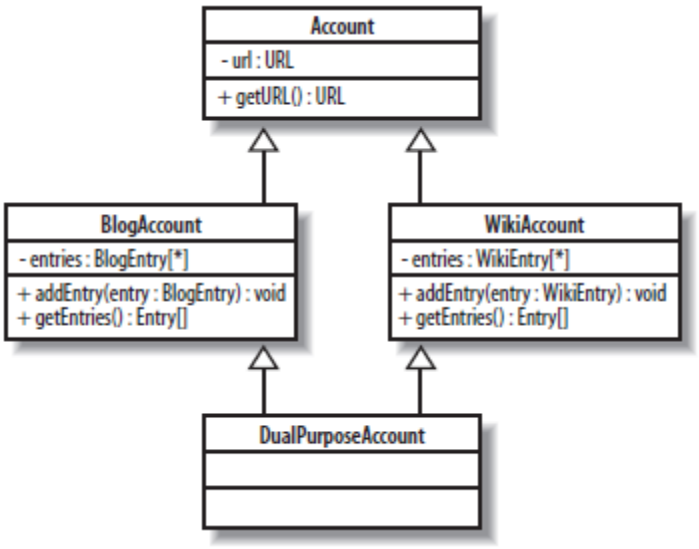
1. *Private,* variabel atau method yang hanya dapat diakses oleh class itu sendiri.
2. *Protected,* variabel atau method dapat digunakan oleh class turunan dari *class parent*.
3. *Public,* variabel atau method dapat digunakan oleh seluruh class.

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. *Depedency,* digunakan jika sebuah *class* menggunakan *class* yang lain, pengguna *dependency* digunakan untuk menunjukan 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.

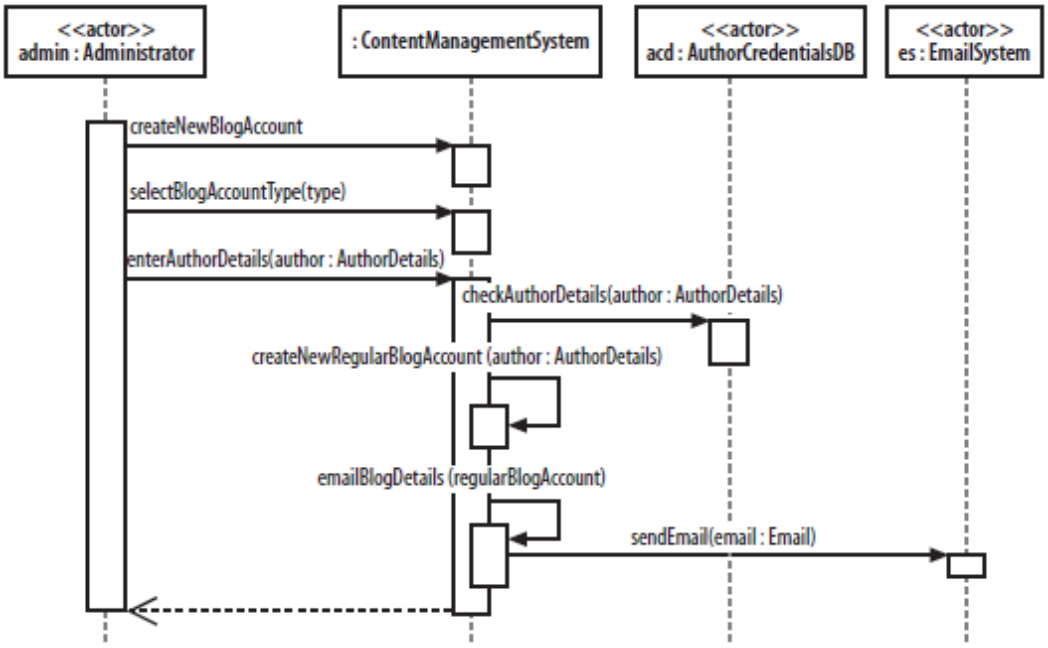


Gambar 2-8 Contoh Class Diagram

Sumber : Russ Miles Kim Hamilton, Buku Learning UML 2.0, O'Reilly Media, 2006

2.1.12.5 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 2-9 Contoh Sequence Diagram

Sumber : Russ Miles Kim Hamilton, Buku Learning UML 2.0, O'Reilly Media, 2006