BAB 2
TINJAUAN PUSTAKA

2.1 Profil DPPKAD Kabupaten Subang

Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kabupaten Subang atau bias disingkat DPPKAD Kabupaten Subang adalah suatu dinas yang bergerak dalam bidang pendapatan, pengelolaan keuangan daerah, serta pengelolaan aset daerah di kabupaten Subang yang beralamat di Jalan S. Parman no. 3 Subang 41211 dengan nomor telepon (0260) 411413.

2.1.1 Sejarah DPPKAD Kabupaten Subang

Sejarah singkat berdirinya Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kabupaten Subang berdasarkan (1) Undang-Undang No. 18 Tahun 1997 tentang Pajak Daerah dan Retibusi Daerah; (2) Undang-Undang No. 19 Tahun 1997 tentang Penagihan Pajak dengan surat paksa sebagaimana telah diubah dengan Undang-Undang No. 19 Tahun 2000; (3) Undang-Undang No. 17 Tahun 2003 tentang Keuangan Negara; (4) Undang-Undang No. 1 Tahun 2004 tentang Pembendaharaan Negara; (5) Undang-Undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah; (6) Peraturan Pemerintah Nomor 41 Tahun 2007 tentang Organisasi Perangkat Daerah; (7) Peraturan Daerah Kabupaten Subang Nomor : 4 Tahun 2008 tentang Pola Organisasi Perangkat Daerah Kabupaten Subang; (8) Peraturan Daerah Nomor 7 Tahun 2008 tentang Organisasi Dan Tata Kerja Dinas Daerah Di Lingkungan Pemerintah Kabupaten Kabupaten Subang.

2.1.2 Visi dan Misi DPPKAD Kabupaten Subang

Visi dan misi dari Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kabupaten Subang.

2.1.2.1 Visi

Terwujudnya pengelolaan keuangan daerah yang dinamis dan akuntabel dalam menunjang visi dan misi Kabupaten Subang.

2.1.2.2 Misi

Adapun misi dari DPPKAD Kabupaten Bandung sebagai berikut :

1. Meningkatkan Pendapatan Daerah.
2. Tertibnya penyusunan anggaran.
3. Terciptanya pelayanan administrasi keuangan.
4. Tersusunnya laporan keuangan yang akuntabel.
5. Tertatanya Aset Daerah.
6. Meningkatnya SDM yang professional.

2.1.3 Logo DPPKAD Kabupaten Subang

Logo Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kabupaten Subang sebagai tempat penelitian dilaksanakan dapat dilihat pada Gambar 2.1.



Sumber Gambar : Pemerintahan Kabupaten Subang [6]

Gambar 2. 1 Logo DPPKAD Kabupaten Subang

Logo Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kabupaten Subang yang terdapat pada Gambar 2.1 menggambarkan karakteristik dari kabupaten Subang tempat penelitian dilaksanakan. Makna dari logo Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah akan diuraikain seperti dibawah ini.

A. Arti Bentuk Logo

Bentuk dari logo Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah memiliki arti sebagai berikut :

1. Perisai bersudut lima menggambarkan makna keselamatan negara, bangsa, agama, masyarakat, dan agama.
2. Pohon Beringin Bergelombang 17 Dengan Akar Tunjang Delapan menggambarkan aspek sejarah Kabupaten Subang (Kutawaringin); kesatuan bangsa yang berjiwa Pancasila dan semangat Proklamasi 17 Agustus 1945; pemerintahan sebagai pelindung rakyat; dan pelaksanaan pembangunan daerah bidang material maupun spiritual.
3. Benteng Berkepala Lima Serta Benteng Bagian Bawah Berbata Empat dan Lima di Bawah Pohon Beringin menggambarkan Pancasila sebagai landasan idiil dan Undang-undang Dasar 1945 yang berkaitan pula dengan makna pembangunan material dan spiritual.
4. Bintang Kuning Bersudut Lima menggambarkan karakteristik masyarakat Kabupaten Subang yang selalu bertaqwa kepada Tuhan yang Maha Esa dan mengangungkan agama.
5. Tulisan “Benteng Pancasila” menggambarkan warga Kabupaten Subang yang senantiasa membentengi Pancasila sebagai landasan idiil negara dari pihak-pihak yang akan menyelewengkannya. Tulisan ini juga menggambarkan tekad masyarakat Kabupaten Subang untuk menjadikan Pancasila sebagai benteng mental dalam mencapai masyarakat adil dan makmur yang diridoi Tuhan yang Maha Esa.
6. Tulisan “Karya Utama Satya Negara” menggambarkan keutamaan karya untuk kepentingan negara, bangsa, dan agama.

B. Arti Warna Logo

Warna dari logo Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah memiliki arti sebagai berikut :

1. Kuning mas pada pinggir perisai, pinggir pohon beringin, dan garis pinggir benteng, serta bintang Menggambarkan keluhuran budi dan kebesaran jiwa.
2. Hijau tua pada dasar perisai Menggambarkan kesuburan tanah.
3. Coklat Menggambarkan kawasan pedataran.
4. Hijau muda Menggambarkan kawasan pegunungan.
5. Biru Menggambarkan kawasan pantai.

2.1.4 Tugas Pokok dan Fungsi

Tugas pokok dari Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah yaitu melaksanakan kewenangan Daerah di bidang pendapatan, pengelolaan keuangan dan aset daerah, yang meliputi :

a. Menyusun dan melaksanakan kebijakan pengelolaan keuangan daerah;
b. Menyusun rancangan APBD dan Perubahan APBD;
c. Melaksanakan pemungutan pendapatan daerah yang telah ditetapkan dengan Peraturan Daerah;
d. Melaksanakan fungsi BUD;
e. Menyusun laporan keuangan daerah dalam rangka pertanggungjawaban pelaksanaan APBD;
f. Melaksanakan tugas pembantuan yang diberikan oleh Pemerintah atau Pemerintah Propinsi Jawa Barat.

Serta memiliki fungsi diantara lain :

a. Perumusan kebijaksanaan dan pedoman pelaksanaan APBD;
b. Pengesahan DPA SKPD/DPPA SKPD;
c. Pengendalian pelaksanaan APBD;
d. Pemberian petunjuk teknis pelaksanaan sistem penerimaan dan pengeluaran kas daerah;
e. Pelaksanaan pemungutan pajak daerah;

f. Penetapan SPD;
g. Penyiapan pelaksanaan pinjaman dan pemberian pinjaman atas nama pemerintah daerah;
h. Pelaksanaan sistem akutansi dan pelaporan keuangan daerah;
i. Penyajian informasi keuangan daerah;
j. Pelaksanaan kebijakan dan pedoman pengelolaan serta penghapusan barang milik daerah;
k. Pengkoordinasian dan pengawasan atas pekerjaan pendapatan daerah;
l. Pengelolaan administrasi umum, meliputi urusan umum, urusan keuangan, urusan kepegawaian dan perlengkapan dinas.

2.1.4.1. Struktur Organisasi Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kabupaten Subang

Struktur organisasi merupakan pengelompokkan pekerjaan berdasarkan jabatan yang dimiliki oleh masing – masing orang di sebuah pemerintahan. Setiap pemerintahan memiliki struktur organisasi yang berbeda – beda disesuaikan dengan bidangnya, salah satunya adalah Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kabupaten Subang. Struktur organisasi yang terdapat di DPPKAD Kabupaten Subang dapat dilihat dibawah ini.

Kepala Dinas membawahkan :

1. Sekretariat membawahkan :
    a. Sub Bagian Umum
    b. Sub Bagian Keuangan
    c. Sub Bagian Kepegawaian
2. Bidang Pendapatan Asli Daerah membawahkan :
    a. Seksi Perhitungan, Penetapan & Angsuran
    b. Seksi Penagihan Keberatan dan Laporan
3. Bidang Perimbangan membawahkan :
    a. Seksi Bagi Hasil Pajak
    b. Seksi Bagi Hasil Bukan Pajak
    c. Seksi Pendapatan Lain - Lain

4. Bidang Anggaran membawahkan :
    a. Seksi Penyusunan Anggaran Bidang Program
    b. Seksi Evaluasi dan Pengendalian
5. Bidang Akutansi membawahkan :
    a. Seksi Akutansi Pendapatan
    b. Seksi Akutansi Belanja
6. Bidang Perbendaharaan membawahkan :
    a. Seksi Pengelolaan Kas
    b. Seksi Pengelolaan Belanja
7. Bidang Aset dan Investasi membawahkan :
    a. Seksi Aset Bergerak
    b. Seksi Aset Tidak Bergerak
    c. Seksi Investasi

2.2 Landasan Teori

Landasan Teori merupakan definisi, konsep yang telah disusun secara sistematis dan dasar yang kuat dalam sebuah penelitian. Landasan teori yang digunakan dalam penyusunan Pembangunan Aplikasi Monitoring Aset Bergerak Menggunakan Teknologi GPS di Dinas Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kabupaten Subang meliputi Aplikasi Monitoring, Pengertian Aplikasi, Pengertian Monitoring, Pengertian Aplikasi Monitoring, Karakteristik Perangkat Mobile, Android, Basis Data, DBMS, SQL, Pemrograman Berorientasi Objek, Konsep Dasar Berorientasi Objek, Geofencing, Jaringan Internet, GPS, Google Maps, Android Studio, PHP, JavaScript, JSON, Database MySQL, *Unified Modeling Language* (UML), Web Server dan Web Service.

2.2.1 Aset Bergerak

Menurut Prof. Subekti, S.H. dalam bukunya, pada sisi lain masih menurut Subekti, suatu benda dihitung termasuk golongan benda yang bergerak karena sifatnya atau karena ditentukan oleh undang-undang. Suatu benda yang bergerak karena sifatnya ialah benda yang tidak tergabung dengan tanah atau dimaksudkan untuk mengikuti tanah atau bangunan, jadi misalnya barang perabot rumah tangga. Tergolong benda yang bergerak karena penetapan undang-undang ialah misalnya *vruchtgebruik* dari suatu benda yang bergerak, lijfrenten, surat-surat sero dari suatu perseroan perdagangan, surat-surat obligasi negara, dan sebagainya. [7]

2.2.2 Aplikasi Monitoring

Aplikasi monitoring pada saat ini sangat dibutuhkan untuk memantau suatu benda ataupun kegiatan karena dapat dapat mempermudah pekerjaan penggunanya yang dimanapun atau kapanpun terutama dalam hal penyampaian informasi. Aplikasi ini dapat melakukan proses pengumpulan dan menganalisis informasi dari penerapan suatu program termasuk mengecek secara reguler untuk melihat apakah kegiatan/program itu berjalan sesuai rencana sehingga masalah yang dilihat /ditemui dapat diatasi.

2.2.3 Pengertian Aplikasi

Menurut Buyens [8] aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas. Jika ingin mengembangkan program aplikasi sendiri, maka untuk menulis program aplikasi tersebut, dibutuhkan suatu bahasa pemrograman, yaitu language software, yang dapat berbentuk assembler, compiler ataupun interpreter. Jadi language software merupakan bahasanya dan program yang ditulis merupakan program aplikasinya. Language software berfungsi agar dapat menulis program dengan bahasa yang lebih mudah, dan akan menterjemahkannya ke dalam bahasa mesin supaya bisa dimengerti oleh komputer. Bila hendak mengembangkan suatu program aplikasi untuk memecahkan permasalahan yang besar dan rumit, maka supaya program aplikasi tersebut dapat berhasil dengan baik, maka dibutuhkan prosedur dan perencanaan yang baik dalam mengembangkannya.

Sekarang, banyak sekali program-program aplikasi yang tersedia dalam bentuk paket-paket program. Ini adalah program-program aplikasi yang sudah ditulis oleh orang lain atau perusahaan-perusahaan perangkat lunak. Beberapa perusahaan perangkat lunak telah memproduksi paket-paket perangkat lunak yang mempunyai reputasi internasional. Program-program paket tersebut dapat diandalkan, dapat memenhi kebutuhan pemakai, dirancang dengan baik, relatif bebas dari kesalahan-kesalahan, user friendly (mudah digunakan), mempunyai dokumentasi manual yang memadai, mampu dikembangkan untuk kebutuhan mendatang, dan didukung perkembangannya. Akan tetapi, bila permasalahannya bersifat khusus dan unik, sehingga tidak ada paket-paket program yang sesuai untuk digunakan, maka dengan terpaksa harus mengembangkan program aplikasi itu sendiri.

2.2.4 Pengertian Monitoring

Monitoring merupakan program yang terintegrasi, bagian penting dipraktek manajemen yang baik dan arena itu merupakan bagian integral di manajemen sehari-hari atau dapat disebut juga sebagai suatu proses mengukur, mencatat, mengumpulkan, memproses dan mengkomunikasikan informasi untuk

membantu pengambilan keputusan manajemen program/proyek. Monitoring atau dalam bahasa Indonesia yaitu pemantauan dapat dijelaskan sebagai kesadaran (*awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu. Monitoring akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan. [9]

2.2.5 *Global Positioning System* (GPS)

GPS (*Global Positioning System*) adalah sistem navigasi berbasis satelit yang dikembangkan oleh U.S. Department of Defense (DoD). pada awal tahun 1970 awalnya GPS dikembangkan sebagai sistem militer untuk memenuhi kebutuhan militer. Namun, yang terakhir dibuat tersedia untuk warga sipil, dan sekarang sistem penggunaan ganda yang dapat diakses oleh pengguna militer dan sipil. GPS memberikan posisi terus menerus dan informasi waktu setiap tempat di dunia di bawah setiap kondisi cuaca. Karena, melayani jumlah pengguna yang tidak terbatas serta digunakan untuk alasan keamanan. GPS menggunakan one-way-ranging (passive) sistem. Artinya, pengguna hanya dapat menerima sinyal satelit.

Satelit-satelit GPS harus selalu berada pada posisi orbit yang tepat untuk menjaga akurasi data yang dikirim ke GPS receiver, sehingga harus selalu dipelihara agar posisinya tepat. Posisi satelit-satelit tersebut selalu dipantau oleh stasiun pengendali. Stasiun-stasiun pengendali di bumi ada di Hawaii, Ascension Islan, Diego Garcia, Kwajalein dan Colorado Spring. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS receiver yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. GPS receiver mengambil informasi tersebut dan melakukan perhitungan triangulation untuk

menentukan lokasi pengguna dengan tepat. GPS receiver membandingkan waktu sinyal dikirim dengan waktu sinyal tersebut diterima untuk mengetahui jarak satelit. Dengan mengetahui jarak tersebut, GPS receiver dapat melakukan perhitungan dan menentukan posisi pengguna dan menampilkan dalam peta elektronik. Setelah menentukan posisi pengguna, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit GPS sangat presisi dalam mengirim informasi karena satelit tersebut menggunakan jam atom. Jam atom yang ada pada satelit merupakan partikel atom yang diisolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam biasa. Keistimewaan GPS adalah mampu bekerja dalam berbagai kondisi cuaca, siang atau malam. Keakuratan sebuah perangkat GPS bisa mencapai 15 meter, bahkan model terbaru yang dilengkapi teknologi Wide Area Augmentation System (WAAS) keakuratannya sampai 3 meter. Karena GPS bekerja mengandalkan satelit, maka penggunaannya disarankan di tempat terbuka. Penggunaan di dalam ruangan, atau di tempat yang menghalangi arah satelit (di angkasa), maka GPS tidak akan bekerja secara akurat dan maksimal. Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi. Semakin banyak sinyal satelit yang dapat diterima maka akan semakin presisi data posisi yang dihasilkan. Selain itu, ketinggian juga mempengaruhi proses kerja GPS, karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikitnya [12].

2.2.6 GPRS

GPRS (General Packet Radio Service) merupakan transmisi berbasis paket untuk GSM yang digunakan untuk transfer data yang berkaitan dengan e-mail, data gambar atau MMS, dan penelusuran internet. GPRS menawarkan laju data yang lebih tinggi. Kanal – kanal radio ganda dapat dialokasikan bagi seorang pengguna dan kanal yang sama dapat pula digunakan dengan berbagi antar pengguna sehingga menjadi sangat efisien. Dari segi biaya, harga mengacu pada

volume pengguna. Penggunanya ditarik biaya dalam kaitannya dengan banyaknya byte yang dikirim atau diterima, tanpa memperdulikan panggilan, dengan demikian dimungkinkan GPRS akan menjadi lebih cenderung dipilih oleh pelanggan untuk mengaksesnya daripada layanan - layanan IP.

GPRS merupakan teknologi baru yang memungkinkan para operator jaringan komunikasi bergerak menawarkan layanan data dengan laju bit yang lebih tinggi dengan tarif rendah, sehingga membuat layanan data menjadi menarik bagi pasar masal. Adapun komponen – komponen utama jaringan GPRS yaitu GGSN (Gateway GPRS Support Node) yaitu gerbang penghubung jaringan GPRS ke jaringan internet, Fungsi dari komponen ini adalah sebagai interface ke public data network, informasi routing, network screening, user screening, address mapping dengan tugas sebagai interface ke jaringan public internet atau mobile service provider. SGSN (Serving GPRS Support Node) atau gerbang penghubung jaringan BTS (Base Transceiver Station) ke jaringan GPRS, untuk mengantarkan paket ke mobile station dalam satu area. GPRS menggunakan sistem komunikasi packet switch sebagai cara untuk menstramisikan datanya. Packet switch sendiri adalah sebuah sistem dimana data yang akan ditransmisikan dibagi menjadi bagian – bagian kecil (paket) lalu ditranmisikan dan diubah kembali menjadi data semula.

2.2.7 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

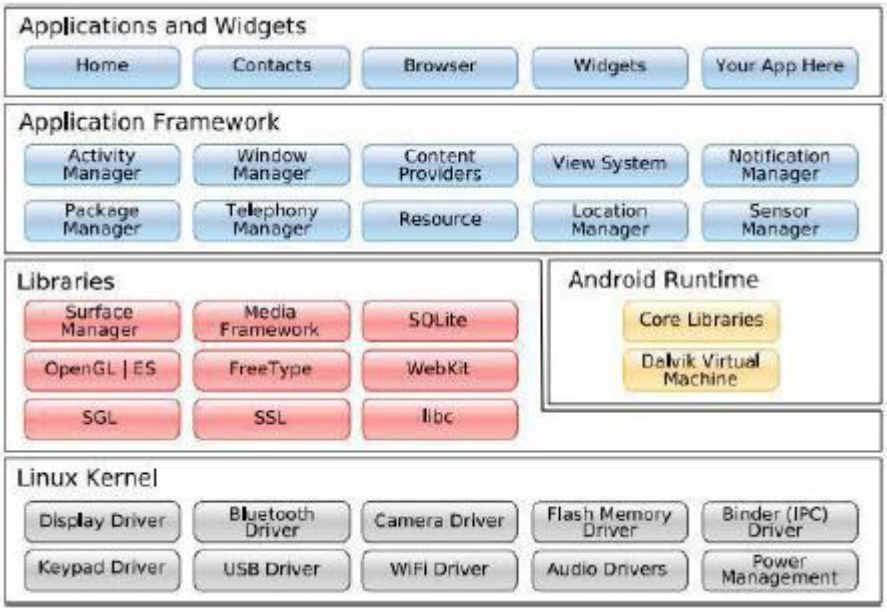
membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga [10].

2.2.7.1 Arsitektur Android

Arsitektur Android dapat digambarkan seperti pada gambar 2.2 dan secara garis besar Arsitektur Android dapat dijelaskan sebagai berikut:

a. *Application* and *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

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 *Views, Content Provider, Resource Manager, Notification Manager, Activity Manager*.

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.

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.

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 [10].

2.2.8 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat [11].

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa file teks ataupun Database Management System (DMBS). Kebutuhan basis data dalam sistem informasi meliputi :

1. Masukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan

2.2.9 DBMS

DBMS (Database Management System) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data [11]. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara bersamaan
4. Mampu menangani backup data.

2.2.10 SQL

SQL (Structured Query Language) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal query yang dimasukkan sama [11].

2.2.11 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan peralngkan lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi

berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientas objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek [11].

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekata terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut. Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas
   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (reusable).
2. Kecepatan pengembangan
   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.
3. Kemudahan pemeliharaan
   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi
   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak

Karena pendekatan pengembangan lebih dekat dengan dunai nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.2.12 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dihubungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancanggan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu [11].

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok daata dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*Class*)

   Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan

mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (relationship) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara tekniks kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satua organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstra. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan. Secara teknis, sebuah kelas saat program dieksekusi makan akan dibuat sebuah objek. Objek dilihat darisegi teknis adalah elemen pada saat runtime yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehinga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari event, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunnia nyata. Contoh metode atau operasi misalnya Read, Write, Move, Copy, dan sebagainya.

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat beruoa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (encapsulation)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain.

9. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

Komunikasi antarobjek dilakukan lewat pesan (message) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. Package

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda.

2.2.13 Internet

Internet adalah suatu jaringan computer global yang terbentuk dari jaringan-jaringan computer local dan regional yang memungkinkan komunikasi data antar computer yang terhubung ke jaringan tersebut. Internet awalnya merupakan suatu rencana dari Departemen Pertahanan Amerika Serikat (US Department of Defense) pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan ARPANET atau *Advanced Research Project Agency Network*. Beberapa universitas di Amerika Serikat diantaranya UCLA, Stanford, UC Santa Barbara dan University of Utah, diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan empat computer di lokasi universitas berbeda tersebut.

2.2.14 *Latitude* dan *Longtitude*

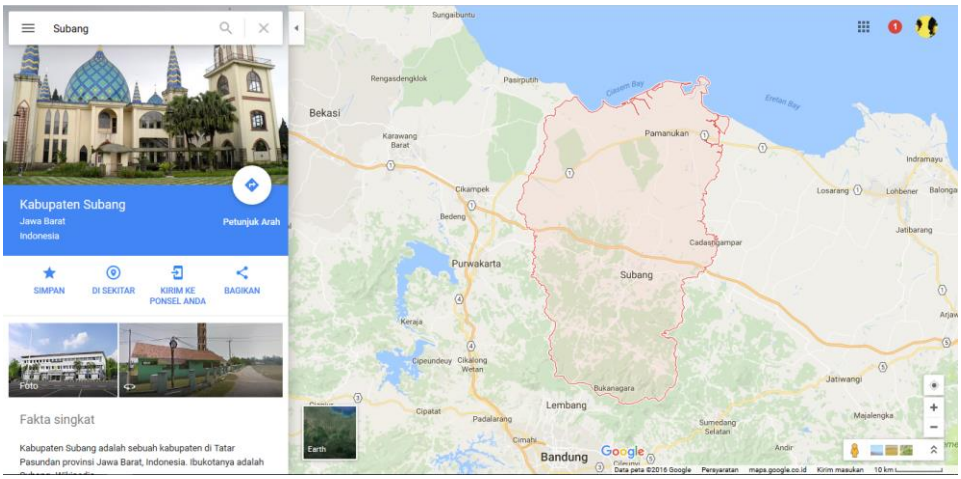
Latitude dan *longitude* adalah sebuah koordinat yang ada di bumi dan digunakan untuk menentukan kedudukan kita di atas bumi. *Latitude* adalah garis yang melintang dari kutub utara dan kutub selatan. Titik 0 adalah sudut ekuator, tanda + menunjukan arah ke atas menuju kutub utara, sedangkan tanda minus 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. *Longitude* adalah garis lintang. Angka dari sudut bundar bumi horisontal. Titik diawali dari 0 ke 180 derajat, dan 0 ke 180 derajat arah sebaliknya. Titik 0 dimulai dari garis negara Inggris, dan jika mengarah ke Indonesia akan menjadi angka positif. Kebalikannya, koordinat *Longitude* minus adalah arah kebalikan [13].

2.2.15 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 2.3. 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 [12].



Gambar 2. 3 Tampilan Google Maps

2.2.16 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya :

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, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine [14].

2.2.17 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara

khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

2.2.18 PHP

PHP atau singkatan dari Personal Home Page merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side. PHP termasuk dalam open source product, sehingga source code PHP dapat diubah dan didistribusikan secara bebas. PHP dapat disisipkan diantara HTML, dan karena bahasa *server-side*, maka bahasa PHP akan dieksekusi di server, yang dikirimkan ke browser adalah “hasil jadi” bentuk HTML, dan kode PHP tidak akan terlihat. PHP dahulu merupakan proyek pribadi dari Rasmus Lerdorf tahun 1994 (dengan dikeluarkannya PHP versi 1 yang digunakan untuk membuat home page pribadi yang dipublikasikan 8 juni 1995). Versi pertama ini berupa kumpulan script PERL. Untuk versi kedua Ramus menulis ulang script-script PERL tersebut menggunakan bahasa C, kemudian menambahkan fasilitas untuk form HTML dan koneksi MYSQL. Adapun PHP didapat dari singkatan PHP Hypertext Preprocessor.

Setelah mengalami perkembangan, Zeev Suraski dan Andi Gutmans (termasuk Rasmus) yang tergabung dalam suatu kelompok open source, maka mereka memulai versi 3 PHP nya, PHP telah menampakkan keunggulan sebagai salah satu bahasa *server scripting* yang handal. Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan oleh kelompok ini PHP disebut sebgai “PHP: Hypertext Preprocessor”. Pada bulan Mei 2000 dirilis PHP versi 4 yang dikeluarkan oleh Zend Engine yaitu sebuah produk yang diciptakan oleh Suraski dan Gutmans. Kehandalan PHP diperlengkap dengan dirilisnya PHP versi 5 pada 13 Juli 2004 yang mendukung OOP serta berbagai manajemen database yang luas

dan penanganan error yang mudah. Sintak yang digunakan berasal dari bahsa C, Java maupun PERL. Sampai tulisan ini dibuat versi PHP yang terbaru adalah versi 5.2.0 yang dirilis pada November 2006.

PHP merupakan bahasa script yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua script PHP dieksekusi pada *server* dimana script tersebut dijalankan. Oleh karena itu, spesifikasi *server* lebih berpengaruh pada eksekusi dari script php daripada spesifikasi *client*. Namun tetap diperhatikan bahwa web yang dihasilkan tentunya harus dapat dibuka oleh browser pada client. Dalam hal ini versi dari html yang digunakan harus didukung oleh browser client. PHP termasuk dalam Open Source Product. Jadi dapat menambah source code dan dapat mendistribusikannya secara bebas. PHP juga diedarkan secara gratis. Kita bisa mendapatkannya secara gratis. PHP juga dapat berjalan diberbagai web *server* semisal IIS, Apache, PWS.

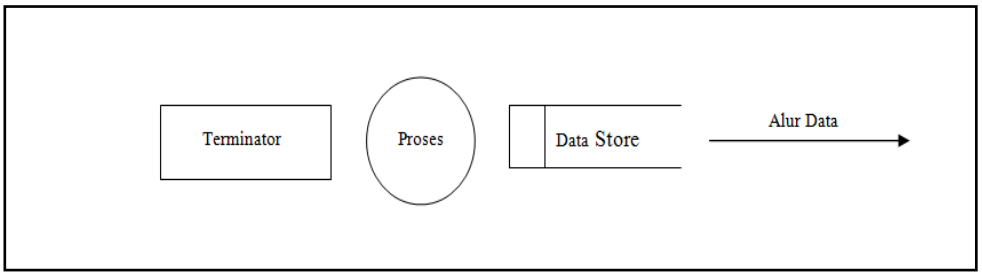
2.2.19 Pemrograman Terstruktur

Pemrograman terstruktur adalah bahasa pemrograman yang mendukung pembuatan program sebagai kumpulan prosedur. Prosedur-prosedur ini dapat saling memanggil dan dipanggil dari manapun dalam program dan dapat mengunakan parameter yang berbeda-beda untuk setiap pemanggilan. Bahasa pemrograman terstruktur adalah pemrograman yang mendukung abstraksi data, pengkodean terstruktur dan kontrol program terstruktur.

2.2.20 *Data Flow Diagram* (DFD)

Menurut Andi Kristanto [15] DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. Sedangkan, menurut Tata Sutabri [16] DFD adalah suatu network yang menggambarkan suatu sistem komputerisasi,

manualisasi atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya. Dibawah ini adalah komponen DFD atau Diagram Arus Data.



Gambar 2. 4 Komponen DFD

2.2.21 JSON

JavaScript Object Notation (JSON) pertama kali dikenalkan oleh Douglas Crockford adalah bagian dari JavaScript, dibuat dengan menggunakan pola dari JavaScript untuk merepresentasikan data terstruktur. Crockford membuat JSON sebagai alternatif dari (eXtended Markup Language) XML untuk mengakses data terstruktur dari JavaScript dengan menggunakan method eval() dan tidak membutuhkan pembuatan dari DOM. Bagian terpenting dari JSON adalah format data bukan bahasa pemrogramannya. JSON bukan bagian dari JavaScript walaupun mereka saling berbagi syntax, JSON juga tidak semata-mata digunakan oleh JavaScript. JSON merupakan data format, jadi JSON memiliki parsers dan serializers yang ada dalam banyak bahasa pemograman. Berikut adalah contoh script JSON dari google api :

```
{ "destination_addresses" : [
      "Jl. Letnan Jenderal S. Parman No.6, Pasirkareumbi,
Kec. Subang, Kabupaten Subang, Jawa Barat 41211,
Indonesia"
   ],
   "origin_addresses" : [
      "Jl. Veteran No.63, Kb. Pisang, Sumur Bandung, Kota
Bandung, Jawa Barat 40112, Indonesia"
   ],
   "rows" : [
      {
         "elements" : [
```

```
            {
               "distance" : {
                  "text" : "123 km",
                  "value" : 123241
               },
               "duration" : {
                  "text" : "1 hour 54 mins",
                  "value" : 6845
               }
            }
         ]
      }
   ]
}
```

2.2.22 FCM (*Firebase Cloud Messagging*)

Firebase Cloud Messaging (FCM) adalah solusi perpesanan lintas-platform yang memungkinkan untuk mengirimkan pesan dan pemberitahuan dengan terpercaya tanpa biaya. Dengan menggunakan FCM, dapat memberi tahu aplikasi klien bahwa email baru atau data lainnya tersedia untuk disinkronkan. Mengirim pemberitahuan untuk mendorong pelibatan kembali dan retensi pengguna. Untuk kasus penggunaan seperti perpesanan instan, pesan dapat mentransfer payload hingga 4 KB ke aplikasi klien.

Fungsi utamanya yaitu, penargetan pesan serba guna, distribusikan pesan kepada aplikasi klien dengan salah satu dari tiga cara — ke satu perangkat, ke grup perangkat, atau ke perangkat yang berlangganan topik. Dukungan pesan data dan pemberitahuan, mengirim pemberitahuan hingga 2 KB, payload data hingga 4KB, dan mengirim pesan dengan pemberitahuan maupun payload data. Perpesanan upstream dari aplikasi klien, Kirimlah kembali pemberitahuan, chat, dan pesan lain dari perangkat ke *server* melalui saluran koneksi FCM yang bisa diandalkan dan hemat baterai.

Implementasi FCM mencakup *server* aplikasi dalam lingkungan yang berinteraksi dengan FCM lewat protokol HTTP atau XMPP, dan aplikasi klien. Selain itu, FCM menyertakan konsol Notifications, yang dapat digunakan untuk mengirim pemberitahuan ke aplikasi klien. Firebase Notifications dibuat pada Firebase Cloud Messaging dan memiliki FCM SDK yang sama untuk pengembangan klien. Untuk uji coba atau mengirim pesan pemasaran atau

keterlibatan dengan penargetan bawaan dan analitik yang andal, dapat menggunakan Notifications. Untuk penerapan dengan persyaratan pesan yang lebih kompleks, FCM adalah pilihan yang tepat.

2.2.23 MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user. MySQL adalah implementasi dari system manajemen basisdata relasional (RDBMS). MySQL dibuah oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL.

Pendiri MySQL adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael "Monty". Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL (General Public License) namun tidak boleh menjadikan produk turunan yang bersifat komersial. Pada saat ini MySQL merupakan database *server* yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database yaitu SQL. SQL (Structured Query Language) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan database lebih user-friendly dibandingan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni. SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemograman seperti C, dan Delphi.

Elemen dari SQL yang paling dasar antara lain pernyataan, nama, tipe data, ekspresi, konstanta dan fungsi bawaan. Pernyataan Perintah dari SQL yang digunakan untuk meminta sebuah tindakan kepada DBMS. Sistem database menyediakan bahasa untuk mendefinisikan database *Data Definition Language* (DDL) serta bahasa untuk memanipulasi database *Data Manipulation Language* (DML) untuk melakukan operasi-operasi tertentu pada database.

a. *Data Definition Language* (DDL)

Kumpulan perintah SQL yang digunakan untuk membangun database, bisa juga memiliki arti lain perintah untuk membuat struktur dasar database dan tabel. Perintah – perintah yang termasuk DDL adalah statement connect, create, show, describe, drop, rename dan alter atau seperti contoh berikut :

```
1. CREATE DATABASE nama_database;
2. SHOW database;
3. USE nama_database;
```

b. *Data Manipulation Language* (DML)

Kumpulan statement yang digunakan sebagai perintah untuk mengelola data dalam sebuah tabel. Perintah DML digunakan setelah menggunakan DDL antara lain seperti statement insert, update, delete dan select atau seperti contoh berikut :

```
1. INSERT INTO nama_tabel VALUES isi_field;
2. SELECT * FROM nama_tabel;
3. UPDATE nama_tabel SET FIELD = nilai WHERE kondisi;
```

2.2.24 *Unified Modeling Language* (UML)

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan kebutuhan, membuat analisis dan desain, erta menggambarkan arsitektur dalam pemrograman berorientasi objek. Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Pengertian berorientasi objek berarti bahwa cara mengorganisasikan perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data perilakunya. Konsep OOAD (Object Oriented Analysis Design) mencakup analisis dan desain sebuah sistem dengan pendekatan objek.

1. Analisis berorientasi objek (OOA)

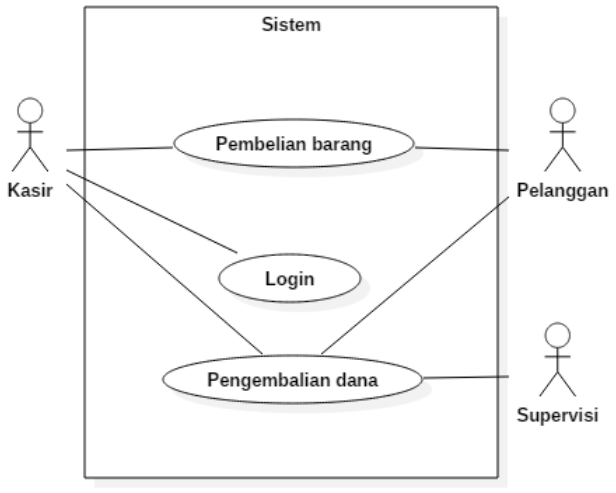
Metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang

ditemui dalam ruang lingkup perusahaan. Unified Modeling Language (UML) adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun dan mendokumentasi *artefacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak), seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML merupakan bahasa standar untuk penulisan blueprint software yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Berikut ini merupakan beberapa bagian dari UML adalah sebagai berikut :

a. *Use Case Diagram*

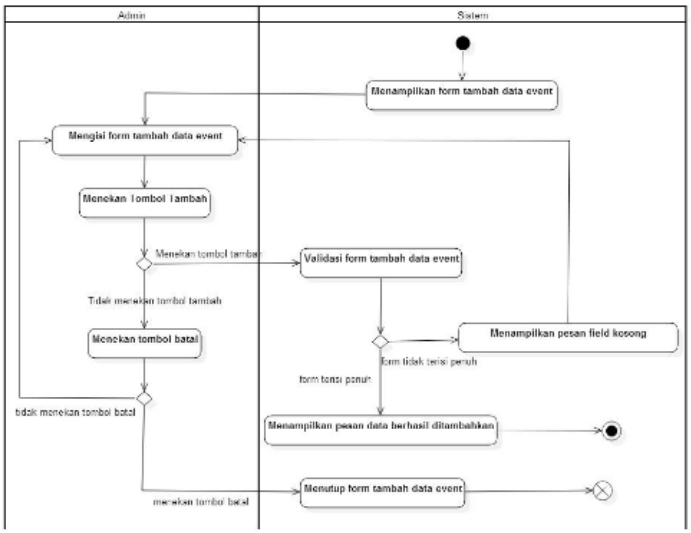
Use case diagram menggambarkan sejumlah eksternal aktor dan hubungannya ke use case yang diberikan oleh sistem.



Gambar 2. 5 Contoh *Use Case Diagram*

b. *Activity Diagram*

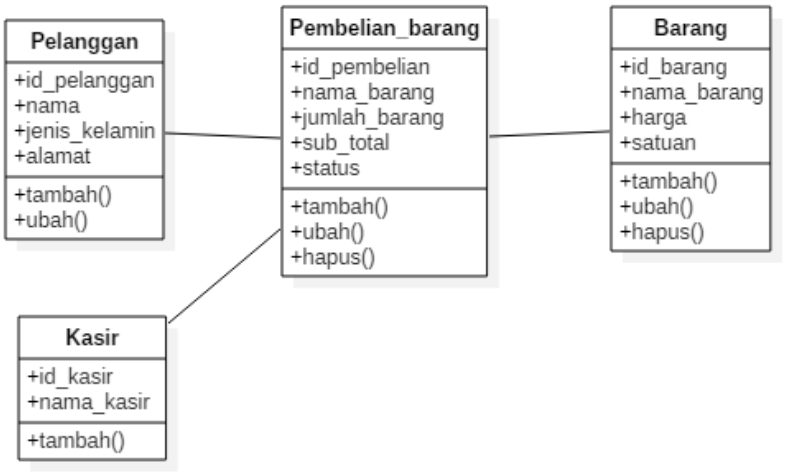
Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainya seperti use case.



Gambar 2. 6 Contoh *Activity Diagram*

c. *Class Diagram*

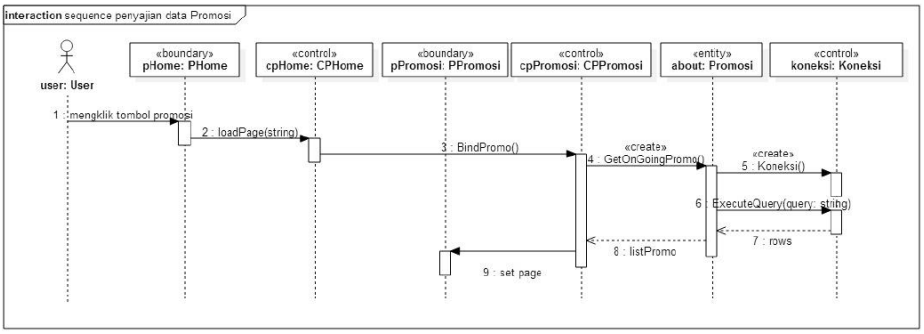
Class diagram menggambarkan struktur dan deskripsi class, package (paket) dan objek beserta hubungan satu sama lain seperti containment (penahanan), pewarisan, asosiasi dan lain-lain.



Gambar 2. 7 Contoh *Class Diagram*

d. *Sequence Diagram*

Sequence Diagram menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.



Gambar 2. 8 Contoh *Sequence Diagram*

2.2.25 *Web Server*

Tahun 1989, Tim Berners-Lee mengajukan pada perusahaannya, CERN (European Organization for Nuclear Research) sebuah proyek yang bertujuan untuk mempermudah pertukaran informasi antar para peneliti dengan menggunakan sistem hyperteks. Sebagai hasil atas implementasi proyek ini, tahun 1990 Berners-Lee menulis dua program komputer yaitu :

a. Sebuah peramban yang dinamainya sebagai WorldWideWeb.

b. *Server* web pertama di dunia, yang kemudian dikenal sebagai CERN httpd, yang berjalan pada sistem operasi NeXTSTEP.

Dari tahun 1991 hingga 1994, kesederhanaan serta efektifitas atas teknologi yang digunakan untuk berkunjung serta bertukar data melalui World Wide Web membuat kedua aplikasi tersebut diadopsi pada sejumlah sistem operasi agar dapat digunakan oleh lebih banyak individu, ataupun kelompok. Awalnya adalah organisasi penelitian, kemudian berkembang dan digunakan di lingkungan pendidikan tinggi, dan akhirnya digunakan dalam industri bisnis. Tahun 1994, Tim Berners-Lee memutuskan untuk membakukan organisasi World Wide Web Consortium (W3C) untuk mengatur pengembangan-pengembangan

lanjut atas teknologi-teknologi terkait lainnya (HTTP, HTML, dan lain-lain) melalui proses standardisasi.

2.2.26 *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler. *Web service* memiliki tiga entitas yaitu :

1. *Service Provider* yang berfungsi untuk menyediakan layanan/service dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.
2. *Service Registry* yang berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.
3. *Service Requestor* sebagai peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

Secara umum, *web service* memiliki tiga operasi yang terlibat didalamnya, yaitu :

1. *Publish/Unpublish*, menerbitkan atau menghapus layanan ke dalam atau dari *registry*.
2. *Find*, *service requestor* mencari dan menemukan layanan yang dibutuhkan.
3. *Bind*, *service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan *binding* ke *service provider* untuk melakukan interaksi dan mengakses layanan atau *service* yang disediakan oleh *service provider*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang

terdapat di dalamnya. Beberapa alasan mengapa digunakannya *webservice* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-upload ke *webserver* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi firewall.