BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah PDAM Tirta Raharja Kabupaten Bandung

Perusahaan Daerah Air Minum (PDAM) Tirta Raharja Kabupaten Bandung didirikan pada tahun 1926 dengan nama *Water Leiding Bedrijf*, yang pada awalnya diperuntukkan memenuhi kebutuhan air bersih bagi komunitas Belanda yang berada di wilayah Cimahi dan Lembang. Hingga pada perkembangannya, pada tahun 1977 berdasarkan Peraturan Daerah No. XVII Tahun 1977 serta disahkan dengan Keputusan Gubernur Jawa Barat No. 510/H.K/011/SK/77, perusahaan ini kemudian ditetapkan menjadi Perusahaan Daerah Air Minum Kabupaten Bandung. Pada akhirnya tahun 2005 PDAM Tirta Raharja Kabupaten Bandung yang tadinya bernama PDAM Kabupaten Bandung terbentuk melalui PERDA Kabupaten Bandung no. 5 Tahun 2005.

Tujuan didirikannya perusahaan ini adalah untuk dapat turut serta mensejahterakan masyarakat Kabupaten Bandung dan sekitarnya melalui air bersih dan sebagai Badan Usaha Milik Daerah (BUMD) mampu memberikan konstribusi bagi Pendapatan Asli Daerah.

Secara administratif, wilayah pelayanan PDAM Tirta Raharja Kabupaten Bandung meliputi 3 daerah otonom yaitu Kabupaten Bandung, Kabupaten Bandung Barat, dan Kota Cimahi. Untuk dapat melayani kebutuhan pelanggan yang berjumlah 60.529 sambungan langganan, PDAM Tirta Raharja

Kabupaten Bandung memiliki 6 Kantor Cabang yang terdiri dari Cabang 1 Soreang, Cabang 2 Ciparay, Cabang 3 Majalaya, Cabang 4 Lembang, Cabang 5 Padalarang, dan Cabang 6 Cimahi.

Visi dan Misi PDAM Tirta Raharja Kabupaten Bandung

Visi : Profesional, handal dengan pelayanan prima menjadi PDAM Termaju.

Misi : Memberikan pelayanan air minum secara baik dengan mengacu pada Kualitas, Kuantitas dan Kontinuitas yang memadai dengan harga yang terjangkau sehingga diharapkan dapat memberikan dampak peningkatan kesehatan masyarakat pengguna air bersih [5].

PDAM Tirta Raharja selalu berusaha untuk mencapai visi dan misi yang telah ditetapkan tersebut. Berbagai usaha dilakukan salah satunya adalah dengan memanfaatkan penggunaan teknologi informasi dan komunikasi dalam aktifitas sehari-hari dilingkup kerja PDAM Tirta Raharja seperti dengan membangun sistem informasi pemeriksaan secara online gangguan sambungan langganan air yang berguna untuk meningkatkan efektifitas dan efisiensi kinerja dilingkup kerja PDAM Tirta Raharja dan untuk lebih mengoptimalkan pelayanan terhadap pelanggan.

2.1.2 Logo PDAM Tirta Raharja Kabupaten Bandung

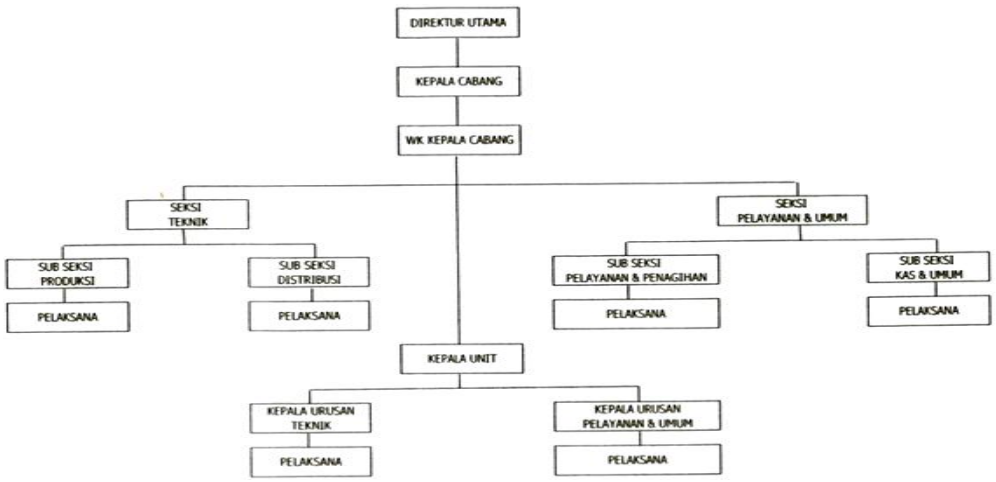


Gambar 2.1 Logo PDAM Tirta Raharja Kabupaten Bandung

2.1.3 Badan Hukum

Perusahaan Daerah Air Minum Tirta Raharja Kabupaten Bandung adalah satu-satunya Badan Usaha Milik Daerah (BUMD), yang mempunyai tugas memberikan pelayanan air bersih untuk masyarakat Kabupaten Bandung, yang didirikan berdasarkan Peraturan Daerah (PERDA) Nomor : XVII tahun 1977 dan disahkan dengan keputusan Gubernur Kepala Daerah Tingkat I Jawa Barat No. 510/HK/011/SK/77 serta diubah terakhir kalinya melalui Peraturan Daerah Nomor 5 tahun 2005 [5].

2.1.4 Struktur Organisasi



Gambar 2.2 Struktur Organisasi

2.1.5 Deskripsi Tugas

a. Direktur Utama

Bertugas memimpin, mengkoordinasikan, mengelola, dan mengendalikan serta merencanakan teknis operasional pengelolaan Perusahaan Daerah sesuai dengan Peraturan Daerah dan Rencana Kerja Anggaran Perusahaan

Daerah serta mempertanggungjawabkan terhadap penggunaan dan pengelolaan kekayaan Daerah yang ditanamkan pada Perusahaan Daerah.

b. Kepala Cabang

Bertugas merencanakan, menyelenggarakan, mengevaluasi, dan merekomendasi pelayanan air minum secara berkesinambungan kepada masyarakat dan pelanggan di cabang dan unit melalui sistem jaringan perpipaan dan terminal lainnya, pemeliharaan seluruh fasilitas perusahaan daerah agar tercapainya kualitas, kuantitas, dan kontinuitas yang optimal, administrasi dan pemliharaan seluruh asset perusahaan daerah, pembinaan kepegawaian serta mempertahankan dan atau meningkatkan penjualan air, efisiensi dan efektifitas penagihan melaui pembinaan hubungan yang harmonis denan seluruh pemangku kepentingan.

c. Wakil Kepala Cabang

Bertugas membantu Kepala Cabang dalam merencanakan, menyelenggarakan, mengevaluasi, dan merekomendasi pelayanan air minum secara berkesinambungan kepada masyarakat dan pelanggan di cabang dan unit melalui sistem jaringan perpipaan dan terminal lainnya, pemeliharaan seluruh fasilitas perusahaan daerah agar tercapainya kualitas, kuantitas, dan kontinuitas yang optimal, administrasi dan pemliharaan seluruh asset perusahaan daerah, pembinaan kepegawaian serta mempertahankan dan atau meningkatkan penjualan air, efisiensi dan efektifitas penagihan melaui pembinaan hubungan yang harmonis denan seluruh pemangku kepentingan.

d. Kepala Unit

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, menganalisa, dan mengevaluasi kegiatan operasional teknik, administrasi, kepegawaian maupun pelayanan untuk terciptanya keselarasan dan keterpaduan pelayanan air minum dalam rangka pelaksanaan tugas dan fungsi di unit secara efektif dan efisien agar terciptanya kualitas, kuantitas, dan kontinuitas pelayanan.

e. Kepala Seksi Teknik

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, menganalisa, dan mengevaluasi proses pengolahan dan pemeliharaan air baku menjadi air minum sesuai dengan kapasitas yang tersedia dan memenuhi standar kesehatan serta mendistribusikan air minum kepada pelanggan dengan kualitas, kuantitas, dan kontinuitas secara optimal serta pemeliharaan dan perbaikan jaringan perpipaan, pemasangan dan penertiban sambungan langganan di cabang sesuai dengan prosedur standar operasi yang ditetapkan.

f. Kepala Seksi Pelayanan dan Umum

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, menganalisa, dan mengevaluasi kegiatan pelayanan langganan, pembacaan meter, kerumahtanggaan, pencatatan, pemeliharaan, dan pengamanan asset, administrasi gudang, pengelolaan kas kecil, administrasi kepegawaian dan transaksi rekening di cabang agar terwujudnya tertib

administrasi dan terciptanya pelayanan yang optimal sesuai dengan target yang ditetapkan oleh perusahaan.

g. Kepala Urusan Teknik

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, dan menganalisa proses pengolahan dan pemeliharaan air baku menjadi air minum sesuai dengan kapasitas yang tersedia dan memenuhi standar kesehatan, mendistribusikan air minum ke pelanggan dengan kualitas, kuantitas, dan kontinuitas secara optimal, serta pemeliharaan dan perbaikan jaringan, pemasangan dan penertiban sambungan langganan di unit sesuai dengan prosedur standar operasi yang ditetapkan.

h. Kepala Urusan Pelayanan dan Umum

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, dan menganalisa kegiatan pelayanan langganan, pembacaan meter, kerumahtanggaan, pencatatan, pemeliharaan dan pengamanan aset, administrasi gudang, pengelolaan kas kecil, administrasi kepegawaian dan transaksi rekening di unit agar terwujudnya tertib administrasi dan terciptanya pelayanan yang optimal sesuai dengan target yang ditetapkan oleh perusahaan.

i. Kepala Sub Seksi Produksi

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, dan menganalisa kualitas air baku, proses produksi, pemeliharaan dan perbaikan instalasi produksi dan fasilitas lainnya, penggunaan bahan

kimia, energi listrik, dan bahan bakar di cabang secara efektif dan efisien sehingga terciptanya kualitas, kuantitas, dan kontinuitas yang optimal.

j. Kepala Sub Seksi Distribusi

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, dan menganalisa pendistribusian air minum kepada pelanggan, pemeliharaan dan perbaikan jaringan perpipaan, survey pelanggan, pemasangan sambungan langganan baru dan penertiban, serta kegiatan materisasi di cabang sesuai dengan ketentuan dan prosedur yang berlaku dalam rangka menjaga kesinambungan pelayanan agar kualitas, kuantitas, dan kontinuitas berjalan secara optimal.

k. Kepala Sub Seksi Pelayanan dan Penagihan

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, dan menganalisa kegiatan pelayanan langganan, pembacaan meter, dan penagihan piutang di cabang dalam upaya terwujudnya tertib administrasi, mempertahankan dan meningkatkan penjualan air, serta terciptanya pelayanan optimal sesuai dengan target yang ditetapkan oleh perusahaan.

l. Kepala Sub Seksi Kas dan Umum

Bertugas merencanakan, melaksanakan, mengatur, mengawasi, dan menganalisa kegiatan transaksi rekening, administrasi perkantoran, administrasi kepegawaian, pengelolaan kas kecil, pengelolaan barang persediaan dan pemeliharaan serta pengamanan asset perusahaan di

cabang sesuai dengan prosedur standar operasi dalam upaya tertib administrasi, efisiensi, dan efektivitas penagihan.

m. Pelaksana Produksi

Bertugas melaksanakan kegiatan proses produksi mulai dari pengolahan air baku menjadi air minum, pemeliharaan, perbaikan dan menjaga kebersihan instalasi produksi dan fasilitas pendukung lainnya, mencatat penggunaan bahan kimia, energy listrik, bahan bakar atau material lainnya di cabang atau unit secara efektif dan efisien sesuai dengan prosedur standar operasi yang telah ditetapkan.

n. Pelaksana Distribusi

Bertugas melaksanakan pendistribusian air, pengaturan tekanan dalam jaringan, pemeliharaan dan perbaikan kebocoran pipa, pemasangan sambungan langganan baru, penertiban sambungan langganan, pencucian jaringan perpipaan, serta meterisasi di cabang atau unit sesuai dengan ketentuan agar kualitas, kuantitas, dan kontinuitas pendistribusian air minum berjalan secara optimal.

o. Pelaksana Pelayanan Langganan

Bertugas melaksanakan seluruh kegiatan administrasi langganan serta tindak lanjut pelayanan kepada pelanggan di cabang atau unit sesuai dengan prosedur standar operasi yang telah ditetapkan agar terciptanya tertib administrasi dan pelayanan pelayanan kepada pelanggan secara optimal.

p. Pelaksana Kas

Bertugas melaksanakan, mengamankan seluruh transaksi kas dan membuat Laporan Penagihan Penagih (LPP) serta mencetak Surat Pemberitahuan Tunggakan (SPT) di cabang atau unit dalam upaya mempertahankan dan emningkatkan efisiensi dan efektivitas penagihan.

q. Pelaksana Umum

Bertugas melaksanakan pekerjaan kerumahtanggaan, pengelolaan kas kecil, dan administrasi perkantoran, barang persediaan, asset, kepegawaian di cabang atau unit untuk menunjang kelancaran kegiatan operasional cabang.

2.2 Landasan Teori

Landasan teori menjelaskan beberapa teori yang berkaitan dengan permasalahan yang dibahas sebagai dasar pemahaman dalam sebuah sistem.

2.2.1 Sistem Informasi

Adapun beberapa penjelasan terkait dengan sistem informasi adalah pengertian sistem, pengertian informasi, dan pengertian sistem informasi itu sendiri [14].

2.2.1.1 Pengertian Sistem

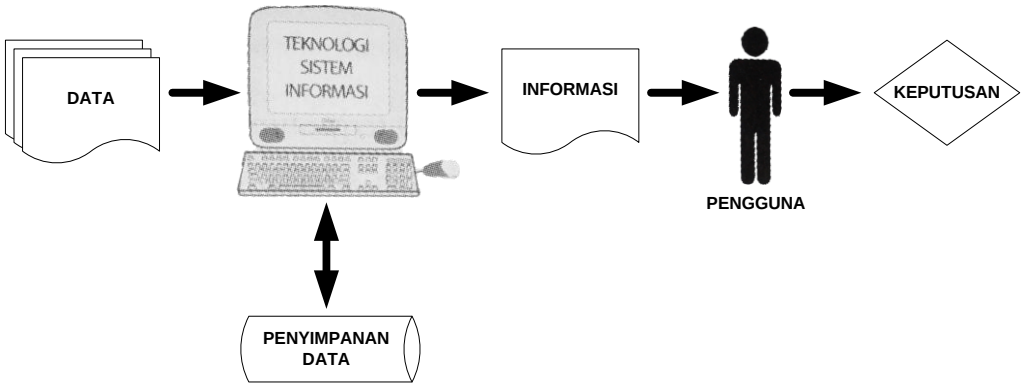
Sistem dapat diartikan sebagai serangkaian komponen-komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu.

2.2.1.2 Pengertian Informasi

Informasi merupakan hasil dari pemrosesan data (fakta) menjadi sesuatu yang bermakna dan bernilai untuk pengambilan keputusan.

2.2.1.3 Pengertian Sistem Informasi

Sistem Informasi bisa disimpulkan sebagai serangkaian komponen berupa manusia, prosedur, data, dan teknologi (seperti komputer) yang digunakan untuk melakukan sebuah proses yang menghasilkan informasi yang bernilai untuk pengambilan keputusan. Sistem informasi melekat dan merupakan infrastruktur penunjang keberhasilan bagi setiap organisasi (dalam mencapai tujuannya).



Gambar 2.3 Interaksi Komponen Sistem Informasi

2.2.2 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan

pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Quacomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan open source pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Sekitar september 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada Januari 2008. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communication, diproduksi oelh Asustek Computer Inc, Garmin Ltd, Softbank, Sonny Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance,* OHA mengumumkan produk perdana mereka Android, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru [11].

Versi-versi dari android:

1. Android versi 1.1

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

2. Android versi 1.5 (*Cupcake*)

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

3. Android versi 1.6 (*Donut*)

   Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, 16 Gestures, dan *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to*

change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. Android versi 2.0/2.1 (*Eclaire*)

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

5. Android versi 2.2 (Froyo: Frozen Yoghurt)

   Android Froyo yaitu android versi 2.2 diluncurkan pada tanggal 20 Mei 2010, dengan perubahan terhadap versi android sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. Android Gingerbread 2.3

   Android Gingerbread diluncurkan pada tanggal 6 Desember 2010. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface)

didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. Android HoneyComb 3.0/3.1

   Android HoneyComb dirancang khusus untuk tablet, disini anda dapat melihat tablet yang memakai android untuk os beserta *harga tablet*. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

8. Android Ice Cream Sandwich (ICS) 4.0

   Android Ice Cream Sandwich diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan

fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

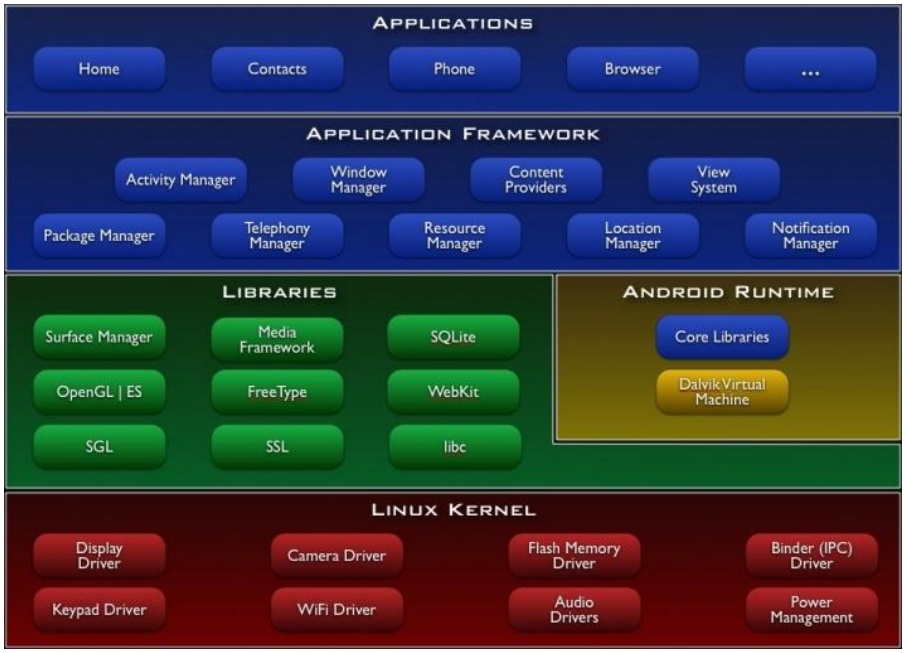
2.2.2.1 Feature

Features yang tersedia pada platform Android saat ini antara lain:

a. Framework Aplikasi yang mendukung penggantian komponen dan *reusable*.
b. Mesin Virtual Dalvik berjalan diatas Linux kernel dan dioptimalkan untuk perangkat mobile.
c. Integrated browser berdasarkan open source engine WebKit.
d. Grafis yang dioptimalkan dan didukung oleh library grafis 2D yang terkostumisasi, grafis 3D berdasarkan spesifikasi openGL ES 1,0 (Opsional akselarasi hardware).
e. SQLite untuk penyimpanan data.
f. Media support yang mendukung audio, video, dan gambar (MPGE, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
g. GSM Telephony (tergantung hardware).
h. Bluetooth, EDGE, 3G, GPS, dan WIFI (tergantung hardware).
i. Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk Eclips IDE.
j. Multi-touch, kemampuan layaknya handset modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.

k. Market, Seperti kebanyakan Handphone yang memiliki tempat penjualan aplikasi, market pada android merupakan katalog aplikasi yang dapat di download dan di install pada Handphone melalui internet.

2.2.2.2 Arsitektur Android



Gambar 2.4 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan 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. Semua aplikasi ditulis menggunakan bahasa pemrograman java.

b. Applications Frameworks

Android adalah "*Open Development Platform"* yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Arsitektur aplikasi dirancang supaya pengembang dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse).*

Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut:

1. *View* yang kaya dan extensible yang dapat digunakan untuk membangun aplikasi, termasuk list, grid, kotak teks, tombol, dan bahkan sebuah embeddable web.
2. *Content provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak help) atau membagi data tersebut.
3. *Resource Manager* yang menyediakan akses ke kode non sumber lokal seperti string, gambar, dan tata letak file.
4. *Notification Manager* yang memungkinkan semua aplikasi menampilkan alert yang bisa dokostumisasi didalam status bar.
5. *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum backstack.

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 Libe dan SSL, serta:

1. *Libraries* media untuk pemutaran media audio dan video.
2. *Libraries* untuk manajemen tampilan.
3. *Libraries* grafik 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 keamanan.
6. *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embedded web view.*
7. *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 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 dalama bahasa java, sementara Dalvik sebagai virtual mesinnya buka *Virtual Machine Java*, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh *Core Libraries.*

2. *Dalvik Virtual Machine:* Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mapu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

e. Linux Kernel

Linux Kernel adalah *layer* dimana inti dari *operating system* Android itu berada. Berisi file-file sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi Android lainnya. *Linux Kernel* yang digunakan Android adalah *linux kernel* versi 2.6.

2.2.3 *Database* (Basis Data)

Dasabase adalah kumpulan file-file yang saling berelasi sehingga membentuk satu bangunan data, relasi tersebut biasa ditunjukan dengan kunci dari tiap file yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam lingkup perusahaan, instansi. Bila terdapat file yang tidak dapat dipadukan atau dihubungkan dengan file lainnya berarti file tersebut bukanlah kelompok dari satu database, ia akan membentuk satu database sendiri [4].

Komponen dalam database:

1. *Entity*: orang, tempat, kejadian atau konsep yang informasinya direkam. Pada bidang Administrasi Siswa misal entitynya adalah siswa, buku, pembayaran, dan nilai test.
2. *Attribute*: Setiap *entity* mempunyai attribute atau sebutan untuk mewakili suatu *entity*. Seorang siswa dapat dilihat dari attributenya,

misalnya nama, nomor siswa, alamat. *Attribute* juga disebut juga sebagai data elemen, data *field*, atau data *item*.

3. *Data Value* (nilai atau isi data): data aktual atau informasi yang disimpan pada tiap data elemen atau *attribute*.
4. *Record/tuple:* kumpulan elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap.
5. *File:* Kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, *attribute* yang sama, namun berbeda beda data *value*nya.

2.2.3.1 *Entity Relationship Diagram* (ERD)

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan ERD. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan ERD, model dapat diuji dengan mengabaikan proses yang dilakukan [1].

2.2.4 *Object Oriented Programming* (OOP)

Pemrograman berorientasi objek merupakan pemrograman yang menjadikan objek sebagai komponen utama dalam sistem. Pemrograman berorientasi objek merupakan satu ide luar biasa dalam bidang pemrograman yang setelah diterapkan ternyata melahirkan bahasa yang lebih maju seperti C++, Java, Phyton, dsb. Perkembangan dari OOP inipun melahirkan konsep pemrograman yang lain berbasis Windows atau GUI *(Graphical User Intarface)*

yang kemudian popular sebai bahasa visual, seperti Visual Basic, Delphi, Visual C, Visual Java, dsb [15].

2.2.4.1 Konsep Objek

Objek dalam pengertian sehari-hari adalah benda, baik yang secara fisik dapat diketahui keberadaannya maupun yang bersifat khayal (virtual). Contohnya, kamera, kalkulator, ponsel, ide, gagasan, dan sebagainya.

Objek dalam pemrograman *(soft object)* adalah suatu komponen dari suatu sistem yang dibangun, misalnya objek matematis, objek jendela *(windows)*, objek tombol, objek kotak gambar, objek kotak teks, dsb.

2.2.4.2 Konsep Kelas

Setiap objek memiliki suatu kelompok. Kelompok ini dalam konsep OOP disebut “kelas” atau “class”. Misalnya Toyota, Daihatsu, dan Honda adalah termasuk kelompok ‘mobil’.

Dalam suatu kelas terdapat dua hal: sifat-sifat *(attribute/field/property)* dan fungsi *(method)* atau kelakuan *(behavior)*. Misalnya kelas mobil memiliki atribut warna body, jumlah tempat duduk, ukuran ban, dsb, kemudian terdapat fungsi seperti pengapian, pengisian aki, percepatan *(accelerator)*, rem, dsb. Pada tingkat pemrograman suatu atribut bisa berupa definisi suatu varabel/field atau definisi fungsi, sedangkan kelakuan bisa berupa implementasi dari suatu fungsi.

Sifat-sifat dari suatu kelas bisa diwariskan *(inheritance)* ke sub-kelasnya, misal sub-kelas mobil sedan mewarisi atribut dari kelas mobil. Fungsi-fungsi kelas juga dapat diwariskan bilamana perlu.

2.2.4.3 Kelas Konstruktor

Konstruktor adalah suatu fungsi khusus yang diberi nama sama dengan nama kelasnya. Pemberian nilai pada atribut kelas dapat dilakukan pada saat objek didefinisikan melalui kelas konstruktor *(class constructor)*.

2.2.4.4 Enkapsulasi

Enkapsulasi *(encapsulation)* adalah suatu kelas sebenarnya yang terpisah dari kelas yang lain, dimana atribut dan fungsi terbungkus dalam kapsul sehingga akses dari luar terhadap isi kelas pada kelas pada dasarnya terbatas.

Akses terhadap kelas bergantung pada *modifier* yang diberikan pada kelas dan isinya. Ada tiga macam modifier yang biasa digunakan, yaitu:

1. *Public*, berarti dapat diakses secara bebas baik dari dalam maupun dari luar kelas.
2. *Private*, berarti akses sangat terbatas, hanya dari dalam kelas itu sendiri, tidak bisa dari luar.
3. *Protected*, berarti akses terbatas, hanya diberikan pada sub-kelas, dan dari dalam kelas.

Enkapsulasi memberi jaminan pada kelas objek agar tidak diganggu oleh pemrogram kecuali yang disediakan untuknya. Dengan demikian program akan lebih bersih dari gangguan meupun kesalahan, termasuk kesalahan dalam pemakaian objek. Contoh enkapsulasi:

```
class belajar{
    public String x =”Pintar”;
    private String y = “Java”;
    protected String z = “OOP”;
}
```

2.2.4.5 Pewarisan

Peristiwa pemakaian kelas objek dengan parameter-parameter lokal yang sesuai dengan definisi antarmuka objek dapat dipandang sebagai pewarisan *(inheritance)* sifat-sifat objek ke program klien. Sebagai contoh disediakan objek bingkai atau frame dengan antarmuka seperti *height* (tinggi) dan *width* (lebar). Dalam hal ini program klien dapat memanggil obejk frame dan men-set tinggi dan lebarnya sesuai kebutuhan. Inheritance berarti pewarisan pada sub-kelas dari suatu kelas, yang bisa dilaksanakan melalui kata kunci *extends.* Contoh pewarisan:

```
public class binatang{
        protected String makanan;
        public void makan(){
                System.out.println(“makan”.+this.makanan);
        }
}
Public class singa extends binatang{
        Public void daging(){
                System.out.println(this.makanan + “daging”);
```

```
        }
    }
```

2.2.4.6 Polimorfisme

Polimorfisme adalah sifat dari kelas objek yang memungkinkan beberapa fungsi dapat memiliki nama yang sama walaupun isi fungsi tersebut berbeda. Sebagai contoh, fungsi konstruktor suatu kelas circle tetapi dengan parameter berbeda, circle dengan parameter x, circle dengan parmeter x,y, dan circle dengan parameter x,y,z. Jadi setiap kelas mempunyai parameter yang berbeda dan dapat digunakan sesuai kebutuhannya masing-masing.

2.2.5 *Unified Modelling Language* (UML)

UML adalah bahasa untuk mendokumentasikan, memodelkan, dan membangun sistem perangkat lunak dalam bentuk diagram visual. UML dicetuskan oleh *Three Amigos* (tiga sekawan), Grady Booch, James Rumbaugh, dan Ivar Jacobson [3].

Versi-versi dari UML:

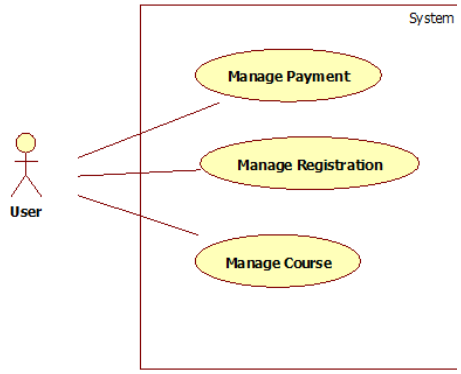
a. UML 1.0 diresmikan pada Januari 1997
b. UML 1.1 diresmikan pada Juli 1997
c. UML 1.2 diresmikan pada Juni 1998
d. UML 1.3 diresmikan pada tahun 1999
e. UML 1.5
f. UML 2.0

Berikut ini adalah jenis-jenis diagram yang umum digunakan :

1. *Use Case Diagram*

   Mendeskripsikan apa yang sistem lakukan tanpa mendeskripsikan bagaimana sistem menyelesaikannya. Dibuat berdasarkan interaksi dan relasi dari individual use case yang berisi aktor, event, dan use case.

   Contoh *use case diagram:*



Gambar 2.5 Contoh *Use Case Diagram*

2. *Use Case Scenario*

   Merupakan hasil penjelasan dari setiap use case. Terbagi menjadi tiga bagian, yaitu: identifikasi dan inisiasi, step perfomed, kondisi akhir. Contoh *use case scenario*:

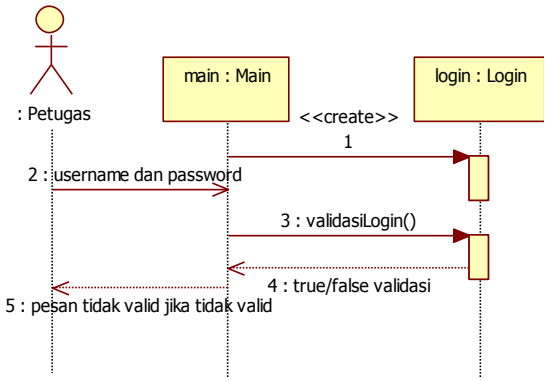
Identifikasi	
Nama *Use Case*	Lihat Peta
Aktor	Pengguna
Tujuan	Melihat peta lokasi
Keadaan Awal	Sistem menampilkan halaman detail_content
Skenario Utama	
Aksi Aktor	**Reaksi sistem**
1. Menekan tombol view_map yang ada pada halaman detail_conten	
	2.Menampilkan map lokasi fasilitas umum sesuai dengan konten yang dipilih.
Skenario Alternatif – Validasi Gagal	
Aksi Aktor	**Reaksi sistem**
	1. Menampilkan pesan bahwa proses gagal dilakukan (”Cek Koneksi Jaringan”) atau (”Nyalakan GPS”)
2. Menekan tombol ”OK”	
	3. Sistem akan *terminate*.
Kondisi Akhir	Menampilkan map yang sesuai diminta.

Gambar 2.6 Contoh *Use Case Scenario*

3. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

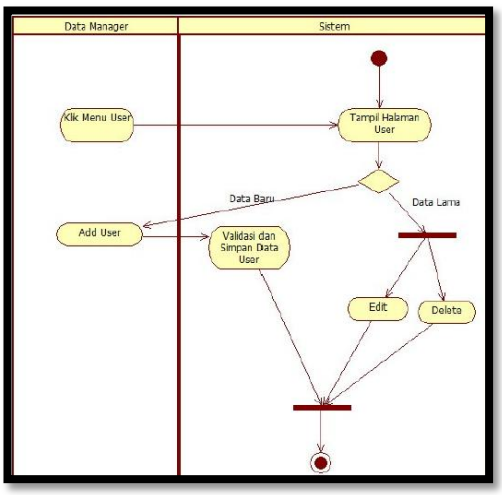
Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output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



Gambar 2.7 Contoh *Sequence Diagram* (Rosa, 2011)

4. *Activity* Diagram

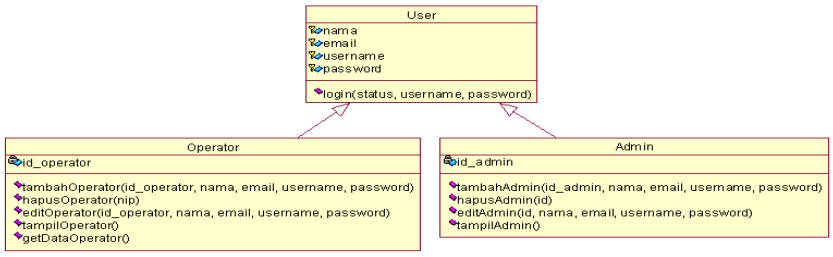
Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses yang dipakai untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip flowchart atau Data Flow Diagram pada perancangan terstruktur. Contoh *activity diagram:*



Gambar 2.8 Contoh *Activity Diagram*

5. *Class* Diagram

Class Diagram menggambarkan keadan suatu sistem dengan menjelaskan keterhubungan antara suatu *class* dengan *class* yang lain yang terdapat pada sistem tersebut. *Class Diagram* bersifat statis, di dalam *class diagram* digambarkan relasi dari masing - masing *class* tetapi tidak menggambarkan apa yang terjadi ketika *class* tersebut berelasi. Contoh *class diagram*:



Gambar 2.9 Contoh *Class Diagram*

2.2.6 Data Flow Diagram (DFD)

DFD atau dikenal juga dengan nama diagram arus data adalah diagram yang menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (melalui lewat telepon, surat, dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya kartu, hard disk, tape, diskette, dan lain sebagainya). DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. DFD dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. Lebih lanjut DFD juga merupakan dokumentasi dari sistem yang baik [2].

2.2.7 Java

Menurut definisi yang diberikan oleh Sun MicroSystem, JAVA adalah bahasa pemrograman yang *simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded,* dan *dinamyc,*.

Java bahasa yang *simple:* Rancangan bahasa java dibuat sedemikian rupa sehingga dengan cepat dapat dikenali dan dipahami oleh pemrogram pemula sekalipun karena mirip dengan bahasa C/C++ dan yang kedua lebih sederhana dari C/C++ dalam beberapa hal yang sulit pada C/C++ sudah tidak ada pada Java, misalnya pemakaian pointer.

Java adalah bahasa yang *object oriented*: Konsep objek diterapkan pada pembuatan program Java, bahkan java menggunakan konsep objek secara murni.

Java adalah bahasa yang *distributed*: Bahasa Java dirancang sedemikian rupa sehingga mendukung pembuatan aplikasi dalam jaringan computer.

Java adalah bahasa yang *robust*: rancangan java dibuat sedemikian rupa hingga mengurangi kemungkinan menjadi beku *(freeze)* ketika dijalankan.

Java adalah bahasa yang s*ecure*: secara otomatis java menerapkan pengamanan terhadap aplikasi sehingga mengurangi kemungkinan terjadinya serangan dari pengguna jaringan.

Java adalah bahasa yang *architecture neutral*: program java yang telah dikompilasi dapat dijalankan

Java adalah bahasa yang *portable*: program java sangat mudah dipindahkan dari satu mesin ke mesin lain tanpa harus dikompilasi ulang.

Java adalah bahasa yang h*igh-performance*: pada hakikatnya program C yang telah dikompilasi berjalan jauh lebih cepat dari Java, namun hasil kompilasi ini belum tentu bisa jalan pada computer yang berbeda arsitektur. Program java dikompilasi menjadi bytecode yang bisa jalan diberbagai mesin, namun ketika dijalankan bytecode ini harus diterjemahkan oleh JVM ke dalam bahasa mesin yang sesuai, sehingga lebih lambat dari C, namun karena java dapat digunakan pada jaringan computer yang kenyataannya tidak terlalu cepat karena hambatan saluran komunikasi maka program Java masih terasa cukup cepat.

Java adalah bahasa yang m*ultithreaded*: Java dirancang untuk menangani berbagai aplikasi yang berjalan secara bersamaan, misalnya

memainkan lagu, sambil melakukan download. Kejadian ini disebut dengan multithreaded.

Java adalah bahasa yang d*ynamic*: bahasa java dirancang untuk beradaptasi dengan lingkungan yang dapat berubah dengan cepat [15].

2.2.8 *HyperText Markup Language* (HTML)

HTML adalah file teks murni yang dapat dibuat dengan editor teks.HTML merupakan dokumen yang disajikan dalam browser web. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam internet. Ada dua cara untuk membuat sebuah web page: dengan HTML editor atau dengan editor teks biasa (misalnya notepad).

Untuk menandai berbagai elemen dalam suatu dokumen HTML, kita menggunakan tag. Tag HTML terdiri atas sebuah kurung sudut kiri (<, tanda lebih kecil), sebuah nama tag, dan sebuah kurung sudut kanan (>, tanda lebih besar). Tag umumnya berpasangan (misalnya <H1> dengan </H1>), tag yang menjadi pasangan selalu diawali dengan karakter garing (/, garis miring). Tag yang pertama menunjukkan tag awal yang berarti awal elemen, dan yang kedua menunjukkan tag ekhir, berarti akhir elemen.

Elemen yang dibutuhkan untuk membuat suatu dokumen HTML dinyatakan dengan tag <html>, <head>, dan <body> berikut tag-tag pasangannya. Setiap dokumen terdiri atas tag head dan body. Elemen head berisi infomasi tentang dokumen tersebut, dan elemen body berisi teks yang sebenarnya tersusun dari link, grafik, paragraph, dan elemen lainnya [13].

Pola dasar HTMl:

```
<html>
<head>
    Informasi tentang dokumen HTML
</head>
<body>
    Informasi yang ditampilkan dalam browser web
</body>
</html>
```

2.2.9 *Hypertext Preprocessor* (PHP)

PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web yang dinamis. Ketika dipanggil dari web browser, program yang ditulis dengan PHP akan di-*parsing* di dalam web server oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke web browser. Karena pemrosesan program PHP dilakukan di lingkungan web server, PHP dikatakan sebagai bahasa sisi server (*server-side*). Oleh sebab itu, PHP tidak akan terlihat pada saat user memilih perintah "View Source" pada web browser yang digunakan.

PHP diciptakan oleh Rasmus Lerdorf, seorang pemrogram C yang andal. Semula PHP hanya digunakan untuk mencatat jumlah pengunjung pada homepagenya. Rasmus adalah salah seorang pendukung open source. Karena itulah ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis [13].

Kelebihan-kelebihan PHP:

1. PHP mudah dibuat dan cepat dijalankan.

2. PHP dapat berjalan dalam web server yang berbeda dan dalam system operasi yang berbeda pula.
3. PHP diterbitkan secara gratis
4. PHP termasuk bahasa embedded (bisa diletakan di dalam tag HTML)
5. Sebagian besar sintaks PHP mirip dengan sintaks dalam bahasa C dan Perl.

2.2.9.1 Bentuk Script PHP

Fungsi-fungsi yang telah ada di dalam PHP tidak case sensitive (membedakan huruf besar dengan huruf kecil), namun variabelnya case sensitive. Kode PHP diawali dengan tanda lebih kecil (<) dan diakhiri dengan tanda lebih besar (>). Ada tiga cara untuk menuliskan script PHP, yaitu:

1. <?
   Script PHP Anda
   ?>
2. <?php
   Script PHP Anda
   ?>
3. <SCRIPT LANGUAGE = “php”>
   Script PHP Anda
   </SCRIPT>

Contoh penggunaan:

```
<html>
    <body>
    Tanggal:
    <?php
        print(date(“d F Y””));
```

```
        ?>
        </body>
</html>
```

2.2.10 *Cascade Style Sheet* (CSS)

CSS merupakan sebuah bahasa yang digunakan untuk mengatur tampilan sebuah dokumen web yang ditulis dalam *markup language*. Dengan kata lain, CSS merupakan sebuah bahasa yang digunakan untuk mengatur tampilan atau desain suatu halaman HTML. Hal ini diperlukan setelah melihat perkembangan HTML menjadi kurang praktis karena web pages terlalu banyak dibebani hal-hal yang berkaitan dengan factor tampilan seperti font dan lain-lain.

Penggunaan dari CSS dapat dilakukan dengan dua cara. Yang pertama yaitu dengan menggabungkan CSS langsung ke dalam satu file *markup* (*internal*), lalu cara yang kedua adalah dengan memanggil CSS tersebut (*eksternal*). Perbedaan dari keduanya adalah cara menggunakan dan cara pemanggilan CSS tersebut. Jika yang digunakan adalah internal CSS, semua kode CSS dan *markup* dimasukkan dalam sebuah file yang sama dan jika menggunakan eksternal CSS maka perlu dibuat sebuah *link* untuk menghubungkan keduanya [13].

2.2.11 JavaScript

Jika kita berbicara dalam konteks *web*, sederhananya, kita dapat memahami JavaScript sebagai bahasa pemrograman yang berjalan khusus untuk di *browser* atau halaman *web* agar halaman *web* menjadi lebih hidup. Kalau

dilihat dari suku katanya terdiri dari dua suku kata, yaitu Java dan Script. Java adalah Bahasa pemrograman berorientasi objek, sedangkan Script adalah serangkaian instruksi program.

Secara fungsional, JavaScript digunakan untuk menyediakan akses *script* pada objek yang dibenamkan (*embedded*). Contoh sederhana dari penggunaan JavaScript adalah membuka halaman *pop up*, fungsi validasi pada form sebelum data dikirimkan ke server, merubah *image* kursor ketika melewati objek tertentu, dan lain lain [13].

Yang harus diperhatikan dalam pengelolaan pemrograman JavaScript, diantaranya:

1. JavaScript adalah "case sensitive", yang artinya JavaScript membedakan huruf besar dan huruf kecil. Jika Anda pernah belajar bahasa pemrograman seperti Turbo C atau C++, maka sama seperti bahasa pemrograman tersebut, dimana huruf T tidak sama dengan huruf t.
2. Menggunakan blok awal "{" dan blok akhir "}".
3. Extension umumnya menggu nakan "*.js".
4. Jika program dalam satu baris terlalu panjang dapat disambung ke baris berikut dengan karakter "\".

2.2.12 JQuery

JQuery merupakan suatu *framework* (library) Javascript yang menekankan bagaimana interaksi antara Javascript dan HTML untuk mempercepat pengembangan web. JQuery project terbentuk pada September

tahun 2009 oleh John Resig. Pada perkembangannya JQuery tidak sekedar sebagai framework Javascript, namun memiliki kehandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak developer web menggunakannya. JQuery memiliki slogan "**Write less, do more**" yang kurang lebih maksudnya adalah kesederhanaan dalam penulisan code, tapi dengan hasil yang lebih banyak [17].

Dari sisi ukurannya, framework JQuery sungguh ramping, hanya sekitar 20 KB dan hanya terdiri dari satu file. Namun demikian, bagi yang menginginkan fungsi lebih, JQuery memungkinkan penambahan fungsionalitas dalam bentuk plugin. Saat ini tersedia ribuan plugin yang dapat diperoleh secara gratis di internet.

2.2.13 *Web Service*

Web Service adalah sekumpulan *application logic* beserta objek-objek dan metode-metode yang dimilikinya yang terletak di suatu server yang terhubung ke internet sehingga dapat diakses menggunakan protocol HTTP dan SOAP (*Simple Object Access Protocol*). Dalam penggunaannya, web service dapat digunakan dari hanya untuk memeriksa data user yang login ke sebuah website ataupun untuk digunakan pada transaksi perbankan online yang rumit.

Tujuan dari teknologi ini adalah untuk memudahkan beberapa aplikasi atau komponennya untuk saling berhubungan dengan aplikasi lain dalam sebuah organisasi maupun diluar organisasi menggunakan standar yang tidak terikat platform (*platform-neutral*) dan tidak terikat akan bahasa pemrograman yang digunakan (*language-neutral*). Hal tersebut dapat terjadi karena penggunaan

XML standar yang didukung oleh banyak perusahaan besar di dunia, yang digunakan untuk bertukar data. Selain daripada itu, penggunaan SOAP menjadikan metode-metode dari objek-objek yang ada dalam sebuah web service dapat di akses dari aplikasi lain seperti halnya aplikasi tersebut mengakses metode lokal [21].

2.2.14 *JavaScript Object Notation* (JSON)

JSON 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 pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [18].

2.2.15 *Global Positioning System* (GPS)

GPS merupakan sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit. Sistem yang pertama kali dikembangkan oleh Departemen Pertahanan Amerika ini dugunakan untuk kepentingan militer maupun sipil (survey dan pemetaan).

Sistem GPS yang nama aslinya NAVSTAR GPS (*Navigation Satellite Timing and Ranging Global Positioning System),* mempunyai tiga segmen yaitu: satelit, pengontrol, dan penerima/pengguna. Satelit GPS yang mengorbit bumi, dengan orbit dan kedudukan yang tetap (koordinatnya pati), seluruhnya

berjumlah 24 buah dimana 21 aktif bekerja dan 3 buah sisanya adalah cadangan [16].

1. Satelit bertugas untuk menerima dan menyimpan data yang ditransmisikan oleh stasiun-stasiun pengontrol, menyimpan dan menjaga informasi waktu berketelitian tinggi (ditentukan dengan jam atomic di satelit), dan memancarkan sinyal dan informasi secara kontinyu ke pesawat penerima (receiver) dari pengguna.
2. Pengontrol bertugas mengendalikan dan mengontrol satelit dari bumi baik untuk mengecek kesehatan satelit, penentuan dan prediksi orbit dan waktu, sinkronisasi antar satelit, dan mengirim data ke satelit.
3. Penerima bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi (posisi tiga dimensi yaitu koordinat di bumi plus ketinggian), arah, jarak, dan waktu yang diperlukan oleh pengguna.

2.2.16 Google Maps

Google Maps merupakan sebuah layanan peta dunia virtual berbasis web yang disediakan Google. Layanan ini gratis dan bisa ditemukan serta diakses di http://maps.google.com. Google Maps menawarkan peta yang dapat digeser (*panned*), diperbesar (*zoom in*), diperkecil (*zoom out*) dan dapat diganti dalam beberapa mode, fitur pencarian rute (*routing*), penunjuk arah dari satu objek peta ke objek yang lain (*direction*), dan juga pencari tempat (*place*).

Google Maps masih berada dalam tahap beta, dan masih terus dikembangkan dengan data yang selalu diperbaharui secara berkala [16].

Ada jenis – jenis mode tampilan yang bisa dipilih dari Google Maps, yaitu:

1. *Map* menampilkan peta dalam bentuk peta garis. Map ditunjukan seperti pada gambar di bawah ini.



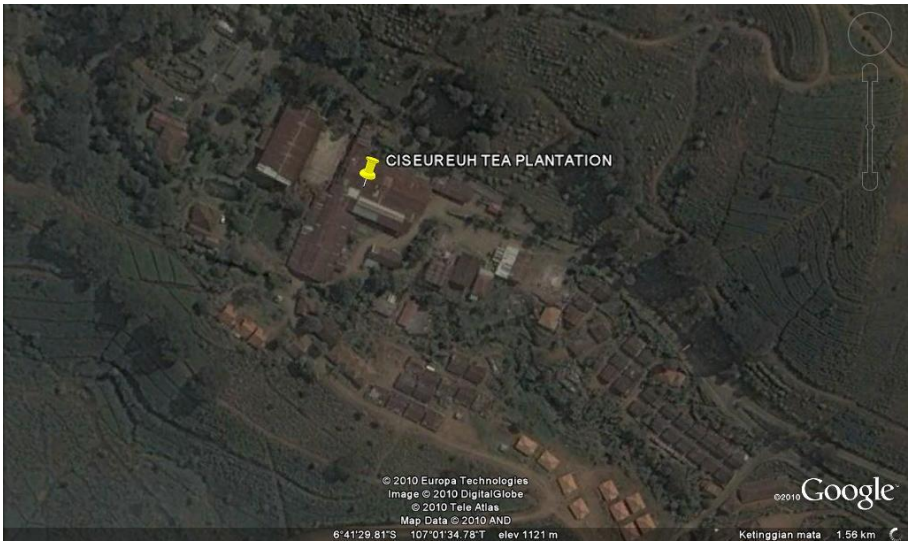
Gambar 2.10 Tampilan Bentuk *Map*

2. *Sattelite* menampilkan peta dalam bentuk citra/foto satelit. Satelit ditunjukan seperti pada gambar di bawah ini.



Gambar 2.11 Tampilan Bentuk *Sattelite*

3. *Earth* menampilkan peta dalam bentuk gabungan map dan satelit. *Earth* ditunjukan seperti pada gambar di bawah ini.



Gambar 2.12 Tampilan Bentuk *Earth*

2.2.17 Google Maps API

Penambahan fitur Google Maps pada suatu web harus menggunakan Google Maps API. Google Maps API adalah *library* JavaScript. Saat ini versi terakhir Google Map API adalah versi 3. Versi ini, diharapkan akan tampil lebih cepat dari versi sebelumnya khususnya untuk browser ponsel [12].

Objek model pada *Google Map API* adalah :

1. *Loading Google Maps API* koneksi *script* yang kita buat ke*server* Google Map API dengan menggunakan *key* yang anda dapatkan pada saat anda mendaftar ke Google Maps API.
2. *Map DOM Elements*

   Untuk menampilkan map pada halaman web, umumnya menggunakan *div tag* untuk menampungnya. Diharuskan membuat sebuah *div tag* dengan nama "*map_canvas*", dan mereferensikan *element* ini ke *Document Object Model* (DOM).

3. GMap2 - *Elementary Object*

   Class javascript yang membuat peta itu ada adalah *class* GMap2, Object dari class ini akan menyediakan sebuah peta di halaman web, *Variable map* akan berikan nilai sebuah object dari *class GMap2*. Fungsi *GMap2* adalah sebagai *constructo*r dan definisinya.

4. Inisialisasi *Map*

   Inisialisasi di proses dengan menggunakan *method setCenter*(). *Method setCenter*() membutuhkan GetLatLng koordinat dan *zoom level,* dan method ini harus segera dikirim sebelum ada pengoperasian lain pada peta, termasuk seting atribut peta itu sendiri.

5. *Loading Map*

   Ketika halaman HTML di *render*, *document object model* (DOM) sudah bisa digunakan, dan semua gambar *external* dan *script* diterima oleh *object document*. Untuk memastikan peta kita dimuat sesudah halaman selesai dimuat oleh *browser*.

6. *Latitudes and Longitudes*

   Objek *GLatLng* menentukan titik koordinat peta yang akan ditampilkan, parameternya terdiri dari lintang/*latitude* dan bujur/*longitude*.

7. Fungsi *Gdirections* adalah *utility* yang menangani masalah pencarian rute optimal dari satu tempat ke tempat lain.

   Cara penulisan Google Map API dengan urutan sebagai berikut:

1. Memasukkan Maps API JavaScript ke dalam HTML kita.
2. Membuat element *div* dengan nama "map_canvas" untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan *event onload*.

Jika ingin mendapatkan fasilitas Google maps pada suatu web, maka harus mendaftarkan web tersebut dengan memberikan alamat web tersebut. Setelah itu Google akan memberikan sebuah API key, key tersebut digunakan pada script dalam tag Javascript

Contoh API key (setiap situs memiliki key yang berbeda):

ABQIAAAA8tt4eKTuBZMVnLJfP2BZrBT2yXp_ZAY8_ufC3CF XhHIE1NvwkxS4Rz1LFzG0odNPtk8VLkdrQF5grA

2.2.18 Pembangun Perangkat Lunak

2.2.18.1 Adobe Dreamweaver CS 5

Dreamweaver merupakan software aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain web secara visual. Macromedia Dreamweaver CS5 merupakan penyempurnaan dari versi sebelumnya dan tentu saja semakin mudah dalam penggunaannya. Oleh karena itu, software ini paling inovatif dan lebih lengkap dibandingkan software web

editor lain. Program Aplikasi Macromedia Dreamweaver CS5 menyertakan banyak perangkat yang berkaitan dengan pengkodean dan fitur.

Keunggulan Macromedia Dreamweaver CS5:

a. Tampilan mudah dimengerti oleh pengguna dari semua tingkatan keahlian, bahkan bagi orang awam sekalipun
b. Dreamweaver merupakan eitor berbasis *WYSIWYG (What you see is what you get)*, artinya apa yang ditampilkan di editor, sama seperti apa yang akan ditampilkan di web browser.
c. Dreamweaver terintegrasi dengan produk macromedia lainnya seperti integrasi dengan Adobe Bussiness catalyst, integrasi Adobe Browser Lab, juga terintegrasi dengan CMS, pengecekan CSS, dan lain-lain.
d. Disertakannya fasilitas pembuatan web album foto, penambahan Flash Paper, dan lain-lain [6].

2.2.18.2 Eclipse

Eclipse adalaha sebuah IDE (Integrated Development Environtment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform independent)*. Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse awalnya diciptakan oleh IBM pada bulan November 2001 dan didukung oleh konsorsium vendor perangkat lunak. Eclipse bersifat *open source* di bawah lisensi dari Eclipse Public Licensi (EPL). Berikut ini adalah versi resmi dari Eclipse [20]:

1. Eclipse Europa Packages (2007 - v 3.3)
2. Eclipse Ganymede Packages (2008 - v 3.4)
3. Eclipse Ganymede SR1 Packages (v 3.4.1)
4. Eclipse Ganymede SR2 Packages (v 3.4.2)
5. Eclipse Galileo Packages (2009 - v 3.5)
6. Eclipse Galileo SR1 Packages (v 3.5.1)
7. Eclipse Galileo SR2 Packages (v 3.5.2)
8. Eclipse Helios Packages (2010 - v 3.6)
9. Eclipse Helios SR1 Packages (v 3.6.1)
10. Eclipse Helios SR2 Packages (v 3.6.2)
11. Eclipse Indigo Packages (2011 - v 3.7)

2.2.18.3 MySQL

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar SQL (*Structured Query Language*).

MySQL merupakan sebuah database server yang *free*¸artinya kita bebas menggunakan database ini untuk keperluan pribadi atau udaha tanpa

harus membeli atau emmbayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius [7].

Kelebihan MySQL:

1. MySQL sebagai DBMS (*Database Management System*)
2. MySQL sebagai RDBMS (*Relational Database Management System*)
3. MySQL adalah sebuah Software database yang *OpenSource*, artinya program ini bersifat gratis atau bebas digunakan oleh siapa saja tanpa harus membeli atau membayar lisensi kepada pembuatnya.
4. MySQL merupakan sebuah database server, jadi dapat menghubungkannya ke media internet sehingga dapat diakses dari jauh.
5. MySQL juga dapat berperan sebagai client, mengakses database pasa server.
6. MySQL mampu menerima query yang bertumpuk dalam satu permintaan atau disebut *Multi-Threading.*
7. MySQL dapat menyimpan data berkapasitas sangat besar hingga berukuran Gigabyte sekalipun.

2.2.18.3.1 *Data Definition Language* (DDL)

DDL adalah sebuah metode query SQL yang berguna untuk mendefinisikan data pada sebuah database, adapun querynya adalah:

1. CREATE: digunakan untuk melakukan pembuatan tabel dan database, contoh: CREATE DATABASE BARU atau
   CREATE TABLE MHS (nama varchar (10)).
2. DROP: digunakan untuk melakukan penghapusan tabel maupun database, contoh: DROP DATABASE BARU atau DROP TABLE MHS.
3. ALTER: digunakan untuk melakukan pengubahan struktur tabel yang telah dibuat, contoh: ALTER TABLE MHS ADD alamat varchar(20).

2.2.18.3.2 *Data Manipulation Language* (DML)

DML adalah sebuah metode query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari query ini adalah untuk melakukan pemanipulasian database yang telah ada atau telah dibuat sebelumnya. Adapun querynya adalah:

1. INSERT: digunakan untuk melakukan penginputan data pada tabel database, contoh: INSERT INTO MHS VALUES (‘budi’, ‘bandung’).
2. UPDATE: digunakan untuk melakukan pengubahan atau peremajaan terhadap data yang ada pada tabel, contoh: UPDATE MHS SET alamat='Jakarta' WHERE nama='budi'.
3. DELETE: digunakan untuk melakukan penghapusan data pada tabel, contoh: DELETE FROM MHS WHERE nama='budi'.

2.2.18.4 *WampServer*

WampServer adalah paket *web server offline* dan diinstal pada sistem operasi Windows. *Wamp* singkatan dari *Windows, Apache, MySQL and*

PHP. Apache adalah sebuah *web server*, MySQL adalah program database server, PHP adalah bahasa scripting yang dapat memanipulasi informasi yang dibuat di database dan menghasilkan halaman web dinamis konten setiap waktu diminta oleh browser. Program lain juga dapat dimasukkan dalam paket, seperti phpMyAdmin yang menyediakan antarmuka pengguna grafis untuk manajer database MySQL.

2.2.18.5 *Java Development Kit* (JDK)

JDK adalah software development kit yang berguna untuk mengembangkan dan menguji program yang ditulis dalam bahasa pemrograman Java yang berjalan pada platform Java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber kompiler java, bundling, debuggers, development libraries dan lain sebagainya [19].

JDK memiliki 3 kelompok pengembangan , yaitu :

a. J2SE (Java 2 Standard Edition, untuk kebutuhan standard)
b. J2EE (Java 2 Enterprise Edition, untuk kebutuhan skala besar)
c. J2ME (Java 2 Mobile Edition, untuk kebutuhan pemrograman mobile)

2.2.18.6 *Software Development Kit* (SDK)

Software Development Kit (SDK) adalah suatu kit atau library dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat

lunak dan biasanya SDK terdiri dari kumpulan tools yang dibutuhkan. Misalnya bahasa pemrograman java, mempunyai SDK yang berisi suatu library yang dapat digunakan untuk membuat suatu aplikasi berbasis java.

2.2.18.7 *Android Development Tool* (ADT)

Android Development Tools (ADT) adalah plugin untuk Eclipse Intergrated Development Environment (IDE) yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android. ADT memperluas kemampuan Eclipse untuk membiarkan para developer lebih cepat dalam membuat proyek baru Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android Framework API, debug aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi. Mengembangkan aplikasi di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi android, karena banyak kemudahan-kemudahan sebagai *tools* yang terintegrasi seperti, custom XML editor, dan debug panel ouput. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.