BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Tempat Penelitian

Pada bagian tinjauan tempat penelitian ini akan dijelaskan beberapa hal mengenai tempat penelitian. Tinjauan penelitian meliputi profil perusahaan, logo perusahaan, visi dan misi, serta struktur organisasi perusahaan.

2.1.1 Profil Perusahaan

Pemerintah pusat dalam mengantur kepariwisataan di seluruh Indonesia mengeluarkan keputusan Presiden No. 30 Tahun 1986 yang berisi tentang pembentukan dewan Pertimbangan Kepariwisataan Nasional tersebut mempunyai tugas utama yaitu membantu Presiden dalam memetapkan kebijaksanaan umum dibidang kepariwisataan nasional. Di dalam keputusan ini dicantumkan pula tentang pembentukkan Direktorat Jendral Pariwisata dalam suatu Struktur Organisasi Departement Perhubungan. Masih pada tahun 1969, pemerintah mengeluarkan intruksi Presiden No. 9 /1969 yang berisi tentang pembentukan Badan Pariwisata Nasional (BAPAPARNAS) dalam rangka untuk menjamin pembinaan dan pembangunan yang efektif dan berlanjut dalam pelaksanaan yang diusahakan oleh pihak pemerintah dan swasta.

Kemudian Pemerintah Daerah Tingkat I Jawa Barat membentuk Dinas Pariwisata Tingkat I Jawa Barat, setelah itu dibentuklah Dinas Pariwisata Kotamadya Tingkat II Bandung berdasarkan peraturan daerah No. II/ 1969, pada tahun 1971 sampai dengan 1987 tercatat dengan nama Kantor Dinas Pariwisata (KAPARDA). Sejak keluarnya peraturan daerah No. II/ 1985 tentang pembentukan Dinas Pariwisata Kotamaya Daerah Tingakat II Bandung, maka sejak tahun 1987 KAPARDA diganti menjadi Dinas Pariwisata Kotamadya Daerah Tingakat II Bandung.[1]

Sejalan dengan adanya otonomi daerah Indonesia, seluruh tingkat Pemerintahan di Indonesia baik di pusat, propinsi, maupun kota dan kabupaten mengalami beberapa perubahan. Bersarkan UU No. 22/ 1999, Dinas Pariwisata daerah Tingkat II Kotamadya mengalami perubahan di antaranya yaitu :

1. Dinas Pariwisata daerah Tingkat II Kotamadya berubah namanya menjadi Dinas Pariwisata Kota Bandung.
2. Dinas Kota Bandung mendapatkan penambahan kewenangan yang lebih luas dalam urusan kepariwisataan di Kota Bandung, terhadap urusan Biro Perjalanan dan Hotel Berbintang.

Sehubungan dengan hal tersebut di atas, dalam upaya memenuhi kewenangan yang iberikan berdasarkan UU No. 22/ 1999 struktur organisasi dan tata kerja Dinas Pariwisata Kota Bandung pun mengalami beberapa Perubahan yang diatur dalm Peraturan Daerah No. 5 tanggal 7 maret 2001 mengenai struktur organisasi Dinas Pariwisata Kota Bandung.

Selain Itu Dinas Kebudayaan Dan Pariwisata Kota Bandung Telah Memiliki Situs Internet Yang Dapat Di Akses Oleh Masyarakat Yaitu www.bandungtourism.com Situs Ini Berisi Informasi Mengenai Kegiatan-Kegiatan Pariwisata Di Kota Bandung.

2.1.2 Logo Perusahaan

Logo dari Dinas Kebudayaan dan Pariwisata Kota Bandung dapat dilihat pada Gambar 2.1.



Gambar 2.1 Logo Dinas Kebudayaan dan Pariwisata Kota Bandung [2]

Logo merupakan hal terpenting dalam suatu instansi atau perusahaan. Karena logo sebagai identitas instansi atau perusahaan agar masyarakatmengetahui keberadaan instansi atau perusahaan tersebut. Sama halnya dengan Dinas Pariwisata dan Kebudayaan Kota Bandung yang meiliki logo sebagi identitas kelembagaan. Namun, Dinas Pariwisata dan Kebudayaan Kota Bandung merupakan lembaga yang berada dibawah naungan Gubernur Jawa Barat yang termasuk dalam perangkat pemerintah provinsi sehingga menggunakan logo provinsi Jawa Barat .

2.1.2.1 Arti Logo

Dari gambar 1.2 dapat dijelaskan makna bentuk dan motif yang terdapat dalam logo tersebut, ialah: [1]

1. Bentuk bulat telur pada lambang Jawa Barat berasal dari bentuk perisai sebagai penjagaan diri.
2. Ditengah-tengah terlihat ada sebilah kujang. Kujang ini adalah senjata khas suku bangsa Sunda yang merupakan penduduk asli Jawa Barat. Lima lubang pada kujang melambangkan dasar Negara Indonesia yaitu Garuda Pancasila.
3. Padi satu tangkai yang terdapat di sisi sebelah kiri melambangkan bahan makanan pokok masyarakat Jawa Barat sekaligus juga melambangkan kesuburan pangan, dan jumlah padi menggambarkan tanggal proklamasi Republik indonesia.
4. Kapas satu tangkai yang berada di sebelah kanan melambangkan kesuburan sandang, dan 8 kuntum bunga menggambarkan bulan proklamasi Republik Indonesia.
5. Gunung yang terdapat di bawah padi dan kapas melambangkan bahwa daerah Jawa Barat terdiri atas daerah pegunungan.
6. Sungai dan terusan yang terdapat di bawah gunung sebelah kiri melambangkan di Jawa Barat banyak terdapat sungai dan saluran air yang sangat berguna untuk pertanian.

7. Petak-petak yang terdapat di bawah gunung sebelah kanan melambangkan banyaknya pesawahan dan perkebunan. Masyarakat Jawa Barat umumnya hidup mengandalkan kesuburan tanahnya yang diolah menjadi lahan pertanian.
8. Dam atau bendungan yang terdapat di tengah-tengah bagian bawah antara gambar sungai dan petak, melambangkan kegiatan di bidang irigasi yang merupakan salah satu perhatian pokok mengingat Jawa Barat merupakan daerah agraris. Hal ini juga melambangkan damdam yang berada di Jawa Barat seperti Waduk Jatiluhur.
9. Motto Jawa Barat Gemah Ripah Repeh Rapih, merupakan kata majemuk yang memiliki arti Jawa barat merupakan daerah yang kaya raya dan subur makmur serta didiami oleh banyak penduduk yang hidup rukun dan damai.

2.1.2.2 Arti Warna Logo

Selain arti dari bentuk logo terdapat pula berbagai warna, yaitu: hijau,kuning, hitam, biru, merah dan putih. Warna - warna tersebut memiliki arti khusus yang dapat dijelaskan sebagai berikut : [1]

1. Warna hijau artinya melambangkan kesuburan dan kemakmuran tanah Jawa Barat.
2. Kuning artinya melambangkan keagungan, kemuliaan, kekayaan.
3. Hitam artinya melambangkan keteguhan dan keabadian.
4. Biru artinya melambangkan ketentraman atau kedamaian.
5. Merah artinya melambangkan keberanian.
6. Putih artinya melambangkan kemurnian, kesucian atau kejujuran.

2.1.3 Visi dan Misi

Berikut ini merupakan visi dan misi dari Dinas Kebudayaan dan Pariwisata Kota Bandung : [2]

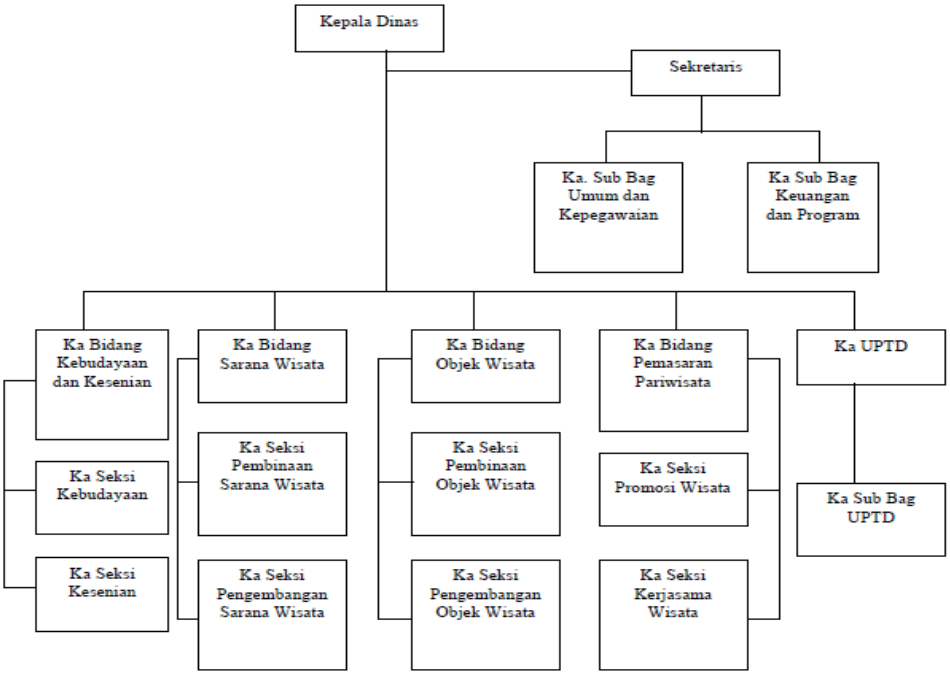
Visi : Memantapkan Kota Bandung sebagai Kota Seni Budaya dan Tujuan Wisata Tahun 2016

Misi :

1. Mengembangkan Sumber Daya Manusia dan Kelembagaan Kepariwisataan yang profesional, berkarakteristik sunda dan berwawasan global;
2. Meningkatkan pelestarian, pemberdayaan, pengembangan dan pemanfaatan kebudayaan dan kesenian;
3. Mengembangkan industri pariwisata yang kreatif, inovatif dengan memperhatikan terlaksananya sapta pesona;
4. Meningkatkan destinasi pariwisata kota yang berdaya saing tinggi baik pada tingkat regional, nasional maupun internasional;
5. Meningkatkan pemasaran melalui kemitraan dan kerjasama budaya dan pariwisata dengan pemangku kepentingan dan/atau Kab/kota/negara lain.

2.1.4 Struktur Organisasi Perusahaan

Berdasarkan Peraturan Daerah Kota Bandung Nomor 13 Tahun 2007 tentang Pembentukan dan Susunan Organisasi Dinas Daerah, Struktur Organisasi Dinas Kebudayaan dan Pariwisata Kota Bandung adalah sebagai berikut:



Gambar 2.2 Struktur Organisasi Dinas Kebudayaan dan Pariwisata Kota Bandung [2]

2.1.5 Deskripsi Pekerjaan Perusahaan

1. Kepala Dinas

1) Kepala Dinas Kebudayaan dan Pariwisata mempunyai tugas pokok melaksanakan urusan pemerintah di bidang kebudayaan dan pariwisata berdasarkan asas otonomi dan pembantuan.
2) Dalam melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Kepala Dinas Kebudayaan dan Pariwisata mempunyai fungsi:
   a. Perumusan kebijkana teknis lingkup kebudayaan dan kesenian, sarana wisata, objek wisata dan pemasaran pariwisata
   b. Penyelenggaraan urusan pemerintah dan pelayanan umum di bidang kebudayaan dan kesenian, sarana wisata, objek wisata dan pemasaran pariwisata;
   c. Pembinaan dan pelaksanaan di bidang kebudayaan dan kesenian, sarana wisata, objek wisata dan pemasaran pariwisata
   d. Pelaksanaan tugas lain yang diberikan Walikota sesuai dengan tugas dan fungsinya.
   e. Pembinaan, monitoring, evaluasi dan pelaporan penyelenggaraan kegiatan Dinas.

2. Sekretariat

1) Sekretariat mempunyai tugas pokok melaksanakan sebagian tugas Dinas Kebudayaan dan Pariwisata lingkup kesekretariatan.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), sekretariat mempunyai fungsi:
   a. Pelaksanaan dan penyusunan rencana kesekretariatan;
   b. Pelaksanaan pelayanan administratif kesekretariatan Dinas yang meliputi administrasi umum dan kepegawaian serta keuangan dan program Dinas;
   c. Pengkoordinasian penyelenggaraan tugas-tugas Bidang; dan
   d. Pembinaan, monitoring, evaluasi, pelaporan dan pengendalian kegiatan kesekretariatan dinas.

3. Sub Bagian Umum dan Kepegawaian

1) Sub Bagian Umum dan Kepegawaian mempunyai tugas pokok melaksanakan sebagian tugas Sekretariat lingkup umum dan kepegawaian.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Sub Bagian Umum dan Kepegawaian mempunyai fungsi:

a. Penyusunan rencana dan program pengelolaan administrasi umum dan administrasi kepegawaian
b. Pelaksanaan pengelolaan administrasi umum yang meliputi pengelolaan naskah dinas, penataan kearsipan dinas, penyelenggaraan kerumahtanggaan dinas, pengelolaan perlengkapan dan administrasi perjalanan Dinas;
c. Pelaksanaan administrasi kepegawaian yang meliputi penyusunan rencana, penyusunan bahan, pemprosesan, pengusulan dan pengelolaan data mutas, cuti, disiplin, pengembangan pegawai dan kesejahteraan pegawai
d. Evaluasi dan pelaporan pelaksanaan kegiatan administrasi umum dan administrasi kepegawaian.

4. Sub Bagian Keuangan dan Program

1) Sub Bagian Keuangan dan Program mempunyai tugas pokok melaksanan sebagian tugas Sekretariat lingkup keuangan dan program.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Sub Bagian Keuangan dan Program mempunyai fungsi
   a. Penyusunan rencana dan program pengelolaan administrasi keuangan dan program Dinas
   b. Pelaksanaan pengelolaan administrasi keuangan meliputi penyusunan vahan dan koordinasi penyusunan rencana anggaran Dinas, koordinasi pengelolaan dan pengendalian keuangan
   c. Pelaksanaan pengelolaan program Dinas meliputi penyusunan vahan dan koordinasi penyusunan rencana dan prgoram kegiatan Dinas serta fasilitasi pengkoordinasian penyusunan laporan
   d. Evaluasi dan pelaporan pelaksanaan lingkup administrasi keuangan dan program Dinas.

5. Bidang Kebudayaan dan Kesenian

1) Bidang Kebudayaan dan Kesenian mempunyai tugas pokok melaksanakan sebagian tugas Dinas lingkup kebudayaan dan kesenian.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Bidang Kebudayaan dan Kesenian mempunyai fungsi:
   a. Penyusunan rencana dan program lingkup kebudayaan dan kesenian;
   b. Penyusunan petunjuk teknis lingkup kebudayaan dan kesenian;
   c. Pelaksanaan lingkup kebudayaan dan kesenian;

d. Pengkajian rekomendasi, pembinaan dan pengawasan penyelenggaraan perfilman serta pengiriman dan penerimaan misi kesenian; dan
e. Pembinaan, monitoring, evaluasi dan pelaporan lingkup kebudayaan dan kesenian.

6. Seksi Kebudayaan

1) Seksi Kebudayaan mempunyai tugas pokok melaksanakan sebagian tugas Bidang Kebudayaan dan Kesenian lingkup kebudayaan.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Seksi Kebudayaan mempunyai fungsi:
   a. Pengumpulan dan penganalisaan data lingkup kebudayaan;
   b. Penyusunan bahan petunjuk teknis lingkup kebudayaan;
   c. Pelaksanaan lingkup kebudayaan yang meliputi kegiatan pelesatarian, pengelolaan dan pemeliharaan sejarah, museum, kepurbakalaan, bahasa, sastra dan penanaman nilai tradisi, pembinaan karakter dan pekerti bangsa, fasilitasi kerjasama dan kemitraan bidang kebudayaan, serta pengembangan jaringan informasi kebudayaan;
   d. Pelaksanaan pengawasan dan pengendalian penyelenggaraan perfilman
   e. Evaluasi dan pelaporan pelaksanaan lingkup kebudayaan.

7. Seksi Kesenian

1) Seksi Kesenian mempunya tugas pokok melaksanakan sebagian tugas Bidang Kebudayaan dan Kesenian lingkup kesenian.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Seksi Kesenian mempunyai fungsi:
   a. Pengumpulan dan penganalisaan data lingkup kesenian;
   b. Penyusunan bahan petunjuk teknis lingkup kesenian;
   c. Pembinaan, pengewasan dan penyiapan bahan rekomendasi penyelenggaraan pengiriman dan penerimaan delegasi asing di bidang kesenian; dan
   d. Evaluasi dan pelaporan pelaksanaan lingkup kesenian.

8. Bidang Sarana Wisata

1) Bidang Sarana Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Pengendalian lingkup Sarana Wisata.

2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Bidang Sarana Wisata mempunyai fungsi:
   a. Penyusunan rencana dan program lingkup pembinaan sarana wisata dan pengembangan sarana wisata;
   b. Penyusunan petunjuk teknis lingkup pembinaan sarana wisata dan pengembangan sarana wisata
   c. Pelaksanaan lingkup pembinaan sarana wisata dan pengembangan sarana wisata; dan
   d. Pembinaan, monitoring, evaluasi dan pelaporan lingkup pembinaan sarana wisata dan pengembangan sarana wisata.

9. Seksi Pembinaan Sarana Wisata

1) Seksi Pembinaan Sarana Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Sarana Wisata lingkup Pembinaan Sarana Wisata.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Seksi Pembinaan Sarana Wisata mempunyai fungsi:
   a. Pengumpulan dan penganalisaan data lingkup pembinaan sarana wisata;
   b. Penyusunan bahan petunjuk teknis lingkup pembinaan sarana wisata;
   c. Pelaksanaan lingkup pembinaan sarana wisata yang meliputi inventarisasi potensi sarana wisata, dan pembinaan pengelolaan sarana wisata; dan
   d. Evaluasi dan pelaporan pelaksanaan lingkup pembinaan sarana wisata.

10. Seksi Pengembangan Sarana Wisata

1) Seksi Pengembangan Sarana Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Sarana Wisata lingkup pengembangan sarana wisata.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Seksi Pengembangan Sarana Wisata mempunyai fungsi:
   a. Pengumpulan dan penganalisaan data lingkup pengembangan sarana wisata;
   b. Penyusunan bahan petunjuk teknis lingkup pengembangan sarana wsiata;
   c. Pelaksanaan lingkup pengembangan sarana wisata yang meliputi pembinaan pengembangan sarana wisata; dan
   d. Evaluasi dan pelaporan pelaksanaan lingkup pengembangan sarana wiata.

11. Bidang Objek Wisata

1) Bidang Objek Wisata mempunyai tugas pokok melaksanakan sebagian tugas Dinas lingkup Objek Wisata.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Bidang Objek Wisata mempunyai fungsi:
   a. Penyusunan rencana dan program lingkup pembinaan objek wisata dan pengembangan objek wisata;
   b. Penyusunan petunjuk teknis lingkup pembinaan objek wisata dan pengembangan objek wisata;
   c. Pelaksanaan lingkup pembinaan objek wisata dan pengembangan objek wisata; dan
   d. Pembinaan, monitoring, evaluasi dan pelaporan lingkup pembinaan objek wista dan pengembangan objek wisata.

12. Seksi Pembinaan Objek Wisata

1) Seksi Pembinaan Objek Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Objek Wisata lingkup pembinaan objek wisata.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Seksi Pembinaan Objek Wisata mempunyai fungsi:
   a. Pengumpulan dan penganalisaan data lingkup pembinaan objek wisata;
   b. Penyusunan bahan petunjuk teknis lingkup pembinaan objek wisata;
   c. Pelaksanaan lingkup pembinaan objek wisata yang meliputi inventarisasi objek wisata, pembinaan dan pengelolaan objek wista; dan
   d. Evaluasi dan pelaporan pelaksanaan lingkup pembinaan objek wisata.

13. Seksi Pengembangan Objek Wisata

1) Seksi Pengembangan Objek Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Objek Wisata lingkup Pengembangan Objek Wisata.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Seksi Pengembangan Objek Wista mempunyai fungsi:
   a. Pengumpulan dan penganalisaan data lingkup pengembangan objek wisata;
   b. Penyusunan bahan petunjuk teknis lingkup pengembangan objek wista;
   c. Pelaksanaan lingkup pengembangan objek wisata yang meliputi melaksanakan kajian dan pengembangan objek wisata; dan

14. Bidang Pemasaran Pariwisata

1) Bidang Pemasaran Pariwisata mempunyai tugas pokok melaksanakan sebagian tugas Dinas lingkup pengwasan dan yustisi pemasaran pariwisata.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Bidang Pemasaran Pariwisata mempunyai fungsi:
   a. Penyusunan rencana dan program lingkup promosi pariwista dan kerjasama wisata;
   b. Penyusunan petunjuk teknis lingkup promosi pariwisata dan kerjasama wisata;
   c. Pelaksanaan lingkup promosi pariwisata dan kerjasama wisata; dan
   d. Pembinaan, monitoring, evaluasi dan pelaporan lingkup promosi pariwisata dan kerjasama wisata.

15. Seksi Promosi Pariwisata

1) Seksi Promosi Pariwisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Pemasaran Pariwisata lingkup promosi pariwisata.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Seksi Promosi Pariwisata mempunyai fungsi:
   a. Pengumpulan dan penganalisaan data lingkup promosi pariwista;
   b. Penyusunan bahan petunjuk teknis lingkup promosi pariwista;
   c. Pelaksanaan lingkup promosi pariwisata yang meliputi fasilitasi penyelenggaraan pameran/event dan sarana pemasaran wisata, dan penyediaan informasi pariwisata; dan
   d. Evaluasi dan pelaporan pelaksanaan lingkup promosi pariwisata.

16. Seksi Kerjasama Wisata

1) Seksi Kerjasama Wisata mempunyai tugas pokok melaksanakan sebagian tugas Bidang Pemasaran Pariwisata lingkup kerjasama wisata.
2) Untuk melaksanakan tugas pokok sebagaimana dimaksud pada ayat (1), Seksi Kerjasama Wisata mempunyai fungsi:
   a. Pengumpulan dan penganalisaan data lingkup kerjasama wisata;
   b. Penyusunan bahan petunjuk teknis lingkup kerjasama wisata;
   c. Pelaksanaan lingkup kerjasama wisata yang meliputi fasilitasi kerjasama dan kemitraan dalam melaksanakan pengembangan dan promosi pariwisata; dan evaluasi dan pelaporan pelaksanaan lingkup kerjasama wisata.

2.2 Landasan Teori

Landasan teori menjelaskan beberapa teori yang berkaitan dengan permasalahan yang akan dibahas sebagai pemahaman sistem yang akan dibangun.

2.2.1 Aplikasi

Aplikasi merupakan program yang berisikan perintah-perintah untuk melaksanakan pengolahan data, aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Aplikasi (*application*) juga adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word dan Microsoft Excel [3].

Dari pengertian di atas dapat disimpulkan bahwa aplikasi merupakan *software* yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Selain itu aplikasi dapat mengintegrasikan berbagai kemampuan komputer.

2.2.2 Android



Gambar 2.3 Logo 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 Aliance* yang merupakan konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, 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 [4].

Di dunia terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service* (GMS) dan kedua adalah yang benar – benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon selular ini diproduksi oleh HTC *Corporation* dan tersedia di pasaran pada 5 Januari 2008. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam progam kerja Android ARM *Holdings*, Atheros *Communication*, diproduksi oleh Asustek *Computer* Inc, Garmin Ltd, Softbank, Sony Ericson, 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 pembaharuan berupa perbaikan *bug* dan penambahan fitur baru.

2.2.2.1 Android SDK (*Software Development Kit*)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java.

2.2.2.2 ADT (*Android Development Tools*)

Android Develpoment Tools adalah plugin yang di desain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya.

2.2.2.3 Arsitektur Android

Secara garis besar Arsitektur Android dapat di jelaskan dan di gambarkan sebagai berikut [4]:

1) *Application* dan *Widgets*

Application dan *Widgets* adalah *layer* dimana *user* berhubungan dengan aplikasi saja, dimana biasanya user men-*download* aplikasi, melakukan instalasi dan menjalankan aplikasi.

2) *Application Frameworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau member kemampuan untuk membangun aplikasi yang inovatif.

3) *Libraries*

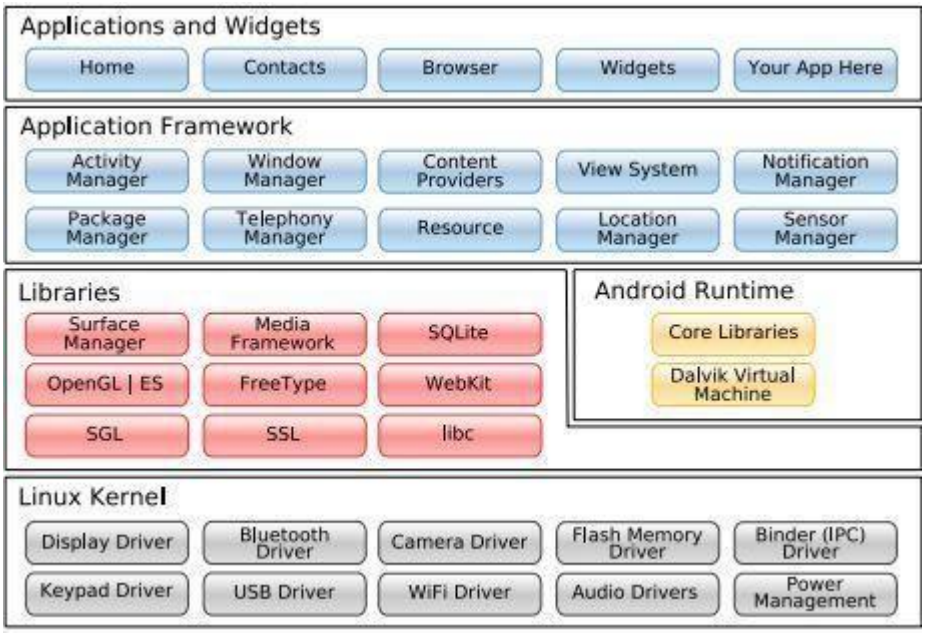
Libraries adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4) Android *Runtime*

Layer yang membuat aplikasi Android dapat djalankan dimana dalam prosesnya menggunakan Implementasi Linux.

5) Linux Kernel

Linux Kernel adalah *layer* dimana inti dari *operating system* dari Android itu berada.



Gambar 2.4 Arsitektur Android

2.2.2.4 Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di-*package* oleh *tools* yang dinamakan "*apt tools*" ke dalam paket Android sehingga menghasilkan file dengan ekstensi apk. Ada empat jenis komponen pada aplikasi Android yaitu :

1) *Activities*
2) *Service*
3) *Broadcast Receiver*
4) *Content Provider*

2.2.2.5 Versi Android

1. Android Versi 1.1
2. Android Versi 1.5 (*Cupcake*)
3. Android Versi 1.6 (*Donut*)
4. Android Versi 2.0/2.1 (*Éclair*)
5. Android Versi 2.2 (*Froyo*)

6. Android Versi 2.3 (*Gingerbread*)
7. Android Versi 3.0 (*Honeycomb*)
8. Android Versi 4.0 (*Ice Cream Sandwich*)
9. Android Versi 4.1 (*Jelly Bean*)
10. Android Versi 4.4 (*KitKat*)
11. Android Versi 5.0 (*Lollipop*)
12. Android Versi 6.0 (*Marshmallow*)

2.2.3 *Firebase Cloud Messaging*



Gambar 2.5 *Firebase*

Firebase Cloud Messaging (FCM) adalah solusi perpesanan lintas-platform yang memungkinkan Anda mengirimkan pesan dan pemberitahuan dengan terpercaya tanpa biaya. Dengan menggunakan FCM, Pengguna bisa memberi tahu aplikasi klien bahwa email baru atau data lainnya tersedia untuk disinkronkan. Pengguna 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. Implementasi FCM mencakup server aplikasi dalam lingkungan pengguna yang berinteraksi dengan FCM lewat protokol HTTP atau XMPP, dan aplikasi klien. Selain itu, FCM menyertakan konsol Notifications, yang dapat Anda gunakan 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, Pengguna bisa menggunakan Notifications. Untuk penerapan dengan persyaratan pesan yang lebih kompleks, FCM adalah pilihan yang tepat.

2.2.3.1 Fungsi utama *Firebase Cloud Messaging* (FCM)

Penargetan pesan serba guna	Distribusikan pesan kepada aplikasi klien Anda 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 Anda melalui saluran koneksi FCM yang bisa diandalkan dan hemat baterai.

2.2.3.2 *Firebase Notifications*

Firebase Notifications adalah layanan gratis yang memungkinkan pemberitahuan pengguna yang ditargetkan untuk pengembang aplikasi seluler. Dibangun di atas *Firebase Cloud Messaging* dan FCM SDK, *Firebase Notifications* (*Notifications*) menyediakan opsi bagi pengembang dan organisasi yang mencari platform pemberitahuan fleksibel yang mengharuskan upaya pengkodean minimal untuk memulai, dan konsol grafis untuk mengirim pesan. Dengan GUI konsol *Notifications*, Anda dapat kembali menumbuhkan interaksi dan mempertahankan basis pengguna, membantu perkembangan aplikasi, dan mendukung kampanye pemasaran.

2.2.3.3 Fungsi Utama *Firebase Notifications*

Notifications mengintegrasikan dengan erat *Firebase Analytics*, sehingga Anda dapat menargetkan pemberitahuan menurut pengguna khusus. Selain itu, Anda bisa menargetkan segmen pengguna yang telah ditetapkan sebelumnya untuk aplikasi, versi, dan bahasa. Dengan pemberitahuan yang ditargetkan ke segmen pengguna, Anda dapat mengontak khalayak pengguna yang tepat untuk pembaruan pada peningkatan yang tersedia, fitur-fitur baru, atau berita lainnya.

Notifications analytics	Menganalisis pelibatan kembali konversi dengan kumpulan *analytics* pemberitahuan bawaan dan analisis corong.
Penargetan pesan serba guna	Targetkan pengguna dalam segmen pengguna yang telah ditetapkan sebelumnya, khalayak *analytics* khusus, klien yang berlangganan ke topik, dan satu perangkat.
Penjadwalan pesan fleksibel	Mengirim pemberitahuan (hingga 2 kb), saat itu juga, atau di waktu mendatang menggunakan waktu lokal klien

2.2.3.4 Cara Kerja *Firebase Notifications*

Mengirim pemberitahuan gunakan GUI konsol *Notifications* untuk menulis dan mengirim pemberitahuan ke semua sasaran pesan yang didukung. *Firebase Cloud Messaging* menangani rute dan penyerahan ini ke perangkat yang ditargetkan.

2.2.4 Java



Gambar 2.6 Logo Java

2.2.4.1 Definisi 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 berbasis *web* [6].

2.2.4.2 Sejarah perkembangan Java

Bahasa pemrograman Java terlahir dari *The Green Project*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road* di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program *Java Oak* pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*7" (*Star Seven*).

Setelah era *Star Seven* selesai, sebuah anak perusahaan Tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto.

Perusahaan baru ini bertambah maju, jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer.

Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinsipirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995.

Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace. Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape.

Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi *release* Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah *Java*).

2.2.4.3 Versi Java

Versi awal Java ditahun 1996 sudah merupakan versi *release* sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya :

1. java.lang : Peruntukan kelas elemen-elemen dasar.
2. java.io : Peruntukan kelas *input* dan *output*, termasuk penggunaan berkas.
3. java.util : Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
4. java.net : Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. java.awt : Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
6. java.applet : Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah *web*.

2.2.4.4 Kelebihan Java

1. ***Multiplatform***. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*/sistem operasi komputer, sesuai dengan prinsip *tulis sekali, jalankan di mana saja*. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin/*bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan.

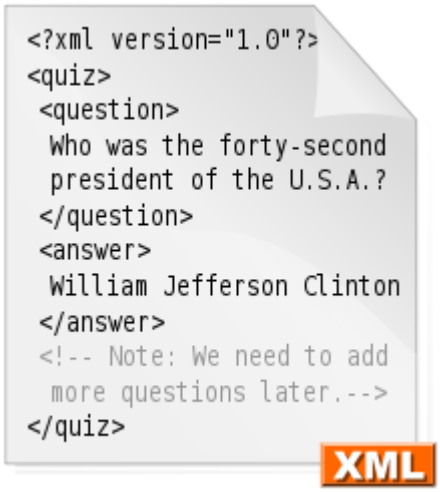
2. **OOP** (*Object Oriented Programming* - Pemrogram Berorientasi Objek)
3. **Perpustakaan Kelas Yang Lengkap**, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
4. **Bergaya C++**, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. **Pengumpulan Sampah Otomatis**, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.2.4.5 Kekurangan Java

1. **Tulis sekali, jalankan di mana saja** - Masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan *platform* lain. Untuk J2SE, misalnya *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.
2. **Mudah didekompilasi**. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-*reverse-engineer*.

3. **Penggunaan memori yang banyak**. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan *Object* Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.2.5 XML



Gambar 2.7 Contoh Dokumen XML

2.2.5.1 Definisi XML

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet [7].

2.2.5.2 Pengenalan XML

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang *view database*, tetapi dengan suatu cara yang standar.

2.2.5.3 Tipe XML

XML memiliki tiga tipe file :

1. XML, merupakan standar format dari struktur berkas (*file*).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD, merupakan standar yang mendefinisikan struktur *database* dalam XML.

2.2.6 GPS (*Global Positioning System*)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem untuk menentukan posisi dan navigasi secara global dengan menggunakan satelit. Sistem yang pertama kali dikembangkan oleh departemen pertahanan Amerika ini digunakan untuk kepentingan militer maupun sipil (survei dan pemetaan).

Sistem GPS, yang nama aslinya adalah NAVSTAR GPS (*Navigaion Satellite Timing and Ranging Global Positioning System*) mempunyai tiga segmen yaitu : satelit, pengontrol, dan penerima/pengguna. Satelit GPS yang mengorbit bum. Dengan orbit dan kedudukan yang tetap (koordinat pasti), seluruhnya berjumlah 24 buah dimana 21 buah aktif berkerja dan 3 buah sisanya adalah cadangan.

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 untuk mengendalikan dan mengontrol satelit dari bumi baik mengecek kesehatan satelit. Penentuan dan prediksi orbit dan waktu, sinkronisasi waktu antar satelit, dan mengirim data ke satelit
3. Penerima bertugas menerima data dari satelit dan memprosesnya untuk menetukan (posisi tiga dimensi yaitu koordinat di bumi dan ketinggian), arah, jarak, dan waktu yang diperluka oleh pengguna. Ada dua macam tipe *receiver* :
    1) Tipe Navigasi Antara lain : *Trimble Ensign*, *Trimble Pathfinder*, *Garmin*, *Sony*, dan lain sebagainya
    2) Tipe Geodetic Antara lain : *Topcon*, *Leica*, *Astech*, *Trimble* seri 4000, dan lain sebagainya

2.2.6.1 Akurasi GPS

Posisi yang ditunjukkan oleh suatu GPS mempunyai faktor kesalahan atau juga disebut tingkat akurasi. Sebagai contoh suatu alat GPS menunjukkan titik koordinat dengan tingkat akurasi 5 meter, itu berarti posisi pengguna bisa berada dalam range radius 5 meter dari titik yang ditunjukkan tersebut. Mengapa tingkat akurasi yang terlihat bisa berubah-ubah? Kadang terlihat 10 meter, 15 meter, atau 5 meter. Ada beberapa hal yang mempengaruhi tingkat akurasi tersebut, antara lain:

1. Kesalahan Ephemeris. Terjadi jika satelit tidak dapat mentransmisikan posisinya di orbit dengan tepat.
2. Keadaan Ionosphere. Ionosphere berada pada jarak sekitar 43-50 mil di atas permukaan bumi.Walaupun GPS receiver berusaha untuk mengkoreksi/memperbaiki faktor keterlambatan yang terjadi tetap saja aktivitas tertentu dari plasma bisa menyebabkan kesalahan perhitungan.
3. Keadaan Troposphere. Troposphere adalah bagian terendah dari atmosfer sampai dengan ketinggian sekitar 11 mil dari permukaan tanah. Variasi pada

temperatur, tekanan, dan kelembaban bisa menyebabkan perbedaan kecepatan penerimaan gelombang radio.

4. Kesalahan Waktu. Karena penempatan jam atom pada setiap GPS receiver tidak berjalan sebagaimana mestinya. Kesalahan waktu dari GPS receiver yang tidak presisi dapat menimbulkan ketidakakurasian.
5. Kesalahan Multipath. Terjadi karena sinyal satelit membentur permukaan keras (seperti bangunan atau tebing) sebelum mencapai GPS receiver. Hal tersebut bisa menyebabkan terjadinya delay sehingga perhitungan jarak menjadi tidak akurat.
6. Buruknya Sinyal Satelit. Keadaan langit yang terhalang akan menyebabkan GPS sulit menerima data satelit. Sebuah sinyal satelit yang pada hari tertentu diterima dengan sangat bagus belum tentu pada hari lain bisa diterima dengan kualitas yang sama walaupun user berdiri pada tempat yang sama. [9]

2.2.6.2 Android dan GPS

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

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

pengembang aplikasi untuk sistem operasi Android termasuk aplikasi-aplikasi GPS. Yang menyenangkan aplikasi-aplikasi tersebut jenisnya beragam dan jumlahnya pun banyak. [9]

2.2.7 Latitude dan Longitude

Latitude dan longitude adalah sebuah koordinat yang ada di bumi dan digunakan untuk menentukan kedudukan kita di atas bumi. [9]

2.2.7.1 Latitude

Latitude adalah garis yang melintang dari kutub utara dan kutub selatan. Titik 0 adalah sudut ekuator, tanda + menunjukan arah ke atas menuju kutub utara, sedangkan tanda minus di koordinat Latitude menuju ke kutub selatan. Titik yang dipakai dari 0 ke 90 derajat arah kutub utara, dan 0 ke -90 derajat ke kutub selatan. [9]

2.2.7.2 Longitude

Longitude adalah garis lintang. Angka dari sudut bundar bumi horisontal. Titik diawali dari 0 ke 180 derajat, dan - ke arah sebaliknya. Titik 0 dimulai dari garis negara Inggris. Mengarah ke Indonesia akan menjadi angka positif. Kebalikannya koordinat Longitude minus adalah arah kebalikan. [9]

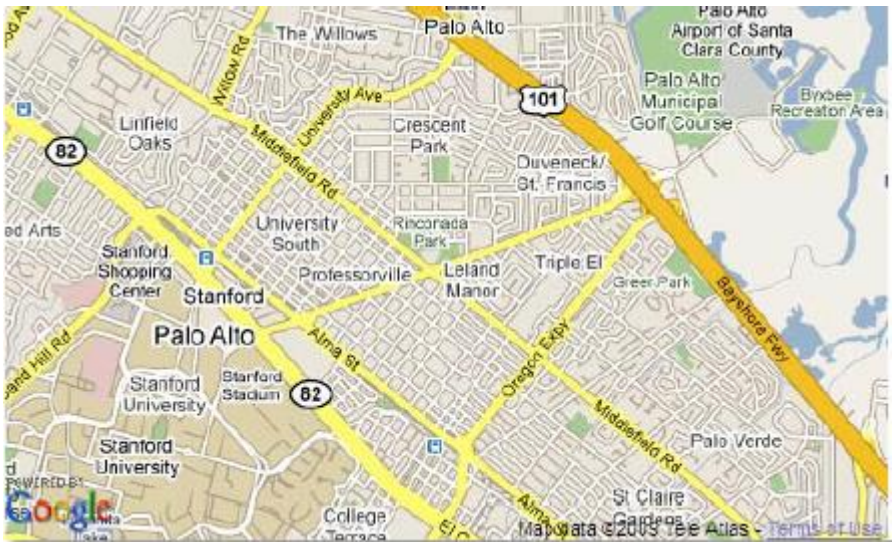
2.2.8 *Google Maps*

Google Maps 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 *Google Maps* pun dapat dipilih, berdasarkan foto asli atau peta gambar rute saja.[10]

GoogleMaps adalah layanan gratis yang diberikan oleh *Google* dan sangat popular. *GoogleMaps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah seperti pada gambar 2. Dengan kata lain, *GoogleMaps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur *GoogleMaps* dalam *web* yang telah kita buat atau

pada blog kita yang berbayar maupun gratis sekalipun dengan *GoogleMaps* API. *GoogleMaps* API adalah suatu library yang berbentuk *JavaScript*.

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



Gambar 2.8 *Google Maps*

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

1. ROADMAP, untuk menampilkan peta biasa 2 dimensi.
2. SATELLITE, untuk menampilkan foto satelit.
3. TERRAIN, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
4. HYBRID, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota). [10]

2.2.9 *Fuzzy Multi-Attribute Decision Making* (FMADM)

Fuzzy Multiple Attribute Decision Making (FMADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan. Pada dasarnya, ada 3 pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif dan pendekatan integrasi antara subyektif dan obyektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan.[11]

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah FMADM salah satunya adalah *Simple Additive Weighting Method* (SAW). Metode SAW sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

2.2.9.1 *Simple Additive Weighting* (SAW)

Metode SAW sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Secara garis besar, SAW digunakan untuk melakukan pembobotan pada data tempat kuliner dan wisata sehingga mempunyasi hasil akhir rekomendasi tempat wisata dan kuliner di kota Bandung.

2.2.9.2 Langkah Penyelesaian *Simple Additive Weighting* (SAW)

Langkah Penyelesaian SAW sebagai berikut :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria(Ci), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai)sebagai solusi.

Formula untuk melakukan normalisasi tersebut adalah :

Rij = $\frac{X\ ij}{Max\ X\ ij}$ Jika J adalah atribut keuntungan (benefit) [*Persamaan 2.1*]

Rij = $\frac{Min\ X\ ij}{X\ ij}$ Jika J adalah atribut biaya (cost) [*Persamaan 2.2*]

Dimana :

rij = rating kinerja ternormalisasi

Max Xij = nilai maksimum dari setiap baris dan kolom

Min Xij = nilai minimum dari setiap baris dan kolom

Xij = baris dan kolom dari matriks

Dengan rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i =1,2,…m dan j = 1,2,…,n.

Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai :

$\mathrm{Vi} = \sum_{j=i}^{n} \mathrm{Wj\ rij}$ [*Persamaan 2.3*]

Dimana :

Vi = Nilai akhir dari alternatif

wj = Bobot yang telah ditentukan

rij = Normalisasi matriks

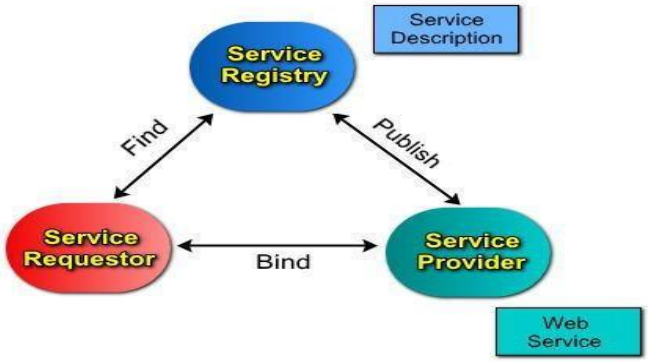
Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.

2.2.10 *Web Service*

Web service adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca oleh mesin. Sistem-sistem lainnya berinteraksi dengan *web service* menggunakan pesan SOAP yang umumnya dikirim melalui HTTP dalam bentuk XML [12].

Definisi diatas diberikan oleh *World Wide Web Consortium* (W3C) yang merupakan badan yang menciptakan dan mengembangkan standar *web service.* Tetapi secara umum, *web service* tidak terbatas hanya pada standar SOAP saja. Salah satu pustaka yang mengulas lengkap tentang *web service* menyebutkan definisi yang lebih umum: *web service* adalah aplikasi yang diakses melalui *internet* menggunakan protokol standar *internet* dan menggunakan XML sebagai format pesannya.

2.2.10.1 Arsitektur *Web service*



Gambar 2.9 Arsitektur *Web Service*

Pada gambar diatas, ada tiga komponen yang membuat *web service* berjalan. Ketiga komponen itu adalah [12]:

1. ***Service provide**r*, merupakan pemilik *Web service* yang berfungsi menyediakan kumpulan operasi dari *Web service*.
2. ***Service requesto**r*, merupakan aplikasi yang bertindak sebagai klien dari *Web service* yang mencari dan memulai interaksi terhadap layanan yang disediakan.

3. ***Service registry***, merupakan tempat dimana *Service provider* mempublikasikan layanannya. Pada arsitektur *Web service*, Service registry bersifat optional. Teknologi *web service* memungkinkan kita dapat menghubungkan berbagai jenis *software* yang memiliki *platform* dan sistem operasi yang berbeda.

2.2.10.2 Jenis-Jenis *Web service*

Jenis *web service* dapat dibagi menjadi dua, yaitu REST dan SOAP.

1. *Representational State Transfer* (REST)

REST adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula REST bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perintah HTTP yang bisa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE*. Balasan yang dikirimkan adalah dalam bentuk XML sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing disisi *Client* [13].

Dalam pengaplikasiannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktivitas yang mengolah *resource* itu. Alasan mengapa REST tidak digunakan dalam skripsi ini karena orientasi pada *resource*nya itu, sedangkan aplikasi lowongan pekerjaan membutuhkan pemanggilan metode yang bisa dikerjakan terhadap kumpulan *resource* data lowongan. Beberapa contoh *web service* yang menggunakan REST adalah: Flickr API (*Application Program Interface*), YouTube API, Amazon API.

2. *Simple Object Access Protokol* (SOAP)

SOAP merupakan protokol standar yang ringan dan ditujukan untuk pertukaran informasi dalam desentralisasi sistem. Protokol ini melakukan pemanggilan metode dalam bahasa XML (*Extensible Markup Language*). Namun, selain itu juga kita dapat menggunakan format pertukaran data lainnya yaitu JSON (*JavaScript Object Notation*). Dengan demikian, SOAP adalah suatu mekanisme yang sederhana untuk melakukan pertukaran struktur dan tipe informasi dalam lingkungan yang tersebar dan terdistribusi menggunakan XML/JSON. Dokumen SOAP yang digunakan untuk melakukan *request* disebut SOAP *request* sedangkan dokumen SOAP yang diperoleh dari web service disebut dengan SOAP *response* [13].

2.2.10.3 *Web service Definition Language* (WSDL)

WSDL merupakan suatu dokumen berbasis XML yang berfungsi untuk menjelaskan fungsi *public* dan metode-metode yang tersedia dalam *web service*. Selain itu, WSDL juga mengatur *end-point* pada pesan-pesan yang berisi dokumen maupun informasi prosedur. Salah satu bagian dokumen WSDL yang cukup penting dalam melakukan interaksi dengan SOAP, HTTP, serta MIME adalah *binding*. SOAP dapat berjalan sendiri tanpa bantuan WSDL, tetapi hal ini akan menyulitkan pengembang di sisi *client* karena tidak tahu harus mengharapkan operasi dan bentuk data apa. Karena itu WSDL diciptakan sebagai pendefinisi sebuah *web service*. Dalam tugas akhir ini, *web service* yang dibangun akan dijabarkan menggunakan WSDL [12].

2.2.10.4 UDDI (*Uniform Description, Discovery, and Integration*)

Dalam penerapannya, *web service* memerlukan sumber dari semua informasi tentang *SOAP message*, cara membuat respons, mengumumkan *service* yang ada, dan sebagainya. Dengan demikian, aplikasi dapat menggunakannya untuk mencari WSDL. Untuk memenuhi hal tersebut, diperlukan UDDI yang menyediakan *service* untuk mencari informasi yang diperlukan.

2.2.10.5 JSON (*Java Script Object Notation*)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. 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. Kelebihan-kelebihan dari JSON adalah [14]:

1. Format Penulisan : Untuk merepresentasikan sebuah struktur data yang rumit dan berbentuk hirarkis penulisan JSON relatif lebih terstruktur dan mudah.
2. Ukuran : Ukuran karakter yang dibutuhkan JSON lebih kecil dibandingkan XML untuk data yang sama.
3. *Browser Parsing* : Proses parsing merupakan proses pengenalan token atau bagian-bagian kecil dalam rangkaian dokumen XML/JSON.

JSON terbuat dari dua struktur :

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

2.2.11 *Web Mining*

Web mining merupakan satu bidang data mining yang memusatkan pada proses *mining* data pada *World Wide Web* (WWW). Pusat informasi ini tentu akan menjadi sumber pengetahuan yang menarik dimana *data mining* dapat diterapkan untuk mendapatkan berbagai pengetahuan yang memiliki nilai manfaat yang tinggi. Berdasarkan tipe data mentah yang diolahnya. Target data dari *Web mining*

adalah menganalisis data dari web, seperti data akses pengunjung, struktur halaman *web*, *format* halaman *web* dan sebagainya. Berdasarkan target analisisnya, *web mining* dibagi menjadi tiga kategori yaitu [15]:

1. *Web structure mining*

Proses yang menggunakan teori *graph* untuk menganalisis simpul (*node*) dan keterhubungan struktur dari situs. Menurut tipe dari struktur web, *web structure mining* terbagi menjadi 2 (dua). Pertama adalah mengekstrak dari pola *hyperlink* di web. Kedua dari *web structure mining* adalah mining terhadap struktur dokumen. Yang dimaksud sebagai struktur dokumen adalah menganalisa struktur dari bahasa yang digunakan dalam web, yaitu bahasa HTML (*Hyper Text Markup Language*), atau XML (*eXtensibel Markup Language*) di dalam halaman.

2. *Web Content Mining*

Proses untuk mendapatkan informasi yang berguna dari isi (*content*) di *web*. Isi (*content*) dapat berupa *text, image, audio* dan *video*. *Web content mining* terkadang disebut sebagai *web text mining*, karena teks merupakan bagian dari *web* yang paling banyak tersedia.

3. *Web Usage Mining*

Teknik *data mining* yang berusaha mengungkap pola penggunaan dari halaman web, dalam rangka coba untuk memahami dan meningkatkan pelayanan kebutuhan dari aplikasi berbasis *web*. Pada jenis *structure* dan *content mining*, yang dianalisa atau digali adalah data didalam *web* itu sendiri, namun pada *web usage mining* yang dianalisa adalah pengguna atau pengunjung dari halaman *web.*

4. Perbedaan Jenis-jenis *Web Mining*

Tabel 2.1 Perbedaan Jenis-jenis *Web Mining*

Topic	***Web Mining***			
	Web structure mining		***Web content mining***	***Web usage mining***
	IR View	***DB View***		
View of Data	*Unstructured, Semistructured*	*Semistructured, Web as DB*	*Links structure*	*Interactivity*
Main Data	*Text documnts Hypertext docs*	*Hypertext docs*	*Links structure*	*-Server logs, -Browser logs*

Representation	*-Bag of words, n-grams -Terms, phrase -Concept or ontology -Relational*	*OEM,Relational*	*Graph*	*-Relational table, -Graph*
Method	*-TFIDF and variants, -Machine Learning, -Statistic (NLP)*	*Propietary algorithms, ILP, Association rules*	*Propietary algorithms*	*-Machine Learning, -Statistical, -(Modified) association rules*
Application Categories	*Categorization, Clustering, Finding extraction rules, Finding pattern in text, User modelling*	*Finding frequent substructure, Website schema discovery*	*Categorization, Clustering*	*-Site construction, addaption and management -Marketing, -User modelling*

2.2.12 MYSQL

MySQL merupakan *software database* yang paling populer di kalangan para pengembang aplikasi *database*. MySQL juga adalah suatu perangkat lunak yang menganut atau mengimplementasikan model basis data relasional maka MySQL disebut sebagai *Relational Database Management System* (RDBMS). MySQL merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*) [11].

Saat ini, MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah, maupun besar. Lisensi MySQL terbagi menjadi dua. Anda dapat menggunakan MySQL sebagai produk *open source* dibawah GNU *General Public License* (gratis) atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada

kenyataannya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik. Beberapa contoh aplikasi yang menggunakan MySQL adalah Joomla, Wordpress, MyBB, phpBB, dan masih banyak lagi.

2.2.12.1 Kelebihan MySQL

Beberapa kelebihan yang menjadikan para pengembang aplikasi *database* memilih untuk menggunakan MySQL dibanding dengan RDBMS yang lain. Banyaknya pengguna MySQL yang kian meningkat secara drastis ini mengakibatkan nama MySQL menjadi melambung tinggi dan menjadikannya se-populer seperti saat sekarang. Berikut ini beberapa keuntungan dari MySQL [11]:

1. Fleksibel

   MySQL dapat digunakan untuk mengembangkan aplikasi *desktop* maupun aplikasi *web* dengan menggunakan teknologi yang bervariasi. Ini berarti bahwa MySQL memiliki fleksibilitas terhadap teknologi yang akan digunakan sebagai pengembang aplikasi, apakah itu PHP, JSP, Java, Delphi, C++, maupun yang lainnya dengan cara menyediakan *plug-in* dan *driver* yang spesifik untuk masing-masing teknologi tersebut. Dalam database MySQL, juga memiliki dukungan terhadap *stored procedure*, fungsi, trigger, *view*, SQL standar ANSI, dll, yang akan mempermudah dan mempercepat proses pengembangan aplikasi.

2. Performa Tinggi

   MySQL memiliki mesin *query* dengan performa tinggi, dengan demikian proses transaksional dapat dilakukan dengan sangat cepat. Hal ini terbukti dengan digunakannya MySQL sebagai *database* dari beberapa aplikasi *web* memiliki *traffic* (lalu lintas) sangat tinggi.

3. Lintas *Platform*

   MySQL dapat digunakan pada *platform* atau lingkungan (dalam hal ini sistem operasi) yang beragam, bisa Microsoft Windows, Linux, atau UNIX. Ini menyebabkan proses migrasi data (bila dibutuhkan) antar sistem operasi dapat dilakukan secara lebih mudah. Misal, jika kita ingin mengganti sistem operasi pada mesin *server*.

4. Gratis

   MySQL dapat digunakan secara gratis. Meskipun demikian, ada juga *software* MySQL yang bersifat komersil. Biasanya yang sudah ditambahi dengan kemampuan spesifik dan mendapat pelayanan dari *technical support.*

5. Proteksi Data yang Andal

   Perlindungan terhadap keamanan data merupakan hal nomor satu yang dilakukan oleh para profesional di bidang *database*. MySQL menyediakan mekanisme yang *powerful* untuk menangani hal tersebut, yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data, dan lain sebagainya.

6. Komunitas Luas

   Karena penggunanya banyak maka MySQL memiliki komunitas yang luas. Hal ini berguna jika kita menemui suatu permasalahan dalam proses pengolahan data menggunakan MySQL. Dengan mengikuti salah satu atau beberapa komunitas tertentu, kita dapat menanyakan atau mendiskusikan permasalahan tersebut melalui forum. Harapannya adalah solusi akan permasalahan tersebut akan cepat diperoleh.

a. Bahasa MySQL

Seperti halnya SQL *engine* yang lain, Perintah dalam SQL diklasifikasikan menjadi lima bagian besar, yaitu [11]:

a. *Data Defenition Language* (DDL), Kumpulan perintah yang berfungsi untuk mendefinisikan *database*, struktur tabel, indeks, dan objek-objek lain dalam *database*. Perintah-perintah yang digolongkan dalam DDL adalah *create, alter* dan *drop*.
b. *Data Manipulation Language* (DML), Kumpulan perintah yang digunakan untuk melakukan manipulasi data di dalam tabel. Perintah-perintah yang digolongkan dalam DML adalah *update, insert,* dan *delete*.
c. *Data Control Language* (DCL), Kumpulan perintah yang digunakan untuk mengontrol (memberi/mencabut) hak akses ke/dari seorang *user*. Dua perintah utama dalam DCL adalah grant dan *revoke*.

d. *Transactional Control Language* (TCL), Kumpulan perintah yang digunakan untuk mengontrol (menyimpan/membatalkan) transaksi data. Perintah-perintah yang digolongkan dalam TCL adalah *commit* dan *rollback. Commit* digunakan untuk menyimpan perubahan-perubahan yang dilakukan terhadap *database* secara permanen, sedangkan *rollback* digunakan untuk membatalkan transaksi atau perubahan-perubahan yang telah dilakukan ke dalam *database*.

e. *Data Query Language* (DQL), Kumpulan perintah yang digunakan untuk melakukan seleksi data.

b. Melihat *Database*

Kita bisa melihat daftar *database* yang ada di MySQL dengan mengetikkan baris perintah berikut : **mysql> SHOW DATABASES;**

MySQL akan menunjukkan *list*/daftar dari database pada *server*. Jika ini adalah sebuah *server* baru (sebagai contoh : jika kamu menginstal *server* ini sendiri) daftarnya harus kelihatan seperti ini :

```
----------------------
| Database |
----------------------
| mysql |
| test |
----------------------
2 rows inset (0.11sec)
```

c. Membuat *Database*

Berikut ini adalah cara membuat sebuah *database* pada MySQL. Sebagai latihan kita membuat *database* dengan nama “Saham_db”, dan perintahnya adalah sebagai berikut :

mysql> create database Saham_db;

Query OK,1 row affected (0.03sec)

d. Menghapus *Database*

Untuk menghapus *database* dapat dilakukan dengan men"*dropping*"nya, dan perintahnya adalah :

mysql> drop database Saham_db;

2.2.13 *Object Oriented Programming*

Secara spesifik, pengertian berorientasi objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, *polymorphism* (polimorfisme) dan *inheritance* (pewarisan) [16].

2.2.13.1 Konsep Dasar Berorientasi *Object*

Pengembangan sistem berorientasi objek berbeda dari pengembangan konvensional yang memandang perangkat lunak sebagai fungsi dan data yang terisolasi. Program perangkat lunak adalah sekumpulan mekanisme yang melakukan aksi-aksi tertentu pada data tertentu. Pada pendekatan konvensional, kebanyaan berfous pada fungs. Namun juga terdapat pendekatan yang fokus pada data terutama pada kubu basis data dan pemodelan informasi. Sementara itu, pandangan berorientasi objek berpusat pada objek yang mengkombinasikan data dan fungsionalitas. Berikut ini adalah beberapa kriteria-kriteria penting pendekatan objek yang harus dipahami tentang metodologi berorientasi objek [16]:

a. *Seamlessness*

   Konsep berorientasi objek dapat diterapkan di seluruh siklus hidup pengembang sistem (*System Developmnent life cycle*) dari analisis, perancangan, sampai implementasi.

b. Kelas

Pendekatan berorientasi objek berdasar pada pentingnya kelas. Kelas adalah elemen perangkat lunak yang mendeskripsikan tipe data abstrak (ADT – *Abstract data type*) dimana implementasinya dapat secara parsial atau total. Implementasi parsial bila di kelas terdapat operas/metode abstrak.

c. *Assertion*

Fitur tipe data abstrak harus secara formal menspesifikasikan properti kelas Assertion berperan pada prakondisi (*precondition*), poskondisi (*postcondition*), invarian kelas. Assertion mendeskripsikan efek fitur-fitur di objek independen terhadap implementasi fitur-fitur itu.

d. Kelas Merupakan Cara Pemodulan

Pendekatan berorientasi objek merupakan teknik pembentukan arsitektur perangkat lunak. Dampak utama penerapannya adalah pada struktur modul dari sistem perangkat lunak. Peran utamanya di sistem berorientasi objek dimainkan oleh kelas. Pada pendekatan murni berorientasi objek maka kelas merupakan satu-satunya pemodulan.

e. Kelas merupakan *type*

Kelas cukup ampuh untuk menghilangkan kebutuhan mekanisme *typing* yang lain. Setiap *type* harus didasarkan pada kelas.

f. *Exception handling*

Sistem perangkat lunak seharusnya mampu menangani kondisi abnormal secara terstruktur dengan baik.

g. *Genericity*

Agar penerapan *typing* menjadi praktis adalah kemampuan untuk mendefinisikan *type-parameterized classes* atau disebut generik.

h. Pewarisan

Pendekatan berorientasi objek harus memungkinkan mendefinisikan kelas yang merupakan turunan kelas yang lain. Pewarisan dapat berupa pewarisan tunggal (*single inheritance*) dan pewarisan jamak (*multiple inheritance*).

i. *Redefinition*

Kita mungkin perlu mengubah implementasi atau properti lain dari fitur-fitur yang diwariskan. Redefinisi dapat mempengaruhi fitur, penanda (tipe argumen dan hasil) dan spesifikasi.

j. *Polymorphism*

Dimungkinkan untuk entitas ditautkan ke objek-objek beragam tipe. *Polymorphism* dikendalikan oleh pewarisan.

k. *Dynamic binding*

Dynamic binding mempunyai pengaruh utama pada aplikasi berorientasi objek. Mekanisme ini memungkinkan pengembang menulis panggilan secara sederhana untuk beberapa panggilan yang dapat dimungkinkan, bergantung pada situasi saat sebenarnnya panggilan itu dieksekusi.

l. *Run-time type interogation*

Perangkat lunak memerlukan mekanisme untuk mengakses objek dengan cara yang aman tanpa melanggar konstrain *static typing.* Mekanisme ini harus dirancang agar tidak emniadakan manfaat-manfaat dari *dynamic binding* dan *polymorphism*.

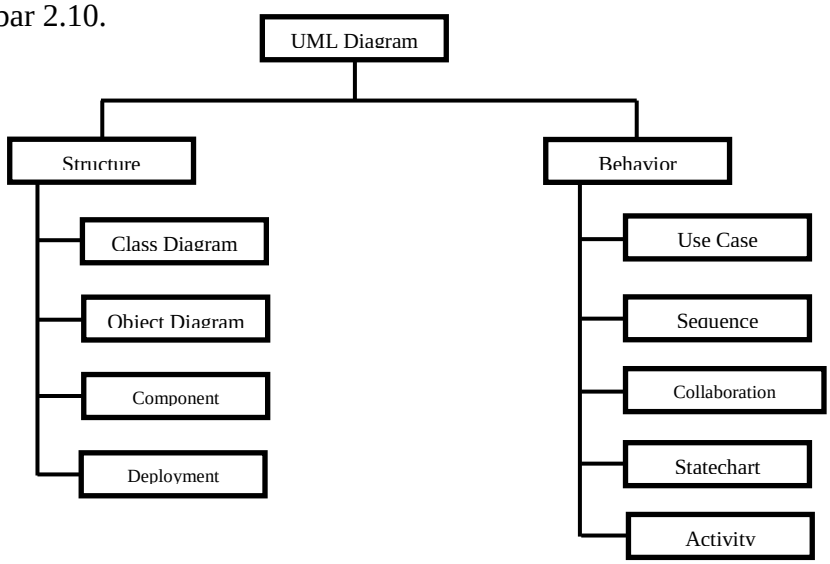
2.2.13.2 UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI). Standar UML dikelola oleh OMG (*Object Management Group*) [16].

UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya yaitu : Grady Booch OOD (*Object Oriented Design*).

2.2.13.3 UML *Diagram*

Pada UML terdiri dari 13 macam diagram yang dikelompokan dalam 2 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.10.



Gambar 2. 1 UML *Diagram*

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

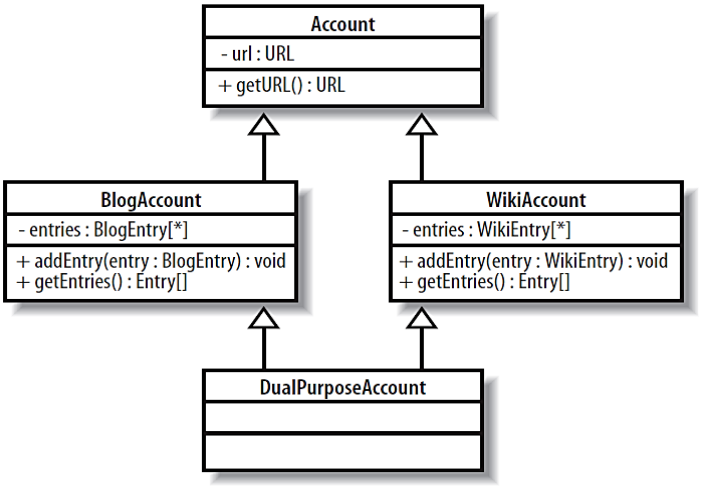
1. *Structure diagrams* yaitu diagram untuk memvisualisasikan, mendokumentasi, menspesifikasikan, dan membangun aspek statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu diagram untuk memvisualisasikan, mendokumentasi, menspesifikasikan, dan membangun aspek dinamis dari sistem yang dimodelkan.

2.2.13.4 *Class Diagram*

Diagram Kelas menunjukkan sekumpulan kelas, interface dan kolaborasi dan keterhubungan. Diagram kelas ditujukan untuk pandangan statik terhadap sistem. Kelas memiliki apa yang disebut nama, atribut, metode atau operasi [16].

1. Nama kelas harus unik. Nama akan menjadi identifier di program, seharusnya sedini mungkin dipilih nama yang memenuhi aturan (semua) bahasa pemrograman.
2. Atribut adalah properti bernama di kelas yang mendeskripsikan *range* nilai yang dipunyai instan kelas. Kelas dapat mempunyai sejumlah atribut atau tidak sama sekali.
3. Operasi atau metode adalah implementasi layanan yang dapat diminta pada sembarang objek kelas untuk mempengaruhi perilaku sistem. Kelas dapat mempunyai sejumlah operasi atau tidak sama sekali.

Berikut Contoh *Class Diagram* dapat dilihat pada Gambar 2.11 [17].



Gambar 2. 2 Contoh *Class Diagram*

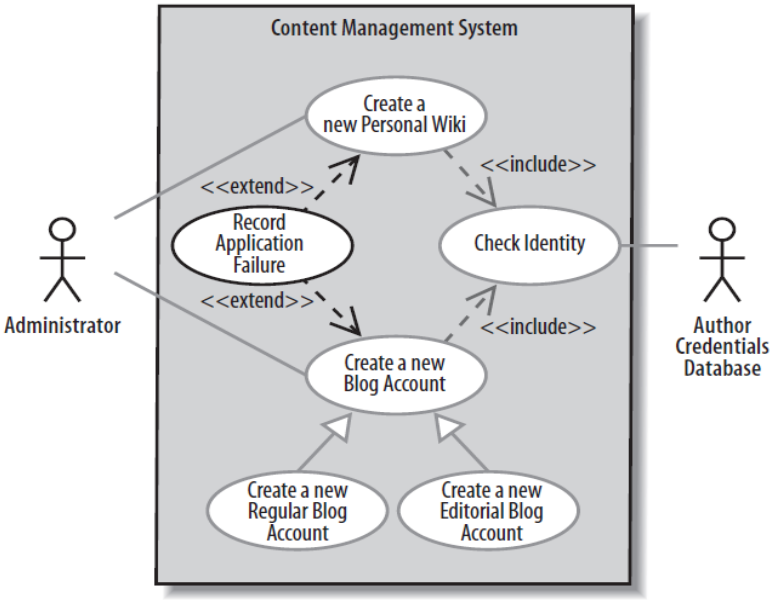
2.2.13.5 *Use Case Diagram*

Use case Diagram merupakan salah satu diagram untuk memodelkan aspek perilakau sistem. Masing-masing diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. Diagram *use case* adalah penting

untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. Diagram-diagram *use case* merupakan pusat pemodelan perilaku sistem, subsistem, dan kelas. Diagram *use case* digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Ada empat hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* [16].

1. Sistem yaitu sesuatu yang hendak kita bangun.
2. Relasi adalah relasi antara aktor dengan *use case*
3. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
4. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut Contoh *Use Case Diagram* dapat dilihat pada Gambar 2.12 [17].



Gambar 2. 3 Contoh *Use Case Diagram*

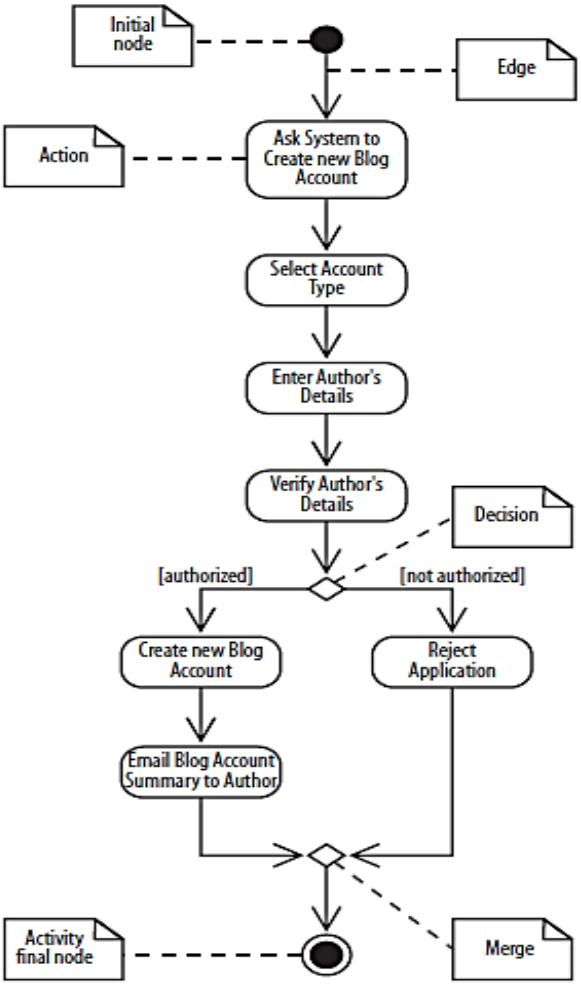
2.2.13.6 *Activity Diagram*

Diagram aktivitas adalah diagram flowchart yang diperluas yang menunjukan aliran kendali satu aktivitas ke aktivitas lain di sistem. Diagram aktivitas ini digunakan untuk memodelkan aspek dinamis sistem. Diagram

aktivitas mendeskripsikan aksi-aksi dan hasilnya. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut [16]:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokkan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut Contoh *Activity Diagram* dapat dilihat pada Gambar 2.13 [17].

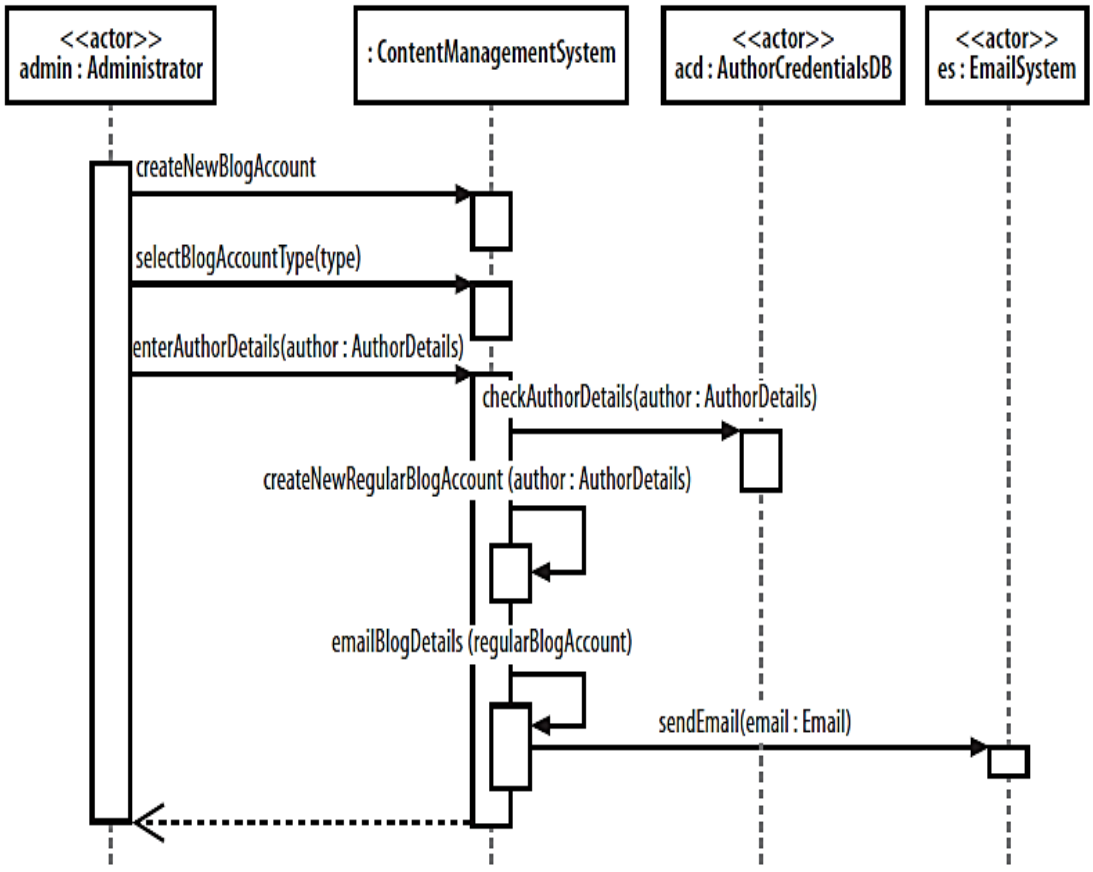


Gambar 2. 4 Contoh *Activity Diagram*

2.2.13.7 *Sequence Diagram*

Diagram sekuen menunjukan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada sisi basis keberurutan waktu dari pesan-pesan yang terjadi. Diagram sekuen menunjukan objek sebagai garis vertikal dan tiap kejadian sebagai panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan. Diagram ini hanya menunjukan barisan kejadian, bukan perwaktuan nyata. Kecuali untuk sistem waktu nyata yang mengharuskan konstrain barisan terjadi [16].

Berikut Contoh *sequence diagram* dapat dilihat pada gambar 2.14 [17].



Gambar 2. 5 Contoh *Sequence Diagram*

2.2.14 Metode Pengujian Sistem

Metode pengujian sistem terdiri dari Pengujian *black box* dan *beta* yang dilakukan untuk mengetahui efektifitas dari perangkat lunak (*software*) yang digunakan.

2.2.14.1 Pengujian *Black Box*

Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian ini dianalogikan seperti melihat suatu kotak hitam yang hanya bisa dilihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. *Black box* testing melakukan evaluasi untuk menemukan kesalahan dalam kategori berikut[20]:

a. Fungsi tidak benar atau hilang.
b. Kesalahan *interface* atau antarmuka.
c. Kesalahan dalam struktur data atau akses *database eksternal*.
d. Kesalahan kinerja atau perilaku dan kesalahan inisialisasi dan terminasi.

2.2.14.2 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan kuesioner mengenai tanggapan siswa terhadap aplikasi yang telah dibangun. Adapun metode penilaian pengujian yang digunakan yaitu metode kuantitatif berdasarkan data sampel dari siswa.

2.2.14.3 Skala Likert

Skala likert adalah suatu skala psikometrik yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama skala Likert. Skala Likert ini merupakan skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena [21].

Dalam skala Likert terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan negative yang berfungsi untuk mengukur sikap negative objek. Skor pernyataan positif dimulai dari 1 untuk sangat tidak setuju (TS), 2 untuk tidak setuju (KS), 3 untuk raguragu (CS), 4 untuk setuju (S), dan 5 untuk sangat setuju (SS). Skor pernyataan negative dimulai dari 1 untuk sangat setuju (SS), 2 untuk setuju (S), 3 untuk raguragu (CS), 4 untuk tidak setuju (KS), dan 5 untuk sangat tidak setuju (TS). Beberapa menghilangkan option "Ragu-ragu" dalam instrument untuk memudahkan dalam melihat angket yang responden isikan. Skala Likert digunakan untuk mengukur kesetujuan dan ketidaksetujuan seseorang terhadap sesuatu rencana program, pelaksanaan program ataupun tingkat keberhasilan suatu program.

Kekurangan Skala Likert :

1. Karena ukuran yang digunakan adalah ukuran ordinal, skala Likert hanya dapat mengurutkan individu dalam skala, tetapi tidak dapat membandingkan berapa kali satu individu lebih baik dari individu yang lain.
2. Kadangkala total skor dari individu tidak memberikan arti yang jelas, karena banyak pola respons terhadap beberapa item akan memberikan skor yang sama. Adanya kelemahan di atas sebenarnya dapat dipikirkan sebagai error dari respons yang terjadi.

Kelebihan Skala Likert :

1. Mudah dibuat dan di terapkan.
2. Skala Likert lebih mudah membuatnya dibanding lain sepertti skala Thurstone.
3. Terdapat kebebasan dalam memasukan pertanyaan- pertanyaan, asalkan sesuai dengan konteks permasalahan yang diteliti.
4. Jawaban suatu item dapat berupa alternative, sehingga informasi mengenai item tersebut diperjelas.
5. Reliabilitas pengukuran bisa diperoleh dengan jumlah item tersebut diperjelas.

2.2.14.4 Kuesioner

Menurut Depdikbud kuesioner atau angket adalah suatu alat pengumpul data yang berupa serangkaian pertanyaan yang diajukan pada *responden* untuk mendapat jawaban. Angket merupakan teknik pengumpulan data yang dilakukan dengan mengadakan komunikasi dengan sumber data. Menurut KBBI (Kamus Besar Bahasa Indonesia) kuesioner merupakan alat riset atau survei yang terdiri atas serangkaian pertanyaan tertulis, bertujuan mendapatkan tanggapan dari kelompok orang terpilih melalui wawancara pribadi atau melalui pos daftar pertanyaan.

Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Selain itu, kuesioner juga cocok digunakan bila jumlah responden cukup besar dan tersebar di wilayah yang luas. kuesioner dapat berupa pertanyaan/pernyataan tertutup atau terbuka, dapat diberikan kepada responden secara langsung atau dikirim melalui pos, atau internet. Bila penelitian dilakukan pada lingkup yang tidak terlalu luas, sehingga kuesioner dapat diantarkan langsung dalam waktu yang tidak terlalu lama, maka pengiriman kuesioner kepada responden tidak perlu melalui pos [22].