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 5 Kantor Cabang yang terdiri dari Cabang 1 Soreang, Cabang 2 Ciparay, Cabang 3 Majalaya, Cabang 4 Padalarang, dan Cabang 5 Cimahi [2].

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 [2].

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

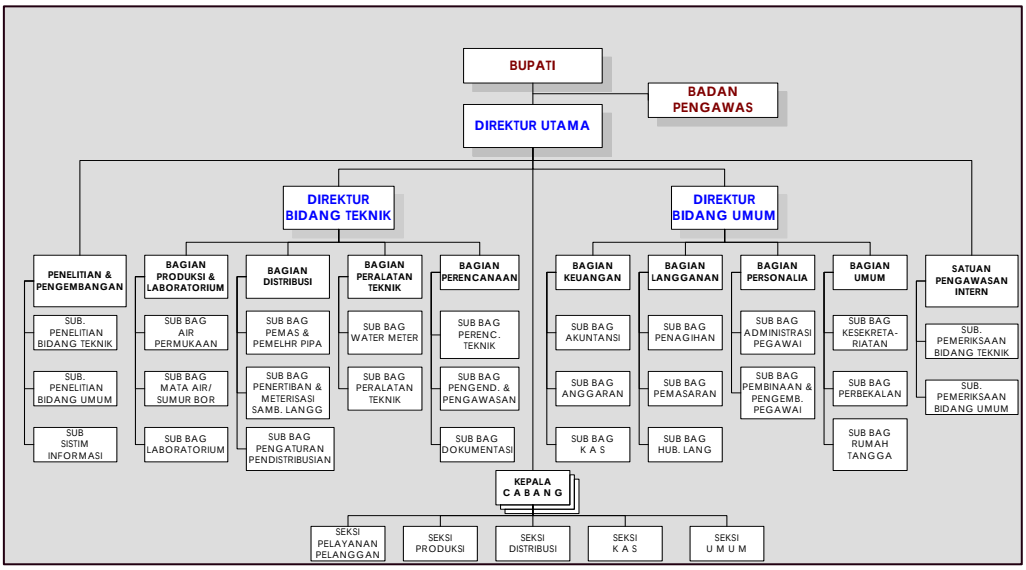
2.1.3 Badan Hukum

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 [2].

2.1.4 Struktur Organisasi

STRUKTUR ORGANISASI

PDAM TIRTA RAHARJA KABUPATEN BANDUNG



Gambar 2.2 Struktur Organisasi PDAM Tirta Raharja Kab. Bandung

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 *list*rik, 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, energi *list*rik, 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 meningkatkan 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.

2.1.6 Profil Cabang

Berikut adalah Profil Cabang dan Kota Pelayanan PDAM Tirta Raharja Kab. Bandung.

Tabel 2.1 Profil Cabang 1 Soreang

Kota Pelayanan	Jml SL	Jml	Sumber	Kapasitas (l/det)
Soreang	10.487	33	Sungai Cisangkuy	167
Banjaran	3.473	11	Sungai Cisangkuy	
Ciwidey	617		MA. Cigadog	4
Pangalengan	3.205	10	MA. Citere	26
Jumlah	**11.782**	**54**		**197**

Tabel 2.2 Profil Cabang 2 Baleendah

Kota Pelayanan	Jml SL	Jml	Sumber	Kapasitas (l/det)
Baleendah	3.370	13	Sungai Citarum	250

Bojongsoang	1.744			
Ciparay	5.607	20		
Dayeuh Kolot	543			
Pacet	273	1	MA. Cilembang	2
Jumlah	**11.737**	**34**		**252**

Tabel 2.3 Profil Cabang 3 Majalaya

Kota Pelayanan	Jml SL	Jml	Sumber	Kapasitas (l/det)
Majalaya	5.731	16	DW & Sungai Citarum	17
Rancaekek	4.731	16		
Cileunyi	469	1	MA. Cihampelas	4
Cicalengka	732	3	DW 2 Los- Majalaya	7
Jumlah	**11.653**	**36**		**28**

Tabel 2.4 Profil Cabang 4 Padalarang

Kota Pelayanan	Jml SL	Jml	Sumber	Kapasitas (l/det)
Cisarua	3.309	12	Sungai Cijanggel	33,5
Lembang	2.339	14	MA. Cikolegede	3
			MA. Pasir Ipis	25,5
Padalarang	1.908	16	DW Lapang Karbit	17
Batujajar	721	5	MA. Cipulus	18
Cililin	1.144	6	Waduk Saguling	20
Cikalong Wetan	351	2	MA. Cibulakan	4,5
Jumlah	**9.772**	**55**		**119,5**

Tabel 2.5 Profil Cabang 5 Cimahi

Kota Pelayanan	Jml SL	Jml	Sumber	Kapasitas (l/det)
Cimahi	15.453	40	Sungai Cijanggel, DW MA Cisintok MA Cikudapati	189
Jumlah	**15.453**	**40**		**189**

2.2 Landasan Teori

2.2.1 Teknologi Komputer *Tablet* (*Tablet PC*)

Tablet PC adalah *laptop* – atau komputer *portable* berbentuk buku. Memiliki layar sentuh atau teknologi *tablet* digital yang memungkinkan pengguna komputer mempergunakan *stylus* atau pulpen digital selain *keyboard* ataupun *mouse* komputer. Istilah ini dipopulerkan oleh Microsoft pada tahun 2001, tetapi *PC tablet* sekarang mengacu pada setiap komputer pribadi yang berukuran *tablet*, pun jika tidak menggunakan Windows melainkan sistem operasi PC yang lain. *Tablet* dapat menggunakan papan ketik *virtual* dan pengenalan tulisan tangan untuk *input* teks melalui layar sentuh [3].

Dilihat dari perkembangannya, sebuah *tablet PC* awalnya hanya rekaan beberapa orang akan *gadget* masa depan. Kemudian jenis perangkat ini mulai di adaptasikan sebagai *e-book reader* hingga akhirnya menjadi seperti yang saat ini banyak di temukan di pasaran.

Peran sistem operasi yang bermain di dalamnya pun menjadi salah satu tonggak kebangkitan perangkat *tablet*. Dengan semakin intuitif dan mampu memuaskan keinginan manusia, kebutuhan akan *tablet PC* pun semakin besar.

Berikut adalah kelebihan dan kekurangan *tablet PC* [4] :

Kelebihan:

1. *Tablet PC* dikenal sebagai *laptop* atau *notebook* ringkas. Penggunaannya mudah dengan layar yang besar. Sehingga, tampilan yang dihasilkan sama seperti *laptop* dan *notebook*.
2. Memiliki *e-book reader* yang mumpuni yang menjamin kenyamanan membaca buku elektronik di manapun dan kapanpun.
3. Cocok untuk mengusir rasa bosan karena *tablet PC* menyediakan berbagai aplikasi yang luar biasa.

Kekurangan:

1. Sebagian orang mengeluhkan saat menerima telepon melalui *tablet PC*. Walaupun produk ini bisa dipakai untuk menelepon, namun ukuran yang besar membuat orang kurang nyaman.
2. Ruang penyimpan data terbatas. Walau disebut sebagai pengganti *laptop* dan *notebook* yang ringkas, namun kapasitas tetap menjadi masalah. Memori yang dimiliki *tablet PC* tak sebesar *laptop* dan *notebook*. Tak heran bila beberapa produk *tablet PC* menawarkan memori eksternal untuk memecahkan masalah ini.
3. Daya tahan baterai tidak sebagus *laptop* dan *notebook*. Hal ini dikarenakan *tablet PC* difungsikan sebagai ponsel dan "agenda" berjalan Anda. Sudah pasti akan terus menyalakan *tablet PC* kapanpun. Inilah yang membuat baterai *tablet PC* cepat habis.



Gambar 2.3 Komputer *tablet* Asus Google Nexus 7 [5]

2.2.1.1 Sistem Operasi di Komputer *Tablet*

Pada *Tablet PC* terdapat beberapa sistem operasi. Berikut adalah 3 *Operating System* yang mempunyai pangsa pasar terbesar :

Top Tablet Operating Systems, Shipments, and Market Share, 2013 Q1 (Shipments in Millions)

Vendor	1Q13 Unit Shipments	1Q13 Market Share	1Q12 Unit Shipments	1Q12 Market Share	Year-over-Year Growth
Android	27.8	56.5%	8.0	39.4%	247.5%
iOS	19.5	39.6%	11.8	58.1%	65.3%
Windows	1.6	3.3%	0.2	1.0%	700.0%
Windows RT	0.2	0.4%	0.0	N/A	N/A
Others	0.1	0.2%	0.2	1.0%	-50.0%
Total	**49.2**	**100.0%**	**20.3**	**100.0%**	**142.4%**

Gambar 2.4 Pangsar Pasar Sistem Operasi di Komputer *Tablet* [6]

1. Android

Google sebagai raksasa teknologi internet menggawangi kemunculan sistem operasi ini. Sejak pertama kemunculannya, android banyak diminati oleh produsen *smartphone* dan *tablet PC* untuk terintregasi dengan perangkat buatan mereka. Tampilan yang menarik dan kemudahan kustomisasi *user interface* (UI) adalah salah satu nilai lebih yang dimiliki sistem ini. Di sistem ini pengguna memiliki akses tak terbatas untuk mengatur bentuk dan tampilan didalamnya.

Android hingga saat ini memiliki beberapa versi dan varian, dimulai dari Android 1.5 (*Cupcake*), Android 1.6 (*Donut*), Android 2.0 (*Éclair*), Android 2.2 (*Froyo*) hingga Android 3.0 (*HoneyComb*) yang khusus diperuntukkan bagi pengguna *tablet PC*.

2. iOS

Sistem operasi buatan Apple ini terkenal dengan tampilan yang elegan dan *Widgets* yang memanjakan mata penggunanya. iPad merupakan satu-satunya *tablet PC* yang menggunakan sistem ini, dan hal tersebut sudah lazim bagi semua sistem operasi buatan Apple yang terkenal "eksklusif".

App store merupakan aplikasi andalan dari iOs, disana pengguna bisa mengunduh beragam aplikasi dan game-game berkualitas dengan grafis yang menawan yang jumlahnya mencapai ratusan. Namun salah satu hal yang menjadi kekurangan *tablet* ini adalah kompabilitasnya dalam menjalankan video berformat flash.

3. Windows 7 *Mobile*

Windows adalah sistem operasi yang tidak mungkin asing di telinga kita. Pada versi *tablet* tidak memiliki perbedaan dengan versi *desktop*, baik dari tampilan maupun dukungan aplikasi yang bisa di*install* di dalamnya. *Tablet PC* dengan sistem ini diharuskan memiliki spesifikasi *hardware* yang tinggi untuk mampu menjalankan sistem ini.

2.2.2 Pengertian Program

Pengertian program adalah ekspresi, pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementaikan dengan menggunakan bahasa pemrograman, sehingga dapat dieksekusi oleh komputer. Program adalah sekumpulan instruksi yang diberikan untuk mengendalikan perangkat keras komputer atau disebut juga sebagai perangkat lunak [7].

2.2.3 Pengertian Aplikasi

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna [8]. Jadi dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang termuat dalam data tersebut tidak mengalami perubahan. Jadi program aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer supaya dapat melakukan pekerjaan sesuai dengan keinginan programmer atau user.

Atau definisi lain aplikasi merupakan kumpulan dari prosedur-prosedur yang digunakan untuk mengolah data menjadi informasi. Misalnya penjumlahan,

klasifikasi, rotasi, koreksi geometri, query, *overlay*, *buffer*, *jointable* dan sebagainya.

2.2.4 Pengertian Survey

Menurut Kamus Besar Bahasa Indonesia, arti kata survey adalah teknik riset dengan memberi batas yg jelas atas data; penyelidikan; peninjauan; [9] Metode survey ialah riset yang diadakan untuk memperoleh fakta-fakta tentang gejala-gejala atas permasalahan yang timbul. Kajiannya tidak perlu mendalam sampai pada tahap menyelidiki kenapa gejala-gejala tersebut ada serta menganalisis hubungan-hubungan atas gejala-gejala.

Tipe informasi yang didapat dari riset survey:

1. Fakta : tempat tinggal responden, usianya, tingkat pendidikan.
2. Persepsi : apa yang diketahui orang tentang dunianya (kita bisa meminta kepada responden untuk menyebutkan nama anggota parlemen atau nama menteri cabinet).
3. Opini : apa pandangan orang tentang dunianya (penilaian tentang kompetensi pimpinan partai politik).
4. Sikap : ini adalah evaluasi yang relative stabil (dukungan pada pemerataan kekayaan, penolakan terhadap olahraga yang berdarah-darah).
5. Laporan perilaku : bagaimana orang bertindak.

2.2.5 Pengukuran Tingkat Kepuasan Pelanggan

2.2.5.1 Pengertian Kepuasan

Kepuasan adalah tingkat perasaan seseorang setelah membandingkan kinerja atau hasil yang dirasakannya dengan harapannya [10]. Jadi, tingkat kepuasan merupakan fungsi dari perbedaan antara kinerja yang dirasakan dengan harapan. Apabila kinerja di bawah harapan, maka pelanggan akan kecewa. Bila kinerja sesuai dengan harapan, pelanggan akan puas. Sedangkan bila kinerja melebihi harapan, pelanggan akan sangat puas. Harapan pelanggan dapat dibentuk oleh pengalaman masa lampau, komentar dari kerabatnya serta janji dan informasi

pemasar dan saingannya. Pelanggan yang puas akan setia lebih lama, kurang sensitif terhadap harga dan memberi komentar yang baik tentang perusahaan.

2.2.5.2 Mengukur Tingkat Kepuasan Pelanggan

Pelanggan memang harus dipuaskan, sebab jika mereka tidak puas akan meninggalkan perusahaan dan menjadi pelanggan pesaing, hal ini akan menyebabkan penurunan penjualan dan pada gilirannya akan menurunkan laba dan bahkan kerugian. Tingkat kepuasan pelanggan sangat tergantung pada mutu suatu produk.

Aspek mutu ini bisa diukur. Pengukuran tingkat kepuasan erat hubungannya dengan mutu produk (barang atau jasa). Pengukuran aspek mutu bermanfaat bagi pimpinan yaitu :

1. Mengetahui dengan baik bagaimana jalannya atau bekerjanya proses bisnis
2. Mengetahui di mana harus melakukan perubahan dalam upaya melakukan perbaikan secara terus-menerus untuk memuaskan pelanggan, terutama untuk hal-hal yang dianggap penting oleh para pelanggan.
3. Menentukan apakah perubahan yang dilakukan mengarah ke perbaikan (*improvement*).

Mengukur tingkat kepuasan para pelanggan sangat perlu walaupun tidak semudah mengukur berat atau tinggi badan pelanggan tersebut. Alat yang dipergunakan untuk mengukur tingkat kepuasan adalah daftar pertanyaan (*quisioner*). Data yang diperoleh berupa jawaban dari para pelanggan terhadap pertanyaan yang diajukan.

2.2.5.3 Cara Pembentukan Kuesioner Kepuasan Pelanggan

Pembentukan daftar pertanyaan kepuasan pelanggan ditempuh dalam 4 (empat) tahap yaitu :

1. Menentukan pertanyaan (butir) yang akan dipergunakan dalam daftar pertanyaan.
2. Memilih bentuk jawaban (*response format*).

Langkah kedua pembentukan skala akan memilih suatu format jawaban untuk daftar pertanyaan. Suatu format jawaban menentukan bagaimana pelanggan dapat menjawab butir-butir dalam pertanyaan. Banyak sekali format jawaban atau metode penskalaan untuk daftar pertanyaan, diantaranya adalah format *check list* dan format tipe *likert* [10].

a. Bentuk *Cheklist*

   Untuk tiap atribut pada kuesioner, responden diijinkan untuk menjawab "ya" atau "tidak". Keuntungan dari metode ini adalah kemudahan untuk menjawabnya.

b. Bentuk *Likert-type*

   Untuk memungkinkan respon dengan tingkat yang beragam, format jawaban dengan skala *Likert* biasa digunakan. Metode ini membagi prosedur skala yang menampilkan sebuah bipolar continuum.

3. Menulis introduksi/pengenalan pada daftar pertanyaan

   Langkah berikutnya adalah menulis pengenalan (introduksi) daftar pertanyaan kepuasan pelanggan. Pengenalan ini harus singkat, dan harus menjelaskan maksud daftar pertanyaan serta memberikan instruksi atau petunjuk cara mengisi daftar pertanyaan.

4. Menentukan isi akhir daftar pertanyaan (memilih beberapa butir yang pokok di antara sekian banyak butir kepuasan yang akan dijadikan ukuran tingkat kepuasan)

2.2.5.4 *Importance Performance Analysis (*IPA*)*

Metode *Importance Performance Analysis (*IPA*)* pertama kali diperkenalkan oleh Martilla dan James (1977) [11] dengan tujuan untuk mengukur hubungan antara persepsi konsumen dan prioritas peningkatan kualitas produk atau jasa yang dikenal pula sebagai *quadrant analysis* [12]. IPA telah diterima secara umum dan dipergunakan pada berbagai bidang kajian karena kemudahan untuk diterapkan dan tampilan hasil analisa yang memudahkan usulan perbaikan kinerja. IPA mempunyai fungsi utama untuk menampilkan informasi berkaitan dengan faktor-faktor pelayanan yang menurut konsumen sangat mempengaruhi

kepuasan dan loyalitas mereka, dan faktor-faktor pelayanan yang menurut konsumen perlu ditingkatkan karena kondisi saat ini belum memuaskan.

Dalam metode ini terdapat 2 buah variabel yang diwakilkan oleh huruf X dan Y, di mana :

X = Tingkat kinerja perusahaan yang dapat memberikan kepuasan para pelanggan

Y = Tingkat kepentingan pelanggan

Adapun rumus yang digunakan adalah :

$$Tki = \frac{Xi}{Yi}\ X\ 100\%$$

Di mana : Tki = Tingkat kesesuaian responden

Xi = Skor penilaian kinerja perusahaan

Yi = Skor penilaian kepentingan pelanggan

Selanjutnya sumbu mendatar X akan diisi oleh skor tingkat pelaksanaan, sedangkan sumbu tegak Y akan diisi oleh skor tingkat kepentingan. Dalam penyederhanaan rumus, maka untuk setiap faktor yang mempengaruhi kepuasan pelanggan dengan :

$$\overline{X} = \frac{\sum Xi}{n} \qquad\qquad \overline{Y} = \frac{\sum Yi}{n}$$

Di mana : $\overline{X}$ = Skor Rata-rata tingkat pelaksanaan/kepuasan

$\overline{Y}$ = Skor rata-rata tingkat kepentingan

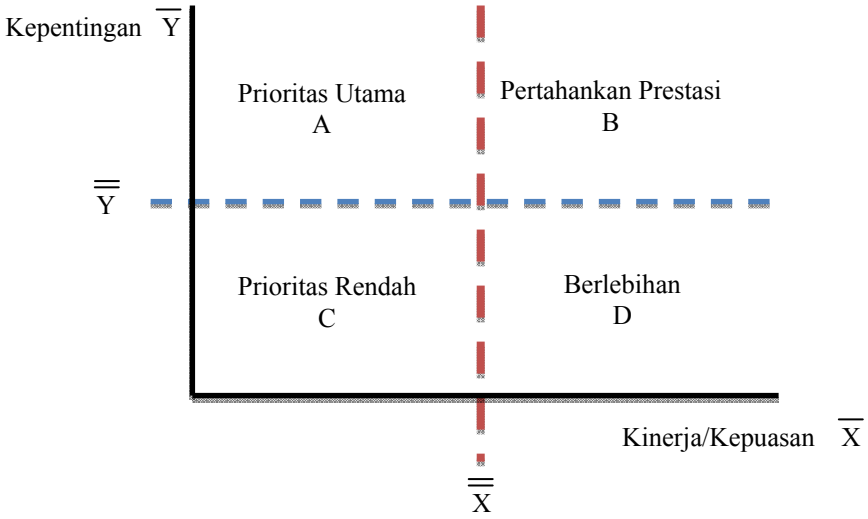
n = jumlah responden

Rumus Selanjutnya :

$$\bar{\bar{X}} = \frac{\sum_{i=1}^{n} \overline{Xi}}{k} \qquad\qquad \bar{\bar{Y}} = \frac{\sum_{i=1}^{n} \overline{Yi}}{k}$$

Di mana k = Banyaknya atribut/fakta yang dapat mempengaruhi kepuasan pelanggan

Selanjutnya tingkat unsur-unsur tersebut akan dijabarkan dan dibagi menjadi empat bagian ke dalam diagram kartesius seperti gambar di bawah ini.



Gambar 2.5 Diagram Kartesius untuk metode *IPA* [10]

Keterangan :

A. Menunjukan faktor atau atribut yang dianggap mempengaruhi kepuasan pelanggan, termasuk unsur-unsur jasa yang dianggap sangat penting, namun manajemen belum melaksanakannya sesuai keinginan pelanggan. Sehingga mengecewakan/tidak puas.
B. Menunjukan unsur jasa pokok yang telah berhasil dilaksanakan perusahaan, untuk itu wajib dipertahankannya. Dianggap sangat penting dan sangat memuaskan.
C. Menunjukan beberapa faktor yang kurang penting pengaruhnya bagi pelanggan, pelaksanaannya oleh perusahaan biasa-biasa saja. Dianggap kurang penting dan kurang memuaskan.
D. Menunjukan faktor yang mempengaruhi pelanggan kurang penting, akan tetapi pelaksanaannya berlebihan. Dianggap kurang penting tetapi sangat memuaskan.

2.2.6 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 [13].

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.

2. 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.

3. 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.

4. 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.

5. 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 copy*, layar antar muka (*User interface*)
didesain ulang, dukungan format video VP8 dan *Web*M, efek audio baru (*reverb, equalization, headphone virtualization* dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

6. 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.

7. 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.

8. Android versi 4.1 (*Jelly Bean*)

Sampai tulisan ini di tulis Versi Android *Jelly Bean* adalah versi Android yang terbaru. Android *Jelly Bean* yaang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input *keyboard*, desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat.Tak ketinggalan Google Now

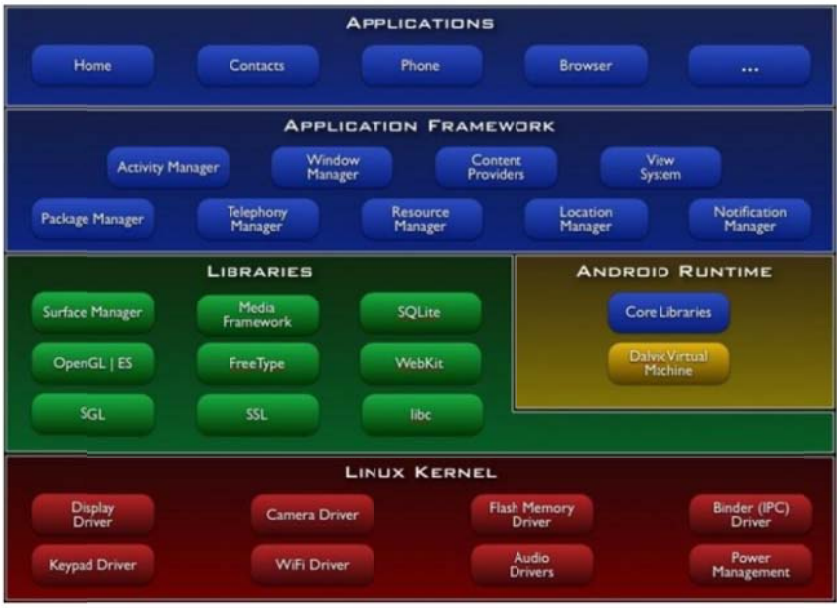
juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga.

2.2.6.1 *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 WebKi*t.
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.

2.2.6.2 Arsitektur Android



Gambar 2.6 Arsitektur Android **[13]**

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. *Application*s *Framework*s

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 Framework*s 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.

2.2.7 *Database* (Basis Data)

Database 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 [14] .

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.

2.2.7.1 *Entity Relationship Diagram (*ERD*)*

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan ERD. ERD ang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antarentitas [15]. Model ini dinyatakan dalam bentuk diagram. Perlu diketahui bahwa ERD tidak mencerminkan bentuk fisik yang nantinya akan disimpan dalam *database*, melainkan hanya bersifat konseptual.

2.2.8 *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 [16].

2.2.8.1 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 [17].

2.2.8.2 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". Kelas adalah pemodelan dari objek yang berisi informasi (aturan) tentang sifat karakteristik (data) dan tingkah laku (method) yang dimiliki oleh objek tersebut [17].

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.

2.2.8.3 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)*.

2.2.8.4 *Enkapsulasi*

Enkapsulasi *(encapsulation)* adalah pengemasan data dan prosedur dalam objek. Pengkapsulan merupakan upaya penyembunyian informasi yang diterapkan pada sehingga data tidak dapat diakses secara langsung oleh pihak di luar objek [8]. 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”; }
```

2.2.8.5 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. Dengan kata lain pewarisan merupakan sifat dalam bahasa berorientasi objek yang memungkinkan sifat-sifat dari suatu kelas diturunkan ke kelas lain [8]. 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”); }
}
```

2.2.8.6 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.

2.2.9 *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 [18].

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

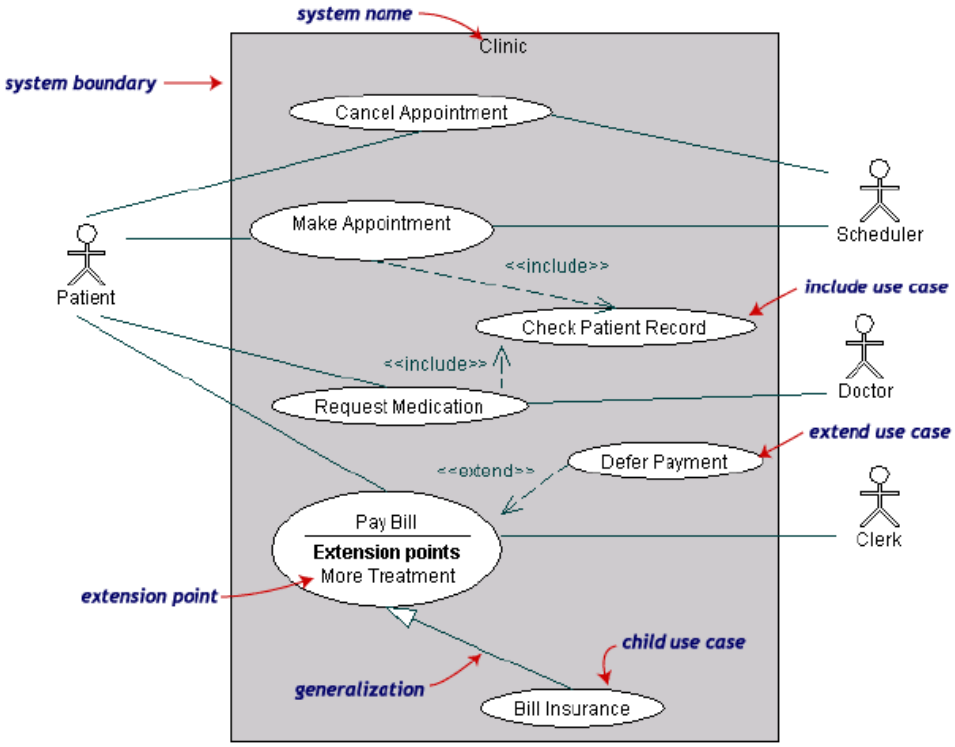
e. UML 1.5

f. UML 2.0

Berikut ini adalah jenis-jenis diagram yang umum digunakan :

1. *Use case Diagram*

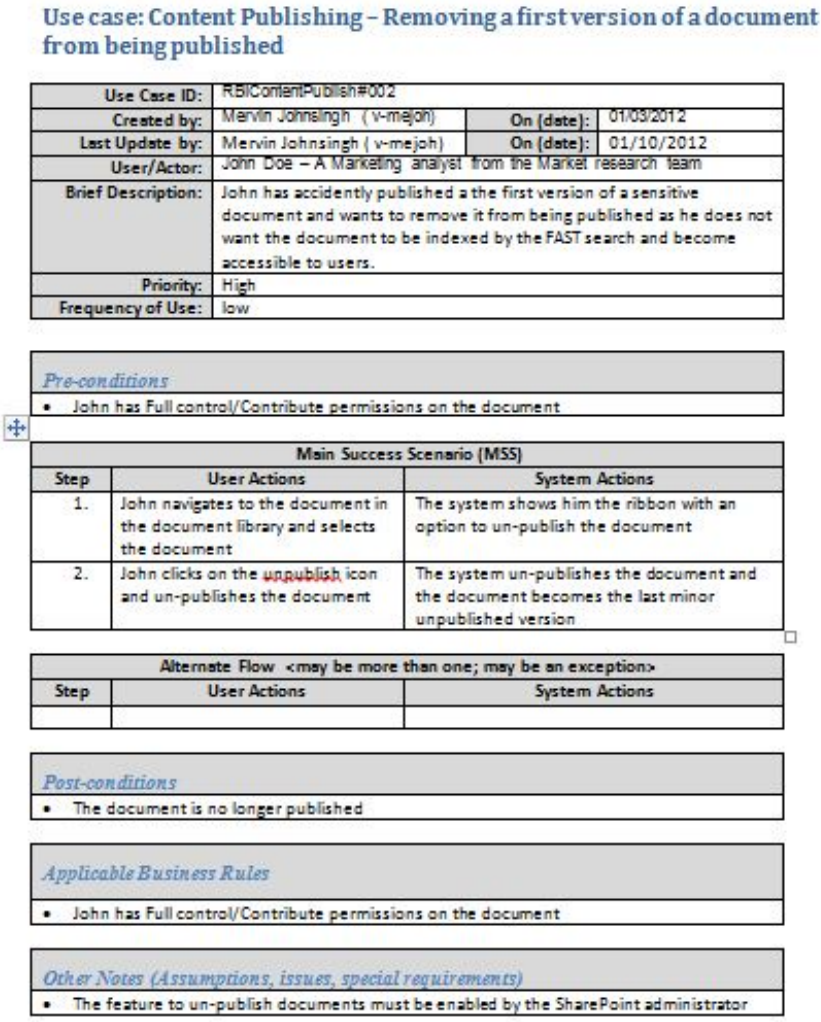
Mendeskripsikan apa yang sistem lakukan tanpa mendeskripsikan bagaimana sistem menyelesaikannya. Dibuat berdasarkan interaksi dan relasi dari individual *use case* yang berisi aktor, *event*, dan *use case*.



Gambar 2.7 *Use case Diagram* [19]

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*:



Use case: Content Publishing – Removing a first version of a document from being published

Use Case ID:	RBIContentPublish#002		
Created by:	Mervin Johnsingh (v-mejoh)	**On (date):**	01/03/2012
Last Update by:	Mervin Johnsingh (v-mejoh)	**On (date):**	01/10/2012
User/Actor:	John Doe – A Marketing analyst from the Market research team		
Brief Description:	John has accidently published a the first version of a sensitive document and wants to remove it from being published as he does not want the document to be indexed by the FAST search and become accessible to users.		
Priority:	High		
Frequency of Use:	low		

Pre-conditions

- John has Full control/Contribute permissions on the document

Main Success Scenario (MSS)		
Step	**User Actions**	**System Actions**
1.	John navigates to the document in the document library and selects the document	The system shows him the ribbon with an option to un-publish the document
2.	John clicks on the unpublish icon and un-publishes the document	The system un-publishes the document and the document becomes the last minor unpublished version

Alternate Flow <may be more than one; may be an exception>		
Step	**User Actions**	**System Actions**

Post-conditions

- The document is no longer published

Applicable Business Rules

- John has Full control/Contribute permissions on the document

Other Notes (Assumptions, issues, special requirements)

- The feature to un-publish documents must be enabled by the SharePoint administrator

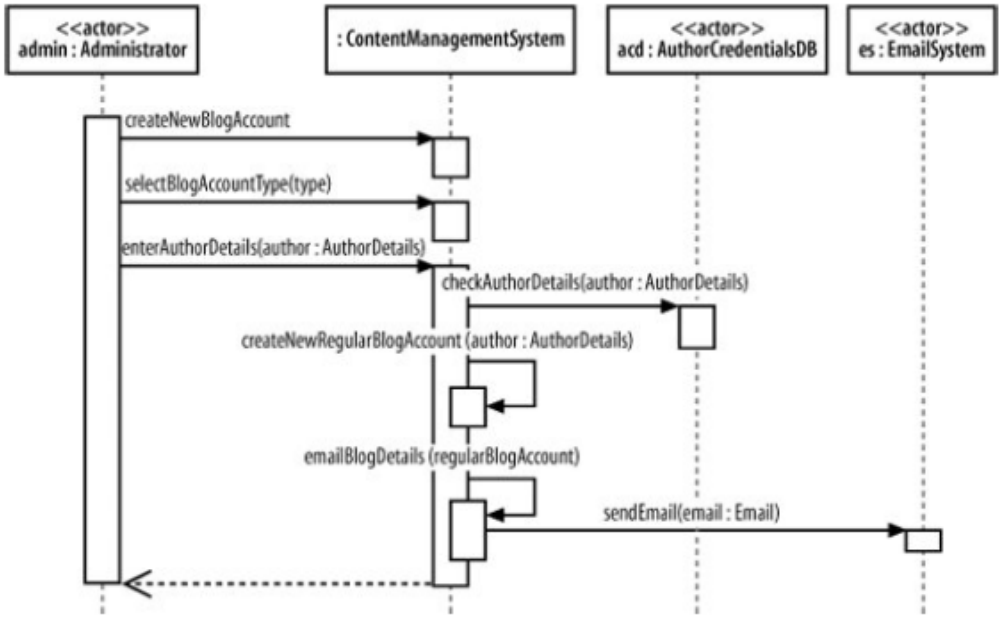
Gambar 2.8 *Use case Scenario* [20]

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

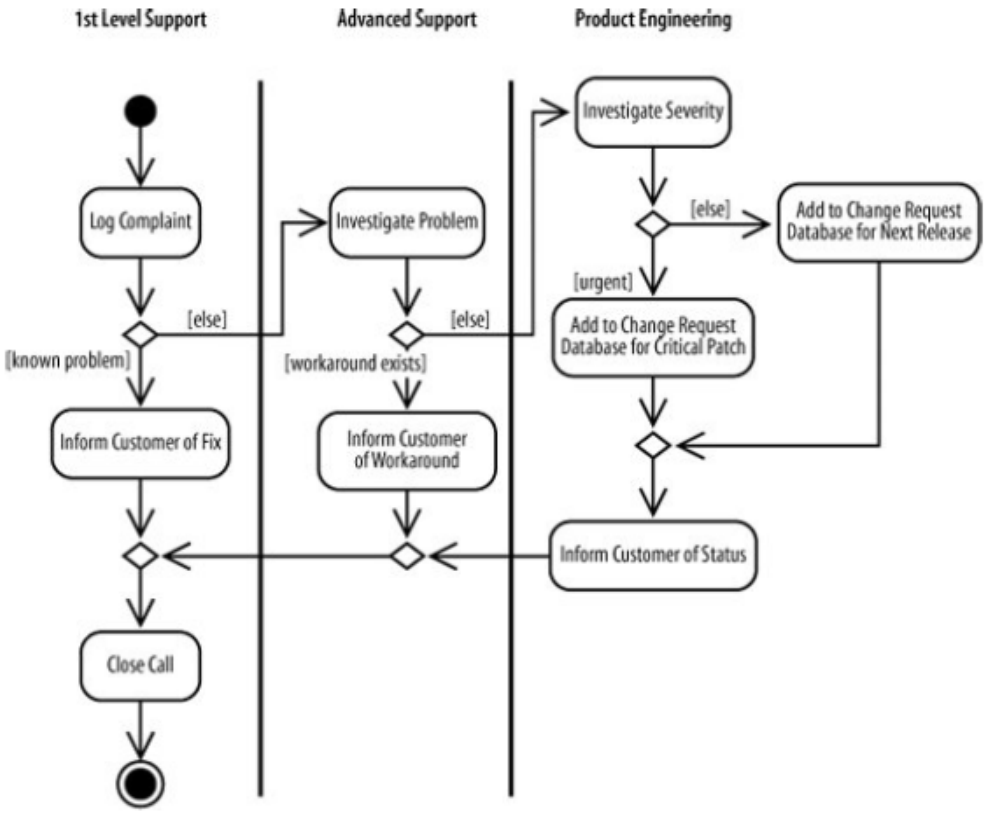
tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Contoh *Sequence diagram:*



Gambar 2.9 *Sequence Diagram* [21]

4. *Activity Diagram*

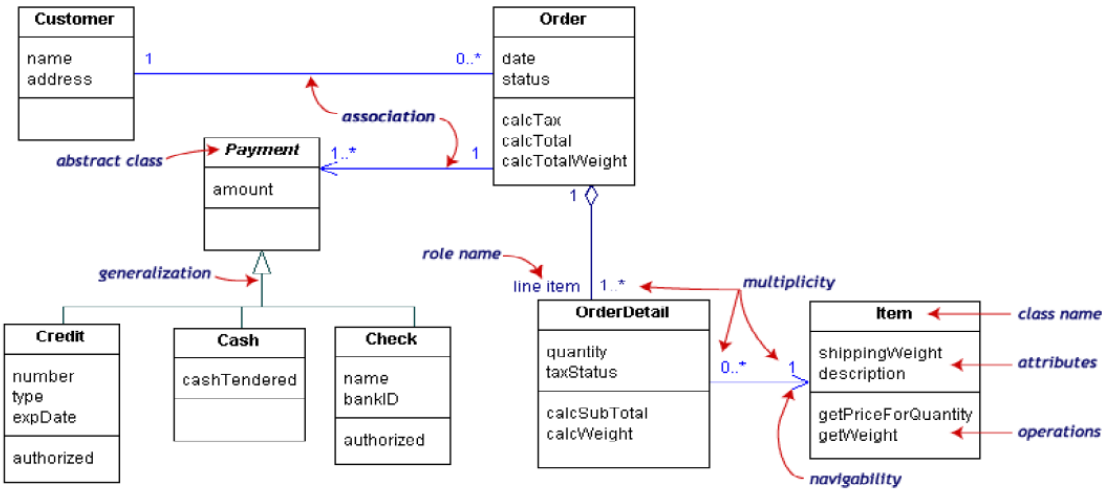
Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses yang dipakai untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Contoh *activity diagram:*



Gambar 2.10 *Activity Diagram* [21]

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 2.11 *Class Diagram* [19]

2.2.10 *Web*

Web merupakan kumpulan informasi pada *server* komputer yang terhubung satu sama lain dalam jaringan internet maupun intranet. Sedangkan aplikasi berbasis *web (web based)* secara prinsip menyerupai aplikasi dalam komputer biasa. Yang membedakan adalah dalam aplikasi *web based* menggunakan tag-tag html sebagai dasar tampilan, sedangkan aplikasi program komputer menggunakan berbagai *platform* bahasa pemrograman. Aplikasi *web* terdiri dari beberapa golongan, diantaranya adalah :

1. Inisialisasi Data Master

   Proses yang dilakukan adalah pengisian satu atau beberapa tabel yang akan digunakan sebagai referensi program aplikasi. Pengisian data master secara umum terdiri dari input data, verifikasi data, proses *insert* data jika data belum terdaftar sebelumnya, atau *update* data jika data telah ada sebelumnya, serta *delete* data jika data hendak dihapus dari tabel master.

2. Operasi *Reporting*

   Operasi mendapatkan data hasil query yang telah diolah sehuingga didapat informasi yang digunakan untuk kepentingan manajerial dan rekapitulasi beberapa transaksi. Selain beberapa golongan tersebut juga terdapat operasi

verifikasi, penampilan profil, penyampaian pesan, pengisian pesan khusus serta kombinasi dari beberap golongan tersebut.

2.2.11 HTML

HTML kependekan dari *Hyer Text Markup Language* adalah file teks murni yang dapat dibuat dengan editor teks sembarang [22]. Dokumen ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan dalam *browser web surfer*. Dokumen ini umumnya berisi informasi atau pun *interface* aplikasi di dalam internet.

Definisi elemen

Dokumen html disusun oleh elemen-elemen. “elemen” merupakan istilah bagi komponen-komponen dasar pembentuk dokumen HTML. Beberapa contoh elemen adalah *head*, *body*, *table*, paragraf, dan *list*.

Definisi tag

Untuk menandai berbagai elemen dalam suatu dokumen HTML, kita menggunakan tag. Tag HTML terdiri atas sebuah kurung kiri (<, tanda lebih lecil), sebuah nama tag, dan sebuah kurung sudut kanan (>, tanda lebih besar). Tag umumnya berpasangan (misalnya <H1> dengan </H2>), tag yang menjadi pasangan selalu diawali dengan karakter garing (/, garis miring). Tag yang pertama menunjukan tag awal yang berarti awal elemen, dan yang kedua menunjukan tag akhir, berarti akhir elemen.

Elemen yang dibutuhkan.

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 informasi tentang dokumen tersebut, dan elemen *body* berisi teks yang sebenarnya yang tersusun dari *link*, grafik, paragraf, dan elemen lainnya

Secara umum dokumen *web* dibagi manjadi dua section (bagian), yaitu *section head* dan *section body*. Sehingga setiap dokumen HTML harus mempunyai pola dasar sebagai berikut :

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

Setiap dokumen HTML harus diawali dengan menuliskan tag <*html*> dan tag </*html*> di akhir dokumen. Tag ini menandai elemen HTML yang berarti dokumen ini adalah dokumen HTML. Dalam satu dokumen hanya ada satu elemen HTML.

2.2.12 *Personal Home Page* (PHP)

PHP adalah kombinasi antara bahasa pemrograman dan aplikasi *server*. PHP merupakan *server-side Script*ing yang digunakan untuk membuat dan menjalankan aplikasi *web server* yang dinamis, interaktif dan mempunyai performansi tinggi [23]. Aplikasi *server* adalah program yang terdiri atas teknik-teknik dalam satu paket yang meliputi :

a. Ketangguhan bahasa pemrograman.
b. Pengaksesan basis data ke media penyimpanan yang permanen.
c. Mendukung internet *protocol*, khusunya HTTP dan *e-mail*.

PHP mampu berhubungan dengan basis data dan dapat diintegrasikan dengan HTML. PHP dapat berjalan dalam *web server* yang berbeda dalam sistem operasi yang berbeda pula. PHP ditulis dalam bahasa C, sehingga sebagian besar sintak PHP mirip dengan C dan *perl*. Beberpa keuntungan menggunakan PHP adalah sebagai berikut :

a. *Open source*, semua *source code* PHP tesedia.
b. PHP diterbitkan secara gratis, tidak ada biaya dalam mengimplementasikan.
c. *Cross-platform*, PHP dapat berjalan dalam sistem operasi Windows 98, Windows NT dan Macintosh.

d. PHP juga dapat berjalan pada *server Microsoft Peronal Web Server*, IIS dan Aphace.
e. *Embedded*-html, karena itu PHP mudah dipelajari.

2.2.13 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 [17].

2.2.14 MySQL

MySQL adalah perangkat lunak *server* basis data yang dapat melayani banyak pengguna, dan dapat melakukan proses dalam waktu yang bersamaan *(multi threaded)* [24]. MySQL juga sangat cepat dan fleksibel dan dapat menyimpan data yang berkapasitas besar. Dalam perancangannya, MySQL mengutamakan kecepatan proses, ketangguhan dan kemudahan bagi penggunannya.

Pada saat ini MySQL adalah *server* basis data yang paling banyak diminati banyak perusahaan di dunia. MySQL menyediakan lingkungan yang mudah dan efisien untuk menyimpan data dan mengakses basis data berkapasitas besar lebih cepat dari pada *server* basis data lainnya. Keuntungan lainya adalah bahwa MySQL *server* tidak membutuhkan spesifikasi *hardware* yang mahal. MySQL *server* memungkinkan membuat aplikasi *client* untuk mengakses dan memanipilasi data di *server* basis data melalui basis data *provider*. Keuntungan utama menggunakan MySQL *server* adalah mudah digunkan, berkecepatan tinggi dan membutuhkan sedikit memori.

2.2.15 *Cascading Style Sheet* (CSS)

CSS merupakan singkatan dari *Cascading Style Sheet* merupakan bahasa pemrograman *web* yang di desain khusus untuk mengendalikan dan membangun

berbagai komponen dalam *web* sehingga tampilan *web* lebih rapi, terstruktur, dan seragam.

Tujuan utama dari CSS adalah untuk memisahkan konten utama dengan tampilan dokumen lainnya. Dengan adanya pemisahan ini, akses konten pada *web* meningkat. *Web* yang menggunakan CSS akan lebih ringan dan mudah untuk dibuka dibandingkan dengan *web* yang tidak menggunakan CSS. Perbedaan ini akan semakin terasa ketika *web* yang dibuka mempunyai data yang banyak [25].

Tujuan lainnya adalah untuk mempercepat pembuatan halaman *web*. Anda hanya perlu membuat satu propoerti dan properti tersebut dapat digunakan pada elemen lainnya, artinya, kita tidak perlu menulis ulang kode program yang digunakan berulang kali.

CSS saat ini dikembangkan oleh *World Wide Web Consortium* atau yang biasa lebih dikenal dengan istilah W3C. Sehingga CSS menjadi bahasa standar dalam pembuatan *web*. CSS bukan menggantikan kode html, tetapi hanya difungsikan sebagai penopang ayau pendukung (pelengkap) dari file html yang berperan dalam penataan kerangka dan *layout*.

Dengan menggunakan CSS, akan banyak keuntungan yang dapat kita peroleh, diantaranya :

1. Memisahkan pembuatan dokumen (CSS & HTML)
2. Mempermudah dan mempersingkat pembuatan dan pemeliharaan dokumen *web*
3. Fleksibel, interaktif, tampilan lebih menarik dan nyaman dipandang
4. Dapat digunakan pada semua *web browser*

2.2.16 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 [22].

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 "\".

2.2.17 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 kode, tapi dengan hasil yang lebih banyak [26].

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.

2.2.18 *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 [27].

2.2.19 *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 [28] .

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.

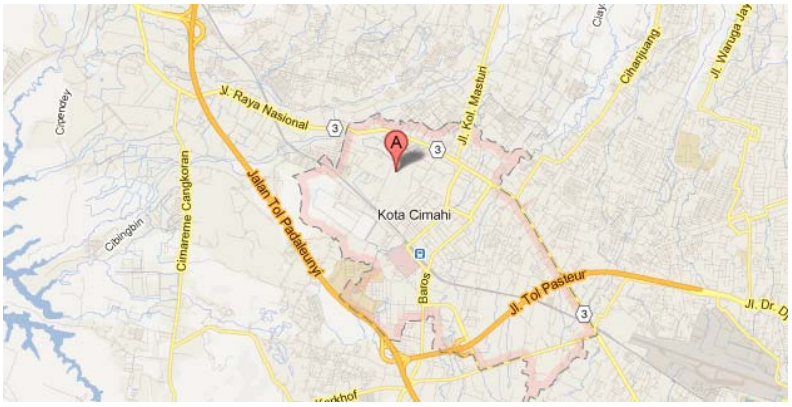
3. Penerima bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi (posisi tiga dimensi yaitu koordinat di bumi plus ketinggian), arah, jarak, dan waktu yang diperlukan oleh pengguna.

2.2.20 *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 [29].

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 2.12 Tampilan Bentuk *Map*

Tampilan Bentuk *Map*

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



Gambar 2.13 Tampilan Bentuk Sattelite

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



Gambar 2.14 Tampilan Bentuk Earth

2.2.21 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 [30].

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_ufC3CFXhHIE1NvwkxS4Rz1LFzG0odNPtk8VLkdrQF5grA

2.2.22 Perangkat Lunak Penunjang

2.2.22.1 Adobe Dreamweaver CS6

Adobe Dreamweaver CS6 adalah program editor HTML professional untuk mendesain, mengkode dan mengembangkan situs, halaman *web*, serta aplikasi *web* .Tampilan *fitur editing* dalam Adobe Dreamweaver CS6 akan membantu dalam membuat sebuah halaman *web* tanpa harus menuliskan kode JavaScript maupun HTML. Dreamweaver juga dapat membantu dalam membangun *dynamic database* untuk aplikasi *web* dengan menggunakan *server* seperti ASP, ASPNet,ColdFusion Markup Language (CFML), JSP dan PHP.

2.2.22.2 Eclipse

Eclipse adalah 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 [31]:

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)

12. Eclipse Juno Sr1 Packages (v 4.2.1)

12. Eclipse Juno Sr2 Packages (v 4.2.2)

2.2.22.3 *Navicat Premium Enterprise*

Navicat adalah *software* yang berfungsi memudahkan kita saat menciptakan (*Data Definition Language*) dan memanipulasi (*Data Manipulation Language*) *database* MySQL, SQL *Server*, SQLite, Oracle atau PostgreSQL, seperti menciptakan tabel, menghapus tabel, memasukan baris, menghapus baris, mengubah baris, menampilkan data, menciptakan user, dan lain sebagainya [32].

Navicat menyediakan sarana untuk memanipulasi *database* dengan mudah dan ramah. Selain itu *Navicat Premium Enterprise* juga mendukung sebagian besar fitur yang ada di MySQL, SQL *Server*, SQLite, Oracle dan PostgreSQL termasuk Stored Prosedur, *Event*, *Trigger*, Fungsi, *View*, dll.

2.2.22.4 *Java Development Kit* (JDK)

JDK adalah Java *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 [33].

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*)

2.2.22.5 *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.

2.2.22.6 *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.

2.2.22.7 Xampp *Server*

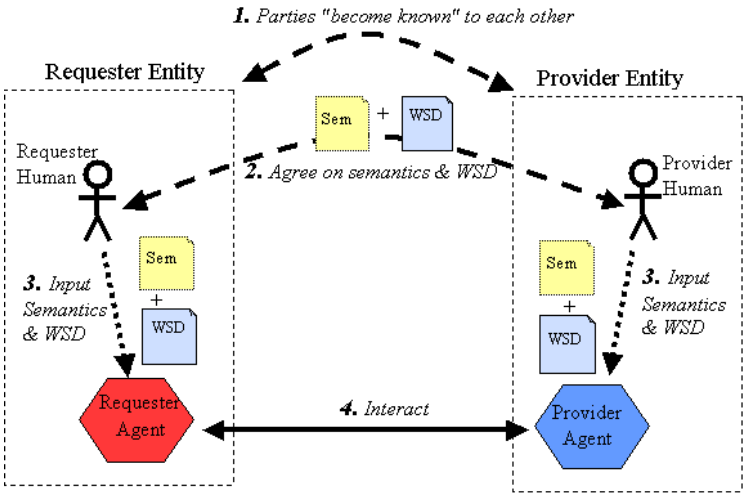
XAMPP *Server* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Yang berfungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP *Server* sebagai *web server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl

[34]. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis

2.2.22.8 *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 *web*site 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 [35].



Gambar 2.15 *The General Process of Engaging a Web Service* [35]

2.2.23 Alat Bantu Pemodelan Sistem

Pemodelan sistem merupakan cara untuk memudahkan dalam melakukan perancangan perangkat lunak. Dimana tujuan dari pemodelan ini adalah untuk memahami perilaku sistem yang akan dirancang dengan lebih mendetail dan menggambarkan sistem secara lengkap dan sederhana sehingga dapat digunakan untuk menjelaskan jalannya sistem.

2.2.23.1 Bagan alir dokumen

Bagan alir dokumen (*document flowmap*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya [36].

2.2.23.2 Diagram Konteks

Diagram Konteks merupakan diagram yang menggambarkan suatu sistem secara garis besar. Diagram Konteks ini menggambarkan hubungan *input* / *output* antara sistem dengan dunia luarnya [37]. Suatu diagram konteks selalu mengandung satu dan hanya satu proses saja. Proses ini mewakili proses dari seluruh sistem.

2.2.23.3 *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) merupakan model grafis dari sistem yang menunjukkan aliran data dalam sistem secara terstruktur dan jelas .DFD memperlihatkan gambaran tentang masukan-proses-keluaran dari suatu sistem/perangkat lunak. Yaitu objek-objek data mengalir ke dalam perangkat lunak, kemudian objek-objek data itu akan ditranformasi oleh elemen-elemen pemrosesan, dan objek-objek data hasilnya akan mengalir keluar dari sistem atau perangkat lunak [38] .

Beberapa Notasi atau simbol yang digunakan dalam DFD adalah sebagai berikut:

1. Kesatuan luar (*External entity*) disimbolkan dengan suatu notasi kotak Entitas ini merupakan kesatuan diluar sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang berada di lingkungan luar lainnya yang akan memberikan input atau menerima *output* dari sistem.
2. Arus data pada DFD disimbolkan dengan suatu notasi panah. simbol panah ini menggambarkan aliran dari data yang mengalir diantara proses, simpanan data, dan entitas baik berupa masukan untuk sistem ataupun hasil dari proses sistem.
3. Proses (*Process*) disimbolkan dengan suatu notasi lingkaran. Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer karena ada arus data yang masuk ke dalam proses untuk dihasilkan aliran dari data yang akan keluar dari proses.
4. Simpanan Data disimbolkan dengan sepasang garis horizontal paralel yang tertutup disalah satu ujungnya.Simpanan Data (*Data Store*) merupakan simpanan data yang dapat berupa sebagai berikut :
   a. Suatu file atau *database* di sistem komputer.
   b. Suatu arsip atau catatan manual.
   c. Suatu kotak tempat data di atas meja.
   d. Suatu tabel acuan manual.
   e. Suatu agenda atau buku.