BAB 2
LANDASAN TEORI

2.1 Gempa Bumi

Gempa bumi adalah peristiwa bergetarnya bumi akibat pelepasan energi di dalam bumi secara tiba-tiba yang ditandai dengan patahnya lapisan batuan pada kerak bumi. Energi yang dihasilkan dipancarkan kesegala arah berupa gelombang gempa bumi sehingga efeknya dapat dirasakan sampai ke permukaan bumi [1].

2.1.1 Penyebab Terjadinya Gempa

Penyebab terjadinya gempa bumi dihasilkan dari pergerakan lempeng-lempeng tektonik. Energi yang dihasilkan dipancarkan kesegala arah berupa gelombang gempa bumi sehingga efeknya dapat dirasakan sampai ke permukaan bumi. Lempeng tektonik merupakan permukaan bumi terpecah menjadi beberapa lempeng tektonik besar. Lempeng tektonik adalah segmen keras kerak bumi yang mengapung diatas *astenosfer* yang cair dan panas. Oleh karena itu, maka lempeng tektonik bebas untuk bergerak dan saling berinteraksi satu sama lain. Daerah perbatasan lempeng-lempeng tektonik, merupakan tempat-tempat yang memiliki kondisi tektonik yang aktif, yang menyebabkan gempa bumi, gunung berapi dan pembentukan dataran tinggi.

2.1.2 Kedalaman dan Kekuatan Pusat Gempa

Kedalaman pusat gempa dapat di kategorikan kedalam tiga bagian, yaitu :

1. Gempa Bumi Dangkal

   Gempa Bumi Dangkal adalah gempa bumi yang pusat gempa berada kurang dari 50 Km dari permukaan bumi. Di Indonesia gempa bumi dangkal letaknya terpencar di beberapa wilayah di Indonesia. Gempa semacam ini dapat menimbulkan kerusakan besar. Makin dangkal pusat gempa , daya perusaknya semakin besar.

2. Gempa Bumi Menengah

   Gempa Bumi Menengah adalah gempa bumi yang pusat gempa yang sumbernya berada antara 50 km - 300 km di bawah permukaan bumi. Di Indonesia gempa bumi menengah terbentang sepanjang Sumatra sebelah

Barat, Jawa sebelah Selatan, selanjutnya Nusa Tenggara antara Sumbawa dan Maluku, akhirnya sepanjang Teluk Tomini, Laut Maluku ke Filipina. Gempa menengah dengan fokus kurang dari 150 km di bawah permukaan masih dapat menimbulkan kerusakan.

3. Gempa Bumi Dalam

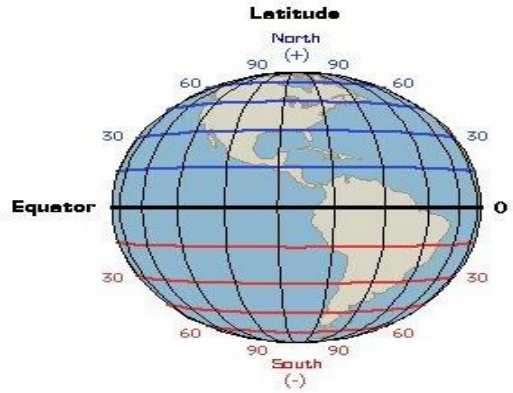
   Gempa Bumi Dalam adalah gempa bumi yang pusatnya berada lebih dari 300 km di bawah permukaan bumi. Di Indonesia gempabumi dalam berada di bawah Laut Jawa, Laut Flores, Laut Banda dan Laut Sulawesi. Gempa dalam tidak membahayakan.

2.1.3 Kekuatan Gempa Bumi

Kekuatan gempa bumi adalah ukuran mutlak yang dikeluarkan oleh pusat gempa. Pendapat ini pertama kali dikemukakan oleh Richter dengan besar antara 0 sampai 9. Selama ini gempa terbesar tercatat sebesar 8,9 skala Richter terjadi di Columbia tahun 1906. Pengukuran kekuatan gempa secara kualitatif yaitu dengan melihat besarnya kerusakan yang diakibatkan oleh gempa. Kerusakan tersebut dapat dikatakan sebagai intensitas gempa bumi.

2.1.4 Garis Lintang

Garis lintang adalah garis maya yang melingkari bumi ditarik dari arah barat hingga ke timur atau sebaliknya, atau sejajar dengan garis khatulistiwa. Garis lintang terus melingkari bumi, dari khatulistiwa hingga ke bagian kutub utara dan kutub selatan bumi. khatulistiwa disebut juga sebagai 0º. Makin ke utara atau ke selatan, angka derajatnya makin besar hingga pada angka 90º pada ujung kutub utara atau kutub selatan. Satuan derajat bisa juga disebut Jam sehingga setiap derajat terbagi menjadi 60 menit dan setiap menit terbagi lagi menjadi 60 detik. Jika misalnya garis lintang suatu tempat tertulis seperti ini 57º 27′ 14”S. Maka dibaca sebagai 57 derajat 27 menit 14 detik Lintang Selatan. Pada sistem pemetaan internasional huruf (U) sebagai Lintang Utara diganti dengan huruf N (North). Sedangkan Lintang Selatan tetap menggunakan huruf (S). Dengan garis lintang inilah yang dapat dijadikan ukuran dalam mengukur sisi utara dan selatan koordinat suatu titik di belahan bumi. Garis equator dapat dilihat pada Gambar 2.1



Gambar 2.1 Garis Equator

Garis lintang dapat dibedakan menjadi 2 wilayah, yaitu utara atau yang biasa disebut lintang utara dan selatan atau yang biasa disebut lintang selatan, dimana nilai koordinat di bagian utara selalu positif dan nilai koordinat di bagian selatan adalah negatif. Nilai-nilai yang dijadikan patokan ukuran garis lintang ini dapat dilihat pada Tabel 2.1

Tabel 2.1 Nilai Ukuran Garis Lintang

GARIS	NILAI
Garis paling atas (kutub utara)	90 derajat
Garis paling tengah (equator)	0 derajat
Garis paling bawah (kutub selatan)	-90 derajat

Bumi memiliki diameter 12.756 Km, dan keliling 40.000 Km. Lingkar bumi sebesar 360^0 garis bujur, berarti setiap 1^0 adalah 111,11 Km. Artinya setiap 1^0 garis bujur dan lintang pada peta mewakili jarak sebesar 111,11 km di permukaan bumi [2]. Nilai dari persamaan derajat kedalam bentuk satuan kilometer (Km) dapat dilihat pada Tabel 2.2

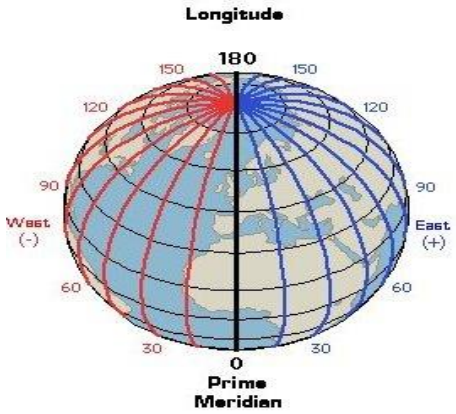
Tabel 2.2 Persamaan Derajat Dalam Bentuk Kilometer

NILAI	KETERANGAN
1^0	111 Km
1^0	$60'$ Menit
$1'$	1,85 Km

2.1.5 Garis Bujur

Garis Bujur adalah garis yang membujur, membagi bola bumi menjadi dua, yaitu bagian barat dan timur. Garis tersebut menghubungkan ke dua kutub dan

melewati kota Greenwich Inggris. Pengukurannya dalam derajat, menit dan detik. Misalnya 5° 10’ 30”. Titik di barat bujur (0°) dinamakan Bujur Barat sedangkan titik di timur (0°) dinamakan Bujur Timur. Kombinasi garis lintang dan garis bujur ini berguna untuk menentukan suatu lokasi di permukaan bumi. Garis Lintang menandakan sumbu ***x*** dan garis bujur menandakan sumbu ***y*** dalam sistem koordinat cartesian. Sama seperti equator pada garis bujur yang berada ditengah dan memiliki nilai (0°), pada longitude, garis tengah yang bernilai (0°) disebut garis bujur. Sedangkan garis yang berada paling kiri memiliki nilai -90 derajat, dan yang paling kanan memiliki nilai 90 derajat, Garis bujur dapat dilihat pada Gambar 2.2



Gambar 2.2 Garis Bujur

Garis bujur juga dibedakan menjadi 2 wilayah, yaitu bujur timur dan bujur barat, dimana koordinat yang berada di timur selalu bernilai negatif, dan sebaliknya yang berada di barat selalu positif. Nilai satuan ukuran derajat menjadi kilometer pada garis lintang juga sama seperti pada garis bujur.

2.2 Teknologi Internet

Perkembangan teknologi informasi dan komunikasi semakin lama semakin pesat karena manusia memilili kecenderungan untuk selalu mencari dan mendapatkan lebih dari apa yang sudah ada. Perangkat teknologi informasi dan komunikasi diciptakan oleh manusia untuk kepentingan manusia juga. Sampai pada puncaknya sekarang ini, keberadaan teknologi informasi dan komunikasi ternyata mampu menjadi pendorong majunya kehidupan manusia dalam berbagai aspek,

terlebih dengan hadirnya teknolgi Internet yang mampu menjadi jemabatan komunikasi antar manusia.

2.2.1 Internet

Internet adalah suatu jaringan super besar yang terdiri dari ribuan jaringan kecil dengan jutaan komputer yang saling terhubung antara pihak bisnis, institusi pendidikan, organisasi pemerintahan dan individual [3].

Dari definisi diatas, dapat disimpulkan bahwa layanan utama Internet sebagai media untuk :

1. Menyebarkan dan memperoleh informasi, umumnya disajikan dalam bentuk *website*, informasi dapat berupa teks, grafik, vidio, atau dalam bentuk file yang dapat di *dwonload.*
2. Berkomunikasi, baik melalui media *chatting* barbasis teks, maupun berkomunikasi suara layaknya menggunakan telepon kabel
3. Bertukar data, salah satunya dengan menggunakan aplikasi FTP, *website*, maupun koneksi *peer to peer*.
4. Remote *Login*, mampu mengeksekusi komputer dari jauh (telnet).

Semua fasilitas yang telah dijabarkan dapat dilakukan dimana saja dan kapan saja serta tidak mengenal batas geografis, asalkan terhubung dengan Internet.

2.2.2 Layanan Aplikasi Internet

Aplikasi Internet yang tersedia saat ini sudah banyak dan terus bertambah seiring dengan kemajuan teknologi infomasi. Dari sekian banyak aplikasi Internet yang ada, yang banyak dikenal dan digunakan antara lain : *Word Wide Web* (www), *Electronik mail*, *Mailing List* (milis), *Newsgroup*, *Internet Relay Chat* (IRC), *File Transfer Protocol* (FTP), Telnet.

1. *Word Wide Web* (WWW)

   Word Wide Web (WWW) adalah aplikasi yang paling menarik di Internet dan seperti *e-mail*, aplikasi ini sangat penting dan banyak digunakan. Dokumen Web tersebut dibuat dengan format *hypertext* dan *hypermedia* dengan menggunakan *Hypertext Markup Language* (HTML). HTML mempunyai kemampuan untuk menghubungkan link sebuah dokumen

dengan dokumen yang lain. Dokumen yang dibuat dengan HTML tersebut dapat memuat teks, gambar, animasi, audio dan video. Aplikasi ini kadang disebut *The world is at your fingertip* karena bisa mendapatkan informasi dengan mudah sekali bukan sekedar teks bahkan gambar maupun multimedia.

2. *Electronic Mail*

   Electronic Mail adalah salah satu kemudahan atau aplikasi yang paling banyak digunakan di Internet. Hal ini karena *email* merupakan alat komunikasi yang paling murah dan cepat. Melalui email, kitas dapat berhubungan dengan siapa saja yang terhubung ke Internet di seluruh dunia dengan biaya relatif terjangkau. Konsep email adalah seperti anda mengirim surat dengan pos biasa, dengan beralamatkan tempat yang anda tuju. Dari pos tersebut akan disampaikan ke pos yang terdekat dengan alamat yang dituju dan akhirnya sampai ke alamat tersebut. Penerima hanya membuka kotak pos saja yang ada di depan rumah, sedangkan pengirim tidak tahu apakah orang yang dituju tersebut sudah menerima surat tersebut, sampai surat itu dibalas. Dengan email data dikirim secara elektronik sehingga sampai di tujuan sangat cepat. Anda juga dapat mengirim file-file seperti program, gambar, grafik dan sebagainya, dan dapat mengirim ke lebih dari satu orang sekaligus dalam satu waktu.

3. *Mailing List* (Milis)

   Mailing List (Milis) adalah aplikasi Internet yang digunakan sebagai sarana diskusi atau bertukar informasi dalam satu kelompok melalui email. Setiap email yang dikirim ke alamat milis akan dikirim ke seluruh alamat email yang terdaftar sebagai anggota milis tersebut. Milis sebenarnya bentuk lain dari email. Fasilitas ini digunakan oleh kelompok-kelompok untuk bertukar informasi dan berdiskusi sesama anggota kelompok. Milis mempunyai sifat yang sama dengan email.

4. *Newsgroup*

   Newsgroup adalah forum perbincangan, atau boleh dibayangkan sebagai suatu tempat di mana terdapat ruangan-ruangan perbincangan yang unik,

dan tiap-tiap ruangan mempunyai topik perbincangan yang berbeda. Di setiap ruangan itu biasanya terdapat lebih dari satu orang yang saling bertukar pendapat atau pikiran.

5. *Internet Realay Chat* (IRC)

   Internet Realay Chat (IRC) adalah aplikasi Internet yang digunakan untuk berbincang-bincang di Internet . Berbincang-bincang di Internet dikenal dengan istilah *chatting*. *Chatting* dilakukan dengan cara mengetik apa yang ingin anda katakan kepada teman *chatting* anda, kemudian mengirimkannya dalam bentuk teks. Melalui *chating* Anda dapat berkenalan dan bercerita dengan orang lain di belahan dunia lain.

6. *File Transfer Protocol* (FTP)

   File Tranfer Protocol (FTP) merupakan salah satu aplikasi dari TCP/IP yang banyak digunakan untuk memindahkan atau mengcopy *file* dari komputer satu ke komputer lainnya. Aplikasi ini adalah aplikasi yang telah dikembangkan sejak awal perkembangan Internet. *File transfer protocol* adalah protokol sekaligus program yang dapat digunakan untuk melakukan operasi *file* dasar pada *remote host* dan untuk mentransfer *file* antar *host*. Sebagai sebuah program, *file transfer protocol* dapat dioperasikan pengguna untuk melakukan perintah *file* secara manual.

7. Telnet

   Telnet adalah aplikasi Internet yang digunakan untuk mengakses komputer yang letaknya jauh. Telnet dapat Anda gunakan apabila Anda mempunyai alamat IP (IP *Address*) dari komputer yang akan diakses dan Anda juga harus mempunyai hak akses.

2.3 *Hypertext Transfer Protocol* (HTTP)

Hypertext Transfer Protocol (HTTP) adalah suatu protokol yang digunakan untuk komunikasi atau mengirim informasi oleh *World Wide Web* (www). *Hypertext Transfer Protocol* mendefenisikan bagaimana suatu pesan dapat diformat dan dikirimkan dari *server* ke *client*. *Client* membuat suatu permintaan sedangkan *server* berguna untuk menyimpan dan membuat *resource* [4].

HTTP merupakan protokol yang bersifat *stateless*, sehingga *server* akan memproses setiap *request* dari *user* secara terpisah dari *request* yang lain, independen terhadap *request* yang sebelumnya. Mekanisme yang terjadi pada HTTP bila suatu client menginginkan layanan dari *server* di bagi dalam 4 langkah yaitu:

1. *Connection Setup*

   Berfungsi mengakses sebuah *server* dengan menggunakan Internet *address* dan *port number*.

2. *Request*

   Berfungsi sebagai mengirimkan *message* beserta informasi.

3. *Response*

   Server mengirimkan *response* kepada *client* sesudah *client* menyelesaikan *request*-nya. *Response message* meliputi informasi dari transaksi dan data yang diminta.

4. *Connection Release*

   Berfungsi sebagai mengakhiri koneksi dengan *server*.

2.3.1 Konsep HTTP Request dan HTTP Response

Untuk mengetahui bagaimana konsep dari HTTP yaitu dapat dilihat bagaimana sebuah *web browser* memproses sebuah *request* dan menerima sebuah *response* dari *web server*. Standar yang digunakan *web browser* pada saat ini adalah HTTP *(HyperText Transfer Protocol)*. HTTP digunakan oleh *web browser* untuk mengirim *request* dari *website* ke *web server* dan kemudian menerima *response* dari *web server*. HTTP *request* berkerja seperti email, yang akan memberitahukan kapan *request* dikirim. *Request* yang di kirim berisi *header* yang memberitahu apa yang harus dilakukan *server* dan bagaimana menangani *request* tersebut. Sebuah *request* akan diterima *web server* kemudian akan menentukan *response* apa yang akan diberikan. Terdapat banyak *request method* yang ada, tetapi yang paling sering digunakan adalah GET dan POST. HTTP *Request* dan HTTP *Respone* keduanya sangat mirip akan tetapi HTTP *Request* dioperasikan di belakang layar dan tanpa memerlukan *refresh* halaman.

2.4 *Application Programing Interface* (API)

Application Programming Interface (API) adalah metode spesifik yang ditentukan oleh sistem operasi komputer atau program aplikasi dimana seorang programmer menulis sebuah program aplikasi dapat membuat permintaan dari sistem operasi atau aplikasi lain. API juga merupakan kumpulan fungsi-fungsi eksternal yang disediakan *library windows* untuk mengatur kemampuan dan tingkah laku setiap element di sehingga dapat dimanfaatkan suatu program untuk meningkatkan kemampuan program tersebut. API menyediakan fungsi dan perintah dengan bahasa yang lebih terstruktur dan lebih mudah untuk dipahami oleh programmer bila dibandingkan dengan *System Calls*, hal ini penting untuk aspek editing dan pengembangan, sehingga programmer dapat mengembangkan sistem dengan mudah. API juga dapat digunakan pada Sistem Operasi mana saja asalkan sudah ada paket-paket API nya. Keuntungan memprogram dengan menggunakan API di antaranya :

1. Programmer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah ter-instal API tersebut. Sedangkan *system call* berbeda antar sistem operasi, dengan catatan dalam implementasinya mungkin saja berbeda.
2. Lebih mudah dimengerti. Karena API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti. Hal ini sangat penting dalam hal editing dan pengembangan.

2.5 *Web Service*

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan [5]. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan dalam bentuk informasi kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan oleh suatu sistem yang menyediakan *web service*.

Dalam implementasinya *web services* tidak mempunyai tampilan, karena *web services* memang termasuk dalam *business services*. Artinya didalam *web services* hanya tersedia fungsi-fungsi yang nantinya dapat digunakan oleh suatu aplikasi.

Dengan menggunakan format dokumen XML, *web services* memungkinkan suatu aplikasi berbicara dengan aplikasi lainnya. *Web Services* dapat diimplementasikan dalam berbagai *platform*, menggunakan bahasa pemrograman apapun dan bisa digunakan oleh berbagai *platform.*

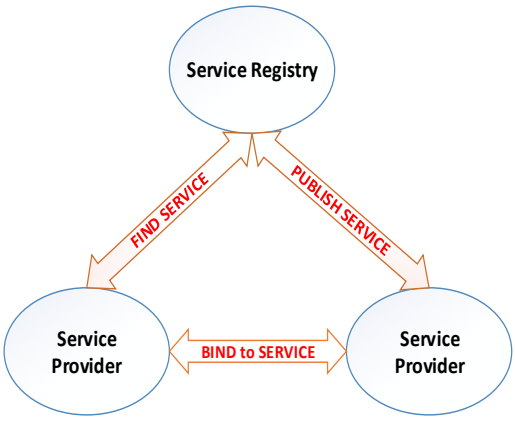
Web service ternyata sangat berbeda dengan *website*. Perbedaan yang paling terlihat adalah *website* dibuat untuk memiliki tampilan yang bagus sedangkan *web service* tidak memiliki tampilan. Karena itu *web service* tidak dibuat untuk berinteraksi langsung dengan *user*. Sesuai dengan kata *service* yang ada pada namanya, *web service* hanya menyediakan layanan, layanan itulah yang kemudian akan digunakan oleh aplikasi lainnya. Perbedaan dari keduanya dapat dilihat pada Tabel 2.3.

Tabel 2.3 Perbedaan Web Site Dan Web Service

WEB SITE	WEB SERVICE
1. Memiliki *web interface*	1. Tidak memiliki *interface* yang bagus.
2. Dibuat untuk ber interaksi langsung dengan user.	2. Dibuat untuk berinteraksi langsung dengan applikasi yang lain baik berbeda sistem operasi atau konsep sekalipun.
3. Dibuat untuk bekerja pada *web browser.*	3. Dibuat untuk bekerja pada semua tipe *client* applikasi atau perangkat *device.*

2.5.1 Arsitektur *Web Service*

Secara umum, arsitektur *web service* dapat dilihat pada Gambar 2.3



Gambar 2.3 Arsitektur *Web Service*

Berikut ini keterangan dari gambar Arsitektur *Web Service* dari gambar 2.3 :

1. *Service Requester* (peminta layanan)

   Berfungsi untuk menyediakan layanan dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.

2. *Service Provider* (penyedia layanan)

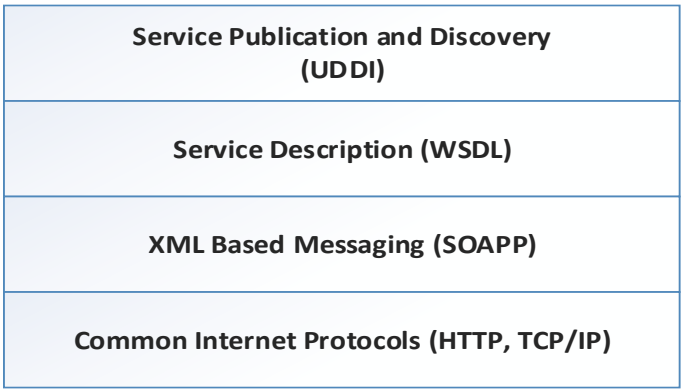
   Berfungsi sebagai lokasi sentral yang mendeskripsikan semua layanan yang telah di *register*.

3. *Service Registry* (daftar layanan)

   Berfungsi sebagai peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

2.5.2 Komponen-Komponen *Web Service*

Web service secara keseluruhan memiliki empat *layer* arsitektur komponen didalamnya, yaitu dapat dilihat pada Gambar 2.4



Gambar 2.4 Komponen *Web Service*

Berikut ini keterangan Komponen *Web Service* dari gambar 2.4 :

1. Layer 1

   Protokol Internet standar yang digunakan sebagai sarana transportasi adalah HTTP dan TCP/IP.

2. Layer 2

   Simple Object Access Protocol (SOAP) berbasiskan XML dan digunakan untuk pertukaran informasi antar sekelompok layanan.

3. Layer 3

   Web Service Definition Languange (WSDL) digunakan untuk mendiskripsikan atribut layanan

4. Layer 4

   Universal Description, Discovery and Integration (UDDI), merupakan direktori pusat untuk deskripsi layanan.

2.5.3 Jenis-Jenis *Web Service*

Saat ini ada dua jenis *web service*, yaitu yang berbasis REST dan berbasis SOAP. *Web Service* berbasis REST cenderung lebih ringan dan lebih mudah dipelajari dibandingkan dengan SOAP. Hal ini menyebabkan terjadinya peralihan dari *web service* berbasis SOAP ke REST.

2.5.3.1 REST (*Representational State Transfer)*

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*. 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*.

2.5.3.2 SOAP (*Simple Object Access Protocol)*

SOAP merupakan protokol untuk pertukaran informasi dengan desentralisasi dan terdistribusi. SOAP merupakan gabungan antara HTTP dengan XML karena SOAP umumnya menggunakan protocol HTTP sebagai sarana transport datanya dan data akan dipertukarkan ditulis dalam format XML. Karena SOAP mengunakan HTTP dan XML maka SOAP memungkinkan pihak-pihak yang mempunyai *platform*, sistem operasi dan perangkat lunak yang berbeda dapat

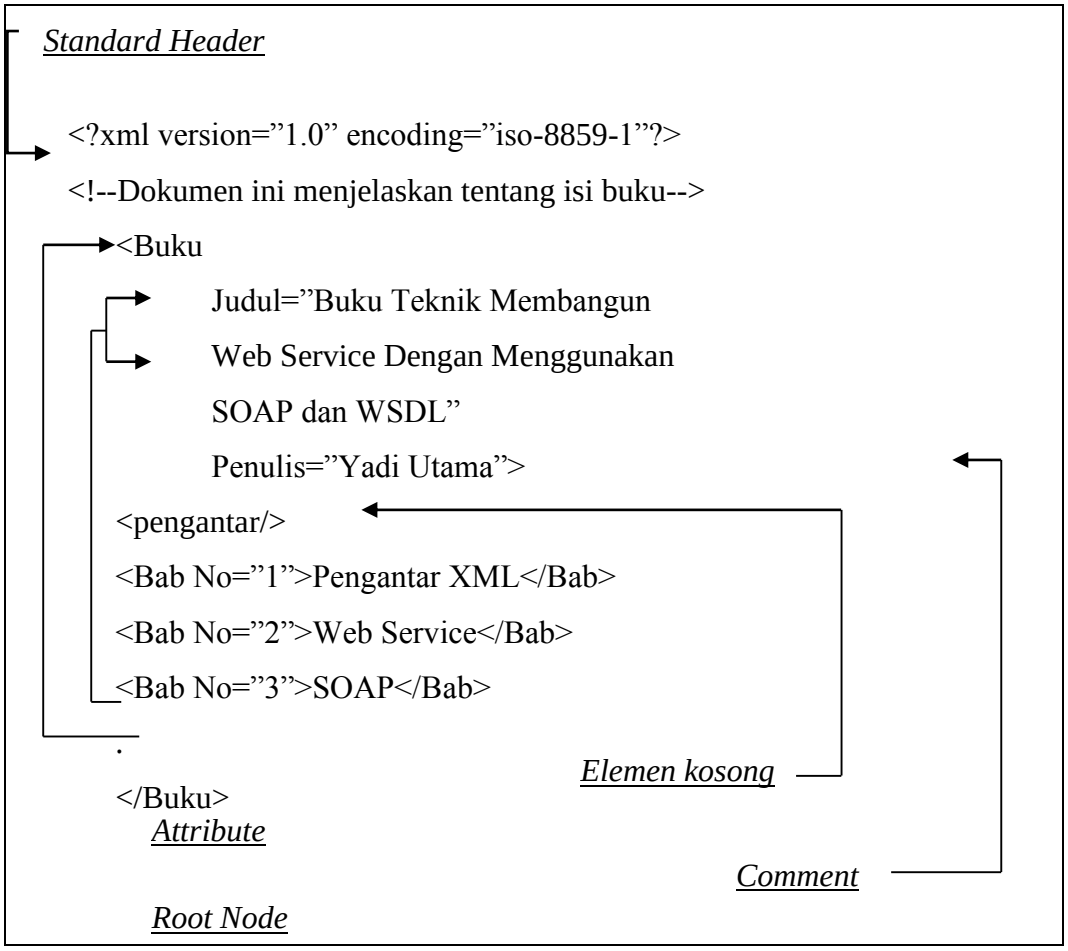
saling mempertukarkan datanya. SOAP mengatur bagaimana *request* dan *respon* dari suatu *web service* bekerja.

2.6 XML (Extensible Markup Language)

XML kependekan dari *Extensible Markup Language*, merupakan sebuah standar untuk *Markup language* yang dikembangkan mulai tahun 1996 dan baru mendapatkan pengakuan dari W3C pada bulan Februari 1998 [6]. *Markup language* itu sendiri merupakan suatu bahasa pemprograman untuk menandai suatu dokumen yang disebut dengan tag agar dokumen tersebut lebih mudah dibaca, dipahami serta menarik.

XML menggunakan sebagian besar karakter dalam set spesifikasi karakter *Unicode*. Sebenarnya merujuk ke set karakter ISO 10646, tetapi biasanya akan ditemukan dua yang digunakan secara bergantian, karena dua set karakter disimpan di sinkron. *Unicode*, set karakter 32-bit, menyediakan karakter standar dan universal yang ditetapkan dan memberikan nomor unik untuk setiap karakter. Hal ini dilakukan dengan cara menggunakan *Unicode*, data dianggap sama tanpa memperhatikan bahasa atau negara. Kedua format *Unicode* yang harus menerima *parser* adalah UTF-8 dan UTF-16, meskipun dapat menggunakan pengkodean karakter lainnya selama mereka sesuai dengan *Unicode*

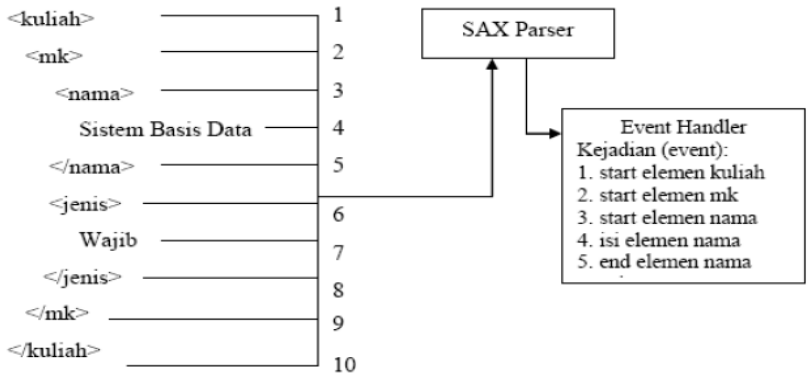
Seperti halnya HTML, XML juga menggunakan elemen yang ditandai dengan tag pembuka (‘<’) dan diakhiri dengan tag penutup (‘>’), atau diawali dengan (‘</ ‘) diakhiri (‘>’) dan atribut elemen dinyatakan dalam tag pembuka (<form name=”isidata”>). Hanya bedanya, HTML medefinisikan dari awal tag dan atribut yang dipakai didalamnya, sedangkan pada XML kita bisa menggunakan tag dan atribut sesuai kehendak kita. Berikut ini adalah contoh Script sebuah struktur dokumen XML, dapat dilihat pada Gambar 2.5



Gambar 2.5 Contoh Script Struktur XML

2.6.1.1 SAX *Parser*

SAX adalah singkatan dari Simple API for XML. API sendiri adalah singkatan dari *Application Program Interface*. SAX parser bekerja berdasarkan apa yang disebut *event based*. SAX Parser berjalan menjelajahi dokumen. Begitu SAX Parser menemukan deklarasi sebuah elemen, akan mencatatnya dan menyimpannya dalam suatu event handler. Event handler inilah yang menyediakan akses ke isi dokumen XML. Gambar 2.6 menjelaskan proses kerja SAX Parser.



Gambar 2.6 Cara Kerja SAX *Parser*

2.7 Rich Sites Summary (RSS)

RSS adalah sebuah file berformat XML yang telah digunakan kebanyakan di situs web berita dan weblog. Singkatan ini biasanya mengarah ke beberapa seperti, *Rich Site Summary (RSS 0.91), RDF Site Summary (RSS 0.9 and 1.0), Really Simple Syndication (RSS 2.0)*. Teknologi yang dibangun dengan RSS mengizinkan kita untuk mengakses kepada situs web yang menyediakan layanan *feed* RSS, biasanya situs web yang isinya selalu diganti secara reguler. Untuk memanfaatkan teknologi ini kita membutuhkan layanan pengumpul. Pengumpul bisa dibayangkan sebagai kotak surat pribadi. Kita kemudian dapat mendaftar ke situs yang ingin kita tahu perubahannya. Namun, berbeda dengan langganan koran atau majalah, untuk berlangganan RSS tidak diperlukan biaya atau gratis. Tapi, kita biasanya hanya mendapatkan satu baris atau sebuah pengantar dari isi situs berikut alamat terkait untuk membaca isi lengkap artikelnya.

2.7.1 Penggunaan *Rich Site Summary* (RSS)

Rich site summary digunakan secara luas oleh komunitas weblog atau *wibsite* untuk menyebar ringkasan tulisan terbaru di jurnal, kadang-kadang juga menyertakan artikel lengkap dan bahkan gambar dan suara. Sekitar 2000, penggunaan RSS meluas di berbagai penerbitan berita termasuk Reuters, CNN, dan BBC. RSS juga digunakan pada hampir semua situs berita atau weblog, dengan berbagai tujuan termasuk sperti pemasaran, *press release,* laporan reguler produk,

atau aktivitas lain yang membutuhkan pemberitahuan periodik dan tentunya publikasi.

Sebuah program komputer yang dikenal sebagai pembaca umpan atau *feed reader* bertindak sebagai pengumpul. Program ini mengecek situs yang menyediakan RSS dan menampilkan berbagai artikel baru yang ditemukan. Tenggang waktu dan siklus pengumpulan RSS biasanya dapat diatur oleh penggunanya. Program pengumpul dapat berupa program komputer atau sebuah layanan web yang tersedia online. Program pengumpul RSS di komputer biasanya berupa aplikasi sendiri yang harus dipasang di komputer sebelum dapat digunakan. Beberapa layanan pengumpul RSS juga menyediakan penggabungan dan juga pencarian. Sebuah web yang menyediakan umpan RSS biasanya ada *link* atau tautan dengan sebuah tombol bertuliskan XML atau RSS.

2.8 *Location Based Service* (LBS)

Location based service adalah kemampuan untuk mencari lokasi geografis dari *mobile divice* dan menyediakan layanan berdasarkan lokasi yang diperoleh. Konsep *location based service* yaitu menghasilkan layanan informasi yang dapat dicocokkan dengan lokasi keberadaan calon penerima informasi. *Location based service* memiliki 5 komponen yaitu :

1. Perangkat *Mobile*

   Perangkat *Mobile* adalah salah satu komponen penting dalam LBS. Perangkat ini berfungsi sebagai alat bantu bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Perangkat *mobile* yang dapat digunakan bisa berupa PDA (*Personal digital assistant)*, *smartphone*, leptop. Selain itu, perangkat *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.

2. Jaringan Komunikasi

   Komponen kedua adalah jaringan komunikasi, komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari perangkat *mobile* untuk kemudian dikirimkan ke penyedia

layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

3. Komponen *Positioning*

   Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi *mobile* atau juga menggunakan *Global Positioning System* (GPS).

4. Penyedia layanan dan aplikasi

   Penyedia layanan merupakan komponen *Location based service* (LBS) yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, dan mencari data.

5. Penyedia data dan Konten

   Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pihak ketiga yang memang memiliki otoritas untuk menyimpannya.

Berdasarkan informasi yang dikirim oleh pengguna layanan LBS bisa dibedakan menjadi dua yaitu :

1. *Pull Services*

   Pengguna secara aktif mengirimkan informasi yang dibutuhkan. Ini sama dengan ketika mengakses sebuah halaman *website* melalui *browser*. *User* memasukkan alamat *website* yang dituju kemudian *user* memperoleh informasi dari halaman *website* yang tampil di *browser*.

2. *Push Services*

Memberikan informasi kepada pengguna yang mana tidak secara langsung diminta oleh pengguna. Karena metode ini di aktifkan berdasarkan sebuah event, yang mana bisa dipicu oleh sesuatu seperti memasuki sebuah wilayah tertentu. Contohnya adalah ketika aplikasi LBS dapat mendeteksi bahwa kita telah memasuki sebuah kota dan secara otomatis palikasi LBS tersebut akan memberikan informasi aktual yang terkait yang terkait kota tersebut.

2.8.1 Global Positioning System (GPS)

Global Positioning System (GPS) adalah suatu sistem navigasi yang memanfaatkan satelit. Penerima *GPS* memperoleh sinyal dari beberapa satelit yang mengorbit bumi. Satelit yang mengitari bumi pada orbit pendek ini terdiri dari 24 susunan satelit, dengan 21 satelit aktif dan 3 buah satelit sebagai cadangan. Dengan susunan orbit tertentu, maka satelit *GPS* bisa diterima di seluruh permukaan bumi dengan penampakan antara 4 sampai 8 buah satelit. *GPS* dapat memberikan informasi posisi dan waktu dengan ketelitian sangat tinggi.

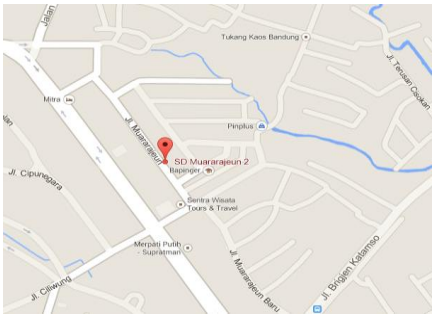
Nama lengkap GPS adalah *NAVSTAR* GPS (*Navigational satellite Timing and Ranging Global Positioning System*), namun lebih sering dikenal sebagai *GPS*. *GPS* mulai diaktifkan untuk umum pada 17 Juli 1995. Sedangkan, *Assisted-Global Positioning System* (*A-GPS*) merupakan penyempurnaan dari *GPS* sebagai satelit penentu posisi di belahan bumi. Satelit *GPS* yang dimiliki bumi mempunyai konstelasi 24 satelit dalam enam *orbit* yang mendekati lingkaran, setiap *orbit* ditempati oleh 4 buah satelit dengan *interval* antara yang tidak sama. *Orbit* satelit *GPS* berinklinasi 550° terhadap bidang *equator* dengan ketinggian rata-rata dari permukaan bumi sekitar 20.200 km.

Prinsip kerja dari *GPS* adalah Satelit *GPS* memberikan informasi kepada *receiver GPS* mengenai jarak atau posisi satelit. Sehingga kita tahu bahwa kita berada pada suatu radius tertentu dari satelit. Bila ada dua satelit maka kita tahu posisi kita, berada pada 2 lokasi, yaitu perpotongan dua radius tadi. *GPS receiver* mampu menghitung tempat yg paling mungkin. Semakin banyak sinyal satelit ditangkap semakin teliti satelit menghitung posisi.

2.9 *Google Maps*

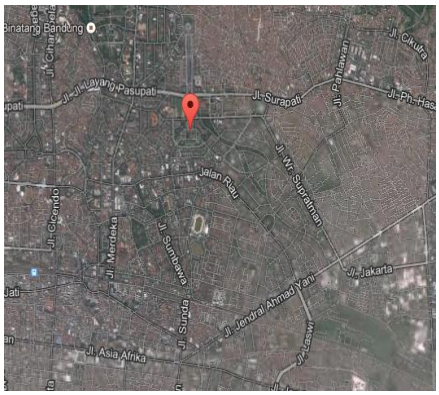
Google map merupakan salah satu layanan yang dikeluarkan oleh Google. *Google Maps* menyediakan layanan berbasis peta yang sangat responsif dan mudah dalam penggunaannya. Dengan menggunakan *google maps*, pengguna dapat dengan mudah mencari suatu lokasi serta dapat melakukan penelusuran rute menuju lokasi yang diinginkan. Ditingkat pemrograman, *Google maps* dapat dikembangkan dengan basis data, semua data yang terkait dengan titik lokasi disimpan dalam tabel dan dapat ditampilkan sesuai keinginan pengguna. Isi tabel yang berisi data posisi peta dapat ditampilkan Dengan menyajikan informasi lokasi yang menggunakan *google maps* [7]. Sementara ini *Google Maps* memiliki dua media untuk menampilkannya diantaranya adalah sebagai berikut :

1. *Maps* menampilkan peta dalam bentuk garis contohnya dapat dilihat pada Gambar 2.7



Gambar 2.7 Contoh tampilan *Maps* Google

2. Satelit yang dapat menampilkan peta dalam bentuk citra satelit seperti yang di tunjukan pada Gambar 2.8



Gambar 2.8 Contoh gambar satelit Google *Maps*

Sampai saat ini Google Maps masih berada dalam tahap beta, dan masih terus dikembangkan dengan data yang selalu diperbarui secara berkala. Google maps merupakan hak cipta Google secara *propriety*, sehingga dalam menggunakannya memerlukan adanya perjanjian, Google membuat mekanisme untuk dapat mengakses google maps dengan koding aplikasi dan dengan sebuah kunci yang dikenal dengan *API Key*.

2.9.1 *Waypoint*

Waypoint adalah titik referensi dalam ruang fisik yang digunakan untuk tujuan navigasi. *Waypoint* merupakan koordinat yang mengidentifikasi titik dalam ruang fisik. Koordinat yang digunakan dapat bervariasi tergantung pada aplikasi. Untuk navigasi darat koordinat berupa bujur dan lintang, sedangkan untuk navigasi udara juga mencakup ketinggian. *Waypoint* biasanya digunakan untuk sistem navigasi pada *Global Positioning System* (GPS) dan jenis-jenis tertentu dari radio navigasi. *Waypoint* yang terletak di permukaan bumi biasanya didefinisikan dalam dua dimensi (misalnya, bujur dan lintang), sedangkan yang digunakan dalam atmosfer bumi atau di luar angkasa didefinisikan dalam setidaknya tiga dimensi atau empat jika waktu merupakan salah satu koordinat untuk beberapa titik yang berada di luar Bumi. *Waypoint* ini digunakan untuk membantu menentukan jalur routing yang tak terlihat untuk navigasi [8]

2.9.2 *Google Maps API*

Google Maps API adalah layanan gratis yang diberikan oleh Google untuk mendukung perkembangan SIG (Sistem Informasi Geografis), sehingga para *develover web* maupun sistem yang lain menggunakan *Google Maps* sebagai penunjang fasilitas di dalam sistemnya. Agar dapat menggunakan Google Maps, dibutuhkan suatu media penghubung antara Google Maps dengan sistem yang dibagun. Maka dikembangkan suatu *interface* guna memfasilitasi hal tersebut. *Google Maps API* adalah antarmuka pemograman aplikasi yang menyediakan berbagai fungsi dan sekumpulan objek dalam bahasa *JavaScript* sehingga citra digital atau peta *Google Maps* dapat ditampilkan pada halaman *website* lain.

Google Maps API dapat menerima *input* kordinat dalam format derajat lintang dan derajat bujur, tanpa mengenal menit dan detik dengan positif untuk lintang utara dan bujur timur, sedangkan untuk lintang selatan dan bujur barat digunakan tanda negatif. Di dalam kordinat Goole Maps, garis lintang dan bujur dikenal dengan nama *latitude* dan *longitude, latitude* merupakan garis yang membentang dari barat selatan ke utara, sedangkan *longitude* merupakan garis yang membentang dari barat ke timur.

2.10 *Euclidean Distance*

Euclidean Distance adalah sebuah metode yang digunakan untuk mengukur jarak (*distance*). *Euclidean Distance* sebenarnya merupakan generalisasi dari *teorema phytagoras* secara formal perhitungan *Euclidian Distance* adalah sebagai berikut :

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad \text{.....(2.1)}$$

Dimana :

d : Nilai Jarak

n : Banyaknya Nilai

Jika terdapat dua buah titik pada sebuah bidang dua dimensi maka untuk mengukur jarak dari kedua titik tersebut dapat digunakan persamaan *phytagoras*.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad \text{.....(2.2)}$$

Dimana :

$x_1 , x_2 =$ Nilai titik x

$y_1 , y_2 =$ Nilai titik y

2.11 Java

Java adalah bahasa pemrograman yang multi *platform* dan multi *device*. Sekali menuliskan sebuah program dengan menggunakan Java, akan dapat menjalankannya hampir di semua komputer dan perangkat lain yang *support* Java, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis Java ini dikompilasikan ke dalam *p-code* dan bisa dijalankan dengan *Java Virtual Machine*. Fungsionalitas dari Java ini dapat berjalan

dengan *platform* sistem operasi yang berbeda karena sifatnya yang umum dan *non-spesifik* [9].

Alasan utama pembentukan bahasa java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan diberbagai macam perangkat elektronik, seperti *microwave oven* dan *remote control*, sehingga Java harus bersifat *portable* atau yang sering disebut dengan *platform* independent atau tidak tergantung pada *platform*. Itulah yang menyebabkan dalam dunia pemrograman Java, dikenal adanya istilah "*write one, run every where*", yang berarti kode program hanya ditulis sekali, namun dapat dijalankan dibawah *platform* manapun, tanpa harus melakukan perubahan kode program. Berikut ini adalah kelebihan dan kekurang Java :

1. Kelebihan Java.

   Kelebihan Java yang pertama tentu saja *multi platform*. Java dapat dijalankan dalam beberapa *platform* komputer dan sistem operasi yang berbeda. Hal ini sesuai dengan slogannya yang udah dibahas sebelumnya. Yang kedua adalah OOP *(Object Oriented Programming)*. Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam Java. Hal ini akan memudahkan pemrograman menjadi lebih mudah.

2. Kekurangan Java

   Kekurangan Java yang adalah penggunaan memori yang cukup banyak, lebih besar daripada bahasa tingkat tinggi sebelum generasi Java. Namun hal ini memang sesuai dengan fitur beragam yang dimiliki oleh Java. Masalah memori ini juga tidak dialami oleh semua pengguna aplikasi Java. yang sudah menggunakan perangkat keras dengan teknologi terbaru tidak merasakan kelambatan dan konsumsi memori Java yang tinggi. Lain halnya dengan yang menggunakan teknologi lama atau komputer yang sudah berumur tua lebih. Namun apapun kelemahan yang dimiliki Java, faktanya adalah Java merupakan bahasa pemrograman yang populer dan digunakan di seluruh dunia saat ini.

2.11.1 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform.* Berikut ini adalah sifat dari Eclipse :

1. *Multi platform* : Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Mulit language* : Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi role* : 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.
4. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source,* yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

2.11.2 *Android Development Tools* (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.

2.12 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc*. membeli *Android Inc*., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance, konsorsium* dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobil*e, dan *Nvidia.* [10].

Berikut ini beberapa dari versi Android yang telah di luncurkan diantaranya:

1. Andorid versi 1.1

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

2. Android versi 1.5 (Cupcake)

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

3. Android versi 1.6 (Donut)

   Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus kamera,

camcorder dan galeri yang dintegrasikan CDMA / EVDO, 802.1x, VPN, Gestures, dan *Text-to-speech engine*; kemampuan dial 10 kontak teknologi *text to change speech*

4. Android versi 2.0 - 2,1 (Eclair)

   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 UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

5. Android versi 2.2 (Froyo: Frozen Youghurt)

   Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain 11 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 menerjemah pada *browser*, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan *auto update* dalam aplikasi *Android Market*.

6. Android versi 2.3 (Gingerbread)

   Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan, peningkatan fungsi copy paste, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb, equalization, headphone virtualization,* dan *bass boost*), dukungan kemampuan *Near Field Communication (NFC),* dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0 (Honeycomb)

   Android Honeycomb dirancang khusus untuk 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.

8. Android 4.0 ICS (Ice Cream Sandwidch)

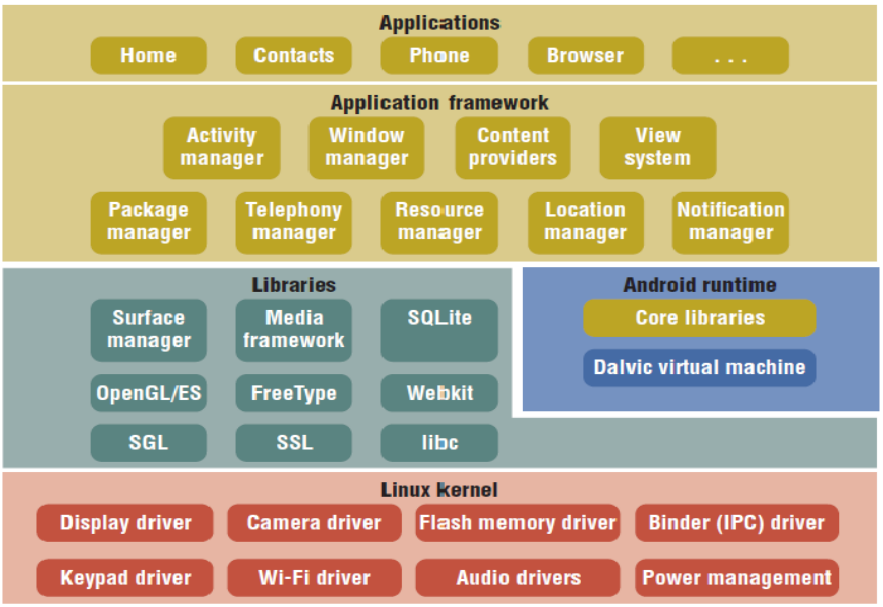
   Pertama kali diliris 19 Oktober 2011. *Smartphone* yang pertama kali menggunakan OS Android ini adalah Samsung Galaxy Nexus. Secara teori semua perangkat selular yang menggunakan versi Android sebelumnya,Ginggerbread dapat di update ke Android versi ICS.

9. Android 4.1 (Jelly Bean)

   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. Sistem operasi *Android Jelly Bean* 4.1 muncul pertama kali dalam produk *tablet Asus*, yakni *Google Nexus 7.*

2.12.1 Arsitektur Android

Google mengibaratkan Android sebagai sebuah tumpukan *software*. Setiap lapisan dari tumpukan tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut ini susunan dari laipsan-lapisan tersebut jika dilihat dari lapisan dasar hingga lapisan teratas. Lihat Gambar 2.9



Gambar 2.9 Arsitektur Android

2.12.1.1 Linux Kernel

Android bukan Linux, akan tetapi Android dibangun diatas Linux Kernel yaitu versi 2.6 sehingga kehandalannya dapat di percaya. Untuk inti sistem *service* linux yang digunakan seperti keamanan, manajemen memori, proses manajemen, *network*, dan *drive* model. Seperti yang terlihat digambar, Linux Kernel menyediakan *Driver* Layer, Kamera, *Keypad*, Kamera, Wifi, *Flash Memory*, Audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan keamanan. Kernel juga bertindak sebagai lapisan abstrak antara hardware dan software stacknya 20.

2.12.1.2 Library

Android terdiri satu *set core library* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core library* bahasa pemograman Java. Setiap aplikasi menjalankan prosesnya sendiri dalam Android. Setiap alikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari mesin virtual dalvik . Dalvik dirancang agar perangkat dapat menjalankan multiple VMs secara efisien. Mesin Virtual Dalvik mengeksekusi file dalam Dalvik executable, yaitu sebuah format yang dioptimalkan untuk memori yang kecil. Dalvik VM berbasis, berjalan dan dikompilasi oleh compiler bahasa Java yang telah

ditransformasikan ke dalam format yang telah di sertakan, Dalvik VM bergantung pada kernel Linux untuk berfungsi seperti threading dan manajemen memori tingkat rendahnya.

2.12.1.3 Application Framework

Pengembang memiliki akses penuh menuju API *framework* yang sama, dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) denga mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini dan aplikasi yang lain mungkin akan memanfaatkan kemampuan ini (sesuai denganbatasan keamanan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna. Semua pengguna merupakan rangkaian *set* layanan dan sistem, termasuk:

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.*

2.12.1.4 Application

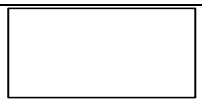
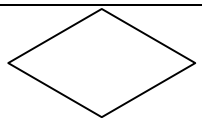
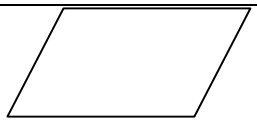
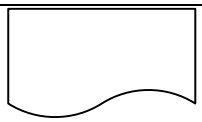
Android telah menyertakan aplikasi inti seperti email elient, SMS, kalender, peta, browser, kontak, dan lain-nya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman Java. Pada layer inilah developer atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada Android semua aplikasi baik aplikasi inti maupun aplikasi pihak ketiga berjalan pada layer aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses

seluruh kemampuan handset, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan *platform* Android ini, Programmer atau developer secara penuh akan bisa mengkostumisasi perangkat Androidnya.

2.13 *Flow Chart*

Flow Chart adalah diagram yang menggambarkan aliran proses dari suatu sistem. *Flow cahart* juga dapat menggambarkan suatu algoritma pemograman. Berikut ini simbol-simbol utama yang digunakan dalam *flow chart* khususnya dalam penggambaran algoritma pemograman

Tabel 2.4 Simbol *Flow Chart*

Simbol	Nama
	Terminal
	Proses
	Pertanyaan / Keputusan
	Input / Output (Pada Layar)
	Output (Dalam bentuk file)

2.14 *Unified Modeling Language* (UML)

Unified Modelling Language (*UML*) adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem [11]. Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa

pemrograman apapun. Tetapi karena *UML* juga menggunakan *class* dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

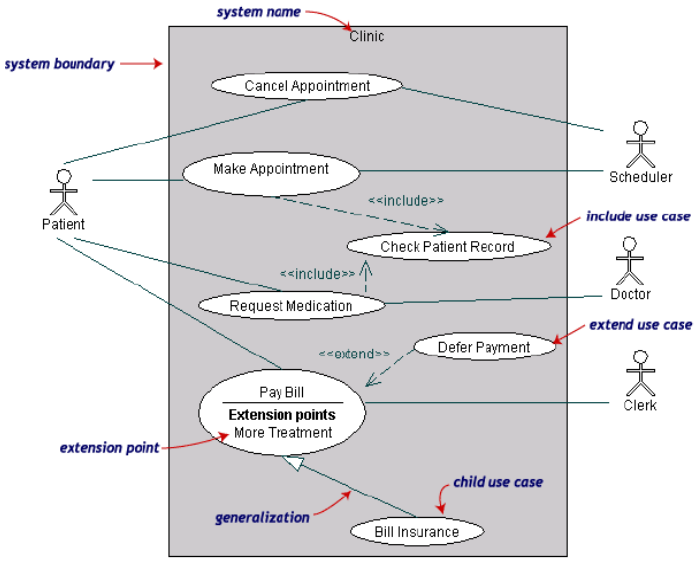
Seperti bahasa-bahasa lainnya, *UML* mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi *UML* terutama diturunkan dari tiga notasi yang telah ada sebelumnya: *Grady Booch* OOD (*Object-Oriented Design*), *Jim Rumbaugh* OMT (*Object Modeling Technique*), dan *Ivar Jacobson* OOSE (*Object-Oriented Software Engineering*). UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. *Use Case Diagram* untuk memodelkan proses bisnis.
2. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar objek.
3. *Collaboration Diagram* untuk memodelkan interaksi antar objek.
4. *State Diagram* untuk memodelkan perilaku objek di dalam sistem.
5. *Activity Diagram* untuk memodelkan perilaku userdan objek di dalam sistem.
6. *Class Diagram* untuk memodelkan struktur kelas.
7. *Component Diagram* untuk memodelkan komponen objek.
8. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

2.14.1 *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem [12]. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal. Contoh *use case diagram* dapat lihat dalam Gambar 2.10



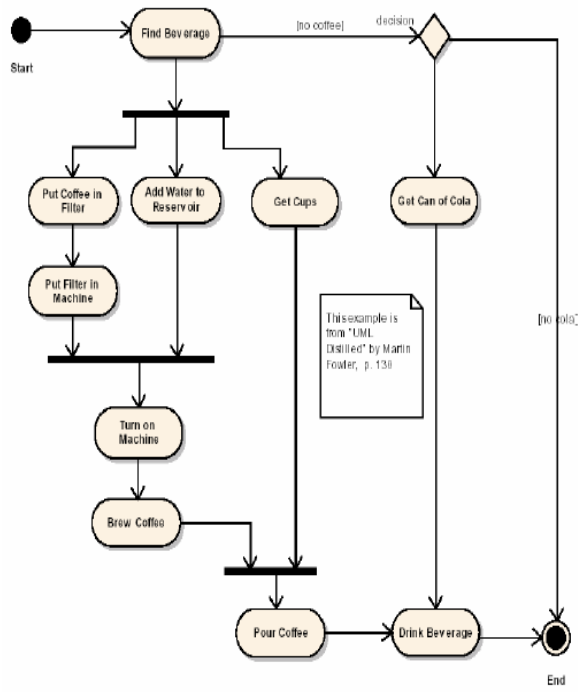
Gambar 2.10 *Use Case Diagram*

2.14.2 *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir [12]. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Contoh *Activity diagram* tanpa *swimlane* dapat di lihat pada Gambar 2.11



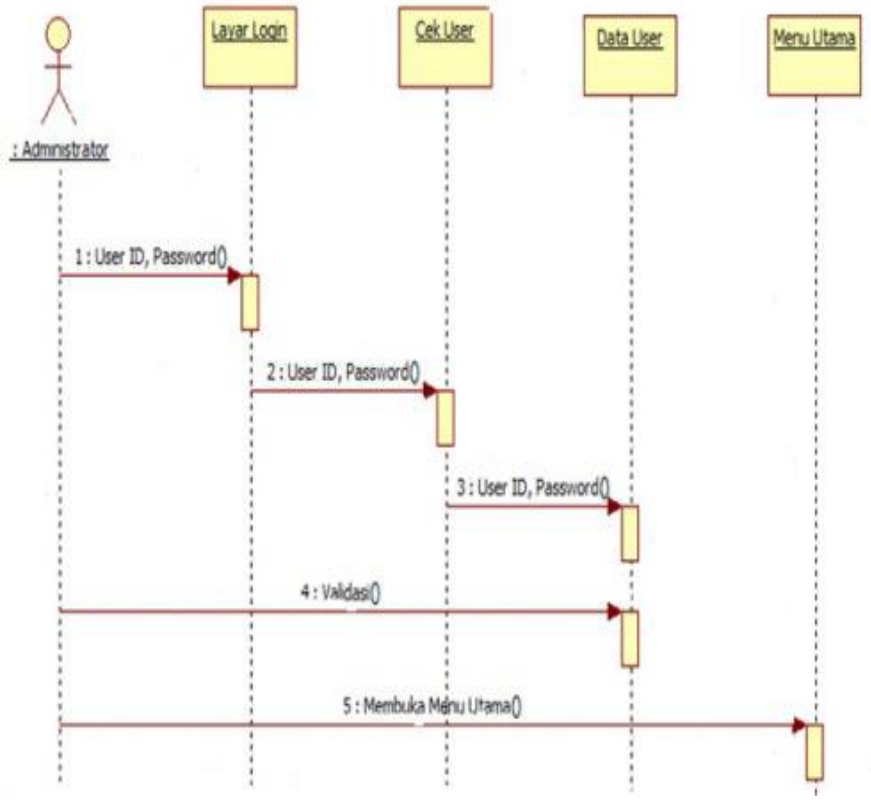
Gambar 2.11 *Activity Diagram*

2.14.3 *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 [12]. *Sequence diagram* terdiri atar 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.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Contoh *Sequence Diagram* dapat dilihat pada Gambar 2.12



Gambar 2.12 ***Sequence Diagram***

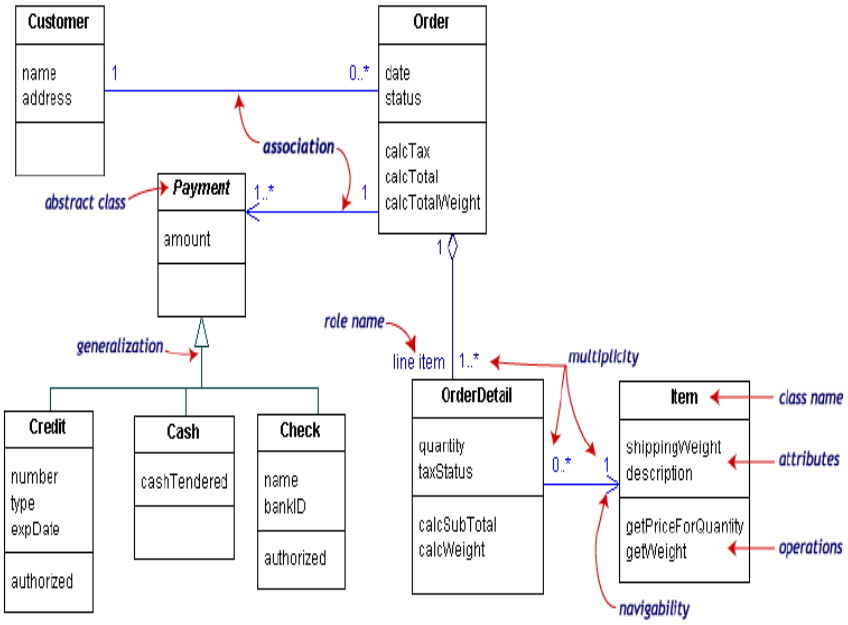
2.14.4 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek [12]. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* sendiri memiliki tiga area pokok yaitu nama, atribut, metoda. Antribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya

3. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Contoh *Class Diagram* dapat di lihat dalam Gambar 2.13



Gambar 2.13 *Class Diagram*