

BAB II

TINJAUAN PUSTAKA

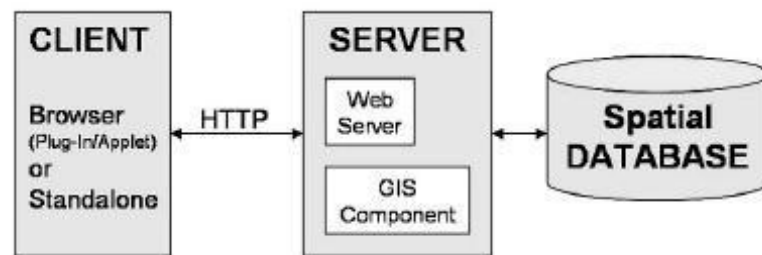
II.1. Sistem Informasi Geografis

Sistem informasi geografis (*Geographic Information System* atau GIS) adalah sistem berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi geografis. Hal ini memungkinkan data dapat diakses penunjukan ke suatu lokasi dalam peta yang tersaji secara digital. Sistem informasi geografis digunakan untuk menangani data spasial atau data tentang keruangan, sistem seperti ini banyak digunakan antara lain untuk pemetaan tanah dan agrikultur, arkeologi dan jaringan listrik. Sistem ini sudah lama diterapkan, sistem informasi geografis berskala nasional yang pertama dioperasikan di Kanada dengan nama CGIS (*Canada Geographic Information System*).

Kebanyakan GIS menggunakan konsep “lapis” (*layer*). Setiap lapisan mewakili satu fitur geografi dalam area yang sama dan selanjutnya semua lapisan bisa saling ditumpuk untuk mendapatkan informasi yang lengkap. Setiap lapisan dapat dibayangkan seperti plastik transparan yang mengandung hanya gambar tertentu. Pemakai bisa memilih transparan-transparan yang dikehendaki dan kemudian saling ditumpangkan sehingga akan diperoleh gambar yang merupakan gabungan dari sejumlah plastik transparan (Abdul Kadir ; 2014 : 121-122).

II.1.1. Arsitektur Sistem Informasi Geografis

Untuk dapat melakukan komunikasi dengan komponen yang berbeda-beda di lingkungan web maka dibutuhkan sebuah *web server*. Karena standart dari geo data berbeda beda dan sangat spesifik maka pengembangan arsitektur system mengikuti arsitektur „*Client Server*“.



Gambar II.1. Arsitektur GIS

(Sumber : Denny Charter ; 2011 : 2)

Gambar diatas menunjukkan arsitektur minimum sebuah system Web GIS. Aplikasi berada disisi client yang berkomunikasi dengan Server sebagai penyedia data melalui web Protokol seperti HTTP (*Hyper Text Transfer Protocol*). Aplikasi seperti ini bisa dikembangkan dengan *web browser* (Mozilla Firefox, Opera, Internet Explorer, dll). Untuk menampilkan dan berinteraksi dengan data GIS, sebuah browser membutuhkan *Pug-In* atau *Java Applet* atau bahkan keduanya. Web Server bertanggung jawab terhadap proses permintaan dari client dan mengirimkan tanggapan terhadap respon tersebut. Dalam arsitektur web, sebuah web server juga mengatur komunikasi dengan server side GIS Komponen. Server side GIS Komponen bertanggung jawab terhadap koneksi kepada database spasial seperti menterjemahkan *query* kedalam SQL dan

membuat representasi yang diteruskan ke server. Dalam kenyataannya Side Server GIS Komponen berupa software libraries yang menawarkan layanan khusus untuk analisis spasial pada data. Selain komponen hal lain yang juga sangat penting adalah aspek fungsional yang terletak di sisi client atau di server (Denny Charter ; 2011 : 1 – 2).

II.1.2. Data Spasial

Sebagian besar data yang akan ditangani dalam SIG merupakan data spasial yaitu sebuah data yang berorientasi geografis, memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang membuatnya berbeda dari data lain, yaitu informasi lokasi (spasial) dan informasi deskriptif (attribute) yang dijelaskan berikut ini :

1. Informasi lokasi (spasial), berkaitan dengan suatu koordinat baik koordinat geografi (lintang dan bujur) dan koordinat XYZ, termasuk diantaranya informasi datum dan proyeksi.
2. Informasi deskriptif (atribut) atau informasi non spasial, suatu lokasi yang memiliki beberapa keterangan yang berkaitan dengannya, contohnya: jenis vegetasi, populasi, luasan, kode pos, dan sebagainya (G. Manjela Eko Hartoyo ; 2010 : 2).

II.1.3. Format Data Spasial

Secara sederhana format dalam bahasa komputer berarti bentuk dan kode penyimpanan data yang berbeda antara file satu dengan lainnya. Dalam SIG, data spasial dapat direpresentasikan dalam dua format, yaitu :

1. Data Raster

Data raster (atau disebut juga dengan sel grid) adalah data yang dihasilkan dari sistem Penginderaan Jauh. Pada data raster, obyek geografis direpresentasikan sebagai struktur sel grid yang disebut dengan pixel (picture element).

2. Data Vektor

Data vektor merupakan bentuk bumi yang direpresentasikan ke dalam kumpulan garis, area (daerah yang dibatasi oleh garis yang berawal dan berakhir pada titik yang sama), titik dan nodes (merupakan titik perpotongan antara dua buah garis). Keuntungan utama dari format data vektor adalah ketepatan dalam merepresentasikan fitur titik, batasan dan garis lurus (G. Manjela Eko Hartoyo ; 2010 : 2)

II.1.4. Sumber Data Spasial

Salah satu syarat SIG adalah data spasial, yang dapat diperoleh dari beberapa sumber antara lain :

1. Peta Analog

Peta analog (antara lain peta topografi, peta tanah dan sebagainya) yaitu peta dalam bentuk cetak. Pada umumnya peta analog dibuat dengan teknik kartografi, kemungkinan besar memiliki referensi spasial seperti koordinat,

skala, arah mata angin dan sebagainya. Dalam tahapan SIG sebagai keperluan sumber data, peta analog dikonversi menjadi peta digital dengan cara format raster diubah menjadi format vektor melalui proses digitasi sehingga dapat menunjukkan koordinat sebenarnya di permukaan bumi.

2. Data Sistem Penginderaan Jauh Data Penginderaan Jauh (antara lain citra satelit, foto udara dan sebagainya), merupakan sumber data yang terpenting bagi SIG karena ketersediaanya secara berkala dan mencakup area tertentu.
3. Data Hasil Pengukuran Lapangan Data pengukuran lapangan yang dihasilkan berdasarkan teknik perhitungan tersendiri, pada umumnya data ini merupakan sumber data atribut contohnya : batas administrasi, batas kepemilikan lahan, batas persil, batas hak perusahaan hutan dan lain-lain.
4. Data GPS (*Global Positioning System*) Teknologi GPS memberikan terobosan penting dalam menyediakan data bagi SIG. Keakuratan pengukuran GPS semakin tinggi dengan berkembangnya teknologi satelit navigasi. Pengolahan data yang bersumber dari GPS biasanya dilakukan dalam format vector (G. Manjela Eko Hartoyo ; 2010 : 4).

II.1.5. Komponen Sistem Informasi Geografis

Menurut Bramantiyo Marjuki (2014 : 2) Sebagai salah satu jenis sistem informasi, SIG mempunyai sub sistem atau komponen yang bekerja secara bersama untuk menghasilkan fungsionalitas SIG. Komponen SIG terdiri dari *hardware*, *software*, data dan metode.

1. *Hardware* atau perangkat keras merupakan media tempat pelaksanaan proses SIG. *Hardware* yang diperlukan dalam sebuah SIG meliputi perangkat keras untuk masukan data, penyimpanan data, pengolahan dan analisa data dan pembuatan keluaran.
2. *Software* atau perangkat lunak merupakan alat pelaksana pekerjaan SIG. *Software* standar SIG harus mempunyai kapabilitas data input, penyimpanan, manajemen data, transformasi dan konversi data, analisa dan pembuatan keluaran.
3. Data atau representasi dari sebuah obyek/fenomena adalah bahan yang dianalisa di dalam SIG. SIG memerlukan sebuah jenis data yang spesifik agar dapat memberikan keluaran seperti fungsionalitasnya. Data yang digunakan dalam SIG adalah data geospasial atau data yang berefrensi geografis (mempunyai informasi lokasi).
4. Metode adalah cara bagaimana data diolah untuk menjadi sebuah informasi. Metode meliputi aspek pemasukan data ke dalam sistem, bagaimana data dikelola dan disimpan, bagaimana data dianalisis, dan bagaimana informasi ditampilkan.

II.1.6. Sistem Informasi Geografis Berbasis Web

Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (*Hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser. Beberapa jenis browser

yang populer saat ini di antaranya : internet Explorer yang diproduksi oleh Microsoft, Mozilla Firefox, Opera dan Safari yang diproduksi oleh Apple. Browser (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen web dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat di dalam aplikasi browser yang biasa disebut web engine. Semua dokumen web ditampilkan oleh browser dengan cara diterjemahkan (M. Rudyanto Arief ; 2011 : 7-8).

II.2. Metode Euclidean Distance

Euclidean distance adalah metrika yang sering digunakan untuk menghitung kesamaan dua vektor. Euclidean distance menghitung akar dari kuadrat perbedaan dua vektor. Semakin besar jarak antara dua vektor, maka tingkat kesamaan atau kemiripannya kecil. Sebaliknya, semakin kecil jarak antara dua vektor, maka tingkat kesamaan atau kemiripannya besar (Sylvia Pretty Tulus ; 2014 : 104).

Rumus Euclidean distance adalah sebagai berikut :

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \dots\dots\dots (1)$$

Dengan :

d(i,j) : jarak dokumen ke-i ke dokumen ke-j

$x_i(n)$: kata ke n di dokumen ke-i.

$x_j(n)$: kata ke n di dokumen ke-j. (Sylvia Pretty Tulus ; 2014 : 104)

II.2.1. Langkah – Langkah Perhitungan Metode Euclidean Distance

Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam Euclidean space. Euclidean space diperkenalkan oleh seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini biasanya diterapkan pada 2 dimensi dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi.

- **Pada 1 dimensi**

Semisal ingin menghitung jarak Euclidean 1 dimensi. Titip pertama adalah 4, titik kedua adalah -10. Caranya adalah kurangkan -10 dengan 4. sehingga menghasilkan -14. Cari nilai absolut dari nilai -14 dengan cara memangkatkannya sehingga mendapat nilai 196. Kemudian diakarkan sehingga mendapatkan nilai 14. Sehingga jarak euclidean dari 2 titik tersebut adalah 14.

- **Pada 2 dimensi**

Caranya hampir sama. Misalkan titik pertama mempunyai kordinat (3,5). Titik kedua ada di kordinat (5,-3). Caranya adalah kurangkan setiap kordinat titik kedua dengan titik yang pertama. Yaitu, (5-3,-3-5) sehingga menjadi (2,-8). Kemudian pangkatnya sehingga memperoleh (4,64). Kemudian tambahkan semuanya sehingga memperoleh nilai $4+4 = 8$. Hasil ini kemudian diakarkan menjadi 2.828. Sehingga jarak euclidean menjadi 2.828. Untuk lebih dari 2 dimensi, caranya sama saja dengan 2 dimensi.

II.3. Quantum GIS

Quantum GIS merupakan salah satu perangkat lunak *open source* di bawah proyek resmi dari *Open Source Geospatial Foundation* (OSGeo) yang dapat dijalankan dalam sistem operasi Windows, Mac OSX, Linux dan Unix. Aplikasi ini menawarkan pengolahan data spasial dengan berbagai format dan fungsionalitas vektor, raster dan *database*. Untuk keperluan analisis spasial, aplikasi ini telah cukup lengkap karena telah terintegrasi dengan perangkat lunak GRASS. Pemanfaatan perangkat lunak Quantum GIS ini dapat digunakan sebagai pilihan alternatif dari *software* SIG komersial seperti ArcView maupun ArcGIS. (Saddam Husein ; 2012 : 93).

II.4. Pengertian Dreamweaver

Macromedia drewamver adalah sebuah perangkat lunak aplikasi untuk mendesain dan membuat halaman web. Dengan menggunakan Dreamweaver 8, ketika membuat sebuah halaman web, Anda tidak perlu lagi mengetik kodekode HTML atau kode-kode lainnya secara manual. Anda cukup melakukan klik beberapa kali, maka simsalabin, halaman web yang Anda inginkan sudah jadi. Selain HTML, Dreamweaver 8 juga mendukung *CSS*, *JavaScript*, *PHP*, *ASP*, dan bahasa pemrograman lainnya untuk membuat web. Hal ini akan sangat menguntungkan Anda. Sebagai contoh, jika dahulu Anda harus mengetikkan kode-kode CSS untuk membuat *Style* tertentu, maka dengan Dreamweaver 8, Anda cukup melakukan klik beberapa kali saja. Dreamweaver 8 adalah versi terbaru dari keluarga Dreamweaver. Versi pertamanya sendiri diluncurkan sekitar

tahun 1994 oleh Macromedia Inc. Dalam versi terbaru ini, banyak sekali fasilitas baru yang ditambahkan. Contohnya, Anda akan dapat membuat dan menggunakan Style dalam CSS dengan mudah dan fleksibel. Panel untuk pengolahan CSS juga sudah diperbarui dan lebih mudah digunakan. Dreamweaver 8 juga menyediakan beberapa template halaman web baru, termasuk *fasilitas Starter Pages*. Akhirnya, selamat memulai mempelajari Dreamweaver 8 (Arief Ramadhan ; 2011 : 2).

II.5. Pengertian PHP

PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membentuk halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web (M. Rudyanto Arief ; 2011 : 43).

II.5.1. Pengertian Apache

Apache adalah paket aplikasi yang digunakan untuk web server yang handal dan stabil. Jika dibandingkan dengan web server lainnya, Apache masih menjadi andalan para webmaster. Perkembangan server ini sangat pesat sehingga hampir semua server web menggunakan Apache. Aplikasi apache dikenal dengan nama

httpd. Jika menggunakan linux redhat/Fedora maka paket ini sudah ada dalam bentuk .RPM (Albertus Dwiyoga ; 2012 : 11).

II.5.2. Pengertian CSS

(CSS) *Cascading Style Sheets* adalah suatu bahasa stylsheet yang digunakan untuk mengatur style suatu dokumen. Pada umumnya CSS dipakai untuk memformat tampilan hakaman web yang dibuat dengan bahasa HTML dan XHTML. CSS memungkinkan web developer untuk memisahkan HTML dari aturan-aturan untuk membentuk tampilan sebuah website. CSS diperkenalkan untuk pengembangan website pada tahun 1996. Nama CSS didapat dari fakta bahwa setiap deklarasi style yang berbeda dapat diletakkan secara berurutan, yang kemudian akan membentuk hubungan parent-child pada setiap style (Sulistiyawan ; 2012 : 32).

II.6. Pengertian Database

Database adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basisdata menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basisdata, dimodifikasi dan dihapus (Janner Simarmata ; 2010 : 1).

II.7. Pengertian MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya. Kepopuleran MySQL mengakses database-nya sehingga mudah untuk digunakan, kinerja query cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan skala menengah-kecil. MySQL juga bersifat *open source* dan *free* (anda tidak perlu membayar untuk menggunakannya) pada berbagai platform (kecuali pada windows, yang bersifat *shareware*).

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pengembangan aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP (M. Rudyanto Arief ; 2011 : 151).

II.8. Normalisasi

Menurut Janner Simarmata (2010 : 76) Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

II.8.1. Bentuk-bentuk Normalisasi

Menurut Janner Simarmata (2010 : 79) Bentuk-bentuk Normalisasi memiliki 5 bentuk yaitu :

1. Bentuk normal tahap pertama (1st Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

ISBN	Thn_Terbit	ID_Pengarang	Nama_Pengarang	ID_Pengarang	Nama_Pengarang
12-1202-19222	1992	K0121	Aris M	K1021	Kosim P
11-1090-29101	2001	K1021	Kosim P		
11-1090-29102	2001	K2091	K Odelia	K0121	Aris M
12-1201-90871	2002	K2092	Renaldi	K2091	K Odelia
13-2089-12910	2001	K2019	Samsuri J		

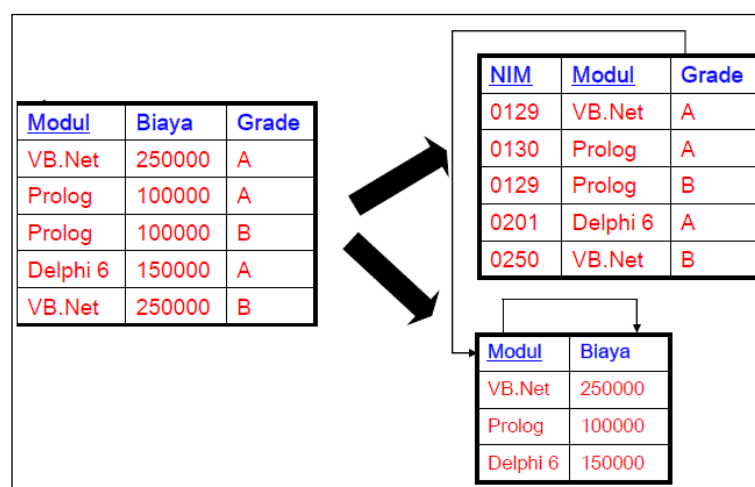
ISBN	Thn_Terbit	ID_Pengarang	Nama_Pengarang
12-1202-19222	1992	K0121	Aris M
12-1202-19222	1992	K1021	Kosim P
11-1090-29101	2001	K1021	Kosim P
11-1090-29102	2001	K2091	K Odelia
11-1090-29102	2001	K0121	Aris M
12-1201-90871	2002	K2092	Renaldi
12-1201-90871	2002	K2091	K Odelia
13-2089-12910	2001	K2019	Samsuri J

Gambar II.2. Normalisasi 1NF

(Sumber : Janner Simarmata ; 2010 : 76)

2. Bentuk normal tahap kedua (2nd normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.



Gambar II.3. Normalisasi 2NF

(Sumber : Janner Simarmata ; 2010 : 76)

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

<u>S</u>	<u>P</u>	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	399
S4	P5	400


Gambar II.4. Normalisasi 3NF

(Sumber : Janner Simarmata ; 2010 : 76)

4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

<u>Modul</u>	<u>Dosen</u>
VB.Net	Ajib
Prolog	Aris
Prolog	Aris
VB Net	Budi
Prolog	Jono
VB.Net	Budi



<u>NIM</u>	<u>Dosen</u>
0129	Ajib
0130	Aris
0129	Aris
0201	Budi
0250	Jono
0260	Budi

<u>Dosen</u>	<u>Modul</u>
Ajib	VB.Net
Aris	Prolog
Jono	Prolog
Budi	VB.Net

Gambar II.5. Normalisasi BCNF

(Sumber : Janner Simarmata ; 2010 : 76)

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*).

FirstName	LastName	Major	Level
Jack	Jones	LIS	Graduate
Lynn	Lee	LIS	Graduate
Mary	Ruiz	Pre-Medicine	Undergraduate
Lynn	Smith	Pre-Law	Undergraduate
Jane	Jones	LIS	Graduate

Gambar II.6. Normalisasi 4NF

(Sumber : Janner Simarmata ; 2010 : 76)

II.9. Unified Modeling Language (UML)

Menurut Sri Dharwiyanti (2013) *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.


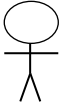


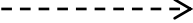
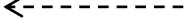
1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. 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 system 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.

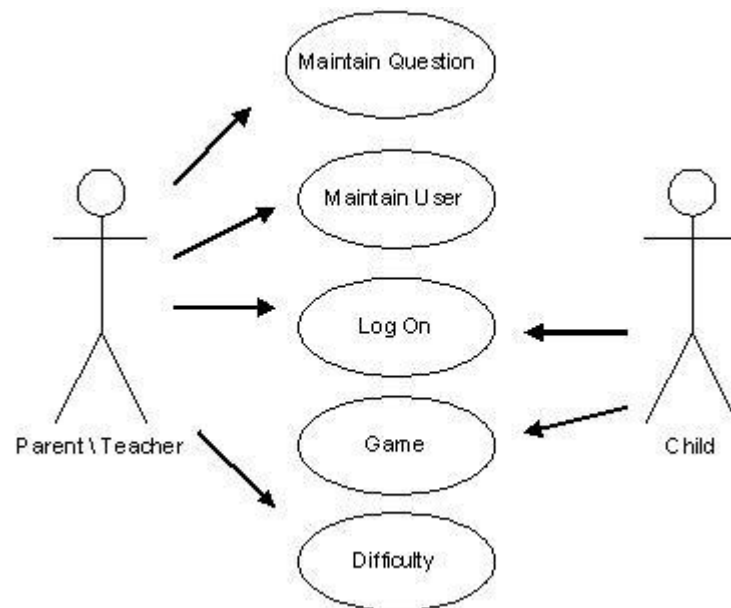
Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasi aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Sri Dharwiyanti ; 2013 : 5)



Gambar II.7. Usecase Diagram

(Sumber : Sri Dharwiyanti ; 2013 : 5)

2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

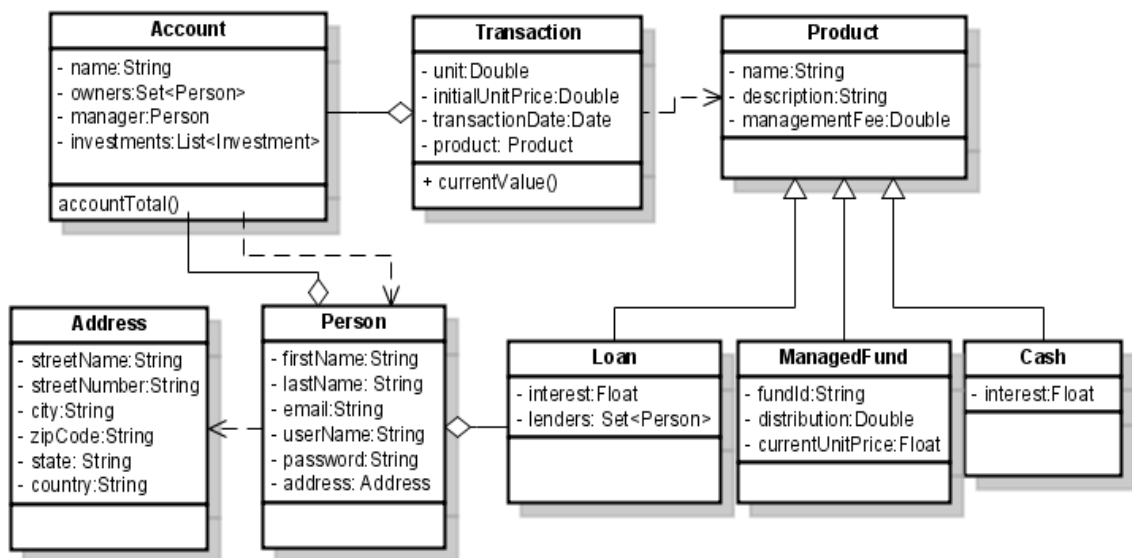
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Tabel II.2. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih

1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 9)



Gambar II.8. Class Diagram

(Sumber : Sri Dharwiyanti ; 2013 : 6)



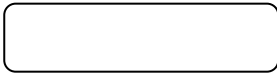
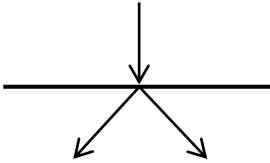
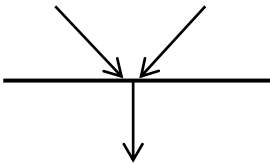
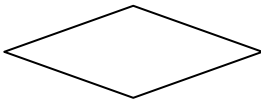

3. Activity Diagram

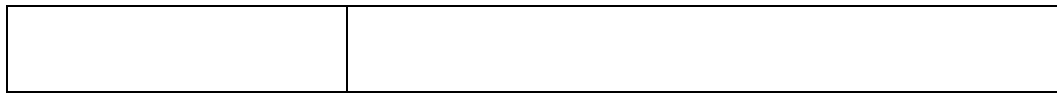
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity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

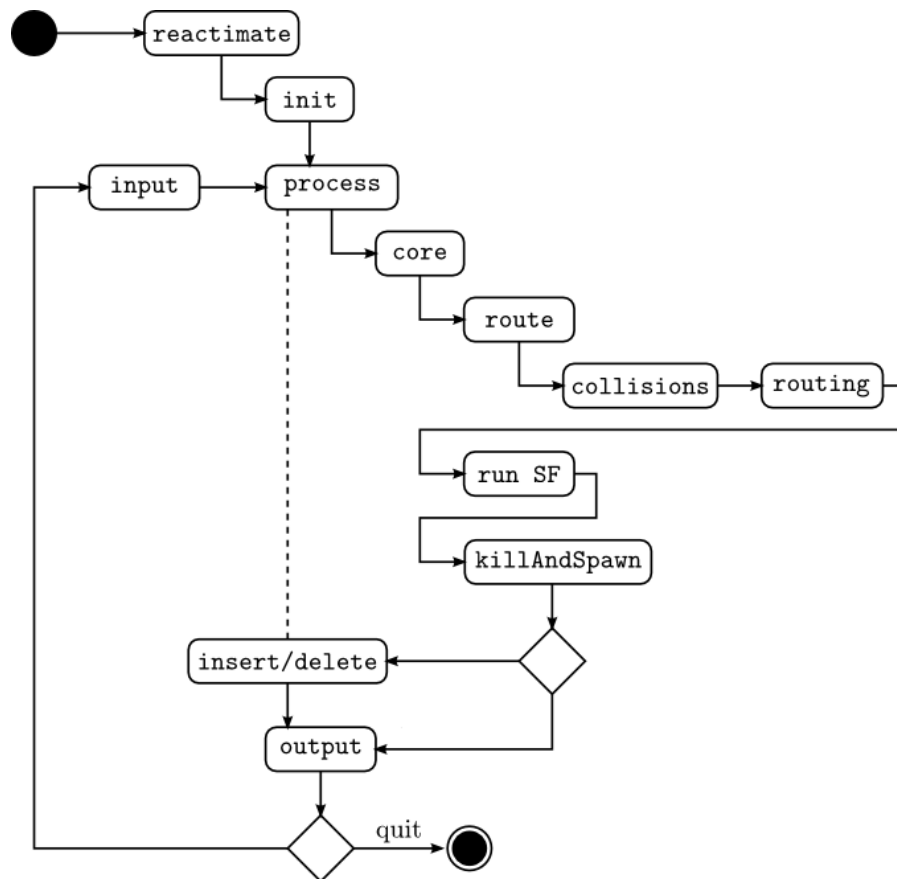
Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas

Tabel II.3. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.



(Sumber : Sri Dharwiyanti ; 2013)



Gambar II.9. Activity Diagram

(Sumber : Sri Dharwiyanti ; 2013 : 8)

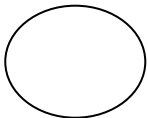
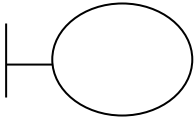
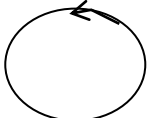

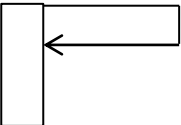
4. Sequence Diagram



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 atas 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.

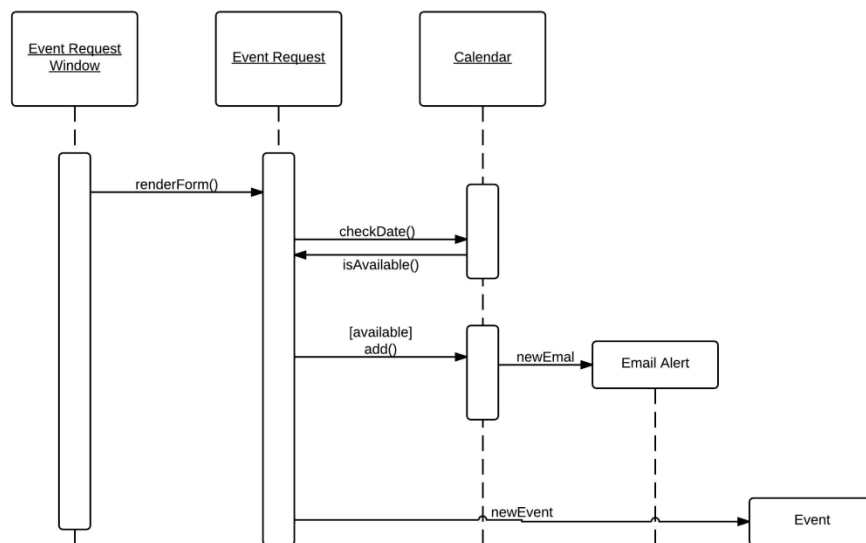
Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

Tabel II.4. Simbol Sequence Diagram

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Sri Dharwiyanti ; 2013 : 9)



Gambar II.10. Sequence Diagram

(Sumber : Sri Dharwiyanti ; 2013 : 9)