

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi Geografis

Berdasarkan penelitian yang dilakukan oleh Qoriani(2012 : 2)GIS atau sistem informasi berbasis pemetaan geografi adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang terkait dengan sistem pemetaan dan analisis terhadap segala sesuatu, serta peristiwa-peristiwa yang terjadi di muka bumi. Teknologi GIS mengintegrasikan operasi pengolahan data berbasis *database* yang biasa digunakan, seperti pengambilan data berdasarkan kebutuhan serta analisis statistik dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar tertentu.

Konsep GIS telah diperkenalkan di Indonesia sejak pertengahan tahun 1980-an, dan kini telah dimanfaatkan di berbagai bidang baik negeri maupun swasta. Kemampuan dasar dari GIS adalah mengintegrasikan berbagai operasi basis data seperti *query*, menganalisisnya, dan menyimpan serta menampilkan dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan GIS dengan sistem informasi lain. Komponen GIS terdiri atas *hardware*, *software*, data, dan *user*. Dengan adanya GIS diharapkan tersedia informasi yang cepat, benar dan akurat tentang keadaan di lingkungannya.

II.1.1. Komponen Sistem Informasi Geografis

Menurut 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.1.2. Data Spasial

Berdasarkan penelitian yang dilakukan oleh Ichsan (2012 : 51) Sebagian besar data yang akan ditangani dalam SIG merupakan data spasial, data yang berorientasi geografis. Data ini memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang berbeda dari data lain, yaitu informasi lokasi (spasial) dan informasi deskriptif (atribut) 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 nonspasial, suatu lokasi yang memiliki beberapa keterangan yang berkaitan dengannya. Contoh jenis vegetasi, populasi, luasan, kode pos, dan sebagainya.

II.1.3. Format Data Spasial

Berdasarkan penelitian yang dilakukan oleh Ichsan (2012 : 51) 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 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 (titik perpotongan antara dua buah garis).

2. Data *raster*

Data *raster* (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*).

II.1.4. Data Vektor

Data Vektor merupakan bentuk bumi yang dipresentasikan 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. Hal ini sangat berguna untuk analisa yang membutuhkan kepetapan posisi, misalnya pada basisdata batas-batas kedaster. Contoh pengguna lainnya adalah untuk mendefinisikan hubungan spasial dari beberapa fitur. Kelemahan data vektor yang utama adalah ketidakmampuannya dalam mengakomodasi perubahan gradual (Marjuki ; 2014 : 5).

II.1.5. 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*). Pada data

raster, resolusi (definisi visual) tergantung pada ukuran pikselnya. Dengan kata lain, resolusi piksel menggambarkan ukuran sebenarnya di permukaan bumi yang diwakili oleh setiap piksel pada citra. Semakin kecil ukuran permukaan bumi yang direpresentasikan oleh satu sel, semakin tinggi resolusinya. Data *raster* sangat baik untuk merepresentasikan batas-batas yang berubah secara gradual, seperti jenis tanah, kelembaban tanah, vegetasi, suhu tanah dan sebagainya. Keterbatasan utama dari data *raster* adalah besarnya ukuran file, semakin tinggi resolusi *grid*nya semakin besar pula ukuran filenya dan sangat tergantung pada kapasitas perangkat keras yang tersedia (Marjuki ; 2014 : 5).

II.3. Metode *Equirectangular Approximation*

Berdasarkan penelitian yang dilakukan oleh Sayed (2014 ; 57) Proyeksi *Equirectangular Approximation* atau sering disebut sebagai *equidistantcylindrical projection*, *geographic projection*, atau *la carte parallélogrammatique projection*, adalah proyeksi sederhana pada peta. Proyeksi peta menggunakan teknik ini diukur dari meridian peta terhadap garis tegak lurus secara vertikal dan dalam ruang yang konstan. Pada proyeksi ini, luas pada permukaan peta tidak selalu sama atau *conformal* yang disebabkan oleh distorsi atas bentuk bumi yang sebenarnya tidak benar-benar bulat. Proyeksi peta dan pengukuran jarak menggunakan formula ini digunakan sebagai standar proyeksi pada dataset global seperti *Cartesia* dan *NASA World Wind* karena keterhubungan antara piksel pada gambar peta dan posisi lokasi geografis bumi cukup sederhana.

Pengukuran jarak menggunakan *Equirectangular Approximation* tepat digunakan jika perangkat komputer yang digunakan tidak memiliki performance yang memadai dan akurasi tidak terlalu penting serta jarak yang diukur masih dalam skala kecil, Teorema pitagoras dapat digunakan pada proyeksi *Equirectangular Approximation* menggunakan formula di bawah ini:

$$\begin{aligned}x &= \Delta\lambda \cdot \cos(\varphi) \\y &= \Delta\varphi \\d &= R \cdot \sqrt{(x^2 + y^2)}\end{aligned}$$

Dimana:

λ : titik koordinat longitudinal

φ : titik koordinat latitude

x: posisi horizontal pada peta

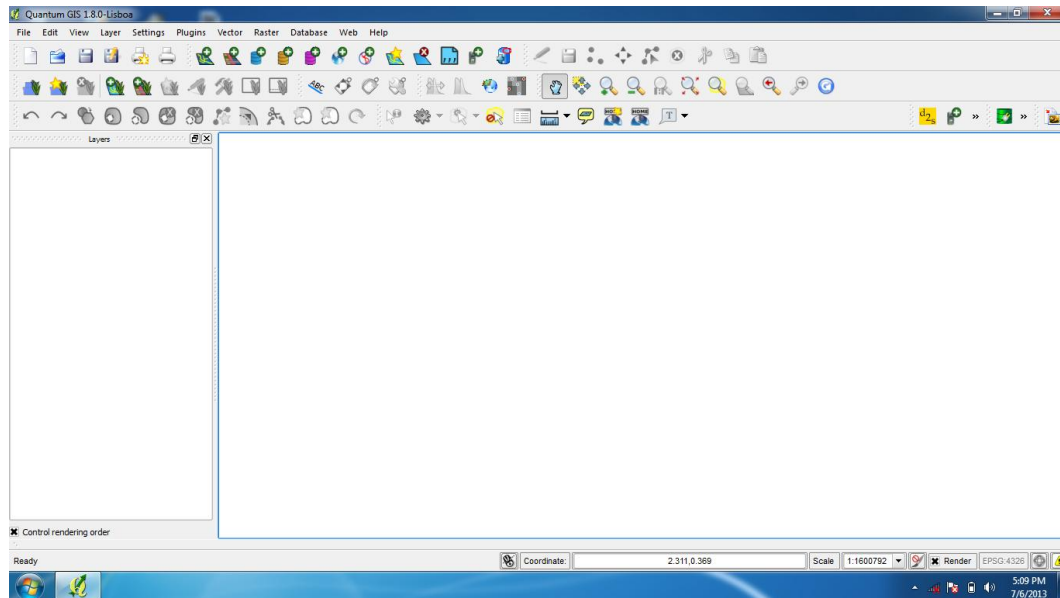
y: posisi vertikal pada peta

d: jarak antara kedua posisi.

II.4. Quantum GIS

Quantum GIS merupakan salah satu perangkat lunak *open source* di bawah proyek resmi dari *OpenSource Geospatial Foundation* (OSGeo) yang dapat dijalankan dalam sistem operasi Windows, MacOSX, Linux dan Unix. Aplikasi ini menawarkan pengolahan data geospasial 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.

Quantum GIS dapat diakses melalui situs resmi yang beralamatkan www.qgis.org (Hussein ; 2012 : 93).



Gambar II.1. Tampilan Quantum GIS
(Sumber : Hussein ; 2012 : 93)

II.4.1. Peta

Peta merupakan bagian penting dari SIG. Peta dalam SIG dapat berfungsi sebagai data masukan, cara representasi data, model untuk melakukan analisa, dan bentuk penampilan informasi. Peta adalah gambaran sebagian atau seluruh muka bumi baik yang terletak di atas maupun di bawah permukaan dan disajikan pada bidang datar pada skala dan proyeksi tertentu (secara matematis). Karena dibatasi oleh skala dan proyeksi maka peta tidak akan pernah selengkap dan sedetail aslinya (bumi), karena itu diperlukan penyederhanaan dan pemilihan unsur yang akan ditampilkan pada peta (Marjuki ; 2014 : 6).

II.4.2. Aplikasi Quantum GIS Dekstop

Berdasarkan penelitian yang dilakukan oleh Ismail (2013 : 1) Quantum GIS Desktop sebagai bagian dari proyek tersebut di atas, memiliki beberapa aplikasi antara lain :

1. Menampilkan data *vector* dan *raster* dalam berbagai format dan proyeksi yang berbeda. Format data yang didukung antara lain :
 - a. *PostGIS* dan *SpatiaLite*,
 - b. Format *vector* yang didukung oleh *OGR library*, termasuk *ESRI shapefiles*, *MapInfo*, *SDTS* and *GML*.
 - c. Format *raster* yang didukung oleh *formats supported by the GDAL library**, seperti *digital elevation models (DEM)*, foto udara, dan citra *landsat*.
 - d. *GRASS locations* dan *mapsets*,
 - e. *spatial data online* yang disediakan oleh *OGC-compliant WMS* , *WMS-C (Tile cache)*, *WFS* and *WFS-T*
2. Aplikasi pemetaan dan pengolahan data spasial, meliputi :
 - a. *on-the-fly reprojection* : Mengelola sistem proyeksi data spasial
 - b. *print composer* : Membuat layout peta
 - c. *overview panel*
 - d. *spatial bookmarks*

- e. *identify/select features* : mengetahui informasi data spasial berdasarkan atribut data
 - f. *edit/view/search attributes* : manipulasi data atribut
 - g. *feature labeling* : memberi label pada tampilan data spasial berdasarkan atribut data
 - h. *vector diagram overlay*
 - i. *symbolology* : memilih dan menyesuaikan symbol dan warna untuk setiap *layer* data spasial
 - j. *graticule layer* : menambahkan grid sebagai informasi posisi dan koordinat
 - k. *map decorations* seperti *north arrow, scale bar and copyright label* : menambah informasi orientasi peta, skala, dan riwayat peta
3. Dukungan membuat, merubah, dan *export* data spasial
- a. Digitasi untuk membuat *layervector*
 - b. *field and rastercalculator*
 - c. *plugin georeferencer*
 - d. *GPS tools* untuk *import* dan *export* data dengan *ekstensi *.gpx, convertformat* data GPS lainnya *formats ke GPX, or down/upload directly to a GPS unit*
4. Analisis data spasial
- a. *map algebrab*
 - b. *analisis terrain*
 - c. *Pemodelan hidrologi*

- d. Analisis jaringan (*network*)
- 5. Publikasi peta melalui jaringan internet menggunakan QGIS *Server* atau *export* kedalam *Mapfile*(UMN *MapServer*)

II.4.3. Kelebihan Quantum GIS

Berdasarkan penelitian yang dilakukan oleh Ismail (2013 : 2) QGIS sebagai alternatif dari sekian banyak perangkat lunak pengolahan data spasial, dengan beberapa kelebihan diantaranya :

1. Gratis : tidak membutuhkan biaya untuk proses instalasi dan penggunaan program
2. Bebas : dapat menambah dan memodifikasi fungsi dalam QGIS
3. Terus berkembang : setiap orang dapat menambah fitur baru dan penyempurnaan aplikasi,
4. Ketersediaan dokumen panduan dan pertolongan : pendukung panduan dan bantuan terhadap permasalahan tersedia *online* dan dapat diunduh dalam bentuk dokumen
5. Multi sistem operasi : dapat diinstal di *MacOS*, *Windows*, *Linux*.

II.5. Pengertian Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah Open

Handset Alliance (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Ichwan ; 2011 : 15).

II.5.1. Platform Android

Platform Android adalah sebuah *software stack* produksi google untuk perangkat *mobile* yang terdiri atas sistem operasi, *middleware*, dan *key applications*. Aplikasi Android dapat dikembangkan melalui *Android Standart Development Kit* (Android SDK) menggunakan sintaks bahasa pemrograman Java. Aplikasi Android nantinya tidak akan berjalan langsung diatas kernel sistem operasi namun berjalan diatas Dalvik, sebuah *virtual machine* yang khusus dirancang untuk digunakan pada sistem *embedded*. Arsitektur system terdiri atas 5 *layer*, pemisahan *layer* bertujuan untuk memberikan abstraksi sehingga memudahkan pengembangan aplikasi, *Layer-layer* tersebut adalah *layer* aplikasi, *layer framework* aplikasi, *layer libraries*, *layer runtime*, dan *layer kernel* (Habibi ; 2011 : 2)

II.6. Teknik Normalisasi

Menurut Simarmata (2010 : 76) Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata 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.7.1. Bentuk-bentuk Normalisasi

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

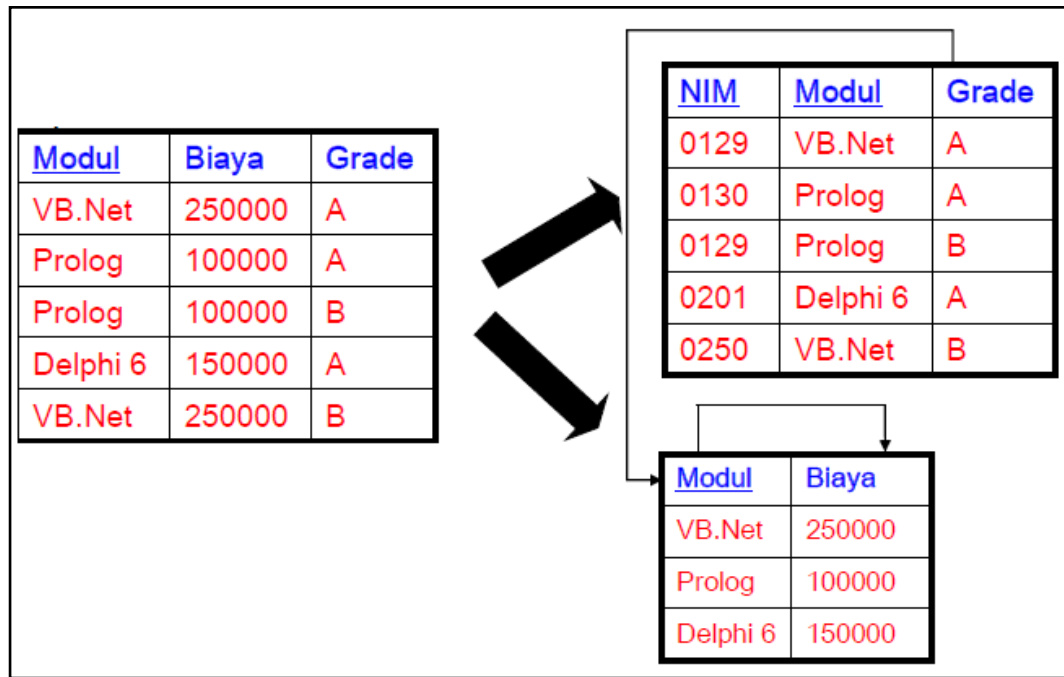
Gambar II.2. Normalisasi 1NF

Sumber : 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 : 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 : 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 : 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 : Simarmata ; 2010 : 76

II.7. *Unified Modeling Language (UML)*

Menurut Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML

saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.


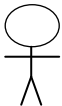
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel III.1. Simbol *Use Case*

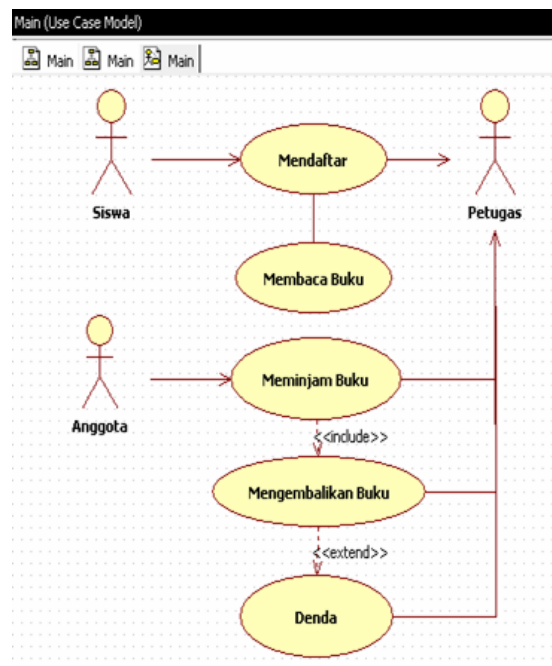
Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang 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</i></p>

	<i>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 mengidikasikan 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 : Gata ; 2013 : 4)

Contoh dari pembuatan *use case diagram* dapat dilihat pada gambar II.8

berikut :




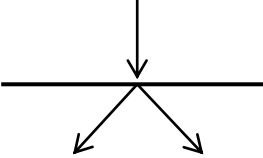
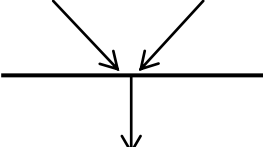
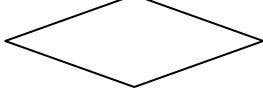



Gambar. II.7. Use Case Diagram
(Sumber : Gata ; 2013 : 4)

2. Diagram Aktivitas (Activity Diagram)

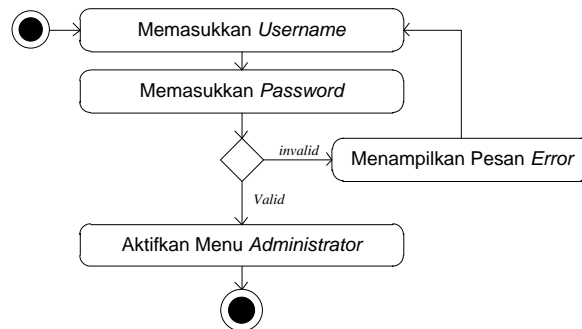
Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel III.2. 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>Activites</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 : Gata ; 2013 : 6)

Contoh dari pembuatan *activity diagram* dapat dilihat pada gambar II.9 berikut :




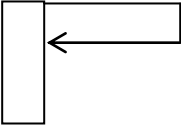


Gambar. II.8. Activity Diagram
(Sumber : Gata ; 2013 : 6)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel III.3. Simbol *Sequence Diagram*

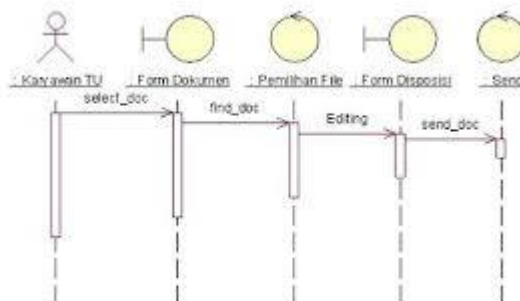
Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara 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.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(*Sumber : Gata ; 2013 : 7*)

Contoh dari pembuatan *sequence diagram* dapat dilihat pada gambar II.10

berikut :



Gambar. II.9.Sequence Diagram

(*Sumber : Gata ; 2013 : 7*)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class diagram secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

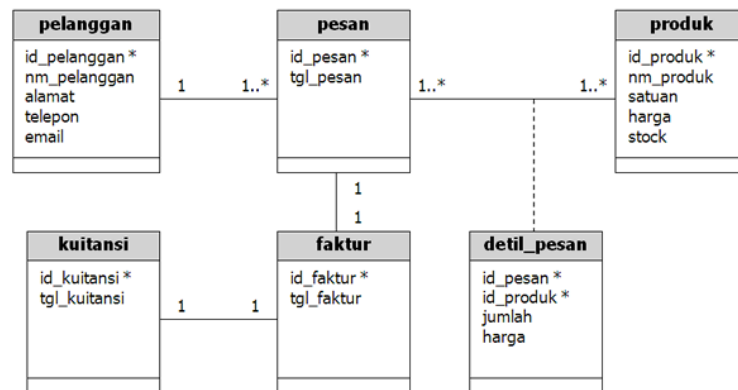
Tabel III.4. *Multiplicity Class Diagram*

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 : Gata ; 2013 : 8*)

Contoh dari pembuatan *use case diagram* dapat dilihat pada gambar II.11

berikut :



Gambar. II.10. *Class Diagram*

(*Sumber : Gata ; 2013 : 8*)