

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Geografis

Pemahaman tentang bumi dimiliki manusia sejak ada di muka bumi ini. Sejak lahir manusia memerlukan berbagai unsur yang ada di bumi. Unsur tersebut seperti udara yang bersih, makanan, pakaian dan pemukiman.

Timbulnya tuntutan pemenuhan berbagai kebutuhan hidup yang tidak diperoleh dari lingkungan tempat tinggalnya dan adanya hasrat keingintahuan tentang benda serta gejala yang ada dipermukaan bumi. Mendorong setiap manusia untuk mengadakan perjalanan ke daerah di luar tempat tinggalnya.

Berkembangnya sistem pengetahuan turut mendorong manusia untuk mengenal alam dan lingkungannya lebih jauh. Misalnya, perdagangan antar daerah telah mendorong manusia untuk mengenal daerah di luar wilayahnya. Dari hasil kunjungan tersebut, mereka dapat mengenal kondisi alam, penduduk, dan kondisi lainnya. Berbagai hasil perjalanannya tersebut kemudian disampaikan kepada orang lain sehingga orang lain tertarik untuk mengunjunginya. Berawal dari perjalanan inilah munculnya ilmu geografi (Hartono ; 2008 : 2).

II.2. Sistem Informasi Geografis

GIS atau sistem informasi berbasis pemetaan dan 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 statistic dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar tertentu (Jurnal Link Vol. 17/No.2/Sept 2012 ; Hersa Farida Qoriani : 2).

Sistem Informasi Geografis (*geographic information system-GIS*) adalah kategori khusus dari DSS yang menggunakan teknologi visualisasi data untuk menganalisis dan menampilkan data untuk perencanaan dan pengambilan keputusan dalam bentuk peta digital. Peranti lunak tersebut merakit, menyimpan, memanipulasi, dan menampilkan secara geografis informasi referensi, menghubungkan data dengan titik, garis, dan bidang pada sebuah peta. GIS mempunyai kemampuan membuat model, memungkinkan manajer untuk mengubah data dan secara otomatis memperbarui scenario bisnis untuk mencari solusi yang lebih baik.

GIS membantu pengambilan keputusan yang membutuhkan pengetahuan tentang distribusi penduduk atau sumber daya lain secara geografis. Sebagai contoh, GIS mungkin digunakan untuk membantu pemerintah negara dan pemerintah lokal menghitung waktu respons bahaya untuk bencana alam, untuk membantu perusahaan eceran mengidentifikasi lokasi pertokoan baru, atau membantu bank mengidentifikasi tempat terbaik untuk membangun cabang atau memasang terminal ATM baru.

Sesi Interaktif Manajemen menjelaskan aplikasi GIS untuk mengendalikan tindak kejahatan. CompStat diciptakan oleh Departemen Kepolisian *New York* untuk mengambil data tentang insiden tindak kejahatan dan aktivitas penegakan hukum di setiap sudut kota. CompStat menggunakan peranti lunak GIS untuk menampilkan data mengenai di mana kejahatan berlangsung dan diklaim berhasil mengurangi jumlah rata-rata tindak kejahatan di *New York* dan kota-kota lain (Kenneth C. Laudan ; 2008 : 167).

II.3. Pengertian Data Spasial Sistem Informasi Geografis

Dalam Sistem Informasi Geografis (SIG) model data yang akan digunakan dari bentuk dunia nyata harus diimplementasikan ke dalam basisdata. Data ini dimasukkan ke dalam komputer yang kemudian memanipulasi objek dasar yang memiliki atribut geometri (entity spasial/entity geografis). Data spasial sendiri pada dasarnya mempunyai banyak pengertian, namun dapat disimpulkan bahwa data spasial adalah suatu entitas data dalam Sistem Informasi Geografis (SIG) yang dapat dikelola, dianalisa dan dapat memetakan informasi objek keruangan beserta data-data atributnya serta dapat disimpan di dalam database dan dapat ditampilkan kedalam suatu sistem yang terpadu sehingga dapat mendukung dalam pengambilan keputusan. Model data raster merupakan salah satu macam dari data spasial yang menampilkan, menempatkan, dan menyimpan data spasial dengan menggunakan struktur matriks atau piksel-piksel yang membentuk grid. Setiap piksel atau sel ini memiliki atribut sendiri, termasuk koordinat yang unik (disudut grid, dipusat grid ataupun ditempat lainnya). Akurasi model data ini sangat bergantung pada resolusi atau ukuran pikselnya di permukaan bumi. Entity spasial

raster disimpan pada layer yang secara fungsionalitas direlasikan dengan unsur-unsur petanya. Sumber entity spasial raster bisa didapatkan dari citra satelit, radar, atau ketinggian digital. (Jurnal Vol. 4, No. 1, Juni 2006 ; Yudha Widiatmoko : 28).

II.4. Pengertian Quantum GIS

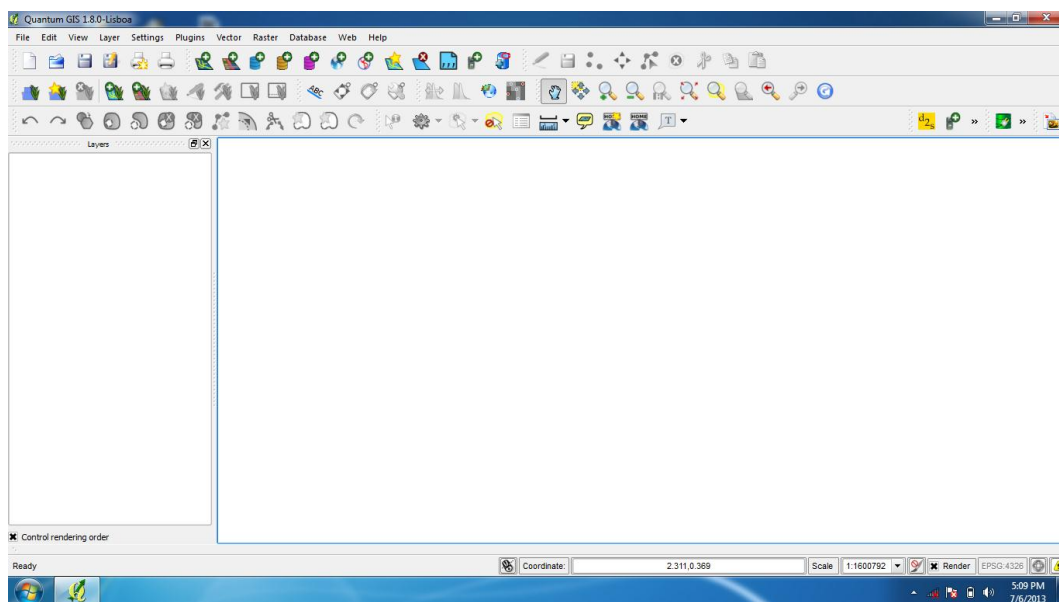
Quantum GIS atau yang sering disingkat menjadi QGIS adalah sebuah aplikasi sistem informasi geografis berbasis *desktop* yang menyediakan fitur untuk menampilkan data, perubahan data dan kemampuan dalam menganalisis data spasial. QGIS dapat berjalan pada sistem operasi Linux, UNIX, Mac OS, dan Windows.

Quantum GIS dapat dibuat dengan bahasa pemrograman C++ dan untuk tampilan grafisnya menggunakan pustaka kode *QT-Library*. Quantum GIS memungkinkan untuk membentuk integrasi pada Plug-In yang dikembangkan dengan C++ maupun Python.

QT-Library menyediakan tampilan grafis yang dapat berjalan secara *Cross-Platform* dalam *Framework* pengembangan aplikasi yang didukung oleh perangkat lunak lainnya. Quantum GIS memungkinkan untuk dihubungkan atau diintegrasikan dengan berbagai paket perangkat lunak GIS yang bersifat Open-Source lainnya, seperti PostGIS, GRASS, dan MapServer untuk memberikan fungsionalitas yang ekstensif kepada penggunaanya. Quantum GIS secara berkesinambungan terus diperbaiki dan dikembangkan oleh grup pengembang

yang aktif dan pengembang sukarela yang secara teratur merilis pembaharuan dan perbaikan pada beberapa kesalahan sistem.

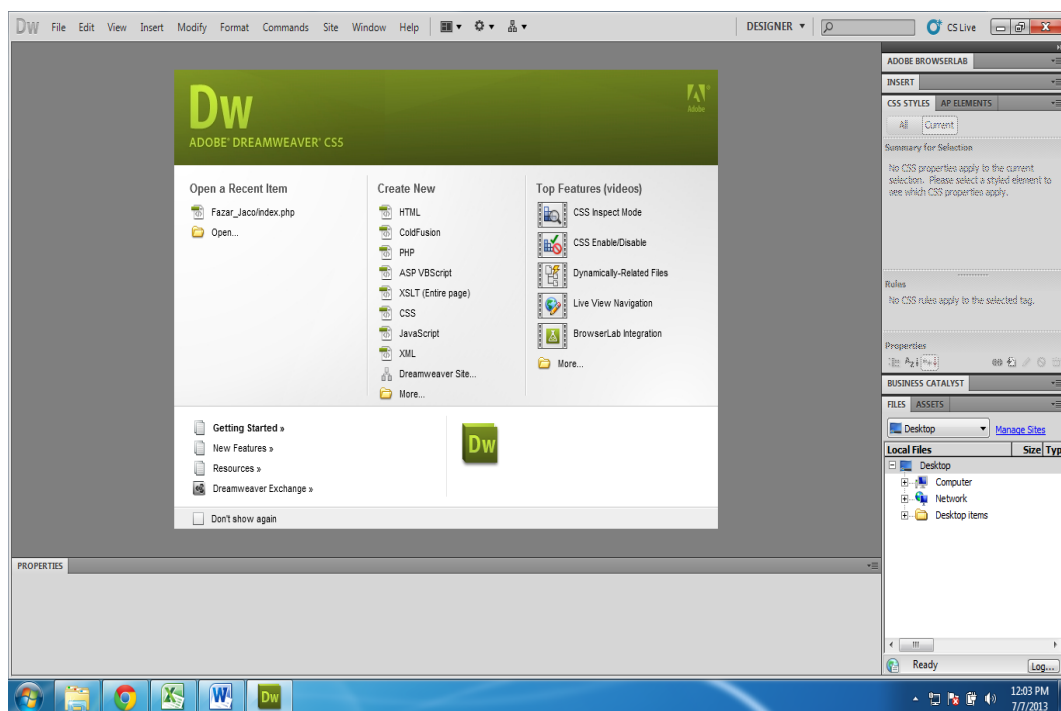
Komponen perangkat lunak GIS dibangun berdasarkan blok-blok sehingga dapat ditambahkan perangkat lunak GIS dan dibentuk dengan baik serta lingkungan pengembangan yang dapat disesuaikan untuk pengguna. Fungsi komponen yang spesifik memberikan dedikasi tugas yang ditambahkan pada lingkungan alat pengembangan GIS, seperti komponen yang memungkinkan untuk memasukkan format data tertentu agar dapat dikonversi, penganalisis data teratur dan perangkat pemrosesan citra, perangkat pengembangan pengguna di sisi lainnya sebagai fungsi yang spesifik (Yupo Chan ; 2011 : 432).



Gambar II.1. Tampilan Quantum GIS
(Sumber : Yupo Chan ; 2011 : 432)

II.5. Pengertian Macromedia Dreamweaver

Macromedia Dreamweaver adalah sebuah *software web design* yang menawarkan cara mendesain *website* dengan dua langkah sekaligus dalam satu waktu, yaitu mendesain dan memprogram. Dreamweaver memiliki satu jendela mini yang disebut *HTML Source*, tempat kode-kode HTML tertulis. Setiap kali kita mendesain *web*, seperti menulis kata-kata, meletakkan gambar, membuat tabel dan proses lainnya, tag-tag HTML akan tertulis secara langsung mengiringi proses pengaturan *website*. Artinya kita memiliki kesempatan untuk mendesain *Website* sekaligus mengenal tag-tag HTML yang membangun *website* itu. Di lain kesempatan kita juga dapat mendesain *website* hanya dengan menulis tag-tag dan teks lain di jendela *HTML Source* dan hasilnya dapat dilihat langsung di layar (M. Suyanto ; 2009 : 244).



Gambar II.2. Tampilan Dreamweaver

(Sumber : M. Suyanto ; 2009 : 244)

II.6. Pengertian PHP

PHP merupakan bahasa *scripting* yang berjalan di sisi *server* (*server-side*). Semua perintah yang ditulis akan dieksekusi oleh *server* dan hasil jadinya dapat dilihat melalui *browser*. Saat ini PHP versi 4 sudah di-*release* di pasaran, mengikuti jejak kesuksesan versi sebelumnya, PHP 3. Selain dapat digunakan untuk berbagai sistem operasi, koneksi *database* yang sangat mudah menyebabkan bahasa *scripting* ini digemari para *programmer web*. Beberapa perintah PHP yang kita pelajari sebatas pada perintah untuk menampilkan *tag-tag* wml, akses *database* MySQL dan pengiriman *email* (Ridwan Sanjaya ; 2009 : 73).

II.7. Pengertian Database

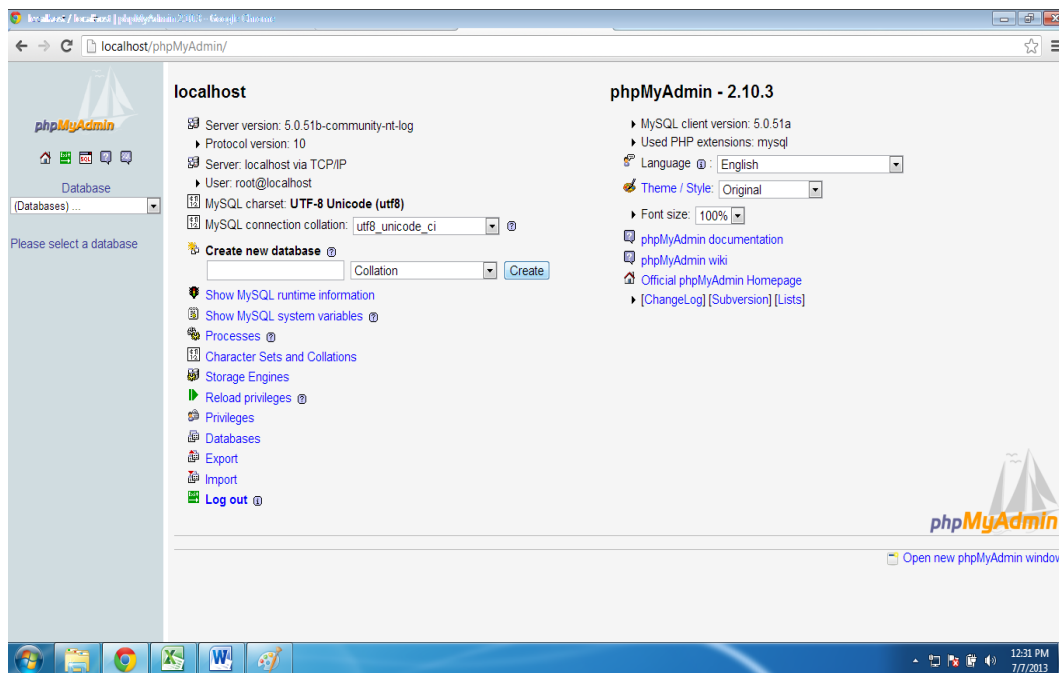
Database adalah sekumpulan *file* data yang saling berhubungan dan diorganisasi sedemikian rupa sehingga memudahkan untuk mendapat dan memproses data. Lingkungan sistem *database* menekankan data yang tidak tergantung (*idenpendent data*) pada aplikasi yang akan menggunakan data. Data adalah kumpulan fakta dasar (mentah) yang terpisah.

Sebuah *database* harus dibuat dengan rapi agar setiap data yang dimasukkan sesuai dengan tempatnya. Sebagai contoh, di sebuah perpustakaan, penyimpanan buku dikelompokkan berdasar jenis atau kategori-kategori tertentu, misalnya kategori buku komputer, buku pertanian dan lain-lain. Kemudian dikelompokkan lagi berdasarkan abjad judul buku. Ini dilakukan agar setiap pengunjung dapat dengan mudah mencari dan mendapatkan buku yang dimaksud (Wahana Komputer ; 2006 : 1).

II.8. Pengertian MySQL

MySQL adalah suatu sistem manajemen basis data relasional (RDBMS-*Relational Database Management System*) yang mampu bekerja dengan cepat, kokoh dan mudah digunakan. Contoh RDBMS lain adalah *Oracle*, *Sybase*. Basis data memungkinkan anda untuk menyimpan, menelusuri, mengurutkan dan mengambil data secara efisien. *Server MySQL* yang akan membantu melakukan fungsionalitas tersebut. Bahasa yang digunakan oleh MySQL tentu saja adalah SQL-standar bahasa basis data relasional di seluruh dunia saat ini.

MySQL dikembangkan, dipasarkan dan disokong oleh sebuah perusahaan Swedia bernama MySQL AB. RDBMS ini berada di bawah bendera GNU GPL sehingga termasuk produk *Open Source* dan sekaligus memiliki lisensi komersial. Apabila menggunakan MySQL sebagai basis data dalam suatu situs Web, anda tidak perlu membayar. Akan tetapi, jika anda ingin membuat produk RDBMS baru dengan basis MySQL dan kemudian menjualnya, anda wajib bertemu muka dengan lisensi komersial (Antonius Nugraha Widhi Pratama ; 2010 : 10).



Gambar II.3. Tampilan MySQL

(Sumber : Antonius Nugraha Widhi Pratama ; 2010 : 10)

II.9. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database*. Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF (Yuniar Supardi, 2010 : 448).

Model entity-relationship pertama kali diperkenalkan oleh Peter Chen pada tahun 1976. Dalam pemodelan ini dilakukan dengan tahapan sebagai berikut:

- a. Memilih entitas-entitas yang akan disusun dalam basis data dan menentukan hubungan antar entitas yang telah dipilih.
- b. Melengkapi atribut-atribut yang sesuai pada entitas dan hubungan sehingga diperoleh bentuk tabel normal penuh (ternormalisasi).





Entitas merupakan sesuatu yang dapat diidentifikasi dalam lingkungan kerja pengguna. Entitas yang diberikan tipe dikelompokkan ke kelas entitas. Perbedaan antara kelas entitas dan instansientitas adalah sebagai berikut:

- a. Kelas entitas adalah kumpulan entitas dan dijelaskan oleh struktur atau format entitas di dalam kelas.
- b. Instansi kelas merupakan bentuk penyajian dari fakta entitas.

Umumnya terdapat banyak instansi entitas di dalam setiap entitas kelas. Setiap entitas kelas memiliki atribut yang menjelaskan karakteristik dari entitas tersebut, sedangkan setiap instansi entitas mempunyai identifikasi yang dapat bernilai unik (mempunyai nilai yang berbeda untuk setiap identifikasinya) atau non-unik (dapat bernilai sama untuk setiap identifikasinya).

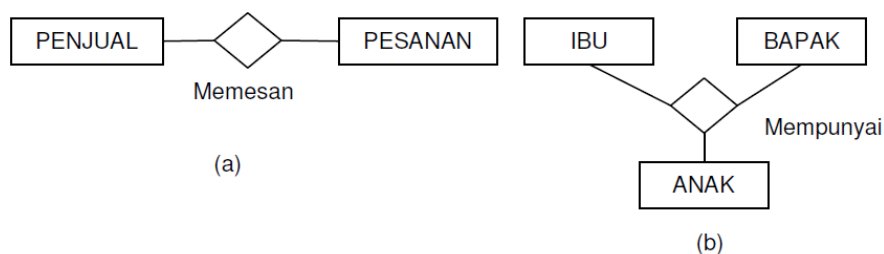
Antara entitas diasosiasikan dalam suatu hubungan (*relationship*). Suatu relasi dapat memiliki beberapa atribut. Jumlah kelas entitas dalam suatu relasi disebut derajat relasi (Haidar Dzacko ; 2007 : 21).

Tabel II.1. Simbol ERD

Notasi	Keterangan
	Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai
	Relasi , menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda
	Atribut , berfungsi mendeskripsikan karakter entitas (Atribut yang berfungsi sebagai Key diberi garis bawah)
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut

(Sumber : Yuniar Supardi ; 2010 : 448)

Gambar di bawah ini merupakan contoh dari relasi berderajat dua dan relasi berderajat tiga



Gambar II.4. (a) Relasi Berderajat Dua, (b) Relasi Berderajat Tiga

(Sumber : Haidar Dzacko ; 2007 : 21)

II.10. Kamus Data

Kamus data (*data dictionary*) mencakup definisi-definisi dari data yang disimpan di dalam basis data dan dikendalikan oleh sistem manajemen basis data. Figur 6.5 menunjukkan hanya satu tabel dalam basis data jadwal. Struktur basis data yang dimuat dalam kamus data adalah kumpulan dari seluruh definisi *field*, definisi tabel, relasi tabel dan hal-hal lainnya. Nama *field* data, jenis data (seperti teks atau angka atau tanggal), nilai-nilai yang valid untuk data dan karakteristik-karakteristik lainnya akan disimpan dalam kamus data. Perubahan-perubahan pada struktur data hanya dilakukan satu kali di dalam kamus data, program-program aplikasi yang mempergunakan data tidak akan ikut terpengaruh (Raymond McLeod ; 2008 : 171).

No.	Nama Kolom	Tipe Data	Lebar	Keterangan
1.	KdBrg	Char	5	Kode Barang
2.	NamaBrg	Varchar	50	Nama Barang
3.	StokBrg	Int	4	Stok Barang
4.	HargaBeli	Money	8	Harga Beli
5.	HargaJual	Money	8	Harga Jual
6.	NoBeli	Int	4	Nomor Beli
7.	TglBeli	DateTime	8	Tanggal Beli
8.	NoBuktiBeli	Varchar	20	Nomor Bukti Beli
9.	TotalBeli	Money	8	Total Beli
10.	KdSupp	Char	5	Kode Supplier
11.	NamaSupp	Varchar	50	Nama Supplier
12.	AlmtSupp	Varchar	100	Alamat Supplier
13.	TelpSupp	Varchar	12	Nomor Telpon Supplier
14.	KdCust	Char	5	Kode Customer
15.	NamaCust	Varchar	50	Nama Customer
16.	AlmtCust	Varchar	100	Alamat Customer
17.	TelpCust	Varchar	15	Nomor Telpon Customer
18.	JumlahBeli	Int	4	Jumlah Beli
19.	SubTotBeli	Money	8	Sub Total Beli
20.	NoJual	Int	4	Nomor Jual
21.	TglJual	DateTime	8	Tanggal Jual
22.	TotalJual	Money	8	Total Jual
23.	JumlahJual	Int	4	Jumlah Jual
24.	SubTotJual	Money	8	Sub Total Jual

Gambar. II.5. Contoh Kamus Data Penjualan Barang

(Sumber : Raymond McLeod ; 2008 : 171)

II.11. Teknik Normalisasi

Proses normalisasi menyediakan cara sistematis untuk meminimalkan terjadinya kerangkapan data di antara relasi dalam perancangan logikal basis data.

Format normalisasi terdiri dari lima bentuk, yaitu:

II.11.1. Bentuk-bentuk Normalisasi

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

Suatu tabel dikatakan sudah 1NF jika telah memenuhi ketentuan sebagai berikut:

- Tidak ada atribut mempunyai nilai berulang atau nilai array
- Tidak mempunyai baris yang rangkap

Bentuk unnormal mengijinkan nilai-nilai pada suatu atribut dapat berulang.

Contoh normalisasi 1NF adalah seperti pada tabel berikut :

Tabel II.2. Normalisasi 1NF

NIM	NamaAwal	NamaAkhir	Fakultas
122233	Asep	Darma	Ilmu Komputer
233323	Angling	Darma	Ilmu Komputer
244455	Bergola	Ijo	Hukum
334343	Jaka	Sembung	Kebidanan
322323	Jaka	Tarub	Hukum

(Sumber : Haidar Dzacko ; 2007 : 14).

2. Bentuk normal tahap kedua (2nd Normal Form)

Relasi dapat dikatakan format normal kedua jika sudah dalam format normal pertama dan diikuti kondisi sebagai berikut:

- Key terdiri dari atribut tunggal
- Setiap atribut nonkey ketergantungan fungsional pada semua key atau tidak terjadinya ketergantungan pada key *composite*.

Misalnya tabel UNIV berada dalam normal kedua dengan mengasumsikan DNO sebagai key, kecuali CRSE. Jika ditentukan CNO

dan SECNO sebagai key composite, atribut nonkey CNAME tergantung hanya pada CNO, bukan pada SECNO, sehingga CNAME tidak secara ketergantungan fungsional penuh terhadap key (CNO, SECNO).

Tabel II.3. Normalisasi 2NF

NIM	NamaAwal	NamaAkhir	Fakultas	Jenjang
122233	Asep	Darma	Ilmu Komputer	S1
233323	Angling	Darma	Ilmu Komputer	S1
244455	Bergola	Ijo	Hukum	S1
334343	Jaka	Sembung	Kebidanan	D3
322323	Jaka	Tarub	Hukum	S1

(Sumber : Haidar Dzacko ; 2007 : 15).

3. Bentuk normal tahap ketiga (3rd Normal Form)

Relasi dikatakan format normal ketiga jika sudah dalam format normal kedua dan tidak ada ketergantungan transitif di antara atribut. Misalnya tabel STUDNT mempunyai atribut SSNO sebagai *key* (2NF). Ketergantungan transitif terjadi di antara DNO dan COLREG. Saat DNO determinan COLREG tanpa melibatkan *key* SSNO. Contohnya, DNO='CS' termasuk COLREG='Arts/Sc.' tidak tergantung oleh atribut SSNO, sehingga STUDNT belum termasuk 3NF. Yang menjadi catatan, ketergantungan transitif tidak akan terjadi jika ada ketergantungan fungsional di antara atribut-atribut *non-key* yang melibatkan *key*.

Tabel II.4. Normalisasi 3NF

Author Last Name	Author First Name	Book Title	Subject	Collection or Library	Building
Berdahl	Robert	The Politics of the Prussian Nobility	History	PCL General Stacks	Perry-Casta Library
Yudof	Mark	Child Abuse and Neglect	Legal Procedures	Law Library	Townes Hall
Harmon	Glynn	Human Memory and Knowledge	Cognitive Psychology	PCL General Stacks	Perry-Casta Library
Graves	Robert	The Golden Fleece	Greek Literature	Classics Library	Waggener Hall
Miksa	Francis	Charles Ammi Cutter	Library Biography	Library and Information Science Collection	Perry-Casta Library
Hunter	David	Music Publishing and Collecting	Music Literature	Fine Arts Library	Fine Arts Building

(Sumber : Haidar Dzacko ; 2007 : 16).

4. Boyce Code Normal Form (BCNF)

BCNF menentukan setiap determinan adalah kunci kandidat (*candidate key*). Misalnya UNIV mempunyai dua determinan yaitu DNO dan DNAME yang merupakan *kunci kandidat* sehingga termasuk ke dalam BCNF. Di lain pihak CRSLST dalam 3NF tetapi tidak dalam BCNF. Atribut komposisinya (CNO, SECNO, SID, OFRNG) sebagai kunci-kunci kandidat dan tidak ada ketergantungan transitif, sehingga CRSLST termasuk ke dalam 3NF. Namun atribut CNO adalah determinan saat SECNO tergantung penuh secara fungsional terhadap CNO, walaupun CNO bukan kunci kandidat, sehingga CRSLST belum termasuk BCNF.

Tabel II.5. Normalisasi BCNF

SSN	Major	Adviser
123-45-6789	Library and Information Science	Dewey
123-45-6789	Public Affairs	Roosevelt
222-33-4444	Library and Information Science	Putnam
555-12-1212	Library and Information Science	Dewey
987-65-4321	Pre-Medicine	Semmelweis
987-65-4321	Biochemistry	Pasteur
123-54-3210	Pre-Law	Hammurabi

(Sumber : Haidar Dzacko ; 2007 : 18).

5. Bentuk Normal Tahap Keempat dan Kelima

Bentuk ini adalah bentuk normal ketiga atau BCNF dengan nilai atribut tidak tergantung pada nilai banyak (*multivalued dependency*). Konsep pada bentuk ini adalah ketergantungan pada gabungan beberapa atribut (*join dependency*) (Haidar Dzacko ; 2007 : 12).

Tabel II.6. Normalisasi

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

(Sumber : Haidar Dzacko ; 2007 : 20).

II.12. *Unified Modeling Language (UML)*

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.


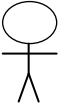

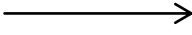
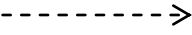
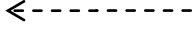
UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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 II.7. Simbol *Use Case*



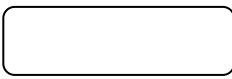
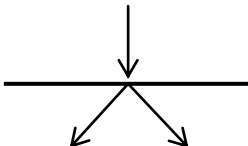
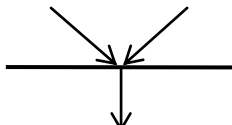
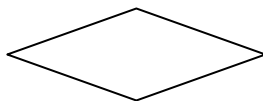

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan 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 : Windu 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 II.8. Simbol *Activity Diagram*

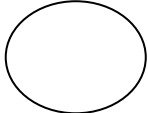
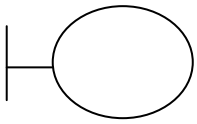
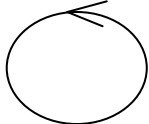
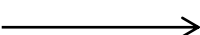
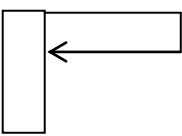


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel 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</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu 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 II.9. 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 : Windu 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*) dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti (Windu Gata ; 2013: 4).

Tabel II.10. *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)