BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Mempelajari suatu sistem akan lebih mengena bila mengetahui terlebih dahulu apakah system itu. Pengertian tentang sistem pertama kali dapat diperoleh dari defenisi sistem itu sendiri. Jika kita perhatikan dengan seksama, diri kita juga terdiri dari sistem yang berfungsi untuk mengantar kita kepada tujuan hidup kita. Sudah banyak ahli yang mengungkapkan berbagai sistem yang bekerja dalam diri manusia, misalnya system kekebalan tubuh untuk menghadapi penyakit cacar dan diptheri. Namun masih banyak pula berbagai sistem yang belum dapat diungkapkan dengan teknologi yang sekarang dimiliki oleh manusia, misalnya sistem kekebalan tubuh untuk menghadapi penyakit AIDS. Contoh sistem lain dalam diri manusia adalah system pernapasan, yang berfungsi untuk menyediakan oksigen bagi tubuh dan untuk mengeluarkan zat asam arang yang merupakan sampah hasil pembakaran di dalam tubuh. (Tata Sutabri ; 2012 : 4).

II.2. Pengertian Sistem Informasi

Sistem informasi bukan merupakan hal yang baru, yang baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi yang memungkinkan manajer merencanakan serta mengendalikan operasi telah ada. Komputer menambahkan satu atau dua dimensi, seperti kecepatan, ketelitian dan

penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri ; 2012 : 38).

II.3. Sistem Informasi Geografis

Sistem Informasi Geografis (*geographicinformationsystem-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 skenario bisnis untuk mencari solusi yang lebih baik.

GIS membantu pengambilan keputusan yang membutuhkan pengetahuan tenatang distribusi penduduk atau sumber daya lain secara geografis. Sebagai contoh, GIS mungkin digunakan untuuk membantu pemerintah Negara dan pemerintah lokal menghitung waktu respon 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 *NewYork* 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 :).

Data Spasial adalah data yang menyimpan komponen-komponen permukaan bumi, seperti : jalan, pemukiman, jenis penggunaan tanah, jenis tanah, dan lain-lain. Model data spasial dibedakan menjadi dua, yaitu : model data raster (Model data yang menampilkan, menempatkan, dan menyimpan spasial dengan menggunakan struktur matriks atau pixel-pixel yang membentuk grid**)** dan model data vector **(**Model data yang menampilkan, menempatkan, dan menyimpan data spasial dengan menggunakan titik, garis-garis, atu kurva atau polygon beserta atribut-atributnya). DataNon Spasial (Tabular/atribut) adalah model data non spasial adalah data yang menyimpan atribut dari kenampakan-kenampakan permukaan bumi tersebut. Misalnya tanah yang memiliki atribut tekstur, kedalaman, struktur pH, dan lainnya (Eriza Siti Mulyani : Jurnal Aplikasi Location Based Service (LBS) Taman Mini Indonesia Indah (TMII) Berbasis Android ; 2011 : 3).

II.4. Intisari Jurnal Sistem Informasi Geografis

Menurut Much Aziz Muslim dalam Jurnal Aplikasi Penentuan Rute Terbaik Berbasis Sistem Informasi Geografis, Perkembangan teknologi SistemInformasi Geografis menyediakan informasiyang merefleksikan seluruh kegiatan di mukabumi. Kegiatan ini meliputi mengambil,mengolah, menyimpan, dan menyampaikan informasi yang diperlukan oleh pengguna. Informasi-informasi yang diperlukan dalam bentuk data-data posisi, koordinat, ruang atau spasial maupun data-data non spasial sangat diperlukan dalam pendukung dasar-dasar untuk memutuskan perencanaan, maupun kegiatan sehari-hari. Sistem informasi geografis dapat diaplikasikan untuk menyelesaikan permasalahan yang berhubungan transportasi dan logistic. Bekerja dengan peta geografis secara mudah dapat dilakukan perhitungan dengan parameter-parameter yang dimasukkan didalam optimalisasi permasalahan di dalam jaringan transportasi.

Dalam kehidupan sehari-hari manusia tidak lepas dengan apa yang disebut *geographicnetworks*, diantaranya adalah jaringan jalan, jaringan sungai, jaringan pipa, rute perjalanan,dan lain sebagainya. Dengan demikian setiap manusia mempunyai permasalahan yang sering dijumpai yaitu pencarian rute yang paling efektif, pembuatan peta arah perjalanan, berdasarkan parameter jarak, waktu dan biaya. Analisis spasial adalah kegiatan untuk menganalisa data keruangan sehingga menghasilkan informasi baru yang dibutuhkan oleh pengguna. Hal ini dilakukan dengan menghubungkan parameter-parameter jarak, waktu dan biaya.

II.5. Pengertian Quantum GIS

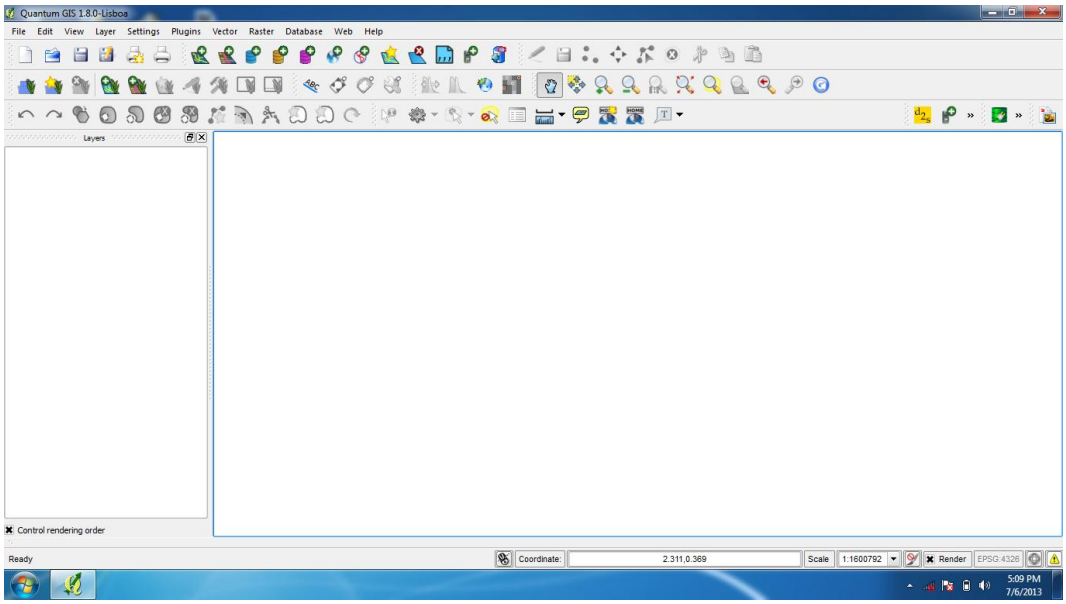
Quantum GIS atau yang sering disingkat menjadi QGIS adalah sebuah aplikasi sistem informasi geografis berbasis *desktop* yang menyediakan fitur untuk menampilkan data, pengubahan data dan kemampuan dalam menganalisis data spasial. QGIS dapat berjalan pada sistem operasi Linux, UNix, Mac OS, dan Windows.

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 penggunanya. 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.6. Pengertian PHP

PHP merupakan suatu bahasa pemrograman sisi server yang dapat anda gunakan untuk membuat halaman Web dinamis. Contoh bahasa yang lain adalah *Microsoft Active Server Page (*ASP) dan *Java Server Page*(JSP). Dalam suatu halaman HTML anda dapat menanamkan kode PHP yang akan dieksekusi setiap kali halaman tersebut dikunjungi. Karena kekayaannya akan fitur yang mempermudah perancangan dan pemrograman Web, PHP memiliki popularitas

yang tinggi. Anda dapat mengecek *survey* popularitas yang dilakukan *netcraft* di URL www.php.net/usage.php.

PHP adalah kependekan dari *Hyper Text Prepocessor* (suatu akronim rekursif) yang dibangun oleh Rasmus Lerdorf pada tahun 1994. Dahulu, pada awal pengembangnnya PHP disebut sebagai kependekan dari *Personal Home Page.* PHP merupakan produk *Open Source* sehingga anda dapat mengakses *source code,* menggunakan dan mengubahnya tanpa harus membayar sepeser pun. Gratis (Antonius Nugraha Widhi Pratama ; 2010 : 9).

II.7. Pengertian Database

Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi.

Database mempunyai kegunaan dalam mengatasi penyusunan dan penyimpanan data, maka seringkali masalah yang dihadapi adalah:

1. Redudansi dan Inkonsistensi data
2. Kesulitan dalam pengaksesan data
3. Isolasi data untuk standarisasi
4. Multi user
5. Keamanan data
6. Integritas data
7. Kebebasan data (Asrianda ; 2008 : 1)

Sebuah *database* harus dibuat dengan rapi agar 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).

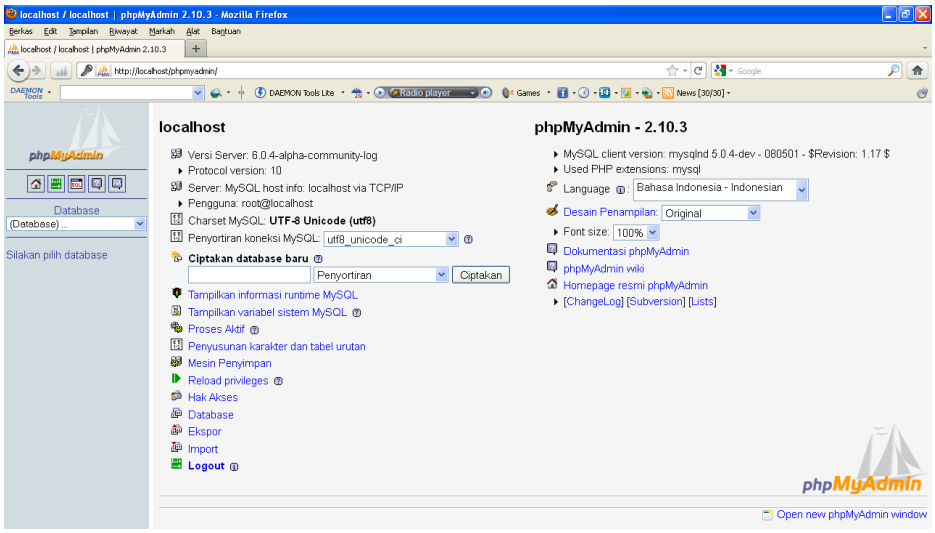
II.8. Pengertian MySQL

Mysql pertama kali dirintis oleh seorang programmer *database* bernama Michael Widenius, yang dapat anda hubungi di emailnya monty@analytikerna.

Mysql database server adalah RDBMS (*Relasional Database Management System*) yang dapat menangani data yang bervolume besar. meskipun begitu, tidak menuntut resource yang besar. Mysql adalah *database* yang paling popular diantara *database* yang lain.

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multiuser*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. Penulis sendiri dalam menjelaskan buku ini menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*General Public License*), yang dapat anda *download* pada alamat resminya http://www.mysql.com. MySQL sudah cukup lama dikembangkan, beberapa fase penting dalam pengembangan MySQL adalah sebagai berikut :

- MySQL dirilis pertama kali secara internal pada 23 Mei 1995
- Versi windows dirilis pada 8 Januari 1998 untuk windows 95 dan windows NT.
- Versi 3.23 : beta dari Juni 2000, dan dirilis pada January 2001.
- Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (unions) (Wahana Komputer ; 2010 : 5).



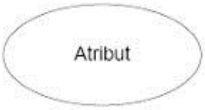
Gambar II.2. Tampilan Awal MySQL
(Sumber : Wahana Komputer ; 2010 : 1)

II.9. *EntityRelationship Diagram* (ERD)

EntityRelationship 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 (Ir. Yuniar Supardi ; 2010 : 448).

Tabel II.1. Simbol ERD

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

(Sumber : Yuniar Supardi ; 2010 : 448)

II.10. Kamus Data

Karena DBMS menyimpan kumpulan beberapa item data yang terpisah yang dapat digunakan pemakai pada beberapa aplikasi secara bersama-sama, adalah penting bahwa beberapa mekanisme digunakan untuk menyediakan informasi mengenai beberapaitem data bersangkutan. Itu adalah fungsi dari kamus data.

Kamus data adalah suatufile yang terpisah yang menyimpan informasi seperti :

a. Nama setiap item/jenis/kolom data.

b. Struktur data untuk tiap item.

c. Program yang menggunakan tiap item

d. Tingkat keamanan untuk setiap item.

Pemakai yang perlu memperoleh informasi dari *database* dapat menuju ke kamus data untuk mendapatkan nama dari item data yang digunakan pada penelusuran (*search*). Dan yang mendesain aplikasi dapat menggunakan kamus untuk menentukan apakah item data sudah disimpan di komputer, dan kalau sudah dengan nama apa item data tersebut dapat dipanggil dan aplikasi apa yang digunakan.

Kamus data berguna khusus bagi perlindungan timbulnya kelebihan data. Tanpa kamus data, pemakai dari lain bagian mungkin menyimpan versi identik dari item data yang sama pada beberapa lokasi, dimana masing-masing item data mempunyai nama yang berbeda.

Operasional komputer dalam organisasi dewasa ini banyak yang sudah menggunakan model kerja jaringan (*network*). Dengan menggunakan data dasar yang sama untuk keperluan informasi masing-masing unit dan manajemen organisasi secara keseluruhan (Zulkifli Amsyah ; 2005 : 382).

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 (1" Normal Form)**

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

   a. Tidak ada atribut mempunyai nilai berulang atau nilai array
   b. Tidak mempunyai baris yang rangkap

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

   Contohnormalisasi 1NF adalahsepertipadatabelberikut :

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 :HaidarDzacko ; 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:

   a. Key terdiri dari atribut tunggal
   b. 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 keycomposite, atribut nonkey CNAME tergantung hanya pada CNO, bukan pada SECNO, sehingga CNAME tidak secara ketergantungan fungsional penuh terhadap key (CNO, SECNO).

Tabel II.3. Normalisasi2NF

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.3. 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 :HaidarDzacko ; 2007 : 16).

4. BoyceCode Normal Form (BCNF)

BCNF menentukan setiap determinan adalah kunci kandidat (*candidatekey*). 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.4. NormalisasiBCNF

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 (*multivaluedependency*). Konsep pada bentuk ini adalah ketergantungan pada gabungan beberapa atribut (*joindependency*) (HaidarDzacko ; 2007 : 12).

Tabel II.5. 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 :HaidarDzacko ; 2007 : 20).

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

UML singkatan dari *UnifiedModellingLangguage* yang berarti bahasa pemodelan standart. (Chonoles; 2003 : 6) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan

konsep *UML* ada aturan–aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam investasi perbankan,lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorentasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensuport para pengembang sistem saat ini. Sebagai perancang

sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Herlawati ; 2011 ; 6).

II.12.1. *Diagram-Diagram UML*

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas (*Class Diagram*). Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umu dijumpai pada pemodelan sistem berorentasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
2. Diagram paket (*PackageDiagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *UseCase*bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adal;ah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram)* bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State)*, transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutam penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram aktivitas (*Activity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram)* bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*runtime)*. Memuat

simpul-simpul berserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing).*

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan.

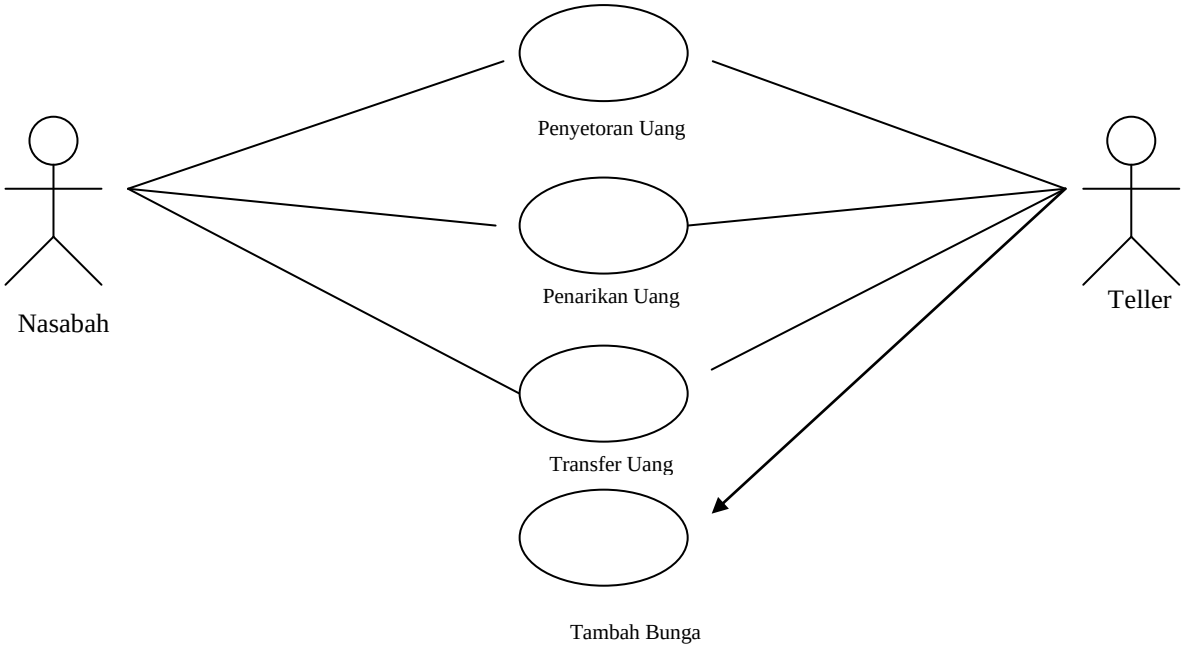
Diagram UseCase (usecase diagram)

UseCase menggambarkan *externalview* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *usecase*dapat dijabarkan dalam diagram, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram.

komponen pembentuk diagram *usecase*adalah :

a. Aktor (*actor),* menggambarkan pihak-pihak yang berperan dalam sistem.
b. *UseCase,* aktivitas/ sarana yang disiapkan oleh bisnis/sistem.
c. Hubungan (*Link),* aktor mana saja yang terlibat dalam *usecase*ini.

Gambar di bawah ini merupakan salah satu contoh bentuk diagram *usecase.*

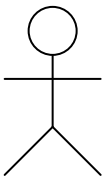


Gambar II.3. Diagram *UseCase*

(Sumber :ProbowoPudjo Widodo ; 2011:17)

1. Aktor

Menurut Chonoles (2003 :17) menyarankan sebelum mebuatusecase dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.

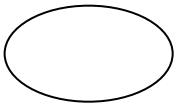


Gambar II.4. Aktor

(Sumber :ProbowoPudjo Widodo ; 2011:17)

2. *UseCase*

Menurut Pilone (2005 : 21) *usecase* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan *usecase* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Usecase* digambarkan dalam bentuk *ellips*/oval



Gambar II.5. Simbol *UseCase*

(Sumber :ProbowoPudjo Widodo ; 2011:22)

Usecase sangat menentukan karakteristik sistem yang kita buat, oleh karena itu Chonoles (2003:22-23) menawarkan cara untuk menghasilkan *usecase*yang baik yakni :

a. Pilihlah nama yang baik

Usecase adalah sebuah *behaviour*(prilaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya lebih detil tambahkan kata benda mengindikasikan dampak aksinya terhadap suatu kelas objek. Oleh karena itu diagram *usecase* seharusnya berhubungan dengan diagram kelas.

b. Ilustrasikan perilaku dengan lengkap.

Usecase dimulai dari inisiasi oleh aktor primer dan berakhir pada aktor dan menghasilkan tujuan. Jangan membuat *usecase* kecuali anda mengetahui tujuannya. Sebagai contoh memilih tempat tidur (*King Size, Queen Size,* atau dobel) saat tamu memesan tidak dapat dijadikan *usecase* karena merupakan bagian dari *usecase* pemesanan kamar dan tidak dapat berdiri sendiri (tidak mungkin tamu memesan kamar tidur jenis king tapi tidak memesan kamar hotel).

c. Identifikasi perilaku dengan lengkap.

Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, *usecase* harus lengkap. Ketika memberi nama pada *usecase*, pilihlah frasa kata kerja yang implikasinya hingga selesai. Misalnya gunakan frasa *reserve a room* (pemesanan kamar) dan jangan *reserving a room* (memesan kamar) karena memesan menggambarkan perilaku yang belum selesai.

d. Menyediakan usecase lawan (*inverse)*

Kita biasanya membutukan *usecase* yang membatalkan tujuan, misalnya pada *usecase*pemesanan kamar, dibutuhkan pula *usecase* pembatalan pesanan kamar.

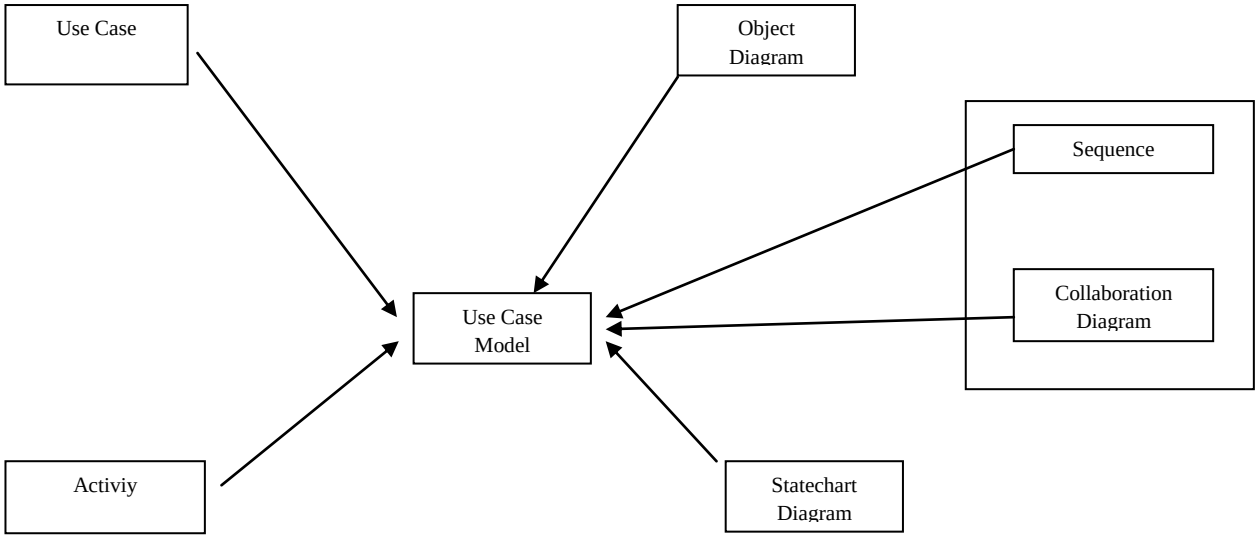
e. Batasi usecase hingga satu perilaku saja.

Kadang kita cenderung membuat *usecase* yang lebih dari satu tujuan aktivitas. Guna menghindari kerancuan, jagalah usecase kita hanya fokus pada satu hal. Misalnya, penggunaan *usecase check in* dan *check out*

dalam satu *usecase* menghasilkan ketidakfokusan, karena memiliki dua perilaku yang berbeda.

3. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model (ProbowoPudji Widodo; 2011 : 37)



Gambar II.6. Hubungan Diagram Kelas Dengan Diagram *UML*lainya

(Sumber :ProbowoPudjo Widodo ; 2011 : 38)

4. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistemdari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukan aktivitas

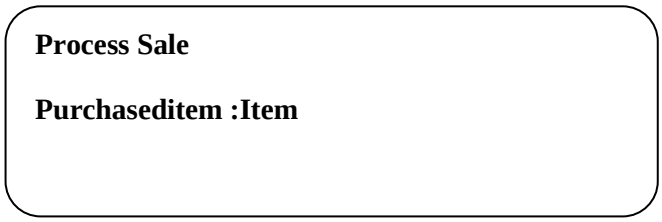
sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call.* Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (ProbowoPudji Widodo ;2011 : 143-145).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi nelakukan langka sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

a. Fungsi Matematika
b. Pemanggilan Perilaku
c. Pemrosesan Data

Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classfier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classfier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.

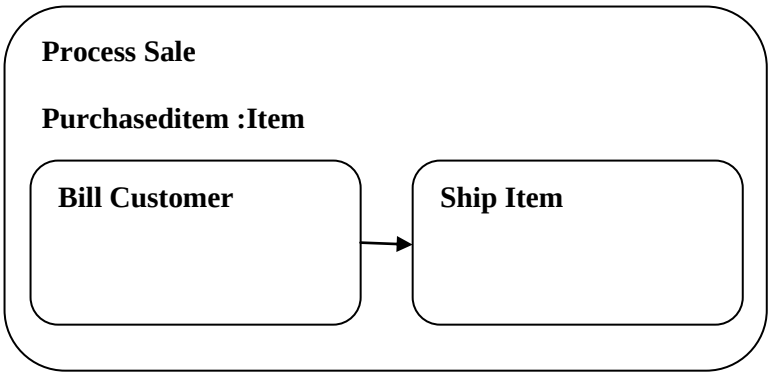
Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.7. Aktivitas serderhana tanpa rincian

(Sumber : ProbowoPudjo Widodo ; 2011:145)

Detail aktivitas dapat dimasukan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.



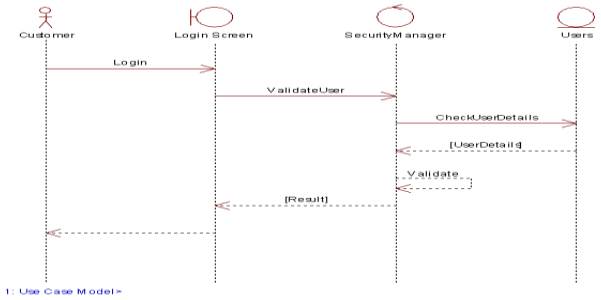
Gambar II.8. Aktivitas dengan detail rincian

(Sumber : ProbowoPudjo Widodo ; 2011:145)

5. *Squence Diagram*

Menurut Douglas (2004 : 174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram),* diagram waktu (*timing diagram)* dan diagram komunikasi (*communication diagram).*

Menurut Pilone (2005 : 174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.9. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya.



Gambar II.9. Diagram Urutan

(Sumber : ProbowoPudjo Widodo ; 2011:175)