

BAB II

LANDASAN TEORI

II.1 Teori Sistem

Menurut Kusrini dan Koniyo (2007 : 5), kata sistem mempunyai beberapa pengertian, tergantung dari sudut mana kata tersebut didefinisikan. Secara garis besar ada dua pendekatan yang dilakukan yaitu :

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang didalam hal ini sistem ini didefinisikan sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu.
2. Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem. Prosedur didefinisikan sebagai urutan operasi kerja (tuliskan-menulis), yang biasanya melibatkan beberapa orang didalam satu atau lebih departemen yang diterapkan untuk menjamin penanganan yang seragam dari transaksi bisnis yang terjadi.

Pendekatan sistem yang lebih menekankan pada elemen-elemen atau komponennya mendefinisikan sistem sebagai sekumpulan elemen-elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Dengan demikian didalam suatu sistem, komponen-komponen ini tidak dapat berdiri sendiri, tetapi sebaliknya, saling berhubungan hingga berbentuk suatu kesatuan hingga tujuan sistem dapat tercapai.

II.1.1 Karakteristik Sistem

Menurut Kusriani (2007 : 6), sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain :

1. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli seberapa kecilnya, selalu mengandung komponen-komponen atau subsistem.

2. Batasan sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*Environment*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut

4. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui perhubungan ini memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya.

5. Masukan sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem (*Process*)

Suatu sistem yang dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

II.1.2 Klasifikasi sistem

Menurut Kusri (2007 : 7), Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya sebagai berikut :

1. Sistem abstrak dan sistem fisik.

Sistem abstrak adalah sistem yang berisi gagasan atau konsep. Misalnya, sistem teologi yang berisi gagasan tentang hubungan manusia dan Tuhan. Sistem fisik merupakan sistem yang secara fisik dapat dilihat. Misalnya sistem komputer, sistem sekolah, sistem akuntansi, dan sistem transportasi.

2. Sistem alamiah dan sistem buatan manusia.

Sistem alamiah adalah sistem yang terjadi karena alam (tidak dibuat manusia). Misalnya, sistem tata surya. Sistem buatan manusia adalah sistem yang dibuat oleh manusia. Misalnya, sistem komputer dan sistem mobil.

3. Sistem tertentu dan sistem tak tentu.

Sistem tertentu adalah sistem yang operasinya dapat diprediksi secara tepat. Misalnya, sistem komputer. Sistem tak tentu adalah sistem yang tak dapat diramal dengan pasti karena mengandung unsur probabilitas. Misalnya, sistem arisan dan sistem sediaan.

4. Sistem tertutup dan sistem terbuka

Sistem tertutup adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Misalnya, reaksi kimia dalam tabung terisolasi. Sistem terbuka adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan.

II.2 Pengertian Sistem Informasi Geografis

Menurut Edy Irwansyah (2013 : 1), Sistem Informasi Geografis (SIG) atau *Geographic Information System* (GIS) adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, menganalisa, mengatur dan menampilkan seluruh jenis data geografis. Akronim GIS terkadang dipakai sebagai istilah untuk *Geographical Information Science* atau *Geospatial Information Studies* yang merupakan ilmu studi atau pekerjaan yang berhubungan dengan *Geographic Information System*. Dalam artian sederhana sistem informasi geografis dapat kita simpulkan sebagai gabungan *kartografi*, analisis statistik dan teknologi sistem basis data (*database*).

II.2.1 Komponen Sistem Informasi Geografis

Menurut Eddy Irwansyah (2013 : 11), komponen-komponen yang membangun sebuah sistem informasi geografis adalah:

a. Komputer dan Perangkat Lunak

Merupakan sistem komputer dan kumpulan piranti lunak yang digunakan untuk mengolah data.

b. Data Spasial

Merupakan data spasial (bereferensi keruangan dan kebumian) yang akan diolah.

c. Manajemen Data dan Analisa Prosedur

Manajemen data dan analisa prosedur oleh *Database Management System*.

d. Manusia

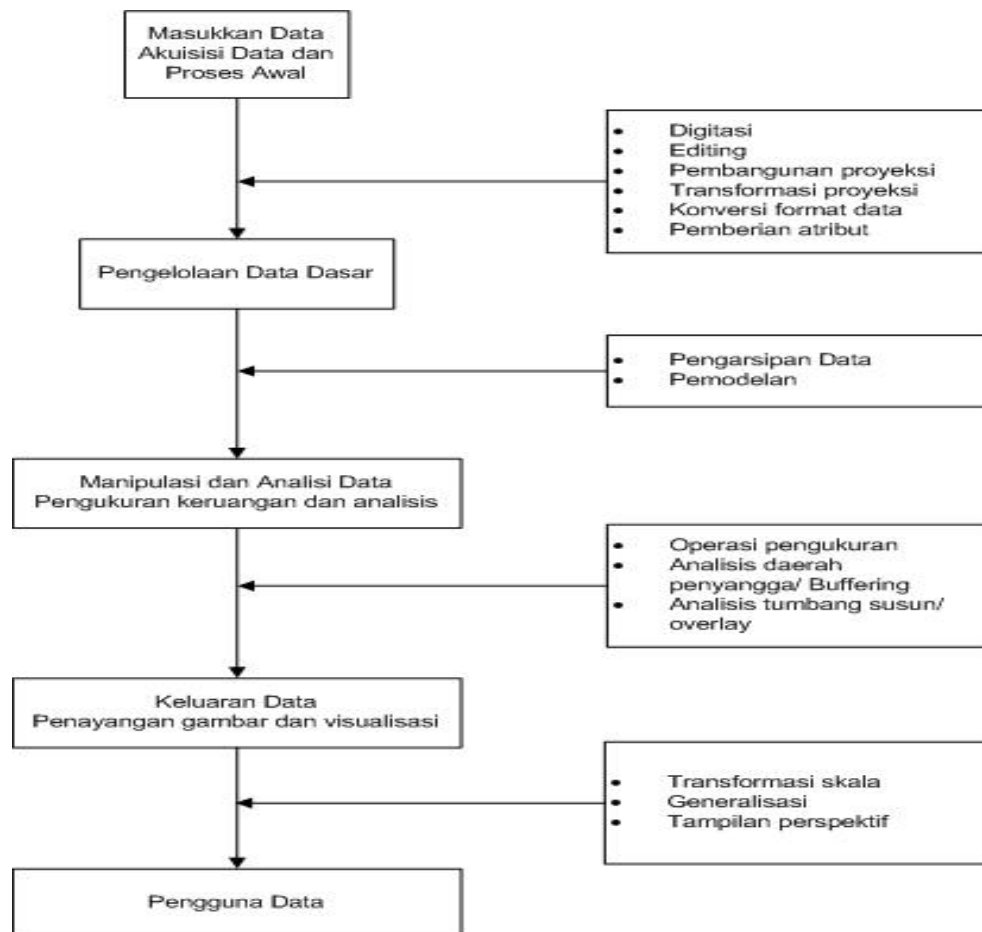
Entitas sumber data manusia yang akan mengoperasikan sistem informasi geografis.

II.2.2 Tujuan Sistem Informasi Geografis

Menurut Dulbahri (1993), Tujuan pokok dari pemanfaatan Sistem Informasi Geografis adalah untuk mempermudah mendapatkan informasi yang telah diolah dan tersimpan sebagai atribut suatu lokasi atau obyek. Ciri utama data yang bisa dimanfaatkan dalam Sistem Informasi Geografis adalah data yang telah terikat dengan lokasi dan merupakan data dasar yang belum dispesifikasi.

II.2.3 Tahapan Kerja dalam Sistem Informasi Geografis

Proses tahapan pengerjaan didalam Sistem Informasi Geografis digambarkan didalam bagan berikut ini :



Gambar II.1. Tahapan Kerja dalam Sistem Informasi Geografis
(Sumber : *Eni A dan Tri H ; 2010*)

Berdasarkan bagan diatas, proses tahap kerjad didalam Sistem Informasi Geografis yaitu :

1. Proses Masukan Data

Proses awal dalam tahapan kerja Sistem Informasi Geografis adalah masukan data yang terdiri atas akuisisi data dan proses awal.

2. Pengelolaan Data

Subsistem selanjutnya adalah pengelolaan data. Dalam subsistem ini dilakukan pengolahan data dasar. Proses-proses yang dilakukan dalam subsistem ini antara lain pengarsipan data dan pemodelan.

3. Manipulasi dan Analisis Data

Melalui proses pemasukan data, peta-peta dasar tersebut diubah menjadi data digital. Setelah dilakukan editing, peta siap digunakan untuk analisis.

4. Keluaran Data

Suatu skala peta sering ditentukan berdasarkan kebutuhan pengguna peta dan media cetak peta. Proses penentuan skala ini bisa dilakukan dengan menggunakan *software* untuk layout peta.

II.3 Pengertian Peta

Menurut Bramantiyo Marjuki (2013 : 6), 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.

II.3.1 Proyeksi Peta

Menurut Bramantiyo Marjuki (2013 : 6), Pada dasarnya bentuk bumi tidak datar tapi mendekati bulat, maka untuk menggambarkan sebagian muka bumi untuk kepentingan pembuatan peta, perlu dilakukan langkah-langkah agar bentuk yang mendekati bulat tersebut dapat didatarkan dan distorsinya dapat terkontrol, untuk itu dilakukan proyeksi ke bidang datar.

II.4 Pengertian Google Maps API

Seperti yang tercatat oleh Gabriel Svennerberg (2010 : 1), *Google Maps API* yang paling populer di internet. Pencatatan yang dilakukan pada bulan Mei 2010 ini menyatakan bahwa 43% aplikasi dan situs *web* yang menggabungkan dua atau lebih sumber data menggunakan *Google Maps API*. Beberapa tujuan dari penggunaan *Google Maps API* adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya. Hampir semua hal yang berhubungan dengan peta dapat memanfaatkan *Google Maps*. *Google Maps* diperkenalkan pada Februari 2005 dan merupakan revolusi bagaimana peta di dalam *web*, yaitu dengan membiarkan user untuk menarik peta sehingga dapat menavigasinya. Solusi peta ini pada saat itu masih baru dan membutuhkan *server* khusus. Beberapa saat setelahnya, ada yang berhasil membobol *Google Maps* untuk digunakan di dalam webnya sendiri. Hal ini membuat *Google Maps* mengambil kesimpulan bahwa mereka membutuhkan API dan pada Juni 2005, *Google Maps API* dirilis secara publik.

II.5 Pengertian Adobe Dreamweaver CS4

Menurut Wahana Komputer (2010 : 2), Adobe Dreamweaver CS4 merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah website, baik secara grafis maupun dengan menuliskan kode sumber secara langsung.

Adobe Dreamweaver CS4 memudahkan pengembang website untuk mengelola halaman-halaman website dan aset-asetnya, baik gambar, animasi *flash*, video, suara dan lain sebagainya. Selain itu, Adobe Dreamweaver CS4 juga menyediakan fasilitas untuk melakukan pemrograman *scripting*, baik ASP(*Active Server Page*), JSP(*Java Server Page*), PHP (*PHP Hypertext Preprocessor*), *JavaScript*, *Cold Fusion*, CSS (*Cascading Style Sheet*), XML (*Extensible Markup Language*) dan lainnya.

II.6 PHP (*PHP Hypertext Preprocessor*)

PHP merupakan singkatan dari *Hypertext Preprocessor*. Menurut Mochamad Joko Adi Wirawan (2008 : 3), PHP merupakan bahasa pemrograman yang paling populer dan banyak digunakan untuk pemrograman web. PHP disebut juga pemrograman *server side* yang artinya program dijalankan pada *server*.

Keuntungan menggunakan PHP antara lain :

1. Kesederhanaan dari bahasa PHP.
2. Siklus pengembangan yang lebih pendek karena pemudahan dalam pembuatan model dan komponen-komponen yang dapat digunakan lagi pada pengembangan berikutnya.
3. Memiliki konektivitas ke *server* basis data.
4. Bersifat *open source* dan tidak bergantung pada *platform* manapun.

II.7 Pengertian Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulan-kumpulan *file* yang saling berkaitan.

Menurut Kusrini (2010 : 2), pengertian Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol).

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

II.7.1. Tujuan Basis Data

Menurut Kusrini (2010 : 2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basisdata yang baik adalah sebagai berikut :

1. Tidak adanya redudansi dan inkonsistensi data

Redudansi terjadi jika suatu informasi disimpan di beberapa tempat. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

2. Kesulitan pengaksesan data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan *query* ataupun dari *tool* yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari *software Database Management Sytem*.

3. *Multiple user*

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian *server* yang bisa diakses dari banyak *client*, sudah menyediakan akses kesemua pengguna dari komputer *client* ke sumber informasi yaitu basis data.

II.7.2. Manfaat/Kelebihan Basis Data

Menurut Kusri (2010 : 5), Banyak manfaat yang diperoleh dengan menggunakan basis data, manfaat/kelebihan basis data dan kelebihan basis data diantaranya adalah :

1. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan matematika. Dengan

perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

2. Kebersamaan pemakai (*sharability*)

Sebuah basis data dapat digunakan oleh banyak *user* dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencacatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama.

3. Pemusatan kontrol data

Karena cukup satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

4. Efisiensi ruang penyimpanan

Dengan pemakaian bersama, tidak perlu menyediakan tempat penyimpanan diberbagai tempat tetapi cukup satu saja, sehingga ini dapat menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

5. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain, dapat menekan ketidakakuratan dalam pemasukan/penyimpanan data.

6. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat dibackup, memilah-milah data mana yang masih diperlukan yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi sebuah organisasi dari lain waktu ke waktu membutuhkan penyimpanan yang semakin besar.

7. Keamanan (*Security*)

Kebanyakan *Database Management System* dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan *password* untuk membatasi orang yang diaksesnya.

8. Kemudahan dalam pembuatan program aplikasi baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari *Database Management System*. Sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur *interface* untuk pengguna.

9. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan tools yang disediakan oleh *Database Management System*.

10. Kebebasan data

Perubahan dapat dilakukan pada level *Database Management System* tanpa harus membongkar kembali program aplikasinya.

11. *User View*

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

II.7.3. Operasi Dasar Database

Menurut Kusri (2010 : 9), Beberapa operasi dasar basis data yaitu :

1. Pembuatan basis data
2. Penghapusan basis data
3. Pembuatan *file*/tabel

4. Penghapusan *file*/tabel
5. Pengubahan tabel
6. Penambahan/pengisian
7. Pengambilan data

II.8. Mysql

Menurut Wahana Komputer (2010 : 5), Mysql adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. Mysql memiliki dua bentuk lisensi yaitu *free software* dan *shareware*. Mysql sudah cukup lama dikembangkan, beberapa fase penting dalam pengembangan mysql adalah sebagai berikut :

1. Mysql dirilis pertama kali secara internal pada 3 Mei 1995
2. Versi Windows dirilis pada 8 Januari 1998 untuk windows 95 dan windows 1995.
3. Versi 2.32, beta dari Juni 2000, dan dirilis pada Januari 2001.
4. Versi 4.0 : Beta dari bulan Juni 2004, dirilis pada bulan Oktober 2005
5. Versi 5.0 : Beta dari Bulan Maret 2005, dirilis pada Bulan Oktober 2005
6. Sun Microsystem membeli Mysql AB pada tanggal 26 Pebruari 2008.
7. Versi 5.1 : Dirilis 27 November 2008

Mysql *database server* adalah *RDBMS (Relational Database Management Sistem)* yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menuntut *resource* yang besar. Mysql adalah *database* yang paling populer diantara *database* lainnya.

II.8.1 Kelebihan dan Keuntungan Memakai Mysql

Menurut Wahana Komputer (2010 : 6), dalam dunia *programming* ada beberapa *database* yang sering digunakan antara lain *Mysql*, *Oracle*, *PostgreSQL*, *MSql*, *Microsoft Sql Server* dan lain-lain. Ketika dibandingkan antara *Mysql* dengan sistem manajemen *database* yang lain, *Mysql* memiliki beberapa kelebihan dan keuntungan dibandingkan *database* lain diantaranya adalah :

1. Banyak ahli berpendapat bawah *Mysql* merupakan *server* tercepat
2. *Mysql* merupakan sistem manajemen *database* yang *open source* yaitu *software* ini bersifat *free* atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
3. *Mysql* mempunyai performa yang tinggi namun *simple*.
4. *Database Mysql* mengerti bahasa SQL (*Structured Query Language*)
5. *Mysql* dapat diakses melalui protocol ODBC (*Open Database Connectivity*) buatan Microsoft. Ini menyebabkan *Mysql* dapat diakses oleh banyak *software*.
6. Semua *client* dapat mengakses *Server* dalam satu waktu tanpa harus menunggu yang lain mengakses *database*.
7. *Mysql* dapat diakses dari semua tempat di internet dengan hak akses tertentu
8. *Mysql* merupakan *database* yang mampu menyimpan data berkapasitas besar, sampai berukuran *gigabyte*.
9. *Mysql* dapat berjalan diberbagai *operating sistem* seperti Linux, Windows, Solaris dan lain-lain.

II.9 Unified Modeling Language (UML)

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

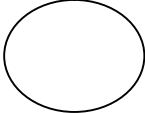
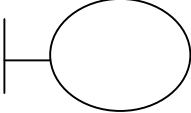
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

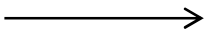
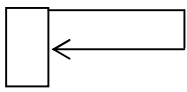
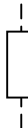
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

a. 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.1. 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)

b. *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*

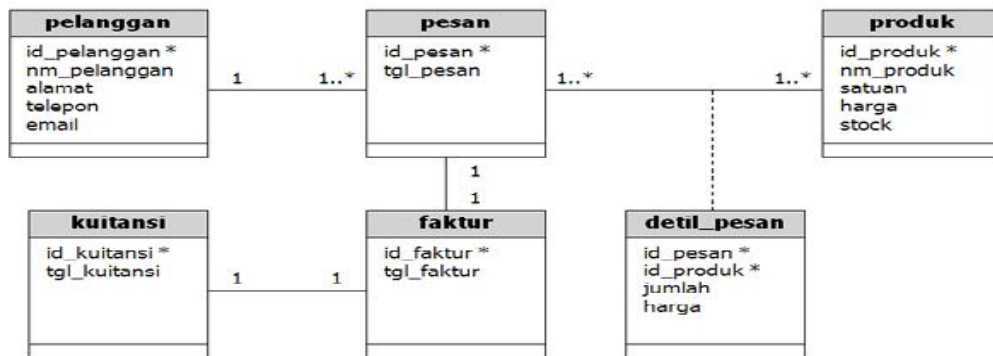
Tabel II.2. *Multiplicity Class Diagram*

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 : 8)

Contoh dari pembuatan use case diagram dapat dilihat pada gambar II.2 berikut :



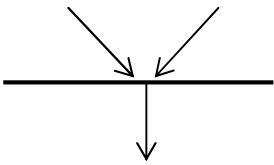
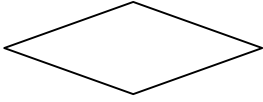
Gambar. II.2 Contoh Class Diagram
(Sumber : Windu Gata ; 2013 : 8)

c. 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.3. Simbol Activity Diagram

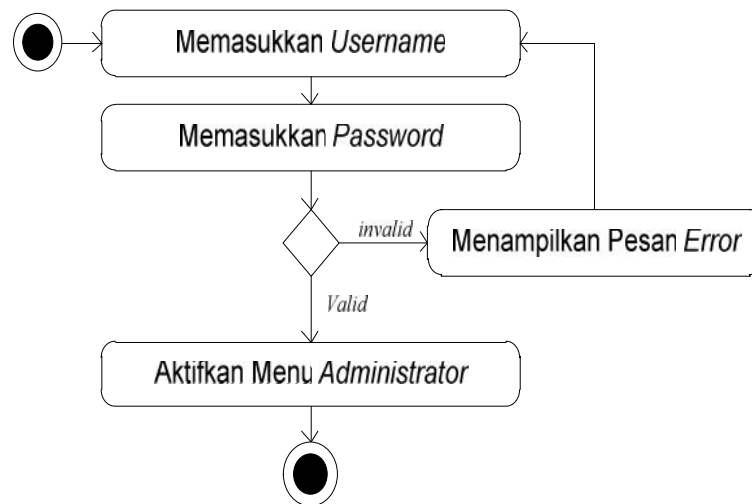
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau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

	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
New Swimlane	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata ; 2013 : 6)

Contoh dari pembuatan activity diagram dapat dilihat pada gambar II.3

berikut :

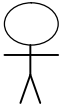
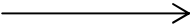
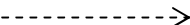
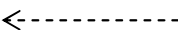


Gambar. II.3 Contoh Activity Diagram
(Sumber : Windu Gata ; 2013 : 6)

d. *Use case Diagram*

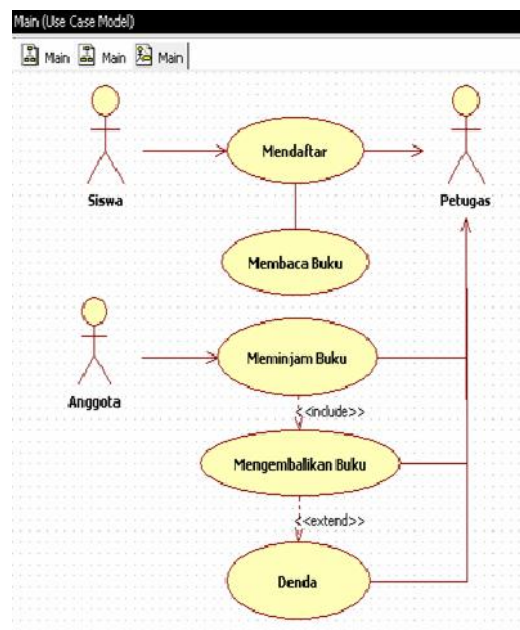
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam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

Tabel II.4. Simbol Use Case Diagram

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 mengidentifikasikan 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 kontrol 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)

Contoh dari pembuatan use case diagram dapat dilihat pada gambar II.4 berikut :



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