

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Pendekatan sistem memberikan banyak manfaat dalam memahami lingkungan. Pendekatan sistem berusaha menjelaskan sesuatu yang dipandang dari sudut sistem serta berusaha menemukan struktur unsur yang membentuk sistem tersebut. Dengan memahami struktur sistem dan proses sistem, seseorang akan dapat menjelaskan mengapa tujuan suatu sistem tidak tercapai. Orang yang ahli pada dasarnya selalu mendekati masalah yang dijumpainya berdasarkan pendekatan sistem. Dari uraian di atas dapat diambil kesimpulan bahwa suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Tata Sutabri ; 2012 : 5).

Model umum sebuah sistem terdiri dari input, proses dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sistem juga memiliki karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai sebuah sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan.

2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan luar sistem.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem.

6. Keluaran Sistem (*Output*)

Hasil energi yang diolah menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

7. Pengolah Sistem (Proses)

Sebuah sistem mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki sasaran dan tujuan yang pasti. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan (Tata Sutabri ; 2012 : 13).

II.2. Pengertian Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya (Tata Sutabri ; 2012 : 22).

Kualitas informasi tergantung dari 3 hal yaitu :

1. Akurat

Informasi harus bebas dari kesalahan-kesalahan yang menyesatkan.

2. Tepat Waktu

Informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan dalam mengambil keputusan.

3. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk orang satu dengan yang lainnya adalah berbeda (Tata Sutabri ; 2012 : 33-34).

II.3. Pengertian Sistem Informasi

Sistem informasi bukan merupakan hal yang baru, yang baru adalah komputernya. 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 dengan laporan-laporan yang diperlukan (Tata Sutabri ; 2012 : 38).

II.4. Sistem Informasi Geografis

Sistem Informasi Geografis atau disingkat SIG dalam bahasa Inggris *Geographic Information System* (disingkat GIS) merupakan sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Atau dalam arti yang lebih sempit adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis atau data geospasial untuk mendukung pengambilan keputusan dalam perencanaan dan pengelolaan suatu wilayah, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah *database*. Para praktisi juga memasukkan orang yang membangun dan mengoperasikannya dan data sebagai bagian dari sistem ini.

Teknologi Sistem Informasi Geografis dapat digunakan untuk investigasi ilmiah, pengelolaan sumber daya, perencanaan pembangunan, kartografi dan perencanaan rute. Misalnya, SIG bisa membantu perencana untuk secara cepat menghitung waktu tanggap darurat saat terjadi bencana alam, atau SIG dapat

digunakan untuk mencari lahan basah (*wetlands*) yang membutuhkan perlindungan dari polusi atau dapat digunakan mencari informasi sebuah tempat khusus dan banyak manfaat lain yang dapat dikembangkan dalam sistem informasi geografis ini (Adam Suseno dan Ricky Agus, ST ; 2012 : 2).

SIG (Sistem Informasi Geografis) merupakan sistem informasi berbasis komputer yang menggabungkan antara unsur peta (geografis) dan informasinya tentang peta tersebut (data atribut) yang dirancang untuk mendapatkan, mengolah, memanipulasi, analisa, memperagakan dan menampilkan data spasial untuk menyelesaikan perencanaan, mengolah dan meneliti permasalahan. Data yang diolah pada SIG ada 2 macam yaitu data data spasial dan data non-spasial. Data spasial adalah data yang berhubungan dengan kondisi geografi misalnya sungai, wilayah administrasi, gedung, jalan raya dan sebagainya. Data spasial didapatkan dari peta, foto udara, citra satelit, data statistik dan lain-lain. Hingga saat ini secara umum persepsi manusia mengenai bentuk representasi *entity* spasial adalah konsep raster dan vector. Sedangkan data non-spasial adalah selain data spasial yaitu data yang berupa text atau angka. Biasanya disebut dengan atribut. Data non-spasial ini akan menerangkan data spasial atau sebagai dasar untuk menggambarkan data spasial. Dari data non-spasial ini nantinya dapat dibentuk data spasial. Misalnya jika ingin menggambarkan peta penyebaran penduduk maka diperlukan data jumlah penduduk dari masing-masing daerah (data non-spasial), dari data tersebut nantinya kita dapat menggambarkan pola penyebaran penduduk untuk masing – masing daerah (Arna Fariza, S.Kom, et al ; Jurnal

Analisa Pesebaran Agen LPG dan SPPBE Berbasis GIS Di Kotamadya Surabaya :2012 : 2).

Sistem Informasi Geografis terdiri dari beberapa komponen utama yang saling berinteraksi untuk merealisasikan suatu tujuan yang ingin dicapai.

Komponen–komponen tersebut adalah sebagai berikut :

1. Perangkat Keras (*Hardware*)

Perangkat keras yang sering digunakan untuk SIG adalah komputer PC (*Personal Computer*). Perangkat keras tambahan berupa perangkat untuk pemasukan data (*input*) seperti *scanner*, *digitizer*, pemrosesan data, media penyimpanan data, dan perangkat untuk mencetak data (*output*) seperti layar monitor, *plotter*, printer dan sebagainya.

2. Perangkat Lunak (*Software*)

Perangkat lunak merupakan komponen untuk pengolahan basis data (*database*), pemrosesan dan analisa hasil keluaran (*output*). Saat ini sudah banyak perangkat lunak (*software*) yang dibuat untuk digunakan dalam proses pengolahan data (spasial dan non-spasial) pada SIG, antara lain: Arc View, Map Info, Arc GIS, QGIS, MySQL, dan lain-lain.

3. Intelegensi Manusia (*Brainware*)

Brainware merupakan kemampuan manusia dalam membangun, mengelola, dan memanfaatkan SIG secara efektif. Bagaimanapun juga manusia merupakan subjek (pelaku) yang mengendalikan seluruh sistem. Selain itu diperlukan pula kemampuan untuk memadukan pengelolaan dengan pemanfaatan SIG, agar SIG dapat digunakan secara efektif dan efisien.

4. Data

SIG dapat mengumpulkan dan menyimpan data dan informasi yang diperlukan baik secara langsung maupun tidak langsung yaitu dengan cara mengimportnya dari perangkat lunak SIG, maupun secara langsung dengan cara mendigitasi data spasialnya dari peta dan memasukkan data atributnya dari tabel-tabel melalui keyboard. SIG merupakan perangkat analisis keruangan dengan kelebihan dapat mengelola data spasial dan data nonspasial sekaligus.

- a. Data atribut atau data non-spasial adalah gambaran data yang terdiri atas informasi yang relevan terhadap suatu lokasi, seperti kedalaman, ketinggian, lokasi penjualan dan lain – lain.
- b. Data spasial adalah data yang mempunyai referensi lokasi terhadap referensi tertentu. Jenis data spasial antara lain peta, foto udara, citra satelit atau data lain yang mempunyai referensi. Data spasial ini terbagi menjadi dua jenis data, yaitu data raster dan data vector (Dhimas Van Er Donna ; Jurnal Membangun Sistem Informasi Geografi Tempat Wisata Berbasis Website Di Kabupaten Bogor : 2010 : 10-11).

II.5. Pengertian Quantum GIS

Quantum GIS (QGIS) adalah *cross-platform* perangkat lunak bebas (*open source*) desktop pada sistem informasi geografis (SIG). Aplikasi ini dapat menyediakan data, melihat, mengedit, dan kemampuan analisis. Quantum GIS berjalan pada sistem operasi yang berbeda termasuk Mac OS X , Linux , UNIX , dan Microsoft Windows. Dalam perizinan, QGIS sebagai perangkat lunak bebas

aplikasi di bawah GPL(*General Public License*), dapat secara bebas dimodifikasi untuk melakukan tugas yang berbeda atau lebih khusus. Quantum GIS memungkinkan penggunaan *shapefiles*, pertanggung, dan Geodatabases pribadi. MapInfo, PostGIS dan beberapa format lain yang didukung di Quantum GIS (Adam Suseno dan Ricky Agus, ST ; 2012 : 15).

II.6. Pengertian Google Map API

Google Maps adalah layanan gratis Google yang cukup populer. Kita dapat menambahkan fitur Google Map dalam web kita sendiri dengan Google Map API. Google Map API merupakan library *JavaScript*. Untuk melakukan pemrograman Google Map API yang kita butuhkan adalah pengetahuan tentang HTML dan *JavaScript* serta koneksi Internet. Kita bisa mulai menulis program Google Map API dengan urutan sebagai berikut :

- Memasukkan Maps API *JavaScript* ke dalam HTML kita.
- Membuat *element div* dengan nama *map_canvas* untuk menampilkan peta.
- Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
- Menuliskan fungsi *JavaScript* untuk membuat objek peta.
- Menginisialisasi peta dalam *tag body* HTML dengan *event onload*. (Umi Laili Yohana ; Jurnal Pemanfaatan Googlemaps Untuk Pemetaan Dan Pencarian Data Perguruan Tinggi Negeri Di Indonesia: 2010 : 2).

II.7. Pengertian Adobe Dreamweaver

Adobe Dreamweaver adalah sebuah aplikasi yang dikembangkan oleh *Adobe Corporation* yang ditujukan untuk mempermudah membuat sebuah situs atau web. Aplikasi ini banyak diminati para pembuat web pemula ataupun yang sudah senior karena banyak fitur yang memang memudahkan untuk membuat situs dan web (Christianus Sigit ; 2013 : 1).

Adobe Dreamweaver sendiri merupakan aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain web secara visual. Aplikasi ini juga biasa dikenal dengan istilah WYSIWYG (*What You See Is What You Get*), yang intinya adalah anda tidak harus berurusan dengan tag-tag HTML untuk membuat sebuah site dan dapat melihat hasil desainnya secara langsung.

Dengan kemampuan fasilitas yang optimal dalam jendela *design* akan memberikan kemudahan untuk mendesain meskipun untuk para Web Designer pemula sekalipun. Kemampuan Adobe Dreamweaver untuk berinteraksi dengan beberapa bahasa pemrograman seperti PHP, ASP, JavaScript, dan yang lainnya juga memberikan fasilitas maksimal kepada desainer web dengan menyertakan bahasa pemrograman di dalamnya (Madcoms ; 2013: 3).

II.8. Pengertian PHP

PHP adalah singkatan dari *PHP Hypertext Preprocessor*. Saat pertama kali dikembangkan oleh programmer bernama Rasmus Lerdoff, PHP awalnya adalah singkatan dari *Personal Home Page Tools*. Namun setelah dikembangkan oleh Zeev Suraski dan Andi Gutmans, dan fiturnya bertambah, maka PHP diubah singkatannya menjadi yang sekarang ini.

Salah satu kelebihan PHP adalah kemudahannya untuk berinteraksi dengan *database*. PHP dapat mendukung beberapa *database* secara langsung tanpa harus menginstal konektor seperti halnya bahasa pemrograman Java. Dengan demikian, PHP sangat fleksibel berhubungan dengan berbagai *database*.

PHP merupakan jenis bahasa *scripting* yang lazim digunakan di halaman web. Artinya, kode ini langsung dimasukkan ke dalam kode HTML. Anda menggunakan tag HTML untuk membungkus bahasa PHP yang ada di file HTML. Cara mengedit kode PHP sama seperti cara mengedit kode HTML, yaitu anda perlu menggunakan software editor teks seperti notepad atau notepad ++ (Tim EMS ; 2014 : 59-61).

II.9. Pengertian Database

Database atau basis data 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.

Dengan memahami defenisi di atas maka istilah basis data dapat dipahami sebagai suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data, disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali, dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal, disimpan sedemikian rupa sehingga proses penambahan,

pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol (Edhy Sutanta ; 2011 ; 29-30).

II.10. Pengertian MySQL

MySQL adalah salah satu program RDBMS yang sangat terkenal. Berikut ini beberapa fitur dari MySQL :

1. MySQL adalah sistem *database* yang lazim digunakan di lingkungan web.
2. MySQL adalah sistem yang berjalan di server.
3. MySQL cocok untuk aplikasi besar dan kecil.
4. MySQL cukup cepat, bisa diandalkan, mudah dipakai.
5. MySQL mendukung SQL standar.
6. MySQL bisa dikompilasi pada berbagai *platform*.
7. MySQL mudah didownload dan digunakan.
8. MySQL dikembangkan, didistribusikan dan didukung secara *open source* oleh *Oracle Corporation* (Tim EMS ; 2014 ; 129).

Walaupun MySQL merupakan standar ANSI, tapi ada beberapa versi SQL yang berbeda-beda. Tapi hampir semuanya mendukung perintah seperti SELECT, UPDATE, DELETE, INSERT, dan WHERE dengan sedikit variasi (Tim EMS ; 2014 ; 131).

II.11. Entity Relationship Diagram (ERD)

Entity Relationship Diagram atau ERD merupakan salah satu alat (*tool*) berbentuk grafis yang populer untuk *desain database*. Sebuah ERD tersusun atas

tiga komponen , yaitu entitas, atribut, dan kerelasian antar entitas (Edhy Sutanta ; 2011 ; 92).

1. Entitas (*Entity*)

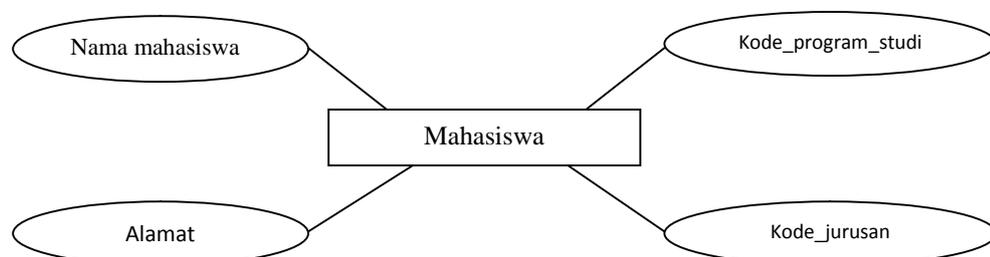
Entitas menunjukkan obyek-obyek dasar yang terkait di dalam sistem. Obyek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan di dalam basis data.

Obyek Dasar	Simbol Entitas
Mahasiswa	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Mahasiswa</div>

Gambar II.1. Contoh Entitas
(*Sumber : Edhy Sutanta ; 2011*)

2. Atribut (*Attribute*)

Merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas.



Gambar II.2. Contoh Atribut pada Entitas Mahasiswa
(*Sumber : Edhy Sutanta ; 2011*)

3. Kerelasian antar entitas (*Relationship*)

Kerelasian antar entitas mendefinisikan hubungan antara dua buah entitas. Kerelasian adalah kejadian atau transaksi yang terjadi diantara dua buah entitas yang keterangannya perlu disimpan di dalam basis data.



Gambar II.3. Contoh Relasi antar Entitas Mahasiswa dan Mata_kuliah
(Sumber : Edhy Sutanta ; 2011)

II.12. Teknik Normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan/ mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data.. Format normalisasi terdiri dari lima bentuk, yaitu:

II.12.1. Bentuk-bentuk Normalisasi

a. Bentuk normal pertama (*first norm form/1NF*)

Relasi disebut sebagai 1NF jika memenuhi kriteria sebagai berikut :

- Jika seluruh atribut dalam relasi bernilai atomik
- Jika seluruh atribut dalam relasi bernilai tunggal
- Jika relasi tidak memuat set atribut berulang
- Jika semua *record* mempunyai sejumlah atribut yang sama.

b. Bentuk normal kedua (*second norm form/2NF*)

Relasi disebut sebagai 2NF jika memenuhi kriteria sebagai berikut :

- Jika memenuhi kriteria 1NF
- Jika semua atribut nonkunci FD atau PK.

c. Bentuk normal ketiga (*third norm form/3NF*)

Relasi disebut sebagai 3NF jika memenuhi kriteria sebagai berikut :

- Jika memenuhi kriteria 2NF
- Jika setiap atribut nonkunci tidak TDF (*non transitive dependency*) terhadap PK.

d. Bentuk normal Boyce Code (BCNF)

Relasi disebut sebagai BCNF jika memenuhi kriteria sebagai berikut :

- Jika memenuhi kriteria 3NF
- Jika semua atribut penentu (determinan) merupakan CK

e. Bentuk Normal Keempat (*forth norm form/4NF*)

Relasi disebut sebagai 4NF jika memenuhi kriteria sebagai berikut :

- Jika memenuhi kriteria BCNF
- Jika setiap atribut di dalamnya tidak mengalami ketergantungan pada banyak nilai. Atau dengan kalimat lain, bahwa semua atribut yang mengalami ketergantungan pada banyak nilai adalah bergantung secara fungsional.

f. Bentuk normal kelima (*fifth norm form/5NF*)

Suatu relasi memenuhi kriteria 5NF jika kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang sederhana (Edhy Sutanta ; 2011 ; 174-179).

II.13. *Unified Modeling Language (UML)*

Menurut Windu 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.

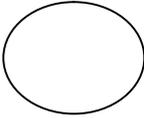
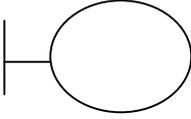
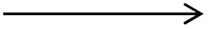
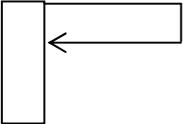
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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

1. Diagram Urutan (*Sequence Diagram*)

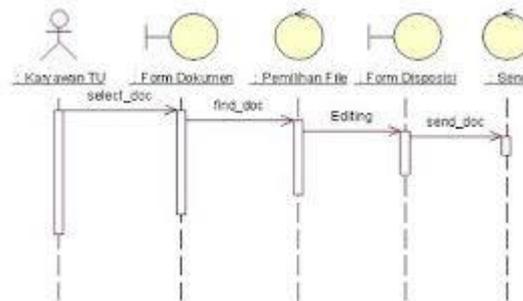
Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

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)

Contoh dari pembuatan *sequence diagram* dapat dilihat pada gambar II.4 berikut :



Gambar. II.4. Sequence Diagram
(Sumber : Windu Gata ; 2013)

2. 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.

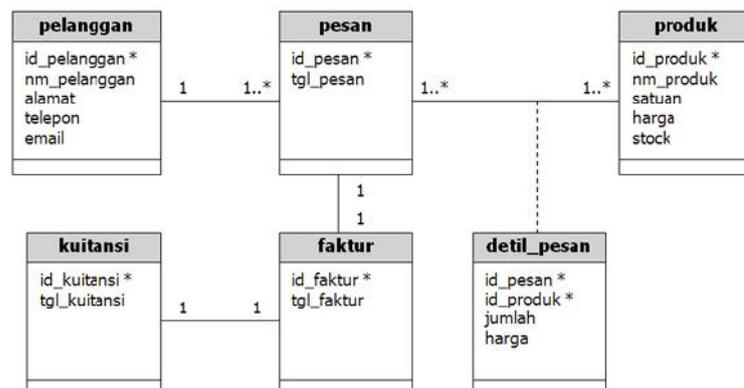
Tabel II.2. Multiplicity Class Diagram

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)

Contoh dari pembuatan *use case diagram* dapat dilihat pada gambar II.5

berikut :



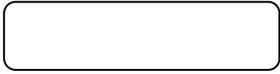
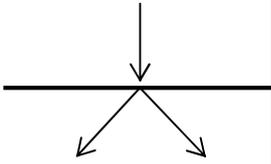
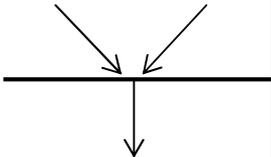
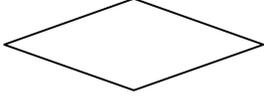
Gambar. II.5. Class Diagram

(Sumber : Windu Gata ; 2013)

3. 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 :

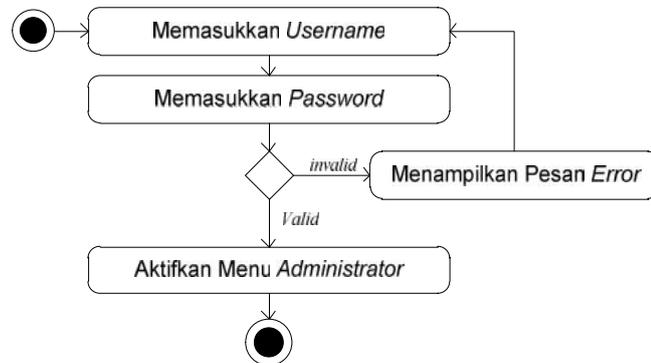
Tabel II.3. Simbol *Activity Diagram*

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 paralel 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 : Windu Gata ; 2013 : 6)

Contoh dari pembuatan *activity diagram* dapat dilihat pada gambar II.6

berikut :

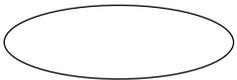
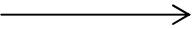
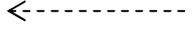


Gambar. II.6. Activity Diagram
(Sumber : Windu Gata ; 2013)

4. 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 :

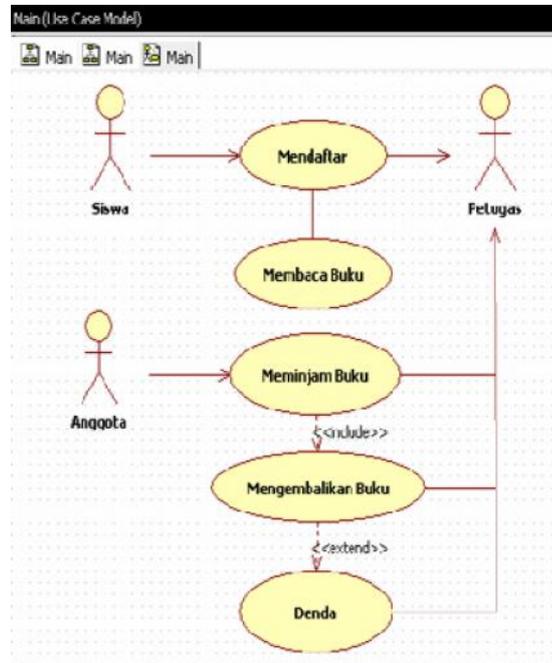
Tabel II.4. Simbol *Use Case*

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 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 : Windu Gata ; 2013)

Contoh dari pembuatan *use case diagram* dapat dilihat pada gambar II.7

berikut :



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