

BAB II

TINJAUAN PUSTAKA

II.1. 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.2. Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau *Geographic Informasi 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 statistic dan teknologi sistem basis data (database).
(Edy Irwansyah ; 2013 : 1).

II.3. Pengertian Web

World Wide Web (WWW) atau yang biasa dikenal sebutan *website* merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. Secara singkat pengertian *web* adalah kumpulan halaman yang menyediakan informasi. *Web* berisi beragam informasi bagia pengguna baik itu informasi yang tidak serius hingga serius maupun informasi gratis hingga berbayar. Berikut bebrapa istilah seputar *website* yang penting untuk diketahui.

1. Penjelajah *web*

Untuk membuka *web page*, kita memerlukan sebuah *web browser* yang dapat digunakan untuk melakukan penjelajahan difunia maya seperti google chrome, safari, mozzila firefox dan opera.

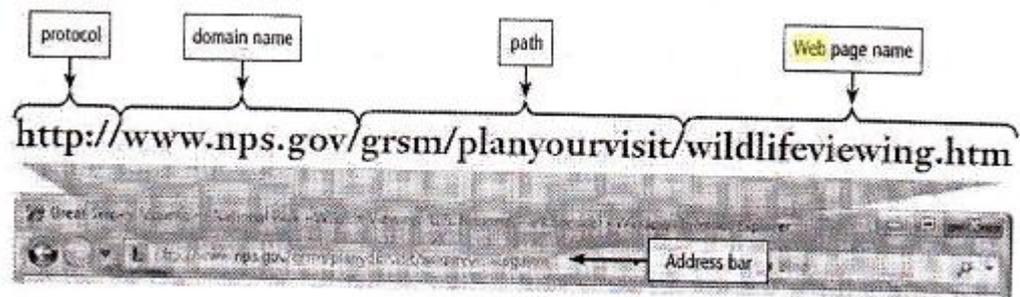


Gambar II.1 Web Browser Populer
(Sumber : *Edy Irwansyah ; 2014 : 34-36*)

2. Alamat Web (*Web Addresses*)

Web page punya sebuah alamat yang unik yang disebut sebagai URL (*Uniform Resource Locator*). URL ini yang dimaksud dengan *web*

address. Setiap halaman mempunyai URL yang berbeda. URL ini bisa terdiri dari perusahaan, militer, pemerintah, edukasi dan lainnya.



Gambar II.2. Alamat Web
(Sumber : *Edy Irwansyah ; 2014 : 34-36*)

3. Navigasi Web

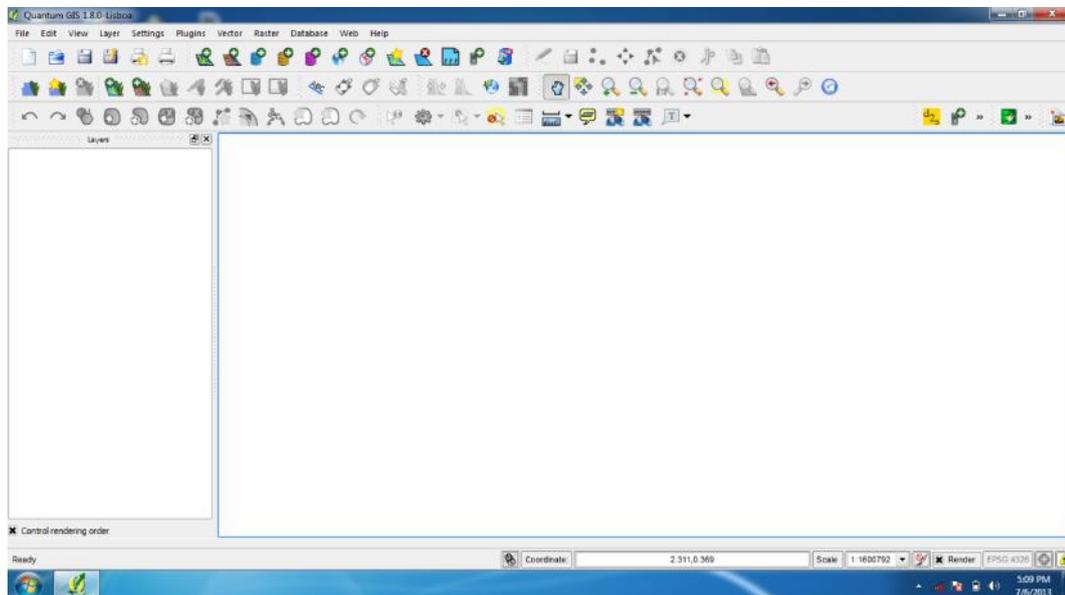
Pada umumnya isi dari *web page* berupa *Hypertext* atau *Hypermedia* yang mengandung *link*, audio, gambar dan video. *Hyperlink* biasanya ditandai dengan kata yang bergaris bawah atau bisa juga berupa gambar. Kebanyakan dari *web browser* menfukung fitur *tabbed browsing* dimana *browser* dapat membuka banyak *tab*.

4. Pencarian Web

Kita dapat menggunakan *search engine*. Search engine membutuhkan sebuah kata kunci untuk mencari isi dari web page. Selain web pages, kita juga dapat mencari gambar, video dan masih banyak multimedia lainnya. Saat www.google.com merupakan search engine yang banyak digunakan oleh pengguna internet diseluruh dunia.(Edy Irwansyah ; 2014 : 34-36).

II.4. Pengertian Quantum GIS

Quantum GIS merupakan salah satu perangkat lunak *open source* di bawah proyek resmi dari *Open Source Geospatial Foundation* (OSGeo) yang dapat dijalankan dalam sistem operasi Windows, Mac OSX, Linux dan Unix. Aplikasi ini menawarkan pengolahan data geospasial dengan berbagai format dan fungsionalitas vektor, raster dan database. Untuk keperluan analisis spasial, aplikasi ini telah cukup lengkap karena telah terintegrasi dengan perangkat lunak GRASS. Pemanfaatan perangkat lunak Quantum GIS ini dapat digunakan sebagai pilihan alternatif dari software SIG komersial seperti ArcView maupun ArcGIS. Quantum GIS dapat diakses melalui situs resmi yang beralamatkan www.qgis.org (Saddam Hussein ; 2012 : 93).



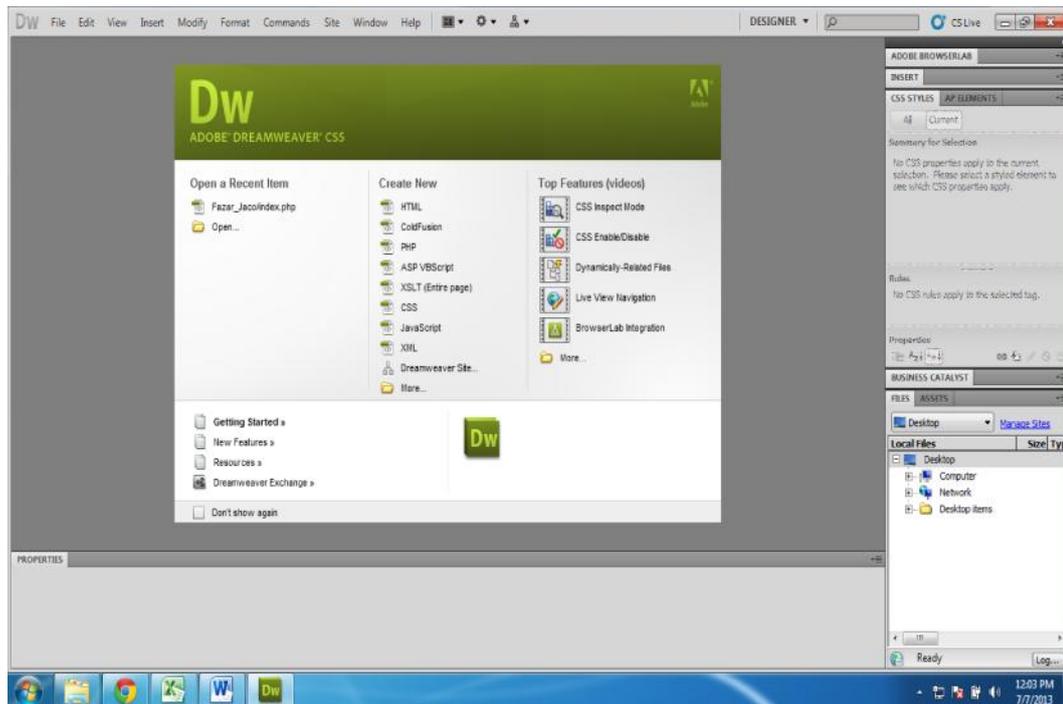
Gambar II.1. Tampilan Quantum GIS
(Sumber : Saddam Hussein ; 2012 : 93)

II.5. Pengertian Macromedia *Dreamweaver*

Macromedia *Dreamweaver* adalah sebuah HTML editor profesional untuk mendesain secara visual dan mengelola situs web maupun halaman web. Bilamana kita menyukai untuk berurusan dengan kode-kode HTML secara manual atau lebih menyukai bekerja dengan lingkungan secara visual dalam melakukan editing, *Dreamweaver* membuatnya menjadi lebih mudah dengan menyediakan tool-tool yang sangat berguna dalam peningkatan kemampuan dan pengalaman kita dalam mendesain web.

Dreamweaver MX dalam hal ini digunakan untuk web desain. *Dreamweaver* MX mengikutsertakan banyak tool untuk kode-kode dalam halaman web beserta fasilitas-fasilitasnya, antara lain : Referensi HTML, CSS dan *Javascript*, *Javascript debugger*, dan editor kode (tampilan kode dan Code inspector) yang mengizinkan kita mengedit kode *Javascript*, XML, dan dokumen teks lain secara langsung dalam *Dreamweaver*. Teknologi *Dreamweaver* Roundtrip HTML mampu mengimpor dokumen HTML tanpa perlu memformat ulang kode tersebut dan kita dapat menggunakan *Dreamweaver* pula untuk membersihkan dan memformat ulang HTML bila kita menginginkannya.

Selain itu *Dreamweaver* juga dilengkapi kemampuan manajemen situs, yang memudahkan kita mengelola keseluruhan elemen yang ada dalam situs. Kita juga dapat melakukan evaluasi situs dengan melakukan pengecekan broken link, kompatibilitas browser, maupun perkiraan waktu download halaman web (Risdiyanto ; 2013 : 2).



Gambar II.2. Tampilan *Dreamweaver*
(Sumber : Risdiyanto ; 2013 : 2)

II.6. Pengertian PHP

PHP singkatan dari PHP : *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada server (*Server Side HTML Embedded Scripting*). PHP adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan. (Anhar ; 2010 : 3).

II.7. Pengertian Database

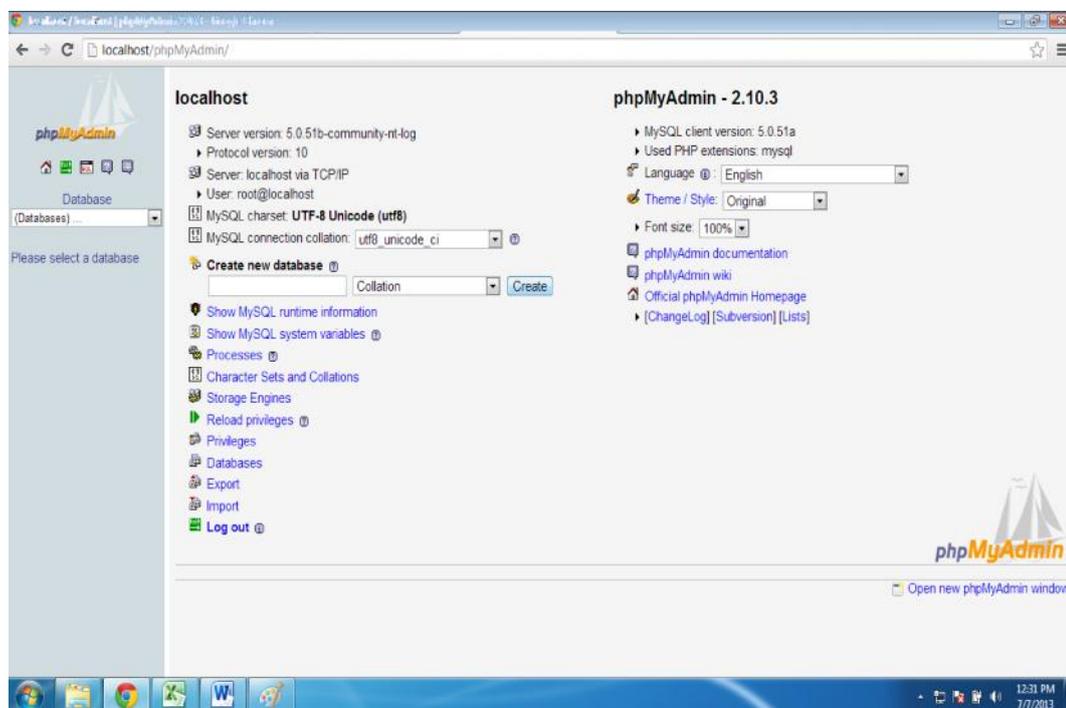
Database adalah sekumpulan *file* data yang saling berhubungan dan diorganisasi sedemikian rupa sehingga memudahkan untuk mendapat dan memproses data. Lingkungan sistem *database* menekankan data yang tidak tergantung (*idenpendent data*) pada aplikasi yang akan menggunakan data. Data adalah kumpulan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

Sebuah *database* harus dibuat dengan rapi agar 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 ; 2010 ; 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, menurutkan 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 ingin membuat produk RDBMS baru dengan basis MySQL dan kemudian menguálnua, anda wajib bertemu mudah dengan lisensi komersial (Antonius Nugraha Widhi Pratama ; 2010 : 10).



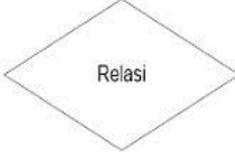
Gambar II.3. Tampilan MySQL
(Sumber : Antonius Nugraha Widhi Pratama ; 2010 : 10)

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 di cermati secara seksama, tool ini mencapai 2NF (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. *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

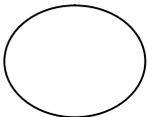
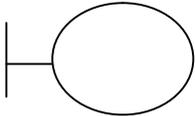
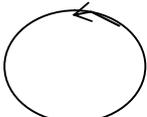
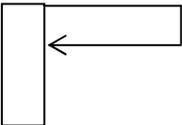
bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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 :

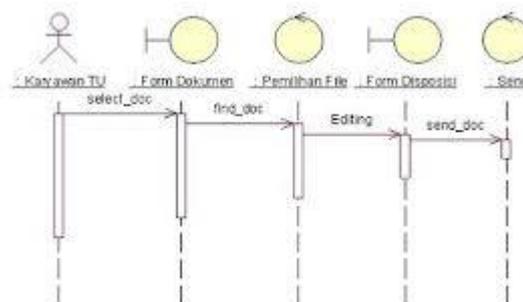
Tabel II.2. Simbol *Sequence Diagram*

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)

Contoh dari pembuatan *sequence diagram* dapat dilihat pada gambar II.6 berikut :



Gambar. II.3. Sequence Diagram
(Sumber : Windu Gata ; 2013 : 7)

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.

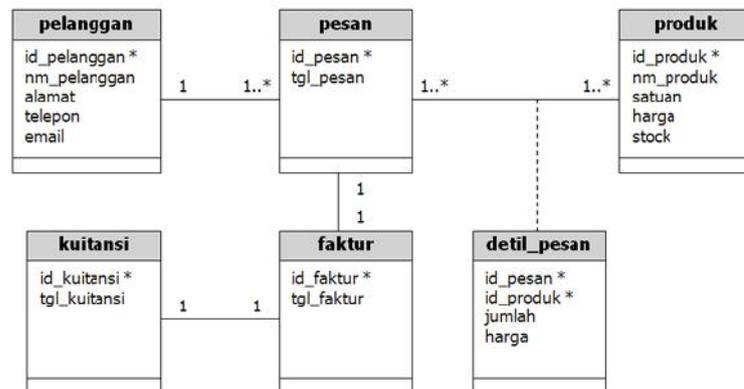
Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata ; 2013 : 8)

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

berikut :

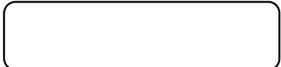
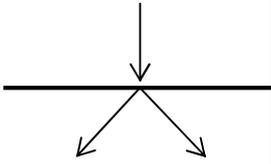
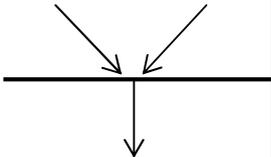
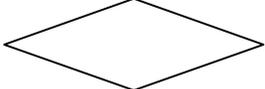


Gambar. II.5. Class Diagram
(Sumber : Windu Gata ; 2013 : 8)

3. Diagram Aktivitas (*Activity Diagram*)

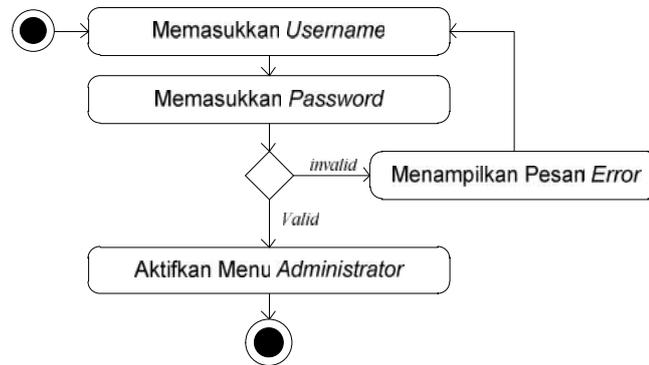
Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.6. 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.5 berikut :

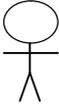


Gambar. II.7. Activity Diagram
(Sumber : Windu Gata ; 2013 : 6)

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.8. Simbol Use Case

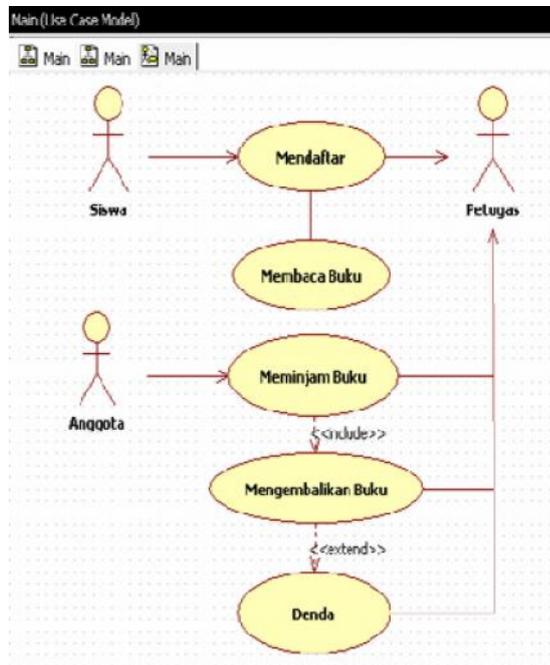
Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa</p>

	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 mengidkasikan aliran data.
—————>	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidnkasikan 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.9. Use Case Diagram
(Sumber : Windu Gata ; 2013 : 4)