

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Perancangan**

Perancangan adalah satu langkah untuk memberikan gambaran secara umum kepada manusia atau pengguna tentang sistem yang diusulkan. Perancangan sistem atau desain secara umum mendefinisikan komponen-komponen sistem informasi pemetaan yang akan dirancang. Tambahannya, model perancangan sesungguhnya secara langsung bertindak sebagai abstraksi implementasi sistem/perangkat lunak dan dengan sendirinya model perancangan suatu saat nanti akan menjadi asupan bagi aktivitas-aktivitas selanjutnya yang kelak akan terdefinisi pada tahap implementasi. Penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi setiap permasalahan dan hambatan yang terjadi serta kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya (Ambon Saragih ; 2015 : 33).

#### **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 (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). (Edy Irwansyah ; 2013 : 1).

Teknologi GIS mengintegrasikan operasi pengolahan data berbasis database yang biasa digunakan saat ini, seperti pengambilan data berdasarkan kebutuhan, serta analisis statistik dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya.

Data yang diolah pada GIS ada 2 macam yaitu data geospasial atau yang biasanya disebut data spasial dan data non-spasial (atribut). Jika pada gambar diatas data atribut tidak digambarkan karena memang dalam GIS yang dipentingkan adalah tampilan data secara spasial. Tetapi sebenarnya pada GIS kadang-kadang juga melibatkan data atribut baik secara langsung maupun secara tidak langsung ( Lisa Ambarwati ; 2013 : 2)

#### **II.4. Haversine Formula**

Metode haversine formula dapat digunakan untuk menghitung jarak antara dua titik, berdasarkan posisi garis lintang (latitude) dan posisi garis bujur (longitude). Metode Haversine Formula tersebut kini sudah mengalami pengembangan, yaitu dengan menggunakan rumus spherical law of cosine sederhana, dimana dengan penghitungan komputer dapat memberikan tingkat presisi yang sangat akurat antar dua titik (Stevian Suryo Saputro; 5-6) :

$$d = \text{acos}(\sin(\text{lat1}) \cdot \sin(\text{lat2}) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \cos(\text{long2} - \text{long1})) \cdot R$$

Keterangan : R = Jari-jari bumi sebesar 6371(km)

D = Jarak (km)

Lat1, lat2 = latitude titik 1 dan 2

Long1, long2 = longitude titik 1 dan 2

#### **II.5. Sistem Koordinat Polar**

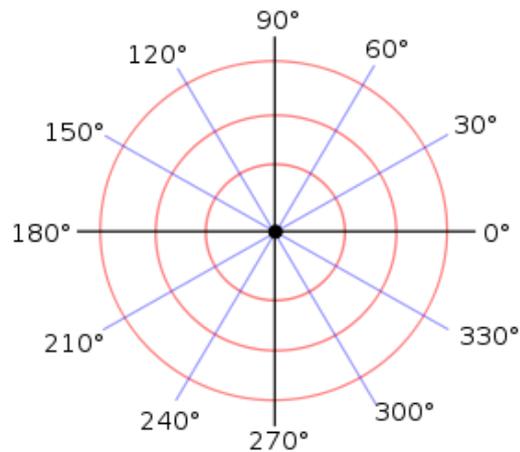
Sistem koordinat polar (sistem koordinat kutub) dalam matematika adalah suatu sistem koordinat 2-dimensi di mana setiap titik pada bidang ditentukan

dengan jarak dari suatu titik yang telah ditetapkan dan suatu sudut dari suatu arah yang telah ditetapkan.

Titik yang telah ditetapkan (analog dengan titik origin dalam sistem koordinat Kartesius) disebut *pole* atau "kutub", dan *ray* atau "sinar" dari kutub pada arah yang telah ditetapkan disebut "aksis polar" (*polar axis*). Jarak dari suatu kutub disebut *radial coordinate* atau *radius*, dan sudutnya disebut *angular coordinate*, *polar angle*, atau *azimuth*.

Konsep sudut dan jari-jari sudah digunakan oleh manusia sejak zaman purba, paling tidak pada milenium pertama SM. Astronom dan astrolog Yunani, Hipparchus, (190–120 SM) menciptakan tabel fungsi chord dengan menyatakan panjang chord bagi setiap sudut, dan ada rujukan mengenai penggunaan koordinat polar olehnya untuk menentukan posisi bintang-bintang. Dalam karyanya *On Spirals*, Archimedes menyatakan Archimedean spiral, suatu fungsi yang jari-jarinya tergantung dari sudut. Namun, karya-karya Yunani tidak berkembang sampai ke suatu sistem koordinat sepenuhnya.

Dari abad ke-8 M dan seterusnya, para astronom mengembangkan metode untuk menghitung arah ke Mekkah (kiblat) dan jaraknya dari semua lokasi di bumi



**Gambar II.1. Grid Polar Dengan Beberapa Sudut Yang Diberi Label  
Dalam Derajat**

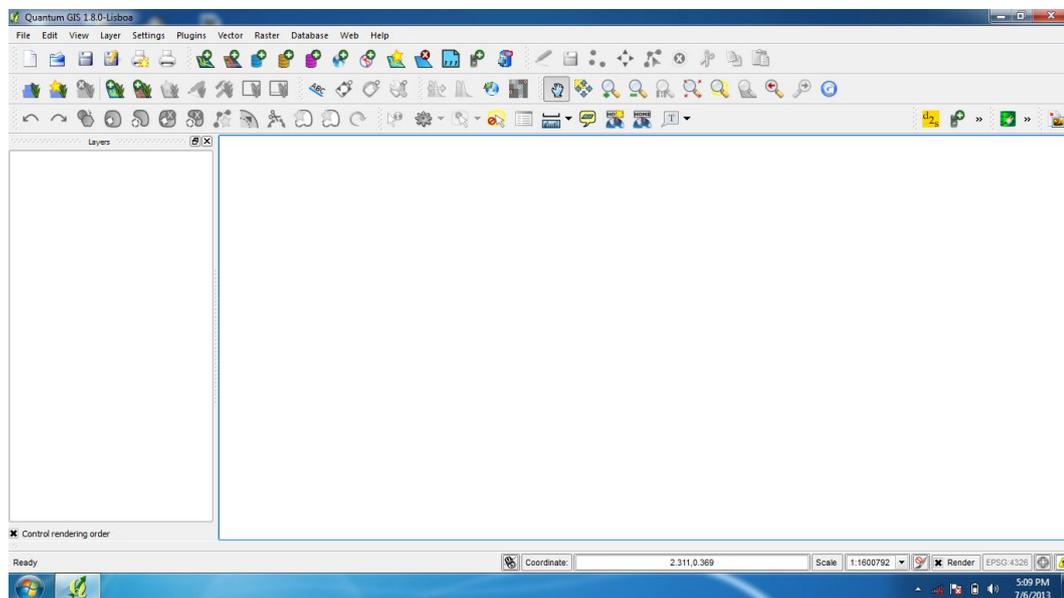
### **II.5.1. Kelebihan dan Kekurangan dari kedua Metode**

Sistem Koordinat Polar ini digunakan apabila kita akan membuat suatu gambar apabila Sistem Koordinat Kartesius tidak memungkinkan lagi. Kelebihan Sistem Koordinat Polaryaitu pemasukan data yang simpel dan mudah namun dengan presisi yang tinggi (akurat). Untuk menggambar suatu titik menggunakan Sistem Koordinat Polar bukan ditentukan dari titik origin seperti pada Sistem Koordinat Kartesius, akan tetapi dari titik penggambaran.

Haversine merupakan formula yang populer digunakan untuk menemukan jarak antara 2 titik. Tetapi belum ada penelitian yang membuktikan dari sekian banyak formula untuk mencari jarak mana yang “terbaik”

## II.6. 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](http://www.qgis.org) (Saddam Hussein ; 2012 : 93).



**Gambar II.2. Tampilan Quantum GIS**

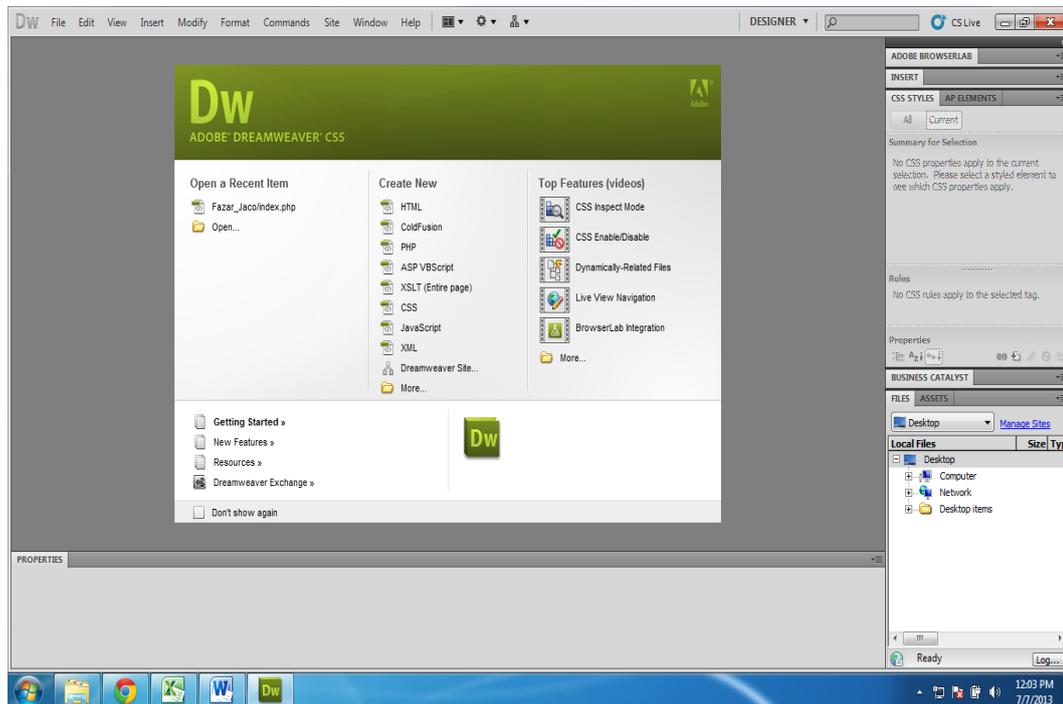
**(Sumber : Saddam Hussein ; 2012 : 93)**

## II.7. 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.3. Tampilan *Dreamweaver***

(Sumber : Risdiyanto ; 2013 : 2)

## II.8. 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 ; 2011 : 3).

## II.9. 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 ; 2012 ; 1).

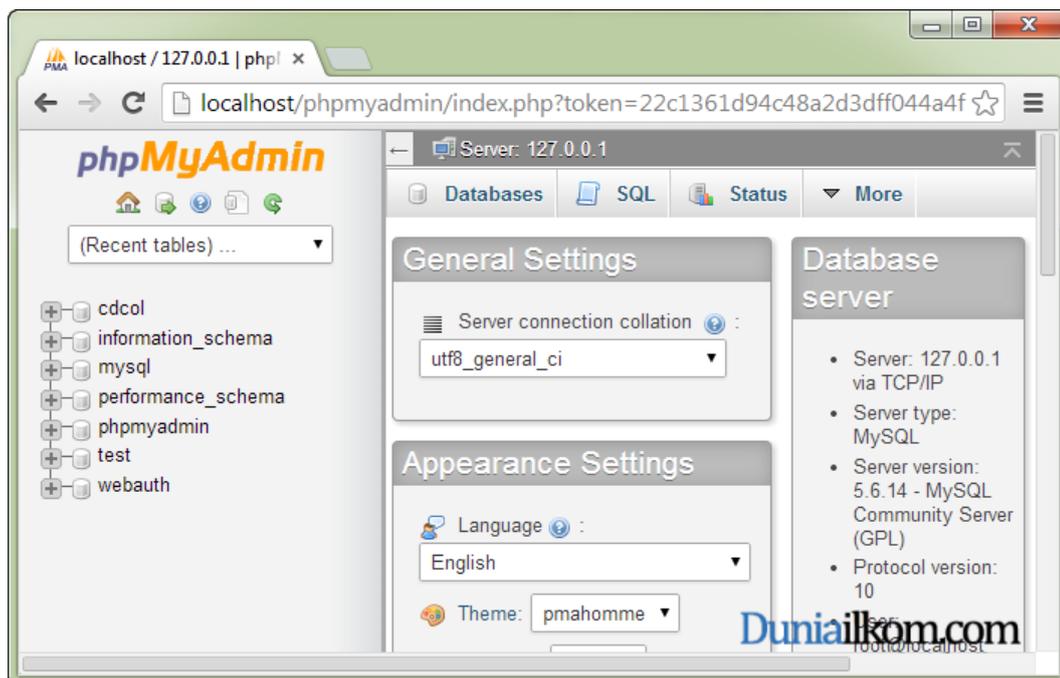
## II.10. Deskripsi sistem

Sistem yang akan penulis kembangkan di sini merupakan sebuah aplikasi pencarian jalur terpendek pemadam kebakaran yang ada di kota Medan. Aplikasi ini berisi tentang informasi peta rekomendasi letak pemadam kebakaran pada peta dengan jarak terpendek antara tempat pemadam kebakaran dengan aplikasi yang di buat ini, dengan aplikasi ini bagian masyarakat kota Medan dapat meringankan tugas pemadam kebakaran dalam menjadwalkan keberangkatan.

## II.11. 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 (Aryanto ; 2016 : 5).



**Gambar II.4. Tampilan MySQL**

(Sumber : Aryanto ; 2016 : 5)

## II.12. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan.

### **II.12.1. Bentuk-bentuk Normalisasi**

#### 1. Bentuk normal tahap pertama (1<sup>st</sup> Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

#### 2. Bentuk normal tahap kedua (2<sup>nd</sup> normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

#### 3. Bentuk normal tahap ketiga (3<sup>rd</sup> normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

#### 4. Boyce Code Normal Form (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

## 5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalued (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata, 2011 : 76).

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

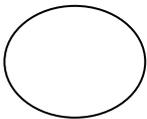
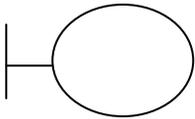
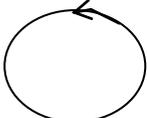
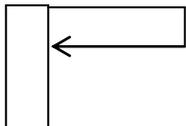
Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

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 :

- Diagram Urutan (*Sequence Diagram*)

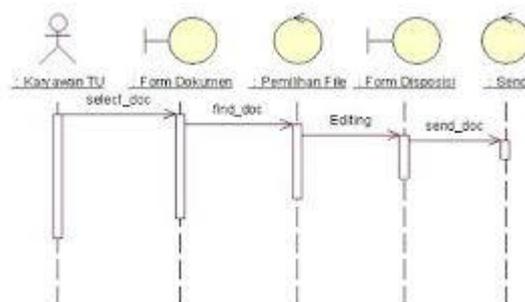
*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

**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)

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



**Gambar. II.5. Sequence Diagram**

**(Sumber : Windu Gata ; 2013 : 7)**

- *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.

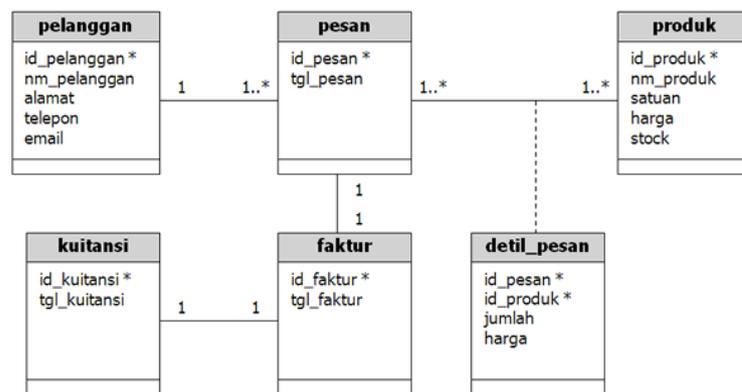
**Tabel II.2. Multiplicity Class Diagram**

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.6

berikut :



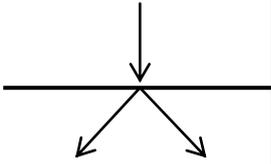
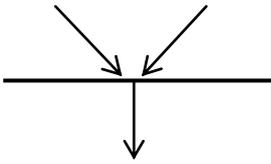
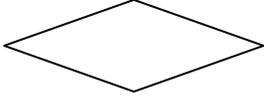
**Gambar. II.6. Class Diagram**

(Sumber : Windu Gata ; 2013 : 8)

- Diagram Aktivitas (*Activity Diagram*)

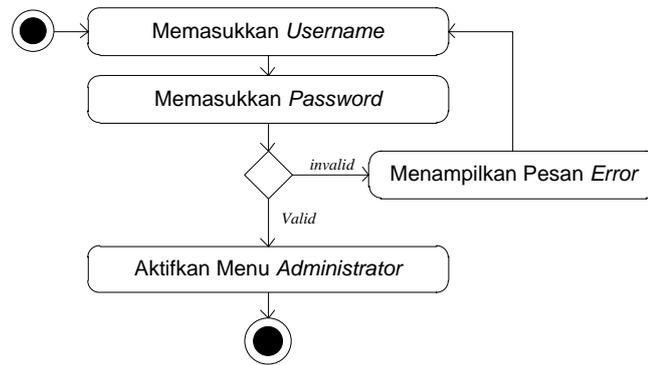
*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

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.7 berikut :



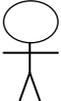
**Gambar. II.7. Activity Diagram**

(Sumber : Windu Gata ; 2013 : 6)

- 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**

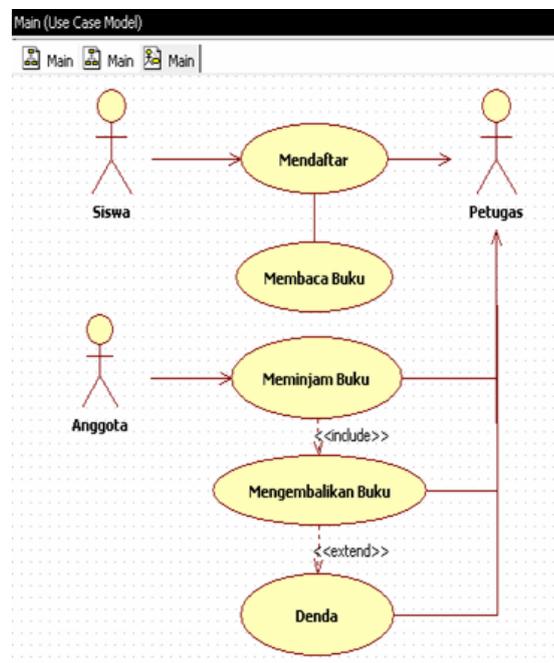
Gambar	Keterangan
	<p>Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan 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 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 : 4)

Contoh dari pembuatan *use case diagram* dapat dilihat pada gambar II.8

berikut :



**Gambar. II.8. Use Case Diagram**

(Sumber : Windu Gata ; 2013 : 4)