BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Teori sistem melahirkan konsep-konsep futuristik. Salah satu konsep yang terkenal adalah konsep sibernetika (*cybernetics*). Konsep kajian ilmiah ini terutama berkaitan dengan upaya menerapkan berbagai disiplin ilmu, yaitu ilmu perilaku, fisika, biologi dan teknik. Oleh karen itu, sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan oleh manusia sehingga studi tentang robotika, kecerdasan buatan (*artificaial intelegance*), dan lain adalah masukan (*input*), pengolahan (*processing*), dan keluaran (*output*).

Konsep lain yang terkandung di dalam definisi tentang sistem adalah konsep sinergi. Konsep ini mengandaikan bahwa di dalam suatu sistem, *output* dari suatu organisasi diharapkan lebih besar dari pada *output* individual atau *output* masing-masing bagian.

Sebuah sistem terdiri atas bagian-bagian atau komponen yang terpadu untuk satu tujuan. Model dasar dari bentuk sistem ini adalah adanya masukan, pengolahan, dan keluaran. Namun demikian sistem ini dapat dikembangkan hingga menyetakan media penyimpanan. Sistem dapat terbuka dan tertutup. Sistem informasi biasanya adalah sistem terbuka, yang berarti bahwa sistem

tersebut dapat menerima berbagai masukan dari lingkungan sekitarnya (Tata Sutabri ; 2012 : 3-4).

II.1.1 Karakteristik Sistem

Adapun karakteristik yang mencirikan suatu sistem, yaitu :

1. Komponen Sistem (*Component*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinterksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen- komponen sistem tersebut dapat berupa suatu bentuk subsistem.

2. Batasan Sistem (*Boundary*)

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environtment)*

   Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat merugikan sistem tersebut.

4. Penghubung Sistem (*Interface*)

   Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*.

5. Masukan Sistem (*Input*)

   Energi yang dimasukan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

   Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini digunakan sebagai masukkan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem yang lain.

7. Pengolahan Sistem (*Procces*)

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*)

   Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan (Tata Sutabri ; 2012 : 12-13).

II.1.2. Informasi

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan

dengan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan. Keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks pengambilan keputusan (Tata Sutabri ; 2012 : 18).

Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Informasi yang disampaikan kepada pemakai mungkin merupakan hasil dari data yang dimasukkan ke dalam pengolahan. Akan tetapi dalam kebanyakan pengambilan keputusan yang kompleks, informasi hanya dapat menambah kemungkinan kepastian atau mengurangi bermacam-macam pilihan (Tata Sutabri ; 2012 : 19).

II.1.3. 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.

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat *manajerial* dengan kegiatan strategi dari suatu orgnanisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri ; 2012 : 36).

II.1.3.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*Building Block*) yang terdiri dari :

1. Blok Masukan (*Input Block*)

   Input mewakili data yang masuk kedalam sistem informasi.

2. Blok Model (*Model Block*)

   Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajeman serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

   Teknologi merupakan *tool box* dalam sisitem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Blok Basis Data (*Database Block*)

   Basis data *(database)* merupakan kumpulan data yang saling berkaitan berhubungan satu sama lainnya, tersimpan diperangkat keras komputer dan meggunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali *(Control Block)*

   Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi (Tata Sutabri ; 2012 : 37).

II.2. Sistem Informasi Geografis

Sistem informasi geografis (SIG) adalah suatu sistem informasi yang berbasis komputer, dirancang untuk bekerja dengan menggunakan data yang memiliki informasi spasial (bereferensi kekurangan). Sistem ini meng-*capture*, mengecek, mengintergasikan, memanipulasikan, dan menampilkan data yang secara spasial mereferensikan kepada kondisi bumi. Tekologi sistem informasi geografis mengintegrasikan operasi-operasi umum database, seperti *query* dan analisa statistik, dengan kemampuan visualisasi yang dimiliki oleh pemetaan.

Kemampuan inilah yang membedakan sistem informasi geografis dengan sistem informasi lainnya yang membuatnya menjadi berguna berbagai kalangan untuk menjelaskan kejadian, merencanakan strategi, dan memprediksi apa yang terjadi.

Sistem ini pertama kali diperkenalkan di Indonesia pada tahun 1972 dengan nama *Data Banks for Development (Rais).* Munculnya istilah sistem informasi geografis seperti sekarang ini setelah dicetuskan oleh *General Assembly* dari *International Geographical Union* di Ottawa Kanada pada tahun 1967. Dikembangkan oleh Roger Tomlison, yang kemudian disebut *CGIS (Canadian*

GIS-GIS Kanada), digunakan untuk menyimpan, menganalisa dan mengolah data yang dikumpulkan untuk Inventarisasi Tanah Kanada *(CLI-Canadian Land Inventory)* sebuah inisiatif untuk mengetahui kemampuan lahan di wilayah pedesaan Kanada dengan memetakan berbagai informasi pada tanah, pertanian, pariwisata, alam bebas, unggas dan penggunaan tanah pada skala 1:250000.

Sejak saat itu sistem informasi geografis berkembang di beberapa benua terutama Amerika, Eropa, Australia, dan Asia. Seperti di negara-negara yang lain, di Indonesia pengembangan sistem informasi geografis dimulai di lingkungan pemerintahan dan militer. Perkembangan sistem informasi geografis menjadi pesat semenjak di tunjang oleh sumber daya yang bergerak di lingkungan akademis (kampus).

Pada sistem informasi geografis dimanfaatkan oleh operator, ada sesuatu yang diharapkan darinya. Salah satu hal yang diperoleh dari sistem informasi geografis adalah kemampuannya dalam menganalisis data spasial. Namun kadang operator sistem informasi geografis tidak memahami apakah dia sudah melaksanakan suatu analisiss spasial ataukah baru sekedar menjalankan suatu prosedur yang ada dalam sebuah perangkat lunak sistem informasi geografs. Mungkin pula sebaliknya, operator mungkin baru sekedar membuat data digitasi hingga layout peta, namun merasa sudah melakukan analisis spasial dengan menggunakan sistem informasi geografis.

Setiap sistem selalu memiliki kelebihan dan kekurangannya masing-masing. Dan sistem informasi geografis juga memiliki kelebihan dan kekurangan.

Adapun kelebihan dan kekurangan dari sistem informasi geografis ini adalah sebagai berikut :

a. Kelebihan

1. Dapat melakukan pengolahan dengan format baik
2. Mengelola data dengan biaya murah jika dibandingkan dengan survei lapangan
3. Data dapat diubah dan diambil dengan cepat karena tersimpan dalam *file* komputer
4. Data yang berbentuk spasial dan non spasial dapat dikelola secara bersama-sama
5. Analisa dapat dilaksanakan dengan efisien
6. Data yang sulit diolah secara manual dapat diolah komputer dan tampil secara tiga dimensi
7. Data berbentuk gambar, peta, atau bagan dapat diperoleh secara cepat dan tepat
8. Mengolah dan menganalisa data, seperti mengubah, menambah, atau menghapus tanpa mengganggu data lain yang telah disusun

b. Kekurangan

1. Tidak banyak diketahui oleh masyarakat awam
2. Jika terjadi kerusakan pada software pengolah data dapat mengakibatkan hilangnya data yang belum sempat tersimpan
3. Peralatan yang dibutuhkan relatif mahal
4. Hampir semua data diolah dengan menggunakan komputer

Dari dunia nyata diambil tiga hal penting yaitu posisi dan klasifikasi, atribut, serta hubungan antar item tersebut. Ketiga hal tersebut diolah sebagai dasar analisa sistem spasial dalam sistem informasi geografis. Dengan dasar tersebut akan dapat diperoleh manfaat dari sistem informasi geografis sebagai berikut :

1. Menjelaskan tentang lokasi atau letak
2. Menjelaskan kondisi ruang
3. Menjelaskan suatu kecendrungan *(trend)*
4. Menjelaskan tentang pola spasial *(spatial pattern)*
5. Pemodelan (Eko Budiyanto ; 2010 : 177).

II.3. *PHP*

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemograman *web server-side* yang bersifat *open source* atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah skrip yang terintegrasi dengan HTML dan berada pada *server* (*server side* HTML *embedded scripting*). Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemograman seperti C, Java, dan Perl serta mudah untuk dipelajari.

PHP diciptakan untuk mempermudah pengembangan *web* dalam menulis halaman *web* dinamis dengan cepat, bahkan lebih dari itu kita dapat mengeksplorasi hal-hal yang luar biasa dalam PHP. Sehingga dengan demikian PHP sangat cocok untuk/bagi para pemula, menengah maupun *expert* sekalipun (Hirin & Virgi ; 2011 : 25).

II.3.1 Sejarah Singkat PHP

Awal Mulanya PHP adalah kependekan dari *Personal Home Page* yang dibuat pada tahun 1995 oleh *Rasmus Leodorf*. Saat itu namanya masih *Form Interpreted*. Pada selanjutnya pembuat PHP merilis kode sumber ke khalayak umum (*open source*) sehingga demikian banyak programmer tertarik untuk mengembangkan PHP.

Akhirnya pada November 1997 direlease PHP 2.0, pada versi ini interpreter PHP sudah diimplementasikan dalam C, serta telah disertakan module-module tambahan atau dalam PHP sering disebut dengan ekstensi. Pada tahun 1997 juga ada andil sebuah perusahaan bernama *Zend*, dimana interpreter PHP ditulis ulang menjadi lebih bersih, cepat, dan lebih baik. Dan akhirnya pada pertengahan tahun 1998,Zend merilis PHP 3.0 dengan digantinya singkatan dari *Personal Home Page* menjadi *Hypertext Preprocessor*.

Pada perkembangan selanjutnya *Zend* terus memegang peranan penting dalam perkembangan PHP, pada pertengahan tahun 1999 PHP 4.0 direlease pada versi inilah mulai banyak orang berbondong-bondong memakai PHP karena kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Seiring dengan perkembangan zaman, banyak bahasa pemograman mulai menerapkan model OOP (*Object Oriented Programming*), tak mau ketinggalan PHP versi 5.0 dirilis pada pertengahan 2004 dengan kemampuan barunya yaitu pemrograman berorientasi objek (Hirin & Virgi ; 2011 : 26-27).

II.4. Basisdata

Basisdata adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan dalam berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basisdata menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan kedalam basisdata, dimodifikasi, dan dihapus.

Basisdata warisan (*legacy database*) merupakan basisdata yang sedang digunakan oleh sebuah perusahaan. Istilah warisan menyatakan bahwa basisdata telah dipakai selama beberapa tahun dan basisdata yang ada tidak sesuai dengan teknologi masa kini. Ketika sebuah perusahaan telah menentukan untuk merancang sebuah basisdata, basisdata yang ada dianggap sebagai basisdata warisan. Contoh Basisdata yang telah kita kenal adalah :

- Buku alamat
- Buku telepon
- Katalog perpustakaan
- Toko buku online
- Peta jalan

Beberapa basisdata diatas merupakan basisdata statis, Sedangkan yang lainnya dinamis. Sebagai contoh, peta jalan adalah basisdata statis yang mengandung informasi seperti kota, arah, jarak, dan sebagainya. Dengan melihat sebuah peta, anda cepat menemukan tujuan relatif terhadap posisi anda sekarang.

Informasi pada peta tidak berubah dalam waktu lama. Buku telepon pun merupakan basisdata statis karena informasi didalamnya hanya dicetak setiap tahun.

Buku alamat adalah contoh basisdata dinamis yang banyak digunakan sehari-hari. Buku alamat merupakan basisdata dinamis karena isinya dapat diubah dengan cepat. Alamat teman baru dapat ditambahkan dan alamat teman lama dapat dihapus dengan mudah (Janner & Iman ; 2010 : 1).

II.4.1. *Normalisasi*

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

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

Pada waktu menormalisasikan basisdata, ada empat tujuan yang harus dicapai, yaitu :

1. Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil dari sistem.
2. Meminimalkan jumlah data berulang dalam basisdata.

3. Membuat basisdata yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan intergitas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada satu tempat.

Tujuan Normalisasi adalah membuat kumpulan tabel relasional yang bebas dari data berulang dan dapat dimodifikasi secara benar dan konsisten.

Ada beberapa langkah dalam normalisasi tabel, yaitu :

1. *Decomposition* (dekomposisi)

   Dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenai dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di-dekomposisi. Dekomposisi akan sering dilakukkan dalam proses normalisasi untu memenuhi syarat-syaratnya.
2. Bentuk Tidak Normal

   Pada bentuk ini semua data yang ada pada tiap entity (diambil atributnya) masih ditampung dalam satu tabel besar. Data pada tabel ini masi ada redudansi dan ada juga yang kosong dan semua masih tidak tertata rapi.
3. Normal Form Pertama

   Pada tahapan ini tabel di dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomic.

4. Normal Form Kedua

   Pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut.

5. Normal Form Ketiga

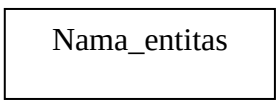
   Pada tahapan ini setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien.

II.4.2. *Entity Relational Diagram (ERD)*

ERD merupakan suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang menghubungkan antara objek satu dengan objek yang lain dalam dunia nyata. Bisa dikatakan bahwa bahan yang digunakan untuk membuat ERD adalah dari objek di dunia nyata. Secara umum ERD terdiri dari 3 komponen, yaitu :

1. Entitas *(Entity)*

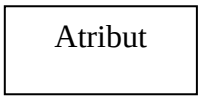
   Merupakan suatu "objek nyata" yang mampu dibedakan dengan objek yang lain. Objek tersebut dapat berupa orang benda ataupun hal yang lainnya. Penggambaran entitas dalam ERD seperti pada gambar II.1



Gambar II.1 : Entitas
(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 18)

2. Atribut *(Attribute)*

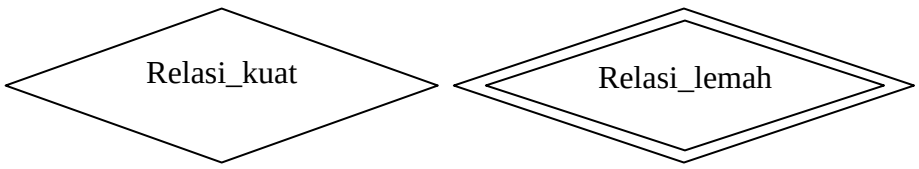
Merupakan semua informasi yang berkaitan dengan entitas. Di dalam dunia pemograman, atribut adalah properti dari suatu objek. Penggambaran atribut dalam ERD seperti pada gambar II.2



Gambar II.2 : Atribut
(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 20)

3. Relasi *(Relationship)*

Belah ketupat merupakan penggambaran hubungan (relasi) anatr entitas atau sering disebut kerelasian. Ada dua macam penggambaran relasi, yakni relasi kuat dan relasi lemah. Relasi kuat biasanya untuk menghubungkan antar entitas kuat, sedangkan realsi lemaha untuk menghubungkan antar entitas kuat dengan entitas lemah. Penggambaran kerelasiaan seperti gambar II.3



Gambar II.3 : Kerelasian
(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

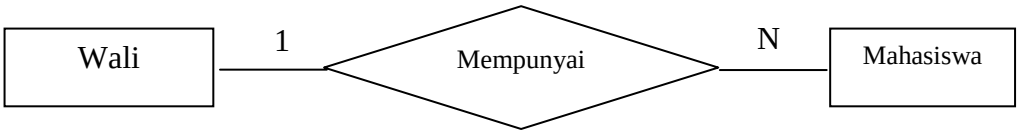
II.4.2.1. Derajat Kardinalitas

Merupakan penjelasan dari tingkat hubungan antar entitas. Ukuran derajat kardinalitas dibagi menjadi tiga macam, yaitu :

1. 1-1 (*one-to-one*), misalnya seorang ketua jurusan hanya memimpin satu jurusan, begitu juga sebaliknya satu jurusan hanya dipimpin seorang ketua jurusan.

Gambar II.4 : 1-1 *(one-to-one)*
(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

2. 1-N (*one-to-many*) atau N-1 (*many-to-one*), misalnya seorang mahasiswa hanya mempunyai seorang wali, tetapi seorang wali bisa menjadi wali banyak mahasiswa.



Gambar II.5 : 1-N *(one-to-many)*
(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 25)

3. N-N (*many-to-many*), misalnya seorang mahasiswa bisa mengambil banyak mata kuliah, begitu juga sebaliknya satu mata kuliah bisa diambil oleh banyak.

Gambar II.6 : N-N *(many-to-many)*
(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 25)

II.5. *MySQL*

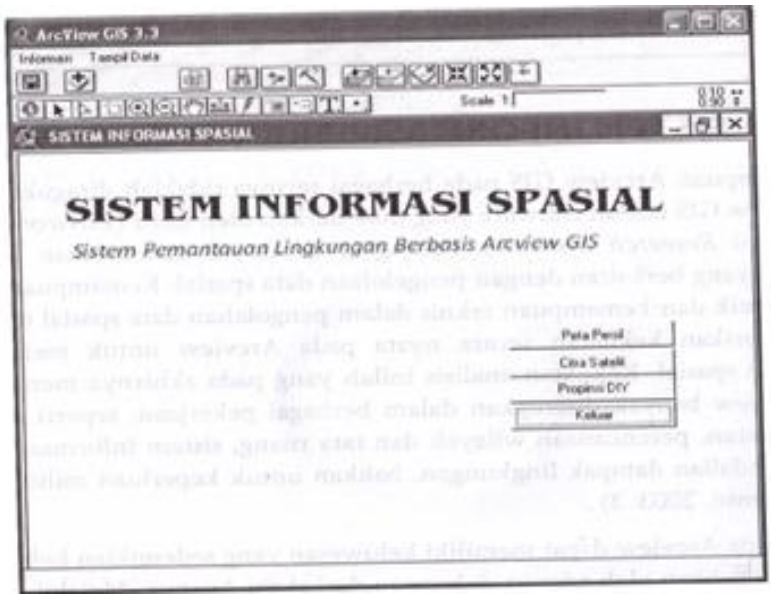
MySQL adalah salah satu perangkat lunak sistem manajemen basisdata (*database*) *SQL* atau sering disebut dengan DBMS (*Database Management System*). Berbeda dengan basisdata konvensional seperti .dat, .dbf, .mdb, *MySQL* memiliki kelebihan yaitu bersifat *multithread*, dan *multi-user* serta mendukung sistem jaringan. *MySQL* didistribusikan secara gratis dibawah lisensi GNU General Public License (GPL). Namun ada juga versi komersial bagi kalangan tertentu yang menginginkannya (Hirin & Virgi ; 2011 : 27-28).

II.6. *ArcView*

Kemampuan *Arcview* GIS pada berbagai serinya tidaklah diragukan lagi. *Arcview* GIS adalah software yang dikeluarkan oleh ESRI *(Environmental Systems Research Institute)*. Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengolahan data spasial. Kemampuan grafis yang baik dan kemampuan teknis dalam pengolahan data spasial tersebut memberikan kekuatan secara nyata pada *Arcview* untuk melakukan analisis spasial. Kekuatan analisis inilah yang pada akhirnya menjadikan *Arcview* banyak diterapkan dalam berbagai pekerjaan, seperti analisis pemasaran, perencanaan wilayah dan tata ruang, sistem informasi persis, pengendalian dampak lingkungan, bahkan untuk keperluan militer. Mengapa *Arcview* dapat memiliki keluwesan yang sedemikian hebat? Hal itu disebabkan oleh adanya dukungan dari skrip *Avenue*. Melalui *avenue* ini dapat dibentuk suatu “kemampuan baru” pada *Arcview*. Tentu saja hal ini membuat *Arcview* menjadi sangat luwes untuk diterapkan pada berbagai permasalahan

spasial. *Avenue* dapat digunakan untuk "merombak" wajah *Arcview* sesuai kebutuhan penggunaanya.

Antarmuka sistem informasi *(interface)* dibentuk dengan memanfaatkan fasilitas *customize* pada perangkat lunak *Arcview* GIS 3.3. Menu dan tombol dibentuk menggunakan teknik kustomasi tersebut. Teknik ini dipilih berdasarkan pada kemudahannya dalam membentuk menu dan berbagai tombol baru.



Gambar II.7. Antarmuka Sistem Informasi Berbasis *Arcview* GIS
(Sumber : Eko Budiyanto ; 2010 : 178)

Dialog designer diperlukan untuk membentuk antarmuka penampil data atribut yang menjadi dasar pemilihan objek. *Dialog designer* yang dipilih adalah bentuk kotak daftar *(listbox).* Dengan menggunakan dialog ini operator akan memilih informasi apa yang akan dicari. Untuk menghubungkan menu dan tombol dengan berbagai aksi yang diinginkan maka perlu dibentuk skrip atau program. Skrip atau program ini dibentuk menggunakan bahasa *Avenue*. Setiap aksi yang diperlukan diuraikan menjadi baris-baris perintah pada skrip *Avenue*

dan selanjutnya dikaitkan ke masing-masing menu atau tombol yang bersangkutan (Eko Budiyanto ; 2010 : 178).

II.7. *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang sebuah model sistem (Yuni Sugiarti; 2013:34).

Unified Modelling Language (UML) biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor.*
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction* diagrams.
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagrams.*
4. Membuat model behavior "yang menggambarkan kebiasaan atau sifat sebuah sistem" dengan *state transition diagrams.*
5. Menyatakan arsitektur implementasi fisik menggunakan *component* dan *development diagrams.*
6. Menyampaikan atau memperluas fungsionality dengan *stereotype* (Yuni Sugiarti; 2013:36).

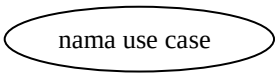
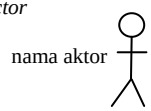
II.7.1. Diagram-Diagram *UML*

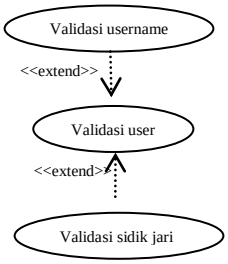
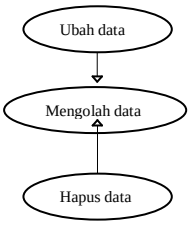
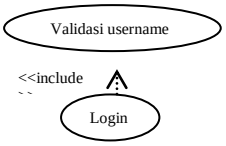
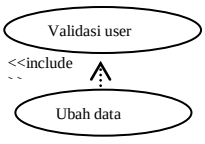
Terdapat sembilan jenis diagram UML, namun Penulis akan menjabarkan empat jenis diantaranya :

1. *Use Case Diagram*

Use Case adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah sistem. Karena sistem pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali susah menjawabnya tatkala dimintai masukan tentang sesuatu. Ide dasarnya adalah bagaimana melibatkan penggunaan sistem di fase – fase awal analisis dan perancangan sistem. Diagram *Use Case* menunjukkan 3 aspek dari sistem yaitu *actor*, *use case* dan sistem/sub sistem *boundary*. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*.

Tabel II.1. *Use Case* Model

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*
Aktor / *actor* 	Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri
Asosiasi / *association*	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case*, atau *usecase* memiliki interasi dengan aktor

Ekstensi / *extend* <<extend>>	Relasi usecase tambahan ke subuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanppa *use case* tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan misal  arah panah mengarah pada *use case* yang ditambahkan
Generalisasi / *generalization*	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainya misalnya :  Arah panah mengarah pada use case yang menjadi generalisasinya (umum)
Menggunakan / *include* / *uses* <<include>> <<uses>>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini Ada 2 sudut pandang yang cukup besar mengenai include di usecase 1. include berarti use case yang ditambahkan akan selalu dipanggil saat use case dijalankan misal pada kasus berikut :  2. Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang di tambahkan telah di jalankan sebelum use case tambahan di jalankan, misal pada kasus berikut :  Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

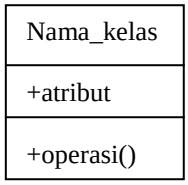
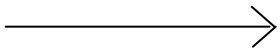
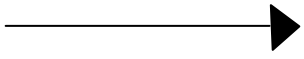
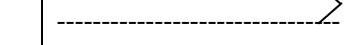
Sumber : Rosa A.S dan M.Shalahuddin (2011 : 101)

2. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

a. Atribut merupakan *varabel-variabel* yang dimiliki oleh suatu kelas.
b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa A.S dan M. Shalahuddin; 2011 : 122).

Tabel II.2. Simbol *Class Diagram*

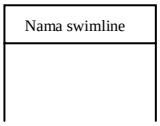
Simbol	**Deskripsi**
Kelas 	Kelas pada struktur sistem
Antarmuka / Interface  Nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
asosiasi / association	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.
Asosiasi berarah / directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / *aggregation*	Relasi antar kelas dengan makna

Sumber : Rosa A.S dan M.Shalahuddin (2011 : 123)

3. *Activity Diagram*

Activity Diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Berikut adalah simbol-simbol yang ada pada *Activity diagram.*

Tabel II.3. Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / decesion 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

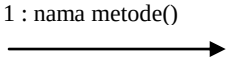
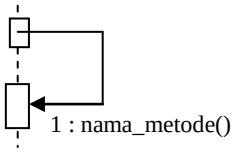
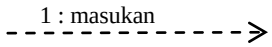
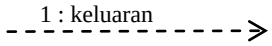
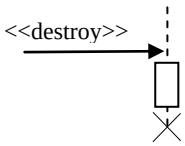
Sumber : Rosa A.S dan M.Shalahuddin (2011 : 134)

4. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. *Diagram* ini menunjukan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case.* Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical.*

Tabel II.4. Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor  nama aktor atau nama aktor tampa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya di nyatakan menggunakan kata benda di awali frase nama aktor
Garis hidup / lifeline	Menyatakan kehidupan suatu objek
Objek nama objek : nama kelas	Menyatakan objek yang berintaraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create <<create>>	Objek yang lain, arah panah mengarah pada objek yang dibuat

Pesan tipe call 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri  Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang di panggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainya, arah panah mengarah pada objek yang dikirimi
Pesan tipe *return* 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasiatau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

Sumber : Rosa A.S dan M.Shalahuddin (2011 : 138)