

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 karena itu, sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan oleh manusia sehingga studi tentang robotika, kecerdasan buatan (*Artificial Intelligence*), 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 *system* terdiri atas bagian-bagian atau komponen yang terpadu untuk satu tujuan. Model dasar dari bentuk sistem ini adalah adanya masukan, pengolahan, dan keluaran. Akan tetapi, sistem ini dapat dikembangkan hingga menyetakan media penyimpanan. Sistem dapat terbuka dan tertutup akan tetapi

sistem informasi biasanya adalah sistem terbuka. Artinya, sistem tersebut dapat menerima beberapa masukan dari lingkungan luarnya. (Tata Sutabri 2012:10-11)

II.1.1 Karakteristik Sistem

Adapun karakteristik yang mencirikan suatu sistem, yaitu :

1. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, 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

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 lain disebut penghubung sistem (*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 masukan 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 deterministic. 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; 2012:20-21)

II.2. Pengertian 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. (Tata Sutabri; 2012:29)

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; 2012:31)

II.3. 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.

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat *manajerial* dengan kegiatan startegi dari suatu orgnaisai untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Tata Sutabri; 2012:46)

II.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 manajemen 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

Basis data (*database*) merupakan kumpulan data yang saling berkaitan berhubungan satu sama lain, tersimpan diperangkat keras computer 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 ; 2012:47)

II.4. Sistem Informasi Geografis

Dalam dunia sistem informasi terdapat banyak model sistem informasi yang bertujuan akhir memberi berbagai macam informasi. Pentingnya informasi ini memberi banyak inspirasi terhadap pembuat model untuk merancang sistem-sistem yang mendekati dunia nyata dengan hasil sedekat mungkin dengan aslinya. Model sistem informasi juga diharapkan dapat digunakan sebagai alat prediksi kejadian dimasa depan dengan mendasarkan pada masa lalu dan masa sekarang. Dari sekian banyak model sistem ini, sistem informasi geografis (SIG) merupakan salah satu model sistem informasi yang digunakan untuk membuat berbagai keputusan, perencanaan, dan analisis.

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

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

II.5. *PHP*

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *Web* dan dapat ditanamkan pada sebuah skrip *HTML*. Bahasa *PHP* dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C++*, *Java*, dan *PHP* 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.5.1 Sejarah Singkat

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 pemrograman 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).

II.6. Basis Data

Basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita pakai 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, modifikasi, dan hapus.

Basis data 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 basis data, basisdata yang ada dianggap sebagai basisdata warisan.

Contoh Basis data yang telah kita pakai adalah :

- a. Buku alamat
- b. Buku telepon
- c. Catalog perpustakaan
- d. Toko buku *online*
- e. Peta Jalan

Beberapa basis data diatas merupakan basis data 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 *relative* 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 telah lama dapat dihapus dengan mudah. (Janner & Iman; 2010 : 1).

II.6.1. *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 table relasional.

Teori normalisasi didasarkan pada konsep bentuk normal. sebuah *table* relasional dikatakan pada bentuk normal tertentu jika *table* 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 integritas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, Anda hanya mengubah pada satu tempat.

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

Ada beberapa langkah dalam normalisasi table, yaitu :

1. *Decomposition*, dekomposisi adalah *proses* mengubah bentuk *table* supaya memenuhi syarat tertentu sebagai *table* yang baik.

Dekomposisi dapat dikatakan berhasil jika *table* yang dikenai dekomposisi bila digabungkan kembali dapat menjadi *table* awal sebelum di-dekomposisi. *Dekomposisi* akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat-syaratnya.

2. Bentuk tidak normal, pada bentuk ini semua data yang ada pada tiap *entity* (diambil atributnya) masih ditampung dalam satu *table* besar. Data pada *table* ini masih ada redundansi dan ada juga yang kosong. Semua masih tidak tertata rapi.
3. Normal Form Pertama, pada tahapan ini *table* di *dekomposisi* dari *table* bentuk tidak normal yang kemudian dipisahkan menjadi *table-table* kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat *atomic*.
4. Normal Form Kedua pada Tahapan ini *table* dianggap memenuhi normal kedua jika pada *table* tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer *table* tersebut.
5. Normal Form Ketiga

Pada tahapan ini setiap atribut pada *table* selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada *table* yang belum efisien. Biasanya penggunaan bentuk normal hanya sampai pada bentuk ketiga, dan *table* yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan.

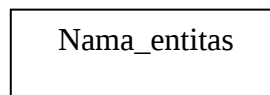
II.6.2. *Entity Relational Diagram (ERD)*

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



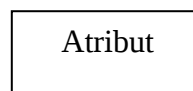
Gambar II.1 : Entitas

(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 19)

1. 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

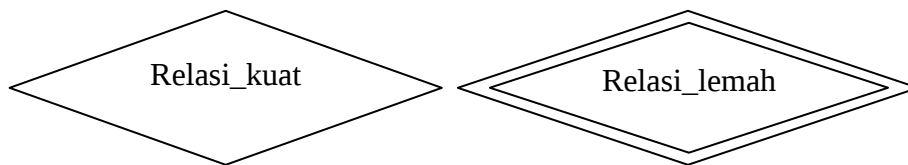


Gambar II.2 : Atribut

(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 20)

2. Relasi (*Relationship*)

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



Gambar II.3 : Kerelasiaan

(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

II.6.2.1. Derajat Kardinalitas

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

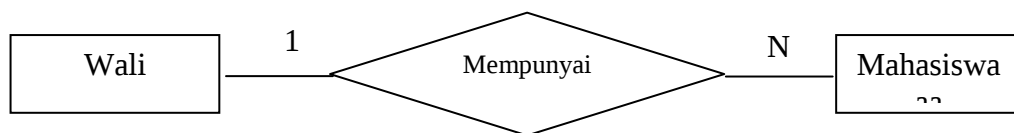
1. 1-1 (*one-to-one*), misalnya seorang ketua jurusan hanya memimpin satu jurusan, begitu juga sebaliknya satu jurusan hanya dipimpin seorang ketua jurusan.



Gambar II.4 : 1-1 Derajat Kardinalitas (*one-to-one*)

(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 24)

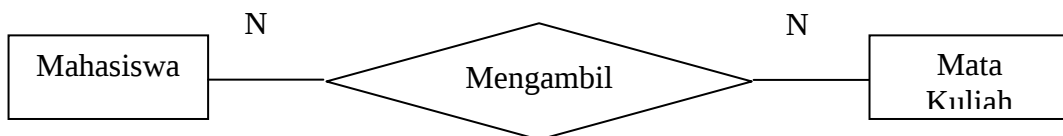
2. 1-N (*one-to-many*) atau N-1 (*many-to-one*), misalnya seorang mahasiswa hanya mempunyai seorang wali, tetapi seorang wali bisa menjadi wali banyak mahasiswa.



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*)

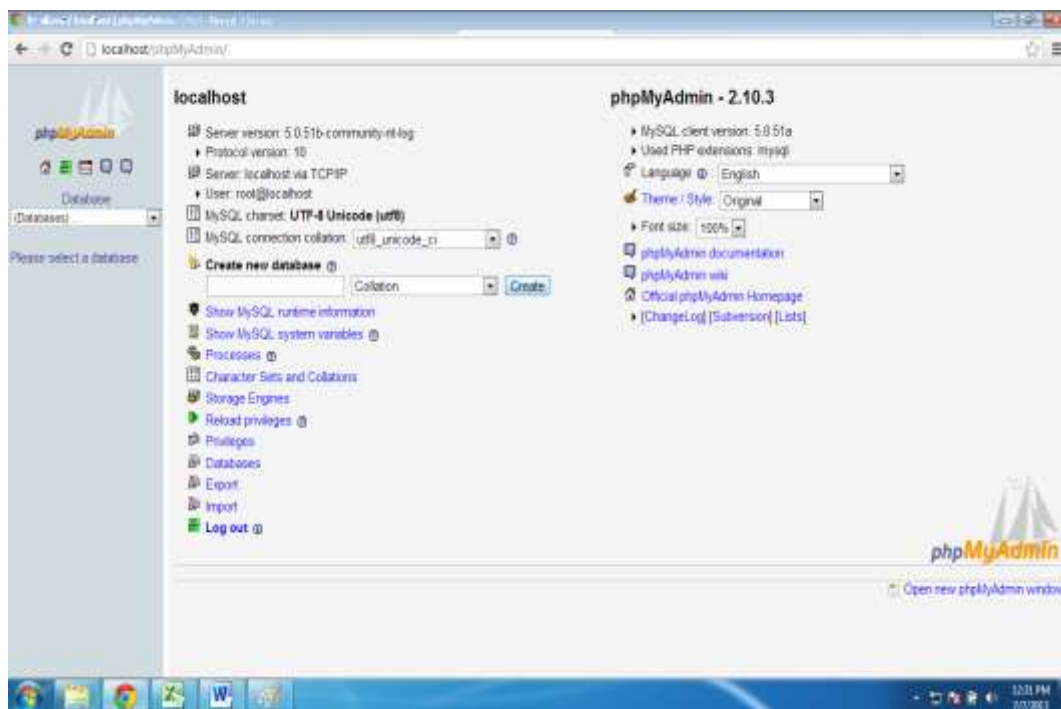
(Sumber : Ema Utami dan Anggit Dwi Hartanto ; 2012 : 25)

II.7. 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

fungsi 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 mengenalnya, anda wajib bertemu mudah dengan lisensi komersial (Antonius Nugraha Widhi Pratama ; 2010 : 10).



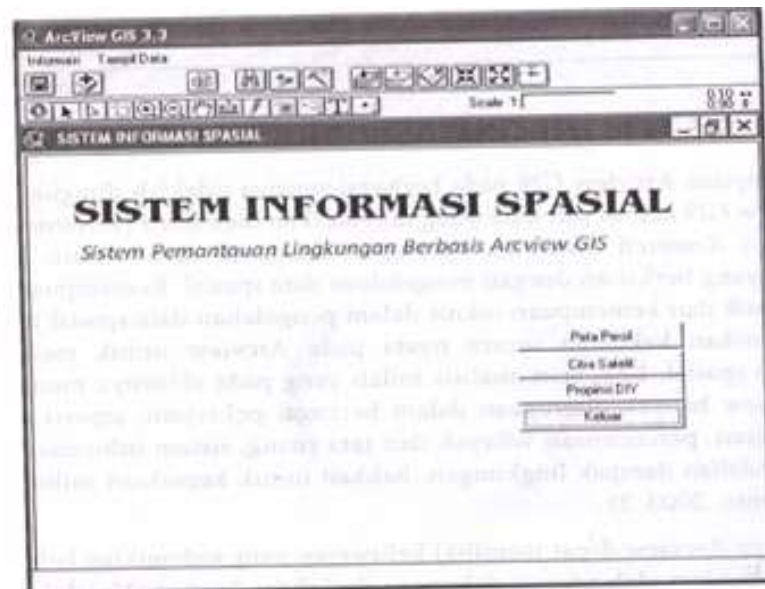
Gambar II.7. Tampilan MySQL

(Sumber : Antonius Nugraha Widhi Pratama ; 2010 : 10)

II.8. *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.8. 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.9. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standard dalam industry untuk visualisasi, merancang dan

mendokumentasikan system piranti 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 and development diagrams
6. Menyampaikan atau memperluas fungsionalitas dengan stereotype.(Yuni Sugiarti; 2013:36)

II.9.1. Diagram-Diagram UML

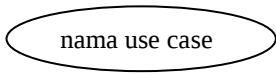
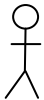
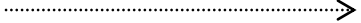
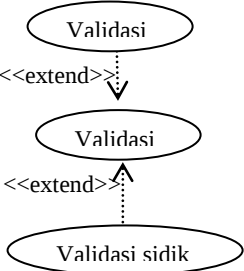
Terdapat sembilan jenis diagram UML, namun Penulis akan menjabarkan empat jenis diantaranya :

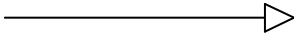
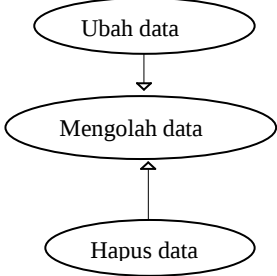
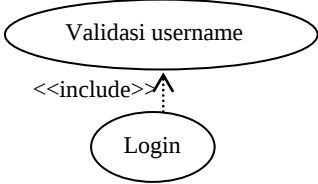
1. Use Case Diagram

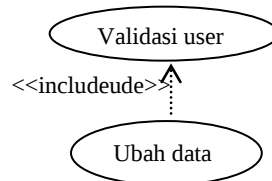
Use Case adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah sistem. Karena sistem pengembangan tradisional

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 <i>use case</i>
Aktor / <i>actor</i> nama aktor 	Orang, proses, atau sistem yang lain berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri
Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> , atau <i>usecase</i> memiliki interaksi dengan aktor
Ekstensi / <i>extend</i> <<extend>> 	Relasi usecase tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanppa <i>use case</i> tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan misal 

	arah panah mengarah pada <i>use case</i> yang ditambahkan
Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya misalnya :  Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
Menggunakan / <i>include</i> / <i>uses</i> <<include>>  <<user>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini Ada 2 sudut pandang yang cukup besar mengenai <i>include</i> di <i>usecase</i> 1. <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> dijalankan misal pada kasus berikut :  2. <i>Include</i> berarti <i>use case</i> 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 :  <pre> graph BT U1([Validasi user]) U2([Ubah data]) U2 -.-> <<include>> U1 </pre> Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.
--	--

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

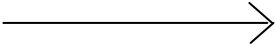
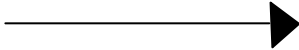
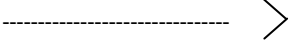
2. Class Diagram

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.

- Atribut merupakan *varabel-variabel* yang dimiliki oleh suatu kelas.
- 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

Tabel 11.2: Simbol Class Diagram	
Simbol	Deskripsi
Kelas Nama_kelas +atribut +operasi()	Kelas pada struktur sistem

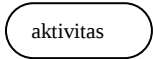
Antarmuka / Interface Nama_interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
asosiasi / association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah / directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
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

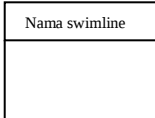
Sumber : Rosa A.S-M.Shalahuddin (2011 : 123)

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 contoh *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

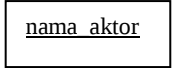
Penggabungan / join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

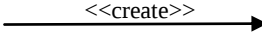
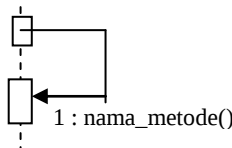
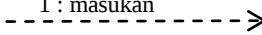
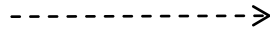
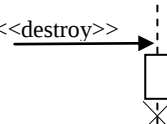
Sumber : Rosa A.S-M.Shalahuddin (2011 : 134)

5. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. *Diagram* ini menunjukkan 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 berinteraksi 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 1 : nama metode() 	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 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainya, arah panah mengarah pada objek yang dikirim
Pesan tipe return 1 : keluaran  Simbol	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima Deskripsi
Pesan tipe destroy 6 <<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-M.Shalahuddin (2011 : 138)

II.10 Macromedia Dreamweaver

Dreamweaver adalah perangkat lunak terkemuka untuk *desain web* yang menyediakan kemampuan visual yang intuitif termasuk pada tingkat kode, yang dapat digunakan untuk membuat dan mengedit *website* HTML serta aplikasi *mobile* seperti *smartphone*, *tablet*, dan perangkat lainnya. Dengan adanya fitur layout Fluid Grid yang dirancang khusus untuk memungkinkan lintas platform, maka akan membuat layout menjadi adaptif atau dapat menyesuaikan dengan browser yang dipakai dan juga dapat melihat hasil sementara desain dengan fasilitas Multiscreen Preview yang telah mengalami banyak perubahan dan peningkatan fungsionalitasnya. (Madcoms; 2013:1).

II.11. Studio Prewedding

Pada awal tahun 2000 foto *prewedding* hanya untuk kalangan sultan, raja, presiden dan para keluarga pejabat, karena dulunya *prewedding* dianggap tidak penting bagi sebagian kalangan masyarakat. Namun semakin berkembangnya teknologi dan mulai bermunculan kamera jenis SLR dan DSLR serta software editing yang membuat dan terjangkau bagi kalangan masyarakat pada saat ini.

Biasanya alasan seseorang melakukan *prewedding* selain untuk kebanggaan foto *prewedding* yang di desain eksotis dan penuh dengan keharmonisan dipercaya dapat menjadi obat paling ampuh ketika kedua mempelai nantinya mengalami proses sulit dalam menjalani rumah tangga. Sehingga *Prewedding* saat ini menjadi suatu hal yang selalu dilakukan oleh masyarakat sebelum melakukan suatu pernikahan.