

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Hal ini menjelaskan bahwa sistem bekerja dalam suatu prosedur yang saling berhubungan satu sama lain untuk menyelesaikan tujuan dan sasaran yang dimaksud. Sistem juga diartikan sebagai sekelompok elemen-elemen yang saling berinteraksi dengan maksud dan tujuan yang sama untuk melaksanakan sasaran yang telah ditentukan (Antonio, Safriadi, 2012:12). Selain itu, dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan yang terus menerus perlu dikembangkan dan dikendalikan.

Sistem Informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan (Antonio, Safriadi, 2012:12). Sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan.

Mengingat sistem informasi merupakan kumpulan dari beberapa elemen yang saling berhubungan untuk mencapai tujuan, maka sistem informasi terdiri dari beberapa komponen. Adapun komponen-komponen sistem informasi adalah sebagai berikut :

1. Perangkat keras (*hardware*), mencakup peranti-peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan penggunaan keluaran sistem informasi.
5. Basis data (*Database*), yaitu sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyampaian data.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

II.2. Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau *Geographic Information System* (GIS) merupakan suatu sistem berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi-informasi geografis (Eddy Prahasta, 2009:1). SIG dirancang untuk mengumpulkan , menyimpan, dan menganalisis objek-objek dan fenomena-fenomena dimana lokasi geografis merupakan

karakteristik yang penting atau kritis untuk dianalisis. Sistem Informasi Geografis merupakan sebuah sistem yang terdiri dari *software* dan *hardware*, data dan pengguna serta institusi untuk menyimpan data yang berhubungan dengan semua fenomena yang ada di muka bumi (Hamidi, 2012:1). Data-data yang berupa detail fakta, kondisi dan informasi disimpan dalam suatu basis data dan akan digunakan untuk berbagai macam keperluan. seperti analisis, manipulasi, penyajian dan sebagainya.

Teknologi SIG mengintegrasikan operasi-operasi umum *database*, seperti *query* dan analisa statistik, dengan kemampuan visualisasi dan analisa yang unik yang dimiliki oleh pemetaan. Kemampuan inilah yang membedakan SIG dengan Sistem Informasi lainnya yang membuatnya menjadi berguna di berbagai kalangan untuk menjelaskan kejadian, merencanakan strategi, dan memprediksi apa yang terjadi.

Dengan demikian, SIG merupakan sebuah sistem yang memiliki empat kemampuan dalam menangani data yang bereferensi geografis (Eddy Prahasta, 2009:1). Adapun keempat kemampuan SIG tersebut adalah sebagai berikut :

1. Masukkan

Kemampuan ini berperan untuk memasukkan data dan mengubah data asli ke bentuk yang dapat diterima dan dipakai dalam SIG. Semua data dasar geografis diubah dulu menjadi data *digital*, sebelum dimasukkan ke komputer. Data *digital* memiliki kelebihan dibandingkan dengan peta (garis, area) karena jumlah data yang disimpan lebih banyak dan pengambilan kembali lebih cepat. Ada dua macam data dasar geografi, yaitu data spasial dan data atribut.

- a. Data spasial (keruangan), yaitu data yang menunjukkan ruang, lokasi atau tempat-tempat di permukaan bumi. Data spasial berasal dari peta analog, foto udara dan penginderaan jauh dalam bentuk cetak kertas.
- b. Data atribut (deskriptis), yaitu data yang terdapat pada ruang atau tempat. Atribut menjelaskan suatu informasi. Data atribut diperoleh dari statistic, sensus, catatan lapangan dan tabular (data yang disimpan dalam bentuk tabel) lainnya. Data atribut dapat dilihat dari segi kualitas, misalnya kekuatan pohon. Dan dapat dilihat dari segi kuantitas, misalnya jumlah pohon.

Data dasar yang dimasukkan dalam SIG diperoleh dari tiga sumber, yaitu data lapangan (*teristris*), data peta dan data penginderaan jauh.

- a. Data lapangan (*teristris*)

Data teristris adalah data yang diperoleh secara langsung melalui hasil pengamatan di lapangan, karena data ini tidak terekam dengan alat penginderaan jauh. Misalnya, batas administrasi, kepadatan penduduk, curah hujan, jenis tanah dan kemiringan lereng.

- b. Data peta

Data peta adalah data yang digunakan sebagai masukan dalam SIG yang diperoleh dari peta, kemudian diubah ke dalam bentuk digital.

- c. Data penginderaan jauh

Data ini merupakan data dalam bentuk citra dan foto udara. Citra adalah gambar permukaan bumi yang diambil melalui satelit.

Sedangkan fotou dara adalah gambar permukaan bumi yang diambil melalui pesawat udara.

2. Keluaran

Komponen keluaran (*output*) berfungsi untuk menayangkan informasi geografi sebagai hasil analisis data dalam proses SIG. Informasi tersebut ditayangkan dalam bentuk peta, tabel, bagan, gambar, grafik dan hasil perhitungan.

3. Manajemen data

Subsistem ini mengorganisasikan data maupun tabel atribut terkait ke dalam sebuah sistem basis data agar mudah di-*retrieve*, di-*update*, dan di-*edit*.

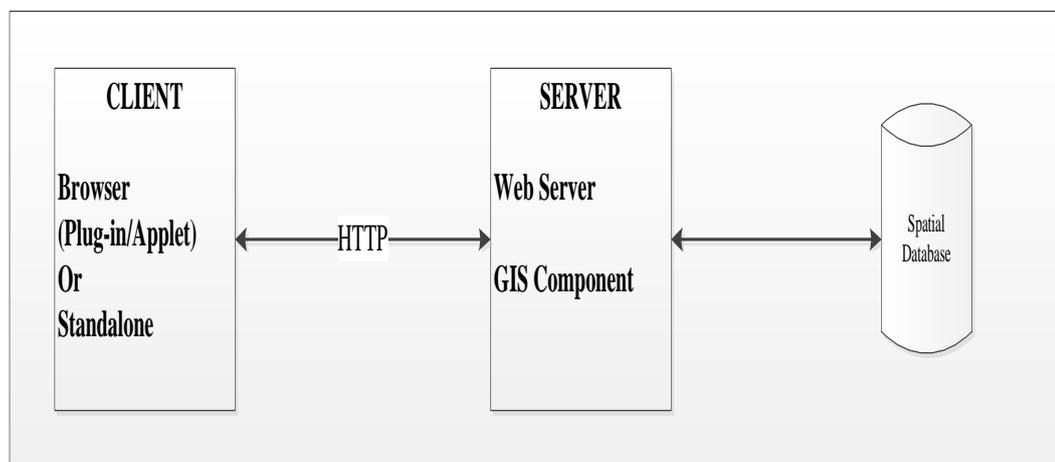
4. Analisis dan manipulasi data

menentukan informasi-informasi yang dihasilkan oleh SIG, melakukan manipulasi dan pemodelan data. Ada beberapa macam analisa data, antara lain:

- a. Analisis lebar, analisis yang dapat menghasilkan gambaran daerah tepian sungai dengan lebar tertentu. Kegunaannya antara lain untuk perencanaan pembangunan bendungan sebagai penanggulangan banjir.
- b. Analisis penjumlahan aritmatika (*arithmetic addition*), analisis ini digunakan untuk menangani peta dengan klasifikasi, hasilnya menunjukkan peta dengan klasifikasi baru.
- c. Analisis garis dan bidang, analisis ini dapat digunakan untuk menentukan wilayah dalam radius tertentu. Misalnya, daerah rawan banjir, daerah rawan gempa dan daerah rawan penyakit.

II.2.1. Arsitektur Web GIS

GIS merupakan suatu sistem yang saling terintegrasi satu sama lain, dan tidak dapat berdiri sendiri. Pada penelitian ini, penulis menggabungkan sistem GIS dengan komponen dalam *web* untuk membangun sebuah *web* GIS. Untuk dapat melakukan komunikasi dengan komponen lain yang berbeda di lingkungan *web* maka dibutuhkan sebuah *server*. Karena standart dari data geografis berbeda-beda dan sangat spesifik, maka pengembangan arsitektur sistem *web* GIS mengikuti arsitektur *client server*.



Gambar II.1. Arsitektur Web GIS

(Sumber : Messas dan Munadi, 2012:3)

Aplikasi berada disisi *client* yang berkomunikasi dengan *server* sebagai penyedia data melalui *web* protocol seperti HTTP (*Hyper Text Transfer Protocol*). Aplikasi seperti ini bisa dikembangkan dengan *web browser* (*Firefox*, *Chrome*, dll). Untuk menampilkan dan berinteraksi dengan data GIS, sebuah *browser* membutuhkan *plug-in* atau *java applet* atau bahkan keduanya. *Web server* bertanggung jawab terhadap proses permintaan dari *client* dan mengirimkan tanggapan terhadap respon tersebut.

II.3. Basis Data

Basis data adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisasikan/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya. Basis data adalah sekumpulan file. Definisi umum dari basis data adalah bahwa basis data merupakan kumpulan dari seluruh data berbasis komputer sebuah organisasi/perusahaan. Definisi basis data yang lebih sempit adalah bahwa basis data merupakan kumpulan data yang berada di bawah kendali peranti perangkat lunak sistem manajemen basis data (Raymond, George, 2008:158).

Sistem basis data adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan dan merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga pemakai mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan (Marlinda, 2004:1).

Sistem basis data terdiri dari beberapa komponen yang saling berhubungan satu sama lain. Adapun komponen-komponen dasar pada sistem basis data adalah sebagai berikut (Marlinda, 2004:2) :

1. Data

Data di dalam sebuah basis data dapat disimpan secara terintegrasi dan dapat pula dipakai secara bersama-sama. Terdapat tiga jenis data, yaitu :

- a. Data Operasional
- b. Data Masukkan

c. Data Keluaran

2. *Hardware* (Perangkat Keras)

Terdiri dari semua peralatan komputer yang digunakan untuk pengelolaan sistem basis data yang berupa: peralatan untuk penyimpanan, peralatan masukan dan keluaran, dan peralatan komunikasi data.

3. *Software* (Perangkat Lunak)

Berfungsi sebagai perantara antara pemakai dengan data fisik pada basis data. Perangkat lunak dalam basis data berupa *Database Management System* (DBMS) atau program aplikasi prosedur.

4. *User* atau Pemakai

Pemakai basis data dibagi atas tiga klasifikasi, yaitu :

a. *Database Administrator* (DBA), merupakan orang atau tim yang bertugas untuk mengelola sistem basis data secara keseluruhan. Adapun tugas dari DBA adalah sebagai berikut :

- 1) Mengontrol DBMS dan *software-software*.
- 2) Memonitor siapa saja yang mengakses basis data.
- 3) Mengatur pemakaian basis data.
- 4) Memeriksa keamanan, *integrity*, *recovery* dan *concurrency*.

b. *Programmer* adalah orang atau tim yang bertugas membuat program aplikasi, misalnya untuk perbankan, administrasi dan lain-lain.

c. *End User* adalah orang yang mengakses basis data melalui terminal dengan menggunakan *query language* atau program aplikasi yang telah dibuat oleh *programmer*.

II.3.1. *Entity Relationship Diagram*

Entity relationship diagram terbentuk karena didukung oleh beberapa komponen yang saling berhubungan satu sama lain. Komponen-komponen yang terdapat pada model *entity relationship* adalah sebagai berikut :

1. *Entity*

Entity adalah suatu yang dapat dibedakan dalam dunia nyata dimana informasi yang berkaitan dengannya dikumpulkan. Simbol *entity* adalah persegi panjang.

2. *Relationship*

Merupakan hubungan yang terjadi antara satu atau lebih *entity*.

3. *Attribute*

Attribute merupakan karakteristik dari *entity* atau *relationship* yang menyediakan penjelasan detail tentang hal tersebut. Nilai *attribute* adalah suatu data yang aktual.

4. Indikator Tipe

Indikator tipe ada dua, yakni : indikator tipe *associative object* dan indikator tipe supertipe.

5. *Cardinality Rasio*

Menjelaskan hubungan batasan jumlah keterhubungan satu *entity* dengan *entity* lainnya atau banyaknya *entity* yang bersesuaian dengan *entity* yang lain melalui *relationship*.

6. Derajat *Relationship*

Derajat *Relationship* menyatakan jumlah *entity* yang berpartisipasi di dalam suatu *relationship*.

7. *Participation Constraint*

Participation Constraint, menjelaskan apakah keberadaan suatu *entity* tergantung pada hubungannya dengan *entity* yang lain.

8. Representasi dari *Entity Set*

Entity set dipresentasikan dalam bentuk tabel dan nama yang *unique*. Setiap tabel terdiri dari sejumlah kolom dan diberi nama yang *unique*.

II.3.2. Normalisasi

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Teori normalisasi secara umum merupakan satu set peraturan yang membenarkan perkara pangkalan data mengenal pasti kes-kes perkumpulan data yang tidak memuaskan dan menentukan hubungan yang boleh ditukar menjadi bentuk yang lebih cekap. Untuk menggunakan normalisasi yang baik, pangkalan data harus mengetahui maksud data-data. Ada beberapa bentuk normal (Marlinda, 2004:122-123), yaitu:

1. Bentuk Tidak Normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan untuk mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terduplikasi. Data tersebut dikumpulkan apa adanya.

2. Bentuk Normal Pertama (1NF)

Suatu relasi 1NF jika dan hanya jika sifat dan setiap relasi atributnya bersifat atomik. Ciri-ciri bentuk normal pertama, yaitu :

- a. Setiap data dibentuk dalam *flat file*.
- b. Tidak ada *set* atribut yang berulang atau bernilai ganda.

c. Tiap *field* hanya satu pengertian.

3. Bentuk Normal Kedua (2NF)

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal pertama. Atribut bukan kunci haruslah bergantung secara fungsi pada *primary key*. Peraturan ini menentukan kebergantungan sepenuhnya. Beberapa sumber teks menjelaskan sebagai kebergantungan secara fungsi dan transitif.

4. Bentuk Normal Ketiga (3NF)

Satu hubungan dikatakan dalam bentuk 3NF jika dan hanya jika ia dalam bentuk 2NF dan setiap atribut tanpa kunci pula bergantung secara tidak transitif dengan kunci primer.

II.4. *Hypertext Preprocessor* (PHP)

Hypertext preprocessor atau PHP adalah salah satu jenis bahasa pemrograman *web* yang open source, sehingga dapat digunakan oleh siapa saja secara cuma-cuma (Wiswakarman, 2009:12). PHP juga merupakan bahasa *server-sidescripting* yang bisa menyatu dengan tag-tag HTML. *Server-sidescripting* adalah sintaks dan perintah-perintah yang dijalankan pada server dan disertakan pada dokumen HTML. PHP berfungsi sebagai bahasa pemrograman yang menjalankan suatu perintah tertentu. PHP mampu mengelolah data pada berbagai *platform database*, namun yang paling ideal dan banyak digunakan adalah menggunakan database MySQL. PHP+MySQL menjadi standar bagi pembuatan web dinamis saat ini, hal ini dikarenakan keduanya *open source*, sehingga bisa digunakan siapa saja dengan bebas.

Dalam penulisan *script* PHP, terdapat *tag* pembuka dan *tag* penutup. *Tag* ini merupakan sebagai identifer PHP itu sendiri. Sehingga ketika PHP digabungkan dengan *scripting* yang lain, seperti HTML dan *Java Script* maka *tag* pembukan dan *tag* penutup inilah yang akan memisahkan PHP dengan bahasa pemograman yang lainnya. Selain *tag* tersebut, PHP juga mempunyai variabel. Variabel dalam PHP tidak seperti variabel dalam bahasa pemograman lain. Variabel PHP tidak membutuhkan deklarasi sebelum digunakan (Muklis, Nugroho, 2009:314). Dengan begitu nilai dapat dimasukkan kapanpun untuk digunakan. Penulisan variabel PHP diawali dengan simbol ”\$”.

Pemograman PHP tidak terlepas dari *control flow*. Hal ini dikarenakan PHP mampu bermain dengan logika. Pada Tabel II.1. akan menjelaskan jenis-jenis *control flow* pada PHP.

Tabel II.1. Control Flow Pada PHP

<i>Statement Type</i>	<i>Keywords</i>
<i>Looping</i>	<i>While, do-while, for.</i>
<i>Decision making</i>	<i>If-else, switch-case</i>
<i>Exception handling</i>	<i>Try-catch, finally, throw</i>
<i>Branching</i>	<i>Break, continue, label, return.</i>

(Sumber : Muklis, Nugroho, 2009:314)

II.5. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek (Adi Nugroho, 2010:6). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan

mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML juga merupakan salah satu alat yang sangat handal di dunia pengembangan system yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Ada beberapa pengklasifikasi (*Classifier*) dan notasi dalam UML (Adi Nugroho, 2010) yang ditunjukkan pada Tabel II.2.dibawah ini.

Tabel II.2. Klasifikasi dan Notasi UML

Pengklasifikasi	Kegunaan	Notasi
<i>Actor</i>	Menggambarkan semua objek diluar sistem yang berinteraksi dengan sistem yang dikembangkan.	
<i>Use Case</i>	Menggambarkan fungsionalitas yang dimiliki sistem.	
Kelas (<i>Class</i>)	Menggambarkan konsep dasar pemodelan sistem.	
Subsistem (<i>Subsystem</i>)	Menggambarkan paket spesifikasi serta implementasi.	
Komponen (<i>Component</i>)	Menggambarkan bagian-bagian fisik sistem/perangkat lunak yang dikembangkan.	
Antarmuka (<i>Interface</i>)	Menggambarkan antarmuka pengiriman pesan (<i>message</i>) antar pengklasifikasi.	
Simpul (<i>Node</i>)	Menggambarkan sumber daya komputasional yang digunakan oleh sistem.	

(Sumber : Adi Nugroho, 2010:16)

Relasi-relasi antar pengklasifikasi yang dikenali UML adalah asosiasi, generalisasi, aliran, dan berbagai jenis kebergantungan termasuk didalamnya realisasi dan penggunaan. Untuk lebih jelasnya dapat dilihat pada Tabel II.3.berikut ini :

Tabel II.3. Relasi-relasi dalam UML

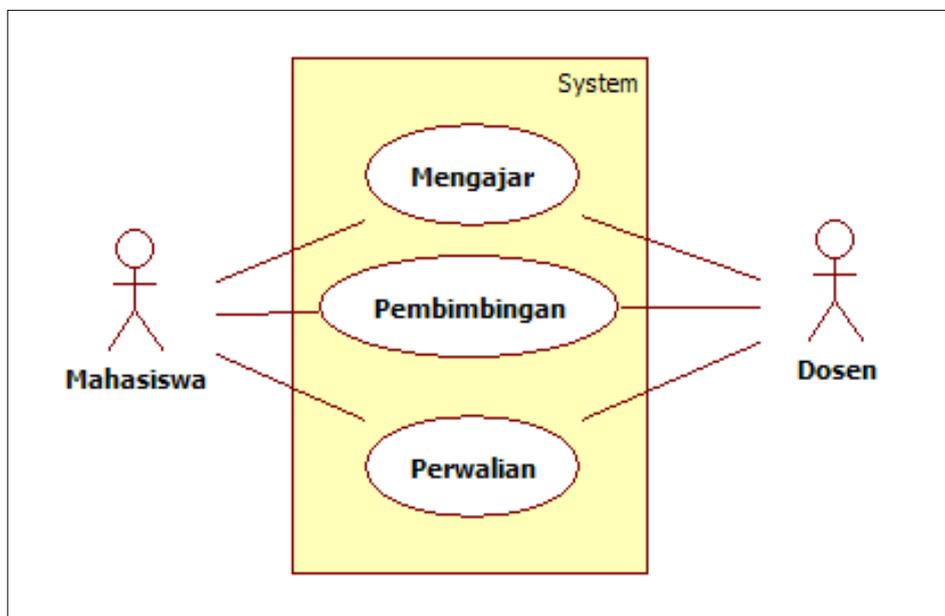
Relasi	Fungsi	Notasi
Asosiasi (<i>Association</i>)	Mendeskripsikan hubungan antar instance suatu kelas.	————
Kerbergantungan (<i>Dependency</i>)	Relasi antar dua elemen model.	----->
Aliran (<i>Flow</i>)	Relasi antar dua versi suatu model.	----->
Generalisasi (<i>Generalization</i>)	Relasi antar pengkalisifikasi yang memiliki deskripsi yang bersifat lebih umum dengan berbagai pengklasifikasi yang lebih spesifik, digunakan dalam struktur pewarisan.	————→
Realisasi (<i>Realization</i>)	Relasi antara spesifikasi dan implementasinya.	-----→
Penggunaan (<i>Usage</i>)	Situasi di mana salah satu elemen membutuhkan elemen yang lainnya agar dapat berfungsi dengan baik.	----->

(Sumber : Adi Nugroho, 2010:23)

II.5.1. Use Case Diagram

Diagram ini bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna. *Use Case Diagram* adalah fungsionalitas atau persyaratan -persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan tersebut menurut pandangan pemakai sistem (Sholih, 2010:21). *Use case* diagram juga dapat diartikan sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara

manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval*.

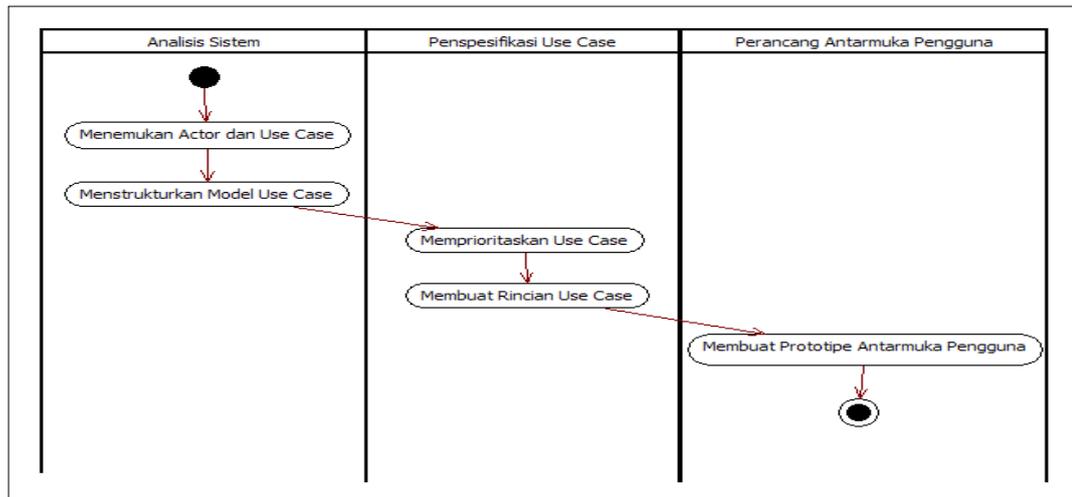


Gambar II.2. Contoh Use Case Diagram

(Sumber : Adi Nugroho, 2010:34)

II.5.2. Activity Diagram

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses (Sulistyorini, 2009:27). Diagram ini sangat mirip dengan sebuah *flowchart* karena dapat dimodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Seringkali bermanfaat bila dibuat sebuah *activity* terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity diagram* juga sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi.



Gambar II.3. Contoh Activity Diagram

(Sumber : Adi Nugroho, 2010:141)

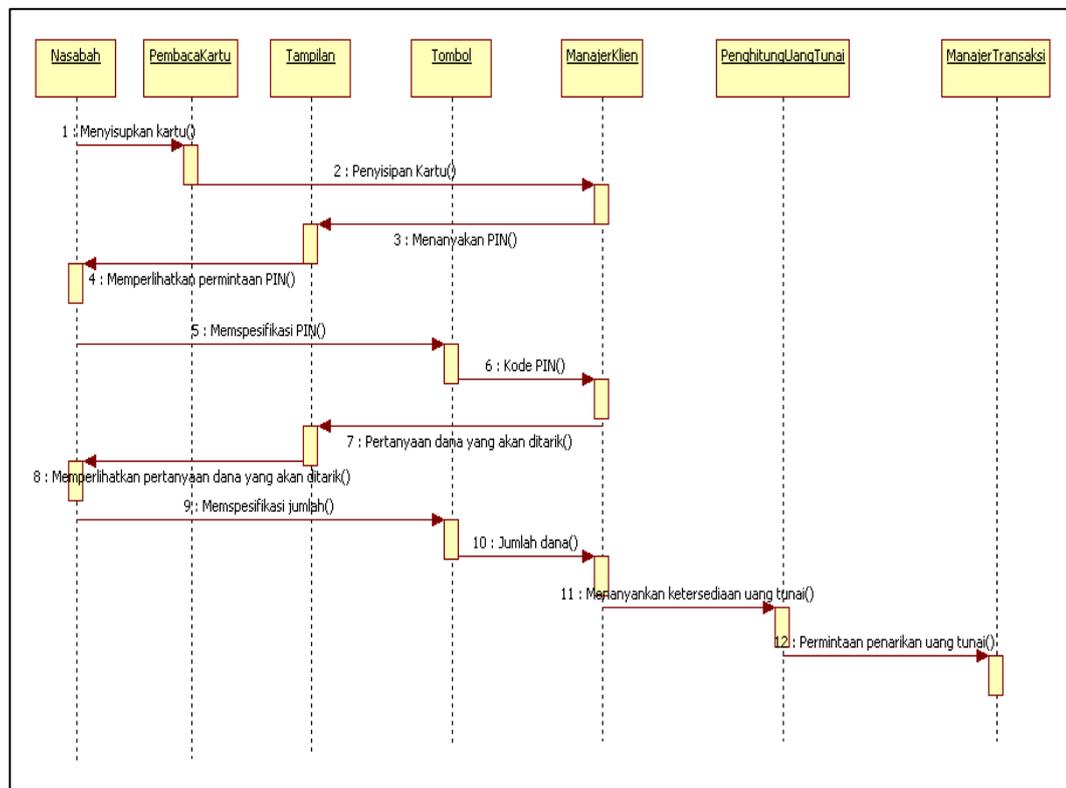
Dapat digunakan *state Chart* diagram untuk memodelkan perilaku dinamis satu kelas atau objek. *State Chart* diagram memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu *state* atau aktivitas ke *state* atau aktivitas lainnya, dan aksi yang menyebabkan perubahan satu *state* lainnya, dan aksi yang menyebabkan perubahan satu *state* atau aktivitas. Diagram aktivitas paling cocok digunakan untuk memodelkan aktivitas dalam suatu proses.

II.5.3. Sequence Diagram

Sequence Diagram memperlihatkan interaksi sebagai diagram dua mantra (dimensi). Mantra vertical adalah sumbu waktu; waktu bertambah dari atas ke bawah sedangkan mantra horizontal memperlihatkan pengklasifikasi yang mempresentasikan objek-objek mandiri yang terlibat dalam kolaborasi.

Diagram Sequence merupakan diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu (Sulistiyorini, 2009:24).

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek ini dalam *use case*.



Gambar II. 4. Contoh *Sequence Diagram*

(Sumber : Adi Nugroho, 2010:109)

Ada beberapa komponen yang terdapat pada *sequence diagram*, yaitu :

1. Objek/*Participant*

Objek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Objek ini diatur dalam urutan guna menyederhanakan diagram. Setiap *participant* terhubung dengan garis titik-titik yang disebut dengan *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili

sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*.

2. *Message*

Sebuah *message* bergerak dari satu *participant* ke *participant* yang lain dan dari satu *lifeline* ke *lifeline*. Sebuah *participant* bisa mengirim sebuah *message* kepada dirinya sendiri. Jika sebuah *participant* mengirimkan sebuah *message synchronous*, maka jawaban atas *message* tersebut akan ditunggu sebelum diproses dengan urusannya. Namun jika *message synchronous* yang dikirimkan, maka jawaban atas *message* tersebut tidak perlu ditunggu.

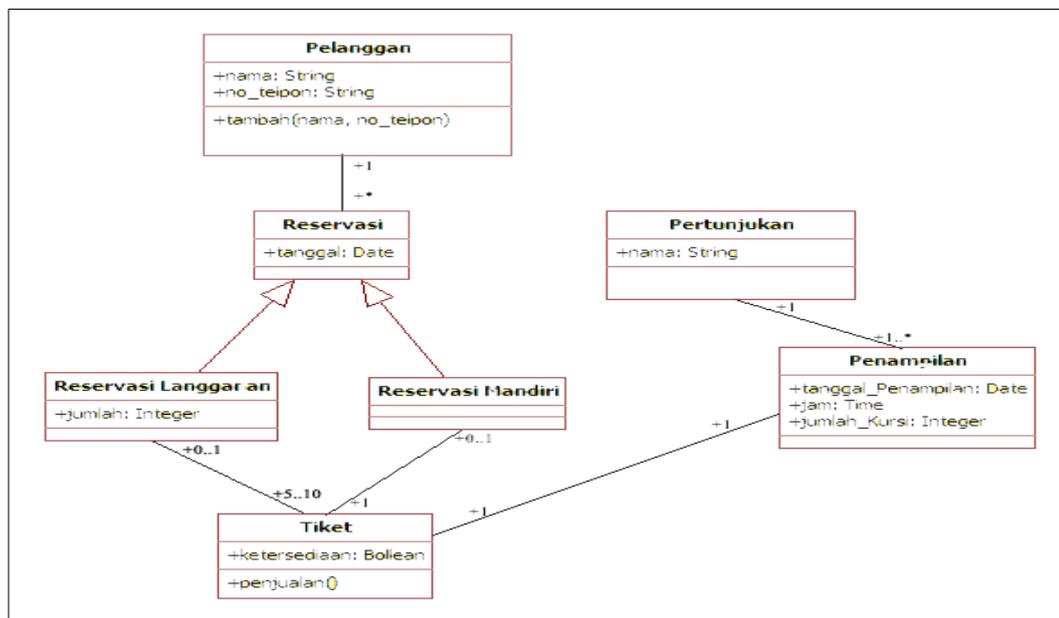
3. *Time*

Time adalah diagram yang mewakili waktu pada arah *vertical*. Waktu dimulai dari atas ke bawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibanding *message* yang lebih dekat ke bawah.

II.5.4. *Class Diagram*

Class Diagram menunjukkan interaksi antar kelas-kelas dalam sistem. Sebuah kelas mengandung informasi dan tingkah laku (*behavior*) yang berkaitan dengan informasi tersebut. *Class Diagram* sesungguhnya merupakan deskripsi dari konsep yang datang dari arah aplikasi atau solusi aplikasi (Adi Nugroho, 2010 :11). Oleh karena itu pengertian kelas sangat penting sebelum merancang diagram kelas. Kelas diagram sebagai satu *set* objek yang memiliki atribut dan perilaku yang sama. *Class diagram* membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak.

Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.



Gambar II.5. Contoh Class Diagram

(Sumber : Adi Nugroho, 2010:12)

II.6. MySQL Server

MySQL adalah database server relasional yang gratis di bawah lesensi GNU (*General Public License*). Dengan sifatnya *Open Source*, memungkinkan juga user untuk melakukan modifikasi pada *source code*-nya untuk memenuhi kebutuhan spesifik. *MySQL* merupakan database server *multi-user* dan *multi-threaded* yang tangguh (*robust*). Dengan memiliki banyak fitur *MySQL* bisa bersaing dengan *database* komersil lainnya. *MySQL* dikembangkan,

disebarluaskan, dan didukung oleh *MySQL* AB yang merupakan perusahaan komersial yang didirikan oleh para pengembang *MySQL* (Wahana, 2010:26).

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data *SQL* (*database management system*) atau *DBMS* yang memiliki konsep *multi-threaded* dan *multi-user* serta menggunakan perintah *SQL* (*Structured Query Language*). *MySQL* merupakan sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat kedalam sebuah sistem aplikasi (*Client*).

II.6.1. Tipe Data *MySQL Server*

MySQL Server mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Terdapat tiga kategori tipe data yang didukung oleh *MySQL Server*, yaitu :

1. Tipe Data Numerik

Data numerik adalah salah satu bentuk data berupa angka, baik berupa bilangan bulat maupun bilangan *real*.

Tabel II.4. Tipe Data Numerik Bilangan Bulat

Tipe Data	Byte	Nilai Minimal	Nilai Maksimal
<i>Tinyint</i>	1	-128	127
		0	255
<i>Smallint</i>	2	-32768	32767
		0	65535
<i>Mediumint</i>	3	-8388608	8388607
		0	16777215
<i>Int/Integer</i>	4	-2147483648	21455483648
		0	4294967295
<i>Bigint</i>	8	-9223372036854775808	9223372036854775807
		0	184467440373709551615

(Sumber : Wahana, 2010:31)

Tabel II.5. Tipe Data Numerik Bilangan Real

Tipe Data	Byte	Keterangan
<i>Float</i> (p)	4 jika $0 \leq p \leq 24$	P mempresentasikan presisi bit.
<i>Float</i>	4	Angka <i>floatingpoint</i> kecil
<i>Double</i>	8	Ukuran normal angka <i>floating point</i>
<i>Decimal</i> (M,D), <i>Numeric</i> (M,D)	Variasi	M adalah jumlah angka digit <i>decimal</i> dan D adalah angka dibelakang <i>decimal</i>
Bit (M)	(M+7)/8	M adalah banyaknya bit setiap nilai.

(Sumber : Wahana, 2010:31)

2. Tipe Data String

Tipe data string dapat menyimpan semua data, baik berupa karakter, angka, waktu maupun tanggal. Data dapat pula merupakan kombinasi karakter dan angka.

Tabel II.6. Tipe Data String

Tipe Data	Byte	Keterangan
<i>Varchar</i>	255	Tipe <i>varchar</i> menyimpan data sebanyak karakter yang diinputkan.
<i>Char</i>	255	Tipe <i>char</i> sama dengan <i>varchar</i> , hanya saja tempat penyimpanan selalu tetap.
<i>Binary</i>	255	<i>Binary</i> mirip dengan <i>char</i> hanya yang disimpan adalah nilai biner dari data yang disimpan.
<i>Varbinary</i>	255	<i>Varbinary</i> sama dengan <i>binary</i> , tetapi keduanya berbeda sebagaimana perbedaaan <i>char</i> dengan <i>varchar</i> .
<i>Enum</i>	N	Tipe data ini disebut juga dengan tipe data validasi. Pada tipe ini data inputan telah dideklarasikan terlebih dahulu.
<i>Set</i>	N	<i>Set</i> memiliki fungsi yang sama dengan <i>enum</i> .

(Sumber : Wahana, 2010:33)

3. Tipe Data Penanggalan dan Waktu

Dalam menangani data tanggal dan waktu (jam), *MySQL* memiliki tipe data tersendiri.

Tabel II.7. Tipe Data Tanggal dan Waktu

Tipe Data	Byte	Keterangan
<i>Datetime</i>	8	Bentuk ini merupakan tipe data yang menyimpan dua tipe, yaitu tanggal dan jam.
<i>Date</i>	3	Tipe ini hanya menyimpan data tanggal dengan format "0000-00-00".
<i>Timestamp</i>	4	Tipe ini ditulis berjajar tanpa ada pembatas. Tipe ini dapat menyimpan tanggal dan jam.
<i>Time</i>	3	Tipe ini hanya dapat menyimpan data jam.
<i>Year</i>	1	Tipe ini hanya menyimpan data tahun saja.

(Sumber : Wahana, 2010:33-34)

II.6.2. Internal dan Portabilitas MySQL Server

Salah satu fitur yang terdapat pada *MySQL* adalah fitur *internal* dan *portabilitas*. Menurut Wahana Komputer (2010:27-28), ada beberapa fitur *internal* dan *portabilitas* sebagai salah satu fitur utama pada *MySQL*, antara lain :

1. *MySQL Server* ditulis dengan bahasa C dan C++.
2. *MySQL Server* diuji secara luas menggunakan *compiler* yang berbeda.
3. *MySQL Server* mampu bekerja pada banyak *platform* berbeda.
4. *MySQL Server* menggunakan GNU *Automake*, *Autoconf*, dan *Libtool* untuk *portabilitas*.
5. Tersedia API untuk C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, dan Tcl untuk mengakses *MySQL Server*.
6. *MySQL Server* secara penuh mendukung *multi-threaded* menggunakan *thread* kernel sehingga mudah menggunakan banyak CPU, jika tersedia.
7. *MySQL Server* menyediakan mesin penyimpanan transaksi dan non-transaksi.
8. *Thread MySQL Server* sangat cepat berdasarkan sistem alokasi memori.

9. Penggabungan tabel *MySQL Server* dapat dilakukan sangat cepat menggunakan optimasi *one sweep multi-join*.

II.7. Pengenalan *ArcView*

ArcView merupakan salah satu perangkat lunak *desktop* Sistem Informasi Geografis (SIG) dan pemetaan yang telah dikembangkan oleh *Environmental Systems Research Institute, Inc* (ESRI). Dengan *ArcView* kita dengan mudah dapat mengelola data, menganalisa dan membuat peta serta laporan yang berkaitan dengan data spasial bereferensi geografis. *ArcView* memiliki beberapa kemampuan (Eddy Prahasta, 2009:1-2), antara lain :

1. Pertukaran data,
2. Melakukan analisis statistik dan operasi-operasi matematis,
3. Menampilkan informasi (basis data) spasial maupun atribut,
4. Menjawab *query* spasial maupun atribut,
5. Melakukan fungsi-fungsi dasar SIG,
6. Membuat peta tematik, dll.

II.7.1. Arsitektur *ArcView*

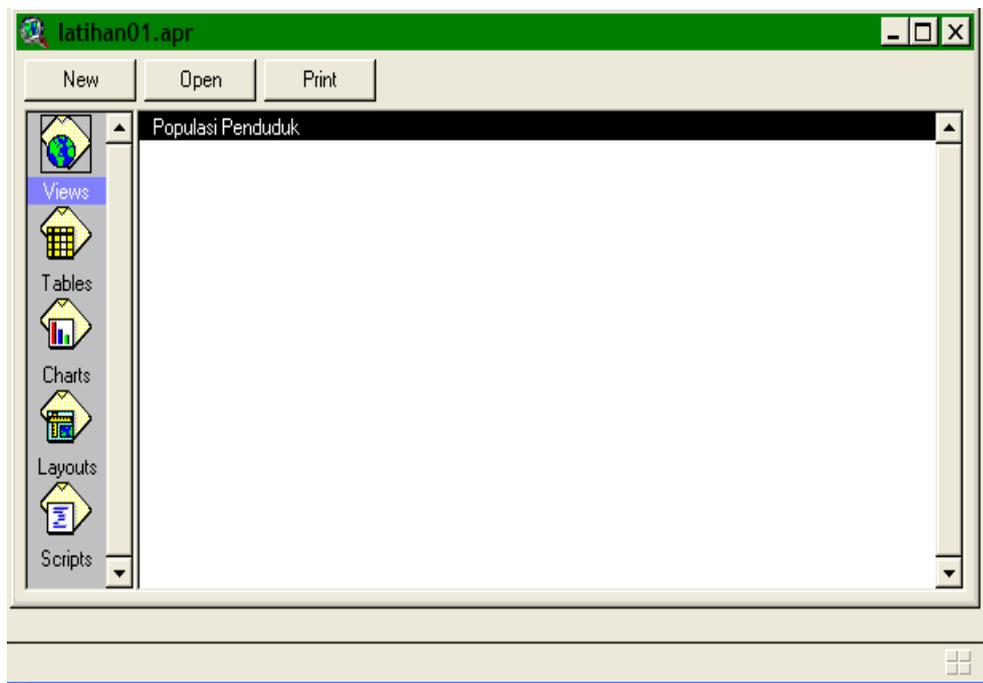
ArcView mengorganisasikan sistem perangkat lunaknya sedemikian rupa sehingga dapat dikelompokkan ke dalam beberapa komponen-komponen penting. Adapun komponen-komponen penting tersebut adalah sebagai berikut :

1. *Project*

Project merupakan suatu unit organisasi tertinggi di dalam *ArcView*. Merupakan kumpulan dari dokumen yang berasosiasi selama satu sesi. Sebuah

project mengorganisasikan suatu file yang kerja yang dapat digunakan untuk menyimpan, mengelompokkan semua komponen-komponen lainnya, seperti *view*, *theme*, *table*, *chart*, *layout* dan *script* dalam satu kesatuan yang utuh. *Layouts* untuk membuat komposisi peta yang akan dicetak dan *scripts* dipakai untuk membuat modul yang berisikan kumpulan perintah *ArcView* yang ditulis menggunakan bahasa pemrograman *Avenue*.

Adapun tampilan *project* dalam *ArcView* dapat di lihat pada gambar berikut ini :



Gambar II.6. Project Windows ArcView

(Sumber : Eddy Prahasta ,2009:5)

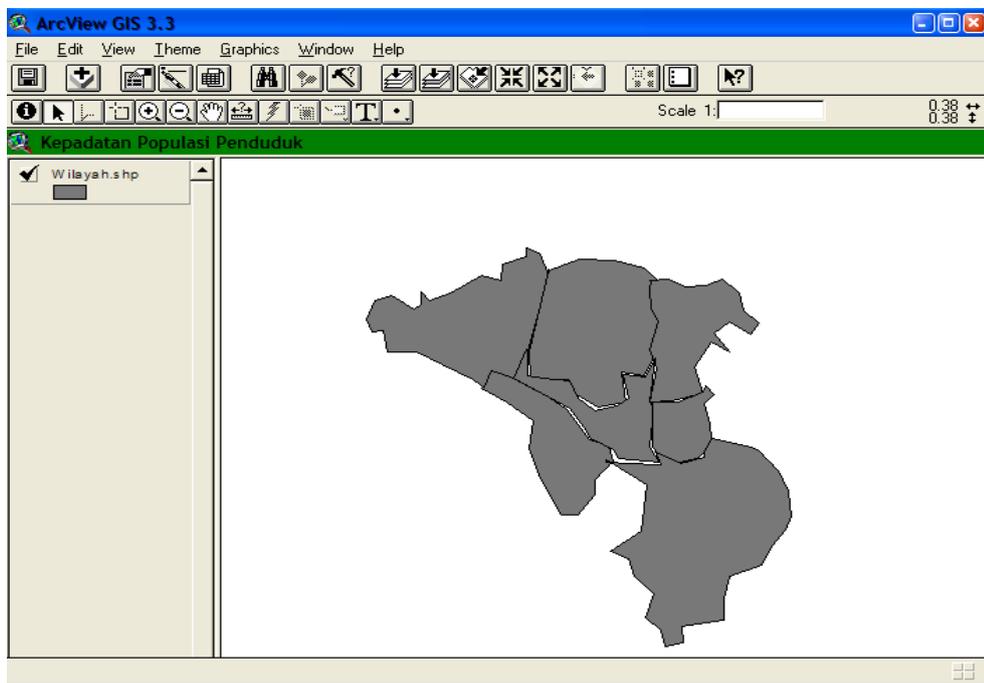
2. Theme

Theme adalah bangunan dasar dalam sistem *ArcView* dan merupakan kumpulan dari beberapa *layer ArcView* yang membentuk ‘tematik’ tertentu.

3. View

View mengorganisasikan *theme* dan dapat menampung beberapa *layer* atau *theme* informasi spasial (titik, garis, *polygon* dan citra *raster*).

Arcview mengendalikan sekelompok *feature* serta atribut di dalam sebuah *theme* dan mengelolanya di dalam sebuah *views*. Sedangkan *theme* menyajikan sekumpulan objek nyata sebagai *feature* peta yang berhubungan dengan atribut. *Feature* dapat berupa titik (*points*), garis (*lines*) maupun *polygon*. Berikut ini adalah tampilan gambar *view* yang menampung *theme*.



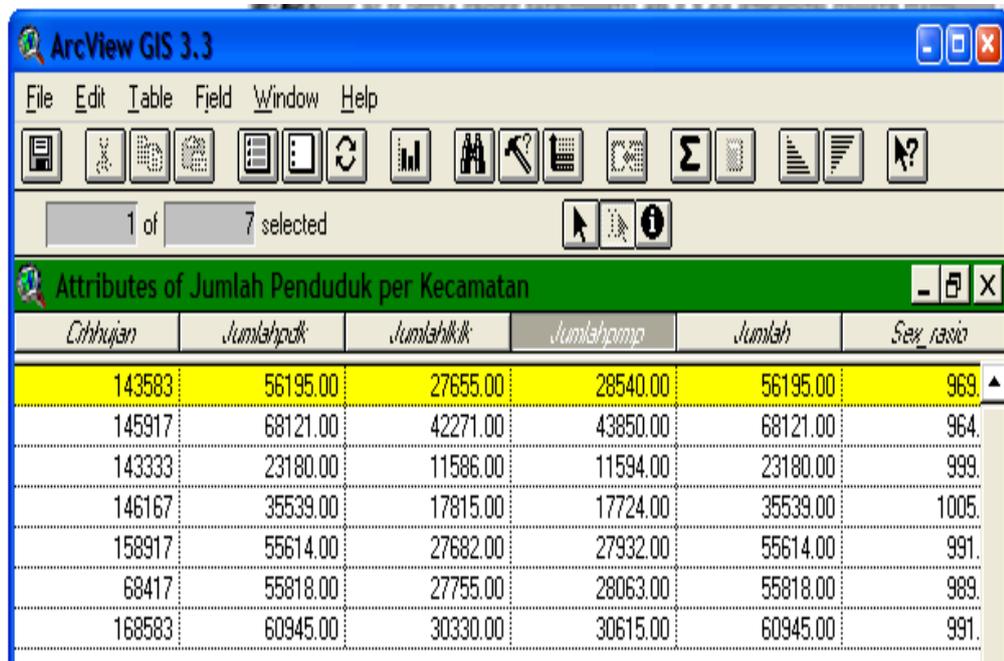
Gambar II.7. Tampilan View ArcView

(Sumber : Eddy Prahasta, 2009)

4. Table

Table digunakan untuk menampilkan informasi tentang *feature* yang ada didalam suatu *view*. Sebagai contoh menjelaskan tentang propinsi Bali disiapkan tabel yang berisi data-data item nama kabupaten, jumlah penduduk

laki-laki, perempuan, total dan sebagainya. Adapun contoh tampilan *table* dalam *ArcView* dapat dilihat pada gambar berikut :



<i>Cekhujaan</i>	<i>Jumlahpadk</i>	<i>Jumlahlkk</i>	<i>Jumlahpmp</i>	<i>Jumlah</i>	<i>Sex_rasio</i>
143583	56195.00	27655.00	28540.00	56195.00	969
145917	68121.00	42271.00	43850.00	68121.00	964
143333	23180.00	11586.00	11594.00	23180.00	999
146167	35539.00	17815.00	17724.00	35539.00	1005
158917	55614.00	27682.00	27932.00	55614.00	991
68417	55818.00	27755.00	28063.00	55818.00	989
168583	60945.00	30330.00	30615.00	60945.00	991

Gambar II.8. Tampilan *Table ArcView*

(Sumber : Eddy Prahasta, 2009:7)

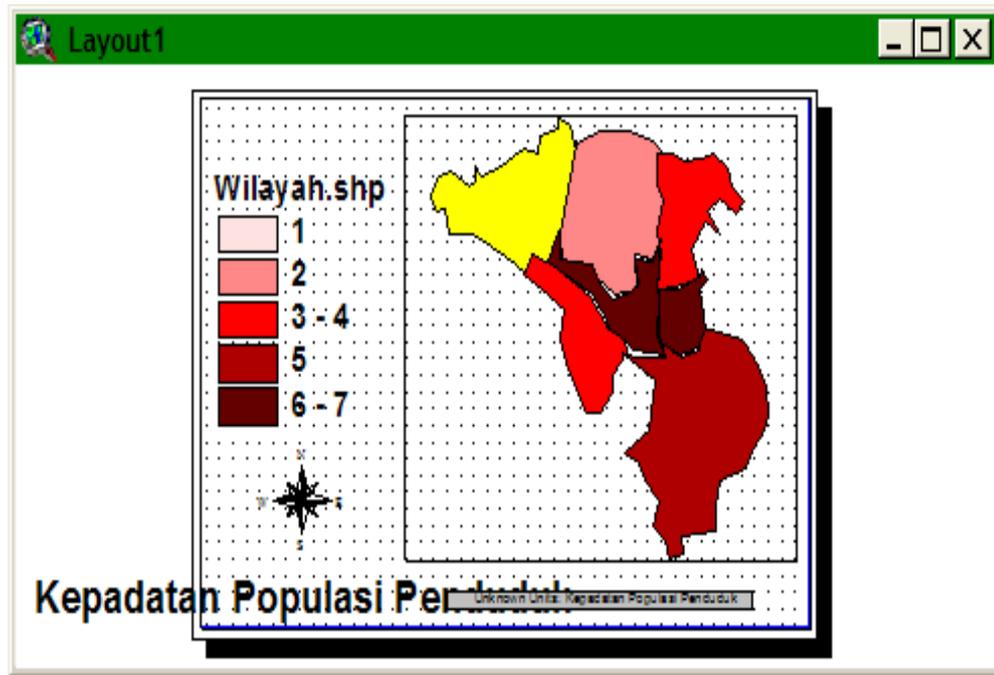
5. *Chart*

Chart merupakan sebuah grafik yang menyajikan data tabular. Di dalam *ArcView* *chart* terintegrasi penuh dengan tabel dan *view* sehingga dapat dilakukan pemilihan *record-record* mana yang akan ditampilkan ke dalam sebuah *chart*. Terdapat enam jenis *chart* yaitu *area*, *bar*, *column* dan *scatter*.

6. *Layout*

Layout digunakan untuk mengintegrasikan dokumen (*view*, *table*, *chart*) dengan elemen-elemen grafik yang lain di dalam suatu *window* tunggal guna membuat peta yang akan dicetak. Dengan *layout* dapat dilakukan proses penataan peta serta merancang letak-letak *property* peta seperti : judul, *legend*,

orientasi, *label* dan sebagainya. Adapun tampilan *layout windows ArcView* dapat dilihat pada gambar berikut ini :



Gambar II.9. Tampilan *Layout*

(Sumber : Eddy Prahasta, 2009:8)

7. *Script*

Script merupakan sebuah bahasa pemrograman dari *ArcView* yang ditulis ke dalam bahasa *Avenue*. Dengan *Avenue*, pengguna dapat memodifikasi tampilan (*user interface*) *ArcView*.