BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain dan terpadu.

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut. (Tata Sutabri ; 2012 : 22)

Sebuah sistem terdiri atas bagian-bagian atau komponen yang terpadu untuk suatu tujuan. Model dasar dari bentuk sistem adalah adanya masukan, pengolahan, dan keluaran. Akan tetapi, sistem ini dapatdikembangkan hingga menyetakan media penyimpanan. Teori sistem menyatakan bahwa setiap unsur pembentukan organisasi adalah penting dan harus mendapat perhatian yang utuh supaya manajer dapat bertindak lebih efektif. (Tata Sutabri ; 2012 : 11)

II.1.1. Karakteristik Sistem

Model umum sebuah sistem adalah input, proses, dan output. Hal ini merupakan konsep sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu, sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem.

Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem *(Components)*

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen tersebut berupa sebuah Subsistem. Setiap Subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

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 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 ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem *(interface)*

   Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface.* Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk

subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem *(Input)*

   Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemelihara *(MaintenanceInput)* dan *sinyal (Signal Input)*.

6. Keluaran Sistem *(Output)*

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

7. Pengolahan Sistem *(Proces)*

   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 *daterministic*. Jika suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang direncanakan. (Tata Sutabri ; 2012 : 21)

II.1.2. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya :

a. Sistem abstrak dan sistem fisik

   Sistem absrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem yang ada secara fisik.

b. Sistem alamiah dan sistem buatan manusia

   Sistem alamiah adalah sistem yang terjadi melalui proses lain tidak dibuat oleh manusia, sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine* sistem, misalnya sistem informasi berbasis komputer.

c. Sistem determinasi dan sistem probabilistik

   Sistem yang beroperasi dengan tingkahlaku yang dapat diprediksi disebut sistem *deterministic*. Sistem komputer adalah contoh dari sistem yang tingkahlakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan, sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilistic*.

d. Sistem terbuka dan sistem tertutup

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tampa campur tangan pihak luar, sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya. (Tata Sutabri ; 2012 : 22)

II.1.3. Daur Hidup Sistem

Siklus hidup sistem (*system life cycle*) adalah proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem karena tugas-tugas tersebut mengikuti pola teratur dan dilakukan secara *top down*. Siklus hidup sistem sering juga disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pembangunan dan pengembangan sistem. Beberapa fase atau tahapan dari daur hidup suatu sistem yaitu :

1. Mengenali adanya kebutuhan

   Kebutuhan dapat terjadi sebagai hasil perkembangan dari organisasi volume yang meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan dari kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektifitasnya.

2. Pembangunan sistem

   Suatu proses seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

3. Pemasangan sistem

   Setelah tahap pembangunan sistem selesai, sistem akan dioperasikan. Pemasangan sistem merupakan tahap yang penting dalam daur hidup sistem. Di dalam peralihan dari tahap pembangunan menuju tahap

operasional terjadi pemasangan sistem yang sebenarnya yang merupakan langkah akhir dari suatu pembangunan sistem.

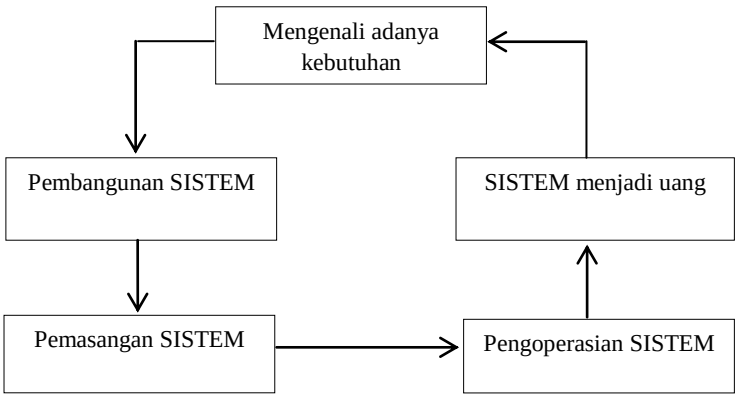
4. Pengoperasian sistem

   Program-progran komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi ditunjang oleh sistem informasi tadi. Ia selalu mengalami perubahan-perubahan itu karena pertumbuhan kegiatan bisnis, perubahan peraturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbaharui.

5. Sistem menjadi uang

   Kadang perubahan yang terjadi begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tibalah saatnya secara ekonomis dan teknik sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

Untuk dapat menggambarkan daur hidup sistem ini, lihat pada gambar dibawah ini.



Gambar II.1. Daur Hidup Distem

Sumber : Tata Sutabri (2012 : 29)

II.2. Pengertian Informasi

Informasi adalah data yang telah di klasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tidak berguna menjadi berguna bagi penerimanya. Nilai informasi berhubungan dengan 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. (TataSutabri ; 2012 : 29)

II.3. Sistem Informasi

Sistem informasi adalah suatu sistem suatu di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu

organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang di perlukan. (Tata Sutabri ; 2012 : 46)

II.4. Sistem Informasi Geografis (SIG)

II.4.1. Pengenalan Sistem Informasi Geografis (SIG)

Sistem informasi geografis merupakan gabungan dari tiga unsur pokok sistem, informasi dan geografis. Dengan demikian, pengertian terhadap ketiga unsur-unsur pokok ini akan sangat membantu dalam memahami SIG.

Istilah Geografis merupakan bagian dari spasial (keruangan). Penggunaan kata geografis mengandung pengertian suatu persoalan atau hal mengenai wilayah di permukaan bumi. Dengan demikian, istilah informasi geografis mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan dimana mengenai posisi di mana suatu objek terletak di permukaan bumi atau informasi mengenai keterangan-keterangan (atribut) objek penting yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui.

SIG dapat dikatakan sebagai satu kesatuan formal yang terdiri dati berbagai sumber daya fisik dan logika yang berkenan dengan objek-objek penting yang terdapat di permukaan bumi. Jadi, SIG juga merupakan sejenis perangkat lunak, perangkat keras (manusia, prosedur, basis data dan fasilitas jaringan komunikasi) yang dapat digunakan untuk memfasilitasi proses pemasukan, penyimpanan, manipulasi, menampilkan dan keluaran data/informasi geografis berikut atribut-atribut terkait. (Eddy Prahasta ; 2009 : 109)

II.4.2. Subsistem SIG

SIG dapat diuraikan menjadi beberapa subsistem sebagai berikut :

1. Data Inputan

   Subsistem ini bertugas untuk mengumpulkan, mempersiapkan, menyimpan data spasial dan atributnya dengan berbagai sumber. Subsitem ini bertanggungjawab dalam mengonversikan atau mentrasformasikan format-format danaslinya ke dalam format *(native)* yang dapat digunakan oleh perangkat SIG yang bersangkutan.

2. Data Output

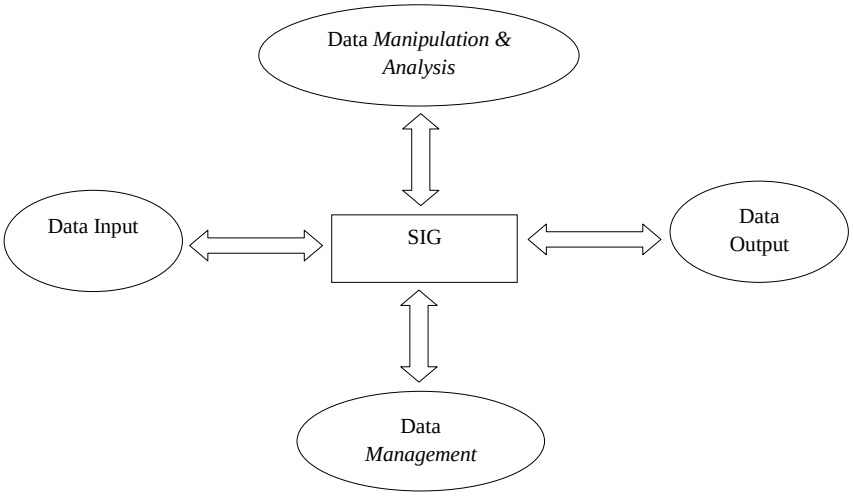
   Subsistem ini bertugas untuk menampilkan dan menghasilkan keluaran (termasuk mengekspornya ke format yang dikehendaki) seluruh atau sebagian basis data (spasial) baik dalam bentuk *softcopy* maupun *hardcopy* seperti halnya tabel, grap *report*, peta dan lainnya sebagainya.

3. Data Management

   Subsistem ini mengorganisasikan baik data spasial maupun tabel-tabel atribut terkait ke dalam sebuah sistem basis data sedemikian rupa hingga mudah dipanggil kembali atau *retrieve* (di *load* ke memori), di *update* dan di *edit*.

4. Data Manipulation & Analysis

   Subsistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Selain itu, sistem ini juga melakukan manipulasi (evaluasi dan penggunaan fungsi-fungsi dan operator matematis dan logika) dan pemodelan data untuk menghasilkan informasi yang diharapkan.



Gambar II.2. Ilustrasi Subsistem SIG

Sumber : Eddy Prahasta (2012 : 119)

II.4.3. Komponen SIG

SIG sebagai sistem terdidri dari beberapa komponen dengan berbagai karakteristiknya sebagai berikut :

1. Perangkat keras

   Pada saat ini SIG sudah tersedia bagi berbagai *platform* perangkat keras mulai dari PC *dekstop, workstation* hingga multi-*user host* yang bahkan dapat digunakan oleh banyak orang secara bersamaan (silmutan) dalam jaringan komputer yang luas, terbesar, berkemampuan tinggi, memiliki ruang penyimpanan (*hardisk*) yang besar dan mempunyai kapasistas memori (RAM) yang besar.Adapun perangkat keras yang sering digunakan untuk aplikasi SIG adalah komputer (PC), *mouse*, monitor (plus VGA-*card* grafik) yang beresolusi tinggi, *digitizer*, *printer*, *poltter*, *receiver* GPS, dan *scanner*.

2. Perangkat lunak

   SIG bisa juga merupakan sistem perangkat lunak yang tersusun modular dimana sistem basis datanya memegang peran kunci. Pada kasus perangkat SIG tertentu, setiap subsistem diinplementasikan dengan menggunakan perangkat lunak yang terdiri dari beberapa modul hingga tidak mengherankan jika ada perangkat SIG yang terdiri dari ratusan modul program (*.exe) yang masing-masing dapat di eksekusi tersendiri.

3. Data dan informasi geografi

   SIG dapat mengumpulkan dan menyimpan data atau informasi yang diperlukan baik secara tidak langsung (dengan cara meng-*import*-nya dari format-format perangkat SIG lain) maupun langsung dengan cara melakukan dijitasi data spasialnya (dijitasi *on-screean* atau *head-ups* di atas tampilan layar monitor, atau manual dengan menggunakan *digitizer*) dari peta analog dan kemudian memasukkan data atributnya dari tabel-tabel atau laporan dengan menggunakan *keyboard*.

4. Manajemen

   Suatu proyek SIG akan berhasil jika dikelola dengan baik dan dikerjakan oleh orang-orang memiliki keahlian (kesesuaian dengan *job-description* yang bersangkutan) yang tepat pada semua tingkatan. (Eddy Prahasta ; 2009 : 120)

II.5. *Unified Modelling Language* (UML)

II.5.1. Pengenalan UML

UML singkatan dari *Unified Modelling Langguage* yang berarti bahasa pemodelan standart. *UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorentasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensuport para pengembang sistem saat ini. (Prabowo Pudjo Widodo, dkk ; 2011 : 6)

II.5.2. Diagram UML

Beberapa literatur menyebutkan bahwa *UML* menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung.

Namun demikian model-model itu dapat dikelompokan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umu dijumpai pada pemodelan sistem berorentasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
2. Diagram paket (*Packagen Diagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* 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 suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram)* bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State)*, transisi kejadian serta aktifitas. Diagram ini terutama penting untuk

memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutam penting pada pemodelan sistem-sistem yang reaktif.

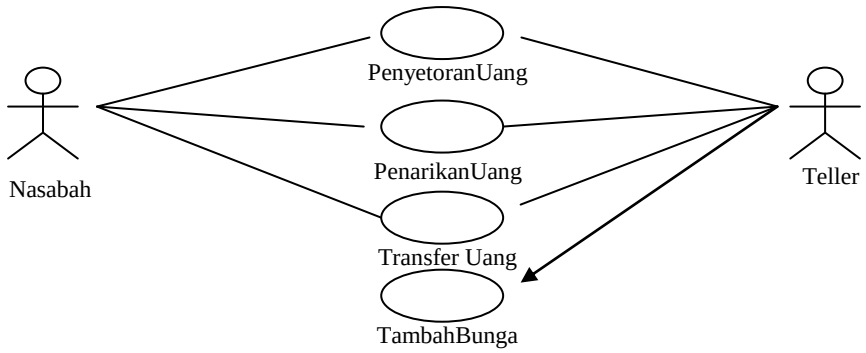
7. Diagram aktivitas (*Activity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram)* bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time).* Memuat simpul-simpul berserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing).* (Prabowo Pudjo Widodo, dkk ; 2011 : 10)

A. Diagram *Use Case* (*use case* diagram)

Use Case menggambarkan *external view* dari sistem yang akan kita buat modelnya. Pooley (2005) dalam buku (Prabowo Pudjo widodo,dkk ; Menggunakan UML ; 2011 : 16) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat. Komponen pembentuk diagram *use case* adalah :

1. Aktor (*actor),* menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use Case*, aktivitas/ sarana yang disiapkan oleh bisnis/sistem.
3. Hubungan (*Link),* aktor mana saja yang terlibat dalam *use case* ini.

Gambar di bawah ini merupakan salah satu contoh bentuk diagram *use case.*

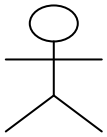


Gambar II.3. Diagram *Use Case*

Sumber : Probowo Pudjo Widodo, dkk (2011 : 17)

1. Aktor

Sebelum membuat *use case* dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder.*

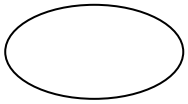


Gambar II.4. Aktor

Sumber : Probowo Pudjo Widodo, dkk (2011 : 17)

2. *Use Case*

Use case menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. *Use case* 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.

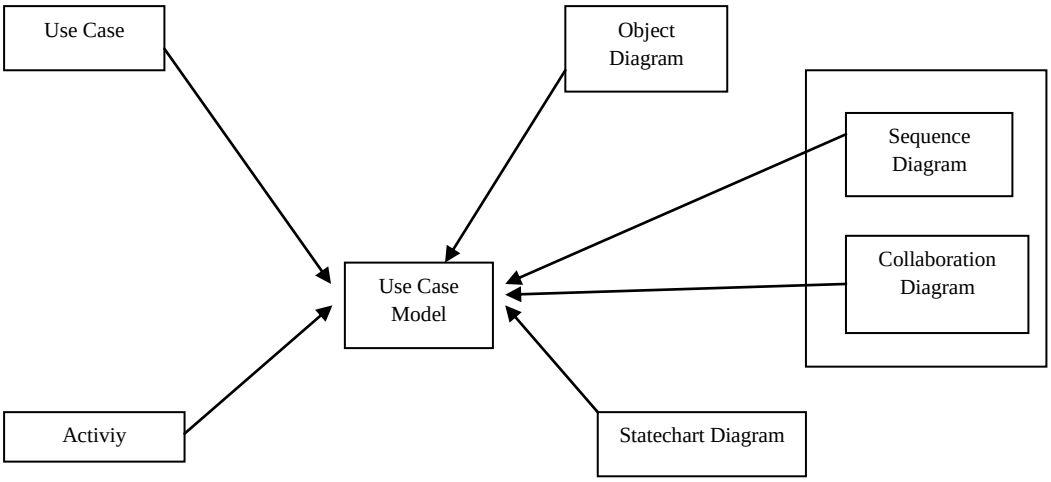


Gambar II.5. Simbol *Use Case*

Sumber : Probowo Pudjo Widodo, dkk (2011 : 22)

B. Diagram Kelas (*Class Diagram*)

Diagram kelas adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini *forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model. (Probowo Pudji Widodo, dkk ; 2011 : 37)



Gambar II.6. Hubungan Diagram Kelas Dengan Diagram *UML* lainya

Sumber : Probowo Pudjo Widodo, dkk (2011 : 38)

C. Diagram Aktivitas (*Activity Diagram*)

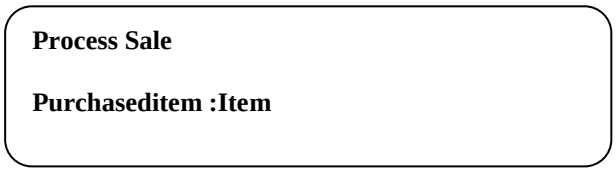
Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistemdari pada bagaimana sistem dirakit. Diagram aktivitas menunjukan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. (Probowo Pudji Widodo, dkk ; 2011 : 143)

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

a. Fungsi Matematika
b. Pemanggilan Perilaku
c. Pemrosesan Data

Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classfier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classfier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku.

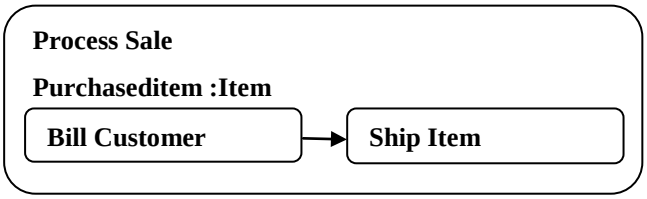
Aktivitas digambarkan dengan persegi panjang tumpul. Namanya ditulis di kiri atas. Parameter yang terlibat dalam aktivitas ditulis dibawahnya.



Gambar II.7. Aktivitas serderhana tanpa rincian

Sumber : Probowo Pudjo Widodo, dkk (2011 : 145)

Detail aktivitas dapat dimasukan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.



Gambar II.8. Aktivitas dengan detail rincian

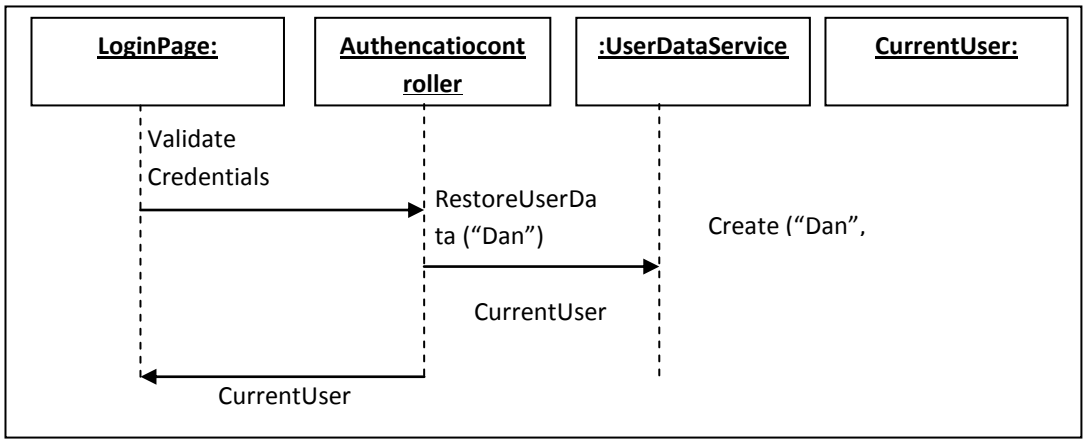
Sumber : Probowo Pudjo Widodo, dkk (2011 : 145)

D. Sequence Diagram

Douglas (2004) dalam buku (Prabowo Pudjo Widodo, dkk ; Menggunakan UML ; 2011 : 174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram),* diagram waktu (*timing diagram)* dan diagram komunikasi (*communication diagram).*

Pilone (2005) dalam buku (Prabowo Pudjo Widodo, dkk ; Menggunakan UML ; 2011 : 174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan.

Contoh diagram urutan dengan notasi-notasinya sebagai berikut :



Gambar II.9. Diagram Urutan

Sumber : Prabowo Pudjo Widodo, dkk (2011 : 175)

II.6. PHP

PHP Hypertext Preprocessor atau sering juga di sebut PHP merupakan bahasa pemograman berbasis *server-side* yang dapat melakukan *parsingscript* php menjadi *script web* sehingga sisi *client* menghasilkan tampilan yang menarik.

PHP berperan sangat besar ketika ingin membuat *website* keren dan dinamis karena melakukan banyak hal, seperti membaca *file*, menulis *file*, menampilkan gambar, animasi *movie*, dan yang paling pokok adalah dapat melakukan koneksi terhadap *database*. (YM Kusuma Aronana, S.T ; 2012 : 86)

Beberapa hal yang terdapat dalam PHP adalah :

1. Kode PHP
   a. *Embedded Script* yaitu *script* php yang di sisipkan ketika dibutuhkan saja, seperti menampilkan *database*, meng-*upload file*, *delete* data, *edit* data dan lain sebagainya.
   b. *Non Embedded Script* adalah kode php murni, karena pembuatannya diawalitag <?php dan diakhiri ?>. Kode html diletakkan didalam php secara utuh atau keseluruhan.
2. Operator

   Operator adalah alat yang digunakan untuk manipulasi data. Didalam php juga dikenal beberapa operator yaitu :

   a. *Arithmetic* Operator
   b. *Assignment* Operator yaitu operator yang mengalikasikan nilai tertentu.
   c. *Comparison* Operators yaitu operator yang dikenal dengan operator pembanding digunakan untuk menghasilkan dua nilai yang hasil akhirnya bernilai true atau *false*.
   d. *Logical* operator yaitu operator yang digunakan untuk mengoperasikan secara logic yaitu dengan *and, or* dan *not*.

II.7. MySQL

MySQL adalah salah satu program dapat digunakan sebagai *database* dan merupakan salah satu *software* untuk *database server* yang banyak digunakan. MySQL bersifat *Open Source* dan menggunakan SQL.

MySQL memiliki beberapa kelebihan, antara lain :

1. MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
2. MySQL memiliki kecepatan yang bagus dalam menangani *query* sederhana.
3. MySQL memiliki operator dan fungsi secara penuh dan mendukung perintah *select* dan *where* dalam perintah *query*.
4. MySQL memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses user dengan sistem perijinan yang detail secara sandi terenkripsi.
5. MySQL mampu mengenai basis data dalam skala besar, dengan jumlah rekaman (*record*) lebih dari 50 juta dan 60 ribu tabel serta kurang lebih 5 miliyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
6. MySQL dapat melakukan koneksi dengan *client*, menggunakan protokol TCP/IP, *Unix Soket* (UNIX), atau *Named Pipers* (NT).
7. MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa.

8. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga,* dan masih banyak lagi.
9. MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan gratis. (Madcoms ; 2011 : 140)

II.8. Kamus Data

Kamus data (data *dictionary*) dipergunakan untuk memperjelas aliran data yang di gambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memilki standar cara penulisannya. Kamus data biasanya berisi :

1. Nama – nama dari data
2. Digunakan pada – merupakan proses – proses yang terkait data.
3. Deskripsi – merupakan deskripsi data
4. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut :

Tabel II.1. Simbol Untuk Menjelaskan Informasi Tambahan

Simbol	Keterangan
=	disusun atau terdiri dari
+	Dan
[|]	baik ...atau...
{}”	n kali diulang/ bernilai banyak
( )	data opsional
...	batas komentar

Sumber : Rosa A.S, dkk (2011 : 68)

II.9. ERD

ERD adalah suatu diagram untuk menggambarkan desain untuk menggambarkan desain konseptual dari model konseptual suatu basis data rasional. ERD juga merupakan gambaran yang menghubungkan antara objek satu dengan objek yang lain dalam dunia nyata. Secara umum ERD terdiri dari 3 komponen, yakni:

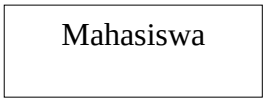
1. Entitas *(Entity)*

   Entitas merupakan suatu objek nyata yang mampu dibedakan dengan objek yang lain. Objek tersebut dapat berupa orang dan benda ataupun hal yang lainnya. Entitas digambarkan sebagai bentuk persegi panjang dengan nama entitas terletak didalamnya. Nama entitas merupakan objek tunggal dan

sedapat mungkin menggunakan nama yang mudah dipahami. Secara umum menurut jenis entitas dibagi menjadi dua yaitu:

a. Entitas Kuat *(Strong Entity)*

Entitas kuat adalah entitas yang dapat berdiri sendiri tanpa bantua entitas yang lain. Dengan kata lain entitas ini bergantung dengan entitas yang lain atau disebut juga dengan entitas induk. Contoh entitas kuat adalah mahasiswa. Simbol dari entitas ini seperti berikut :



Gambar II.10. Entitas Kuat

Sumber : Ema Utami, dkk (2012 : 20)

b. Entitas Lemah *(Weak Entity)*

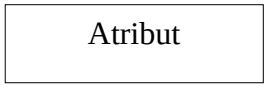
Entitas lemah adalah entitas yang tidak dapat berdidri sendiri tanpa bantuan entitas yang lain. Dengan kata lain entitas lemah merupakan hasil pembentukan dari entitas kuat, sehingga entitas lemah akan muncul setelah adanya entitas kuat. Entitas lemah juga dikenal dengan entitas anak. Simbol entitas lemah seperti berikut :

Gambar II.11. Entitas Lemah

Sumber : Ema Utami, dkk (2012 : 20)

2. Atribut *(Attribute)*

Atribut merupakan suatu informasi yang berkaitan dengan entitas. Di dalam dunia pemograman, atribut adalah properti dari suatu objek. Sebagai contoh, jika entitas adalah manusia/orang maka atributnya adalah rambut, mata hidung, tangan dan lain-lain. Atribut digambarkan dengan suatu lingkaran dengan nama atribut di tulis di tengahnya seperti berikut :

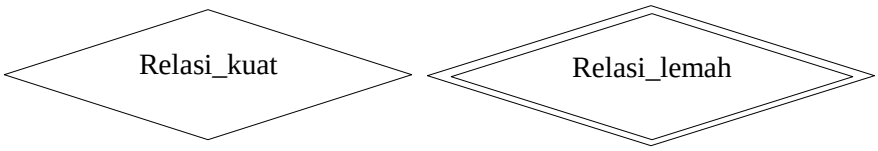


Gambar II.12. Atribut

Sumber : Ema Utami, dkk (2012 : 20)

3. Relasi *(Relationship)*

Belah ketupat merupakan penggambaran hubungan (relasi) antarentitas atau sering disebut kerelasian. Ada dua macam penggambaran relasi, yaitu relasi kuat dan lemah. Relasi yang kuat biasanya untuk menghubungkan antarentitas kuat, sedangkan relasi lemah untuk menghubungkan antar entitas kuat dengan yang lemah. Penggambaran kerelasian seperti berikut:



Gambar II.13. Kerelasian

Sumber : Ema Utami, dkk (2012 : 24)

Ada tiga macam relasi menurut derajatnya yaitu *unary* merupakan relasi yang menghubungkan satu entitas, *binary* merupakan relasi yang menghubungkan

dua entitas, *ternary* merupakan relasi yang menghubungkan lebih dari dua entitas. Untuk menghubungkan entitas-kerelasian-entitas digunakan garis lurus seperti berikut :

Gambar II.14. Kerelasian Antarentitas

Sumber : Ema Utami, dkk (2012 : 24)

Dalam kelerasian *binary* terdapat kardinalitas atau derajat hubungan antarentitas. Derajat kardinalitas merupakan penjelasan dari tingkat hubungan antarentitas. Ukuran derajat kardinalitas dibagi menjadi tiga macam yaitu :

a. 1-1 *(one-to-one)*

Derajat kardinalitas 1-1 terjadi jika satu entitas A hanya mempunyai hubungan dengan satu entitas B, ataupun sebaliknya. Sebagai contoh seorang ketua jurusan hanya memimpin satu jurusan, begitu juga sebaliknya satu jurusan hanya dipimpin seorang ketua jurusan.

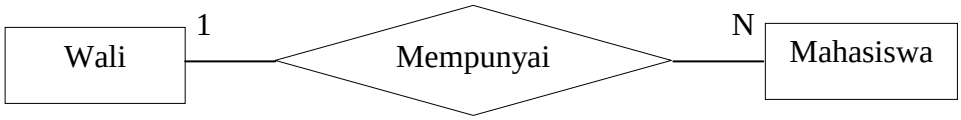
Gambar II.15. Derajat Kardinalitas 1-1 *(one-to-one)*

Sumber : Ema Utami, dkk (2012 : 24)

b. 1-N *(one-to-many)*

Derajat kardinalitas 1-N terjadi jika satu entitas A mempunyai lebih dari satu hubungan ke entitas B, ataupun sebaliknya. Sebagai contoh seorang

mahasiswa hanya mempunyai seorang wali, tetapi seorang wali bisa menjadi banyak wali mahasiswa.



Gambar II.16. Derajat Kardinalitas 1-N *(one-to-many)*

Sumber : Ema Utami, dkk (2012 : 25)

c. N-N *(many-to-many)*

Derajat kardinalitas N-N terjadi jika satu entitas A mempunyai lebih dari satu hubungan ke entitas B, sebaliknya satu entitas B mempunyai lebih dari satu hubungan ke entitas A. Sebagai contoh seorang mahasiswa bisa mengambil banyak mata kuliah, begitu juga sebaliknya satu mata kuliah bisa diambil oleh banyak mahasiswa.

Gambar II.17. Derajat Kardinalitas N-N *(many-to-many)*

Sumber : Ema Utami, dkk (2012 : 25)

II.10. Normalisasi

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel yang menunjukkan entitas sekaligus relasinya. Melalui normalisasi,kita akan mendisain basis data relasional menjadi suatu set data yang memenuhi kriteria berikut ini :

1. Memuat semua data penting yang dapat di sediakan oleh basis data.
2. Memiliki *redudancy* data yang sedikit mungkin.
3. Akomodasi multi *value* untuk tipe data yang diperlukan.
4. Mengizinkan *update* data yang efisien dalam basis data.
5. Terhindar dari bahaya kehilangan data yang tidak dikenal.

Tujuan utama dari normalisasi (umumnya minimal sampai pada level normalisasi ketiga) adalah mencegah terjadinya seperti berikut:

a. *Insertion Anomaly* merupakan kesalahan penambahan data ke dalam basis data
b. *Delection Anomaly* merupakan kesalahan dalam menghapus data yang ada di dalam basis data.
c. *Update Anomaly* merupakan kesalahan dalam mengubah data, baik dalam hal penambahan, penghapusan, atau keduanya.

Tujuan dari normalisasi adalah mengurangi kemungkinan terjadinya anomali yang terjadi dalam basis data. (Ema Utami, dkk ; 2012 : 40)

II.10.1. Bentuk – Bentuk Normalisasi

1. Bentuk Normalisasi Pertama (1NF)

   Suatu tabel dikatakan dalam bentuk normal pertama apabila :

   a. Tidak ada baris data yang terduplikat atau berulang dalam tabel.
   b. Setiap sel memiliki nilai tunggal artinya tidak ada perulangan dalam tabel.
   c. Data dalam kolom (atribut dan field) memiliki tipe data yang sejenis.

2. Bentuk Normalisasi Kedua(2NF)

   Tabel dalam keadaan 2NF apabila tabel sudah dalam keadaan 1NF dan semua atribut yang bukan kunci bergantung semua kunci dalam tabel. Dengan kata lain, 2NF bertujuan untuk menghilangkan ketergantungan parsial.

3. Bentuk Normalisasi Ketiga (3NF)

   Tabel dalam keadaan 3NF apabila tabel sudah dalam keadaan 2NF dan dalam tabel tersebut tidak ada ketergantungan transitif. Artinya sebuah *field* dapat menjadi atribut bisa pada suatu relasi tetapi menjadi kunci pada relasi lain. Setiap atribut yang bukan kunci haruslah bergantung hanya pada *primary key*.

4. Bentuk Normalisasi *Boyce-Codd* (BCNF)

   Tabel dalam keadaan 3NF dan setiap determain merupakan kunci kandidat. Determain adalah suatu atribut/*field* atau gabungan atribut dimana beberapa atribut lain bergantung pada atribut tersebut. Pada tahap BCNF, kuta harus menghilangkan kunci kandidat yang bukan merupakan determain.

5. Bentuk Normalisasi Keempat (4NF)

   Tabel dalam keadaan BCNF dan tidak ada ketergantungan multi *value*.

6. Bentuk Normalisasi Kelima (5NF)

   Tabel dalam keadaan 4NF dan setiap ketergantungan join dalam tabel merupakan akibat dari kunci kandidat tabel.

7. Bentuk Normalisasi *Domain-Key* (DKNF)

   Tabel dikatakan dalam keadaan DKNF jika setiap *constraint* tabel merupakan akibat dari definisi kunci – kunci dan domain.

II.11. Arcview

ArcView adalah software dikeluarkankan oleh *ESRI* (*Environmental Systems Research Institute*). Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengolahan data spasial. Kemampuan grafisyang 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 persil, pengendalian dampak lingkungan, bahkan untuk keperluan militer. (Eko Budiyanto ; 2010 : 176)

Avenue adalah sebuah skrip atau bahasa pemograman berorientrasi objek (*Objeck Oriented Programming*) (Esri,1996:5). Dengan *avenue* ini dapat dibentuk sebuah *interface* baru pada *arcview*, otomatis pekerjaan-pekerjaan yang bersifat berulang (*repetitif*), ataupun membuat sebuah alur analisis spasial khusus yang belum terdapat pada *arcview* tersebut. *Avenue* banyak digunakan untuk membnetuk sistem informasi aplikatif pada suatu lembaga atau intansi dengan berbasis *arcview* GIS.

Antarmuka sistem informasi (*interface*) dibentuk memanfaatkan fasilitas *customize* pada perangkat lunak *arcview* GIS 3.3. Menu dan tombol dibentuk

menggunakan teknik kustomasi tersebut. Teknik ini dipilih dengan didasarkan pada kemudahannya dalam membentuk menu dan tombol baru. Untuk menghubungkan menu dan tombol dengan berbagai aksi yang diinginkan maka perlu dibentuk skrip atau program menggunakan bahasa *avenue*.

Dalam sebuah sistem informasi ditampilkan berbagai data seperti data *shapefile* dengan berbagai skala, data atribut yang ditampilkan pada sebuah *form* ataupun data penginderaan jauh seperti citra satelit atau foto udara. Berbagai data tersebut saling melengkapi dan memperkuat informasi yang ditampilkan pada sistem informasi tersebut. Data spasial seperti peta dan citra akan memberikan informasi letak atau sebarannya secara spasial. Data atribut akan memberikan uraian data yang berkaitan dengan objek yang dipilih. Data citra diturunkan setelah melalui beberapa prosedur pengolahan citra. Pengolahan citra dilakukan dengan menggunakan berbagai perangkat lunak pengolahan citra. Citra tersebut diregistrasi untuk memberikan presisi secara spasial. Dalam sistem informasi ini citra digunakan bersama dengan data *shapefile*. Citra dan *shapefile* di-*obverlay*-kan untuk memberikan ketepatan lokasi pengambilan data.

Beberapa bagian *Arcview* yang cukup penting antara lain adalah :

1. *Project*

   Merupakan kumpulan dari dokumen yang berasosiasi selama satu sesi *Arcview*. Setiap *project* memiliki lima komponen pokok yaitu *views, tables, charts, layouts* dan *scripts*. *Views* digunakan untuk mengelola data grafis. Sedangkan *table* untuk manajemen data *atribut*, *charts* untuk mengelola grafik (bukan data grafis), *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*.

2. *Theme*

   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*. Contoh *feature* yang berupa titik adalah sekolah, pos polisi, rumah sakit. Untuk *feature* garis antara lain adalah jalan raya, jalan tol, sungai. Sedangkan sawah, danau, lahan parkir, wilayah administrasi pemerintahan merupakan sebuah *fiture polygon*.

3. *Views*

   View merupakan sebuah peta interaktif yang dapat digunakan untuk menampilkan, memeriksa, memilih dan menganalisa data grafis. *View* tidak menyimpan data grafis yang sebenarnya, tetapi hanya membuat referensi tentang data grafis mana saja yang terlibat. Ini mengakibatkan *view* bersifat dinamis. *View* merupakan kumpulan dari *theme*.

4. *Table*

   Tabel digunakan untuk menampilkan informasi tentang fature yang ada di dalam 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.

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.

7. *Script*

   Script merupakan sebuah bahasa pemrograman dari *Arcview* yang ditulis ke dalam bahasa *Avenue*.