

## BAB II

### TINJAUAN PUSTAKA

#### II.1. Konsep Dasar Sistem

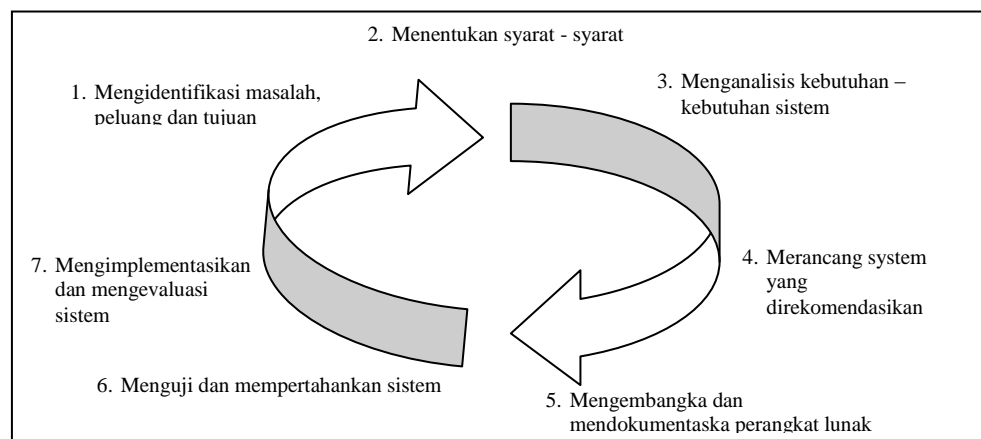
##### II.1.1. Sistem

Sistem merupakan sekumpulan objek, ide, berikut saling keterkaitannya (inter-relasi) di dalam (usaha) mencapai suatu tujuan. Atau, dengan kata lain, sistem dapat disebutkan sebagai kumpulan komponen (sub-sistem fisik maupun non-fisik/logika) yang saling berhubungan satu sama lainnya dan bekerja sama secara harmonis untuk mencapai suatu tujuan. (Eddy Prahasta:2009:78)

##### II.1.2.1. Siklus Hidup Pengembangan Sistem

Siklus hidup pengembangan sistem ini adalah pengembangan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan sangat baik melalui pengembangan siklus kegiatan penganalisisan dan pemakaian secara spesifik. (Kusrini:2008:8)

Adapun siklus hidup pengembangan sistem data pada gambar II.1 :



**Gambar II.1. Siklus Hidup Pengembangan Sistem**

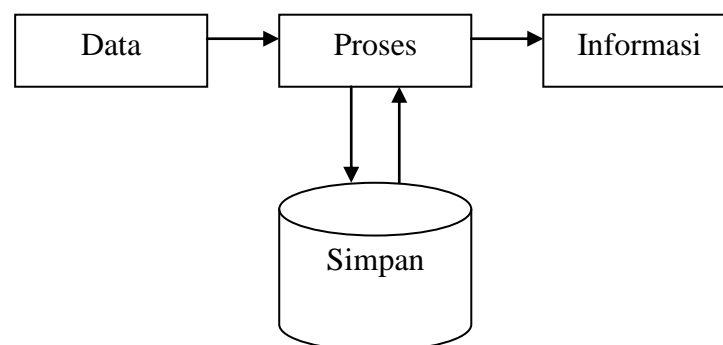
*Sumber : Kusrini (2008 : 9)*

## II.2. Informasi

Dalam bukunya Gordon B.Davis menjelaskan bahwa “Informasi adalah data yang telah diproses kedalam suatu bentuk yang mempunyai arti bagi sipenerima dan mempunyai nilai nyata dan terasa bagi keputusan saat itu atau keputusan mendatang”. (Tata Sutabri:2012:1)

Menurut Hoffer, dkk (2005) “Informasi adalah data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakannya”. (Abdul Kadir:2009:3)

Jadi informasi dapat didefinisikan sebagai data yang telah diolah menjadi suatu bentuk yang berarti bagi penerima dan bermakna dalam pengambilan keputusan.



**Gambar II.2. Pemrosesan Data**

**Sumber : Tata Sutabri (2012:2)**

## II.3. Sistem Informasi

Menurut Eddy Prahasta (2009 : 93), Sistem informasi adalah sekumpulan komponen-komponen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi terkait untuk mendukung proses pengambilan keputusan, koordinasi, dan pengendalian.

Komponen-komponen sistem informasi adalah sebagai berikut:

1. Perangkat keras (*hardware*)

Perangkat keras (*hardware*) terdiri dari monitor, CPU, keyboard, mouse dan harddisk.

2. Perangkat lunak (*software*)

Perangkat lunak (*software*) berupa program-program aplikasi yang akan digunakan, yaitu merupakan kumpulan dari perintah atau fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.

3. Data

Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.

4. Prosedur

Prosedur merupakan dokumentasi prosedur atau proses sistem, tata cara atau penuntun operasional (aplikasi) dan teknis

5. Manusia

Manusia adalah pengguna dari sistem informasi.

#### **II.4. SIG (Sistem Informasi Geografis)**

Definisi SIG (Sistem Informasi Geografis) selalu berkembang, bertambah dan bervariasi. Hal ini terlihat dari banyaknya definisi SIG yang telah beredar. Selain itu, SIG juga merupakan suatu kajian ilmu dan teknologi yang relatif baru,

digunakan oleh berbagai bidang disiplin ilmu, dan berkembang dengan cepat. Berikut ini adalah beberapa definisi SIG (Eddy Prahasta; 2009:116).

1. Aronoff (1989), mendefinisikan SIG sebagai suatu sistem berbasis komputer yang memiliki kemampuan dalam menangani data bereferensi geografi yaitu pemasukan data, manajemen data (penyimpanan dan pemanggilan kembali), manipulasi dan analisis data, serta keluaran sebagai hasil akhir (output). Hasil akhir (output) dapat dijadikan acuan dalam pengambilan keputusan pada masalah yang berhubungan dengan geografi.
2. Menurut Gistut (1994), SIG adalah sistem yang dapat mendukung pengambilan keputusan spasial dan mampu mengintegrasikan deskripsi-deskripsi lokasi dengan karakteristik-karakteristik fenomena yang ditemukan di lokasi tersebut. SIG yang lengkap mencakup metodologi dan teknologi yang diperlukan yaitu data spasial perangkat keras, perangkat lunak dan struktur organisasi Gistut.
3. Burrough (1986) mendefinisikan SIG adalah sistem berbasis komputer yang digunakan untuk memasukan, menyimpan, mengelola, menganalisis dan mengaktifkan kembali data yang mempunyai referensi keruangan untuk berbagai tujuan yang berkaitan dengan pemetaan dan perencanaan.

Dari defenisi-definisi tersebut diatas, diambil satu buah definisi yang dapat mewakili SIG secara umum yaitu sistem informasi yang digunakan untuk memasukkan, menyimpan, memanggil kembali, mengolah, menganalisa dan menghasilkan data bereferensi geografi atau data geospasial, untuk mendukung pengambilan keputusan dalam perencanaan dan pengolahan. Selain itu, juga dapat

diambil kesimpulan bahwa SIG terdiri atas beberapa subsistem yaitu: data *input*, data *output*, data *management*, data manipulasi dan analisis.

#### **II.4.1. Subsistem Sistem Informasi Geografis**

Menurut (Prahasta, 2005), SIG dapat diuraikan beberapa subsistem sebagai berikut :

1. *Data Input*

Subsistem ini bertugas untuk mengumpulkan, mempersiapkan, dan menyimpan data spasial dan atributnya dari berbagai sumber. Sub-sistem juga bertanggung jawab dalam mengonversikan atau mentransformasikan format-format data aslinya ke dalam format yang dapat digunakan oleh perangkat SIG yang bersangkutan.

2. *Data Output*

Subsistem ini bertugas untuk menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data spasial dalam bentuk *softcopy* maupun *hardcopy* seperti : tabel, grafik, *report*, peta, dan lain-lain.

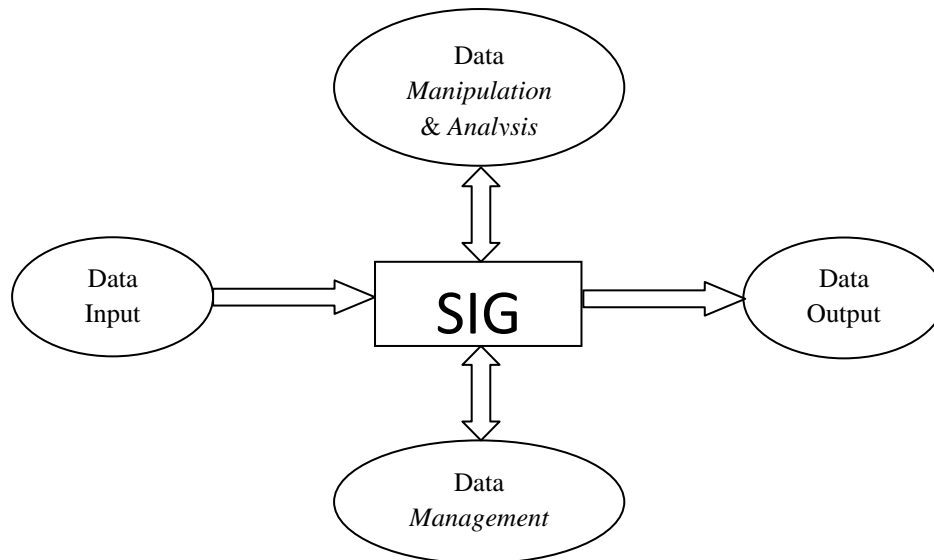
3. *Data Management*

Subsistem ini mengorganisasikan baik data spasial maupun atribut terkait kedalam sebuah sistem basis data sedemikian rupa hingga mudah dipanggil, di-*update*, dan di-*edit*.

4. *Data Manipulation & Analysis*

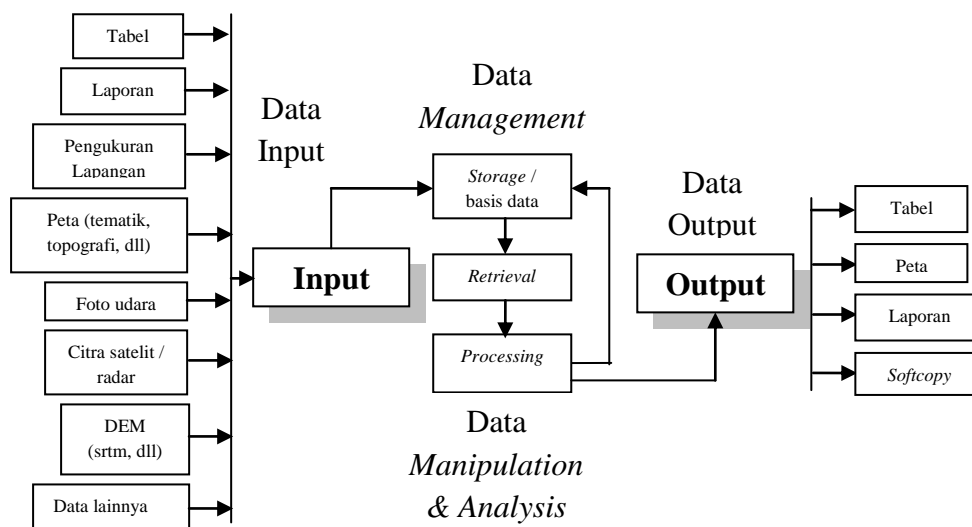
Subsistem ini menentukan informasi - informasi yang dapat dihasilkan oleh SIG. Selain itu, sub-sistem ini juga melakukan manipulasi (evaluasi

dan penggunaan fungsi-fungsi dan operator matematis & logika) dan pemodelan data untuk menghasilkan informasi yang diharapkan.



**Gambar II.3. Ilustrasi Sub-Sistem SIG**  
*Sumber : Eddy Prahasta (2009 : 119)*

Jika beberapa sub-sistem SIG pada gambar II.3 di atas diperjelas berdasarkan uraian jenis masukan, proses, dan jenis keluaran yang ada didalamnya, maka sub-sistem SIG di atas juga dapat digambarkan sebagai berikut:



**Gambar II.4: Ilustrasi Uraian Sub-Sistem SIG**  
*Sumber : Eddy Prahasta (2009 : 119)*

#### II.4.2. Komponen SIG

SIG merupakan salah satu sistem yang kompleks dan pada umumnya juga (selain yang *stand-alone*) terintegrasi dengan lingkungan sistem komputer lainnya di tingkat fungsional dan jaringan (*network*). SIG sebagai sistem terdiri dari beberapa komponen (sebagai berikut) dengan berbagai karakteristiknya:

- a) Perangkat keras. Pada saat ini SIG sudah tersedia bagi berbagai *platform* perangkat keras; mulai dari kelas PC *desktop*, *workstation* hingga *multi-user host* yang bahkan dapat digunakan oleh banyak orang secara bersamaan (simultan) dalam jaringan komputer yang luas, tersebar, berkemampuan tinggi, memiliki ruang penyimpanan (*harddisk*) yang besar. Walaupun demikian, fungsionalitas SIG tidak terikat secara ketat pada karakteristik - karakteristik fisik perangkat kerasnya sehingga keterbatasan memori pada suatu PC-pun dapat diatasi. Adapun perangkat keras yang sering digunakan untuk aplikasi SIG dalam komputer (PC), *mouse*, monitor (plus VGA-card grafik) yang beresolusi tinggi, *digitizer*, *printer*, *plotter*, *receiver GPS*, dan *scanner*.
- b) Perangkat lunak. Dari sudut pandang yang lain, SIG bisa juga merupakan sistem perangkat lunak yang tersusun secara modular dimana sistem basis datanya memegang peranan kunci. Pada kasus perangkat SIG tertentu, setiap sub-sistem diimplementasikan 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 dieksekusi tersendiri.

- c) Data & 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 lunak SIG yang lain) maupun secara langsung dengan cara melakukan digitasi data spasialnya (digitasi *on-screen* 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*.
- d) 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.

## II.5. Peta

Peta merupakan alat utama dalam ilmu geografis, selain foto udara dan citra satelit. Melalui peta seseorang dapat mengamati ketampakan permukaan bumi lebih luas dari batas pandangan manusia.

Menurut Hartono (2007 : 2), Peta adalah suatu gambaran unsur-unsur ketampakan abstrak dari permukaan bumi yang digambarkan pada suatu bidang datar dan diperkecil atau diskalakan.

Secara umum fungsi peta sebagai berikut :

1. Menunjukkan posisi atau lokasi relatif (letak suatu tempat dalam hubungannya dengan tempat lain dipermukaan bumi).

2. Memperlihatkan ukuran (dari peta dapat diukur luas daerah dan jarak-jarak diatas permukaan bumi).
3. Memperlihatkan bentuk (misalnya bentuk benua, gunung).
4. Mengumpulkan dan menyeleksi data-data dari suatu dan menyajikannya diatas peta.

## II.6. ArcView

Menurut Eddy Prahasta (2009) program ArcView merupakan salah satu perangkat lunak SIG yang populer dan paling banyak digunakan untuk mengelola data spasial. *Arcmap* dibuat oleh *Enviromental System Research Intitute (ESRI)*. Arcview berfungsi untuk mengolah data, menganalisa, dan membuat peta serta laporan yang berkaitan dengan data spasial bereferensi geografis. Beberapa bagian ArcView yang cukup penting adalah *project, theme, view, table, chart, layout, script*.

## II.7. Aplikasi Web

Kepopuleran Internet di seluruh dunia mendorong aplikasi *web* semakin diminati. Dengan menggunakan aplikasi web, Anda hanya perlu menempatkan aplikasi dalam sebuah *server* dan dengan sendirinya aplikasi tersebut dapat diakses dari mana pun, sepanjang pemakai dapat mengakses *web server-nya*. Aplikasi *web* adalah sebuah jenis aplikasi yang diakses melalui browser, misalnya *Internet Explorer* dan *Mozilla Firefox*. (Abdul Kadir;2009:2)

## **II.8. CorelDraw**

*CorelDraw* adalah *software* grafis serbaguna yang biasa dipakai untuk ilustrasi dan publikasi. Sehingga banyak digunakan untuk aplikasi percetakan di media kertas, kain, outdoor, elektronik pemetaan lokasi, dll.

Software sejenis antara lain *Adobe Illustrator* dan *Macromedia Freehand*. Para profesional grafis dan ilustrasi lebih memilih *illustrator* dibandingkan *CorelDraw* karena akurasi warna. Namun banyak fitur dan kemudahan di *CorelDraw* yang tidak memiliki pesaingnya. (Hendi Hendratman; 2009:1)

*CorelDraw* merupakan salah satu *software* pengolah gambar vektor populer yang dibuat oleh *corel corporation* untuk keperluan visualisasi *image*. Aplikasi *CorelDraw* telah berkembang menjadi *CorelDraw X5* yang lebih dinamis sehingga mudah digunakan. Berbagai fitur baru pun ditambah di antaranya *Tables, Independent Layer, Live Text Preview, New Font, New Template, Raw Camera Support, Mirroring Paragraph Text, CorelPower Trace*, dan masih banyak lagi sehingga kreativitas anda pun meningkat. (Wahana Komputer;2011:1)

## **II.9. Unified Modeling Language (UML)**

UML adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Menurut pencetusnya, UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah system. Karena tergolong bahasa

visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari system yang sedang dimodelkan. Memahami UML itu sebagai bahasa visual itu penting, karena penekanan tersebut membedakannya dengan bahasa pemrograman yang lebih dekat ke mesin.

Unified Modelling Language (UML) biasa digunakan untuk:



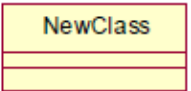


1. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model *behaviour* “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 *functionality* dengan stereotypes.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembangan sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram

yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. (Yuni Sugiarti; 2013:37)

Dalam merancang UML (*Unified Modelling Language*) ada notasi-notasi yang telah ditentukan. Notasi-notasi dalam UML terdapat pada table II.1

**Tabel II.1. Daftar Notasi UML**

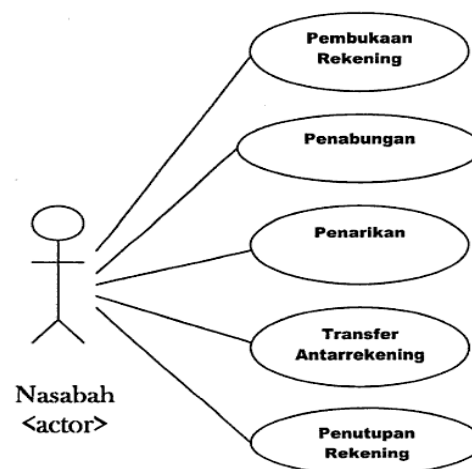
Simbol	Arti	Fungsi
	Notasi Actor	Actor menggambarkan segala pengguna software aplikasi (user).
	Notasi Use Case	Use case menjelaskan urutan kegiatan yang dilakukan actor dan system untuk mencapai suatu tujuan tertentu.
	Notasi Class	Class merupakan pembentuk utama dari system berorientasi obyek, karena class menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama.
	Notasi Interface	Interface merupakan kumpulan operasi tanpa implementasi dari suatu Class.
	Notasi Interaction	Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.

**Sumber : Munawar (2005:23)**

Dalam metode UML terdapat beberapa diagram sebagai berikut:

### 1. Use Case Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke use case. Dibawah ini salah satu contoh diagram *use case* pada mesin ATM.



**Gambar.II.5. Use Case Diagram**  
**Sumber : (Adi Nugroho: 2009)**

### 2. Class Diagram

*Class* diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk

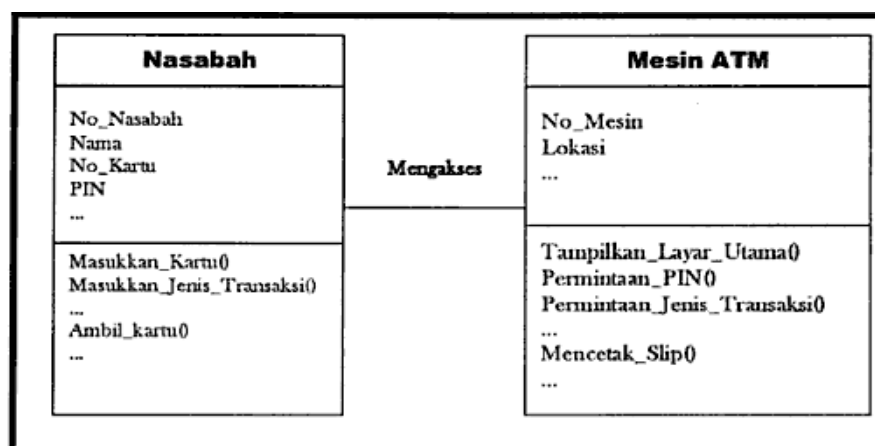
mencapai suatu tujuan. *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

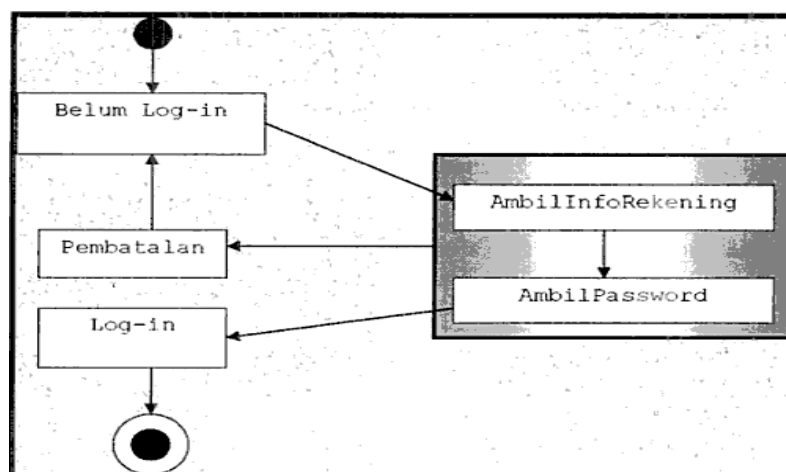
- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- b. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
- c. *Public*, dapat dipanggil oleh siapa saja



**Gambar.II.6. Class Diagram**  
**Sumber : (Adi Nugroho: 2009)**

### 3. Statechart Diagram

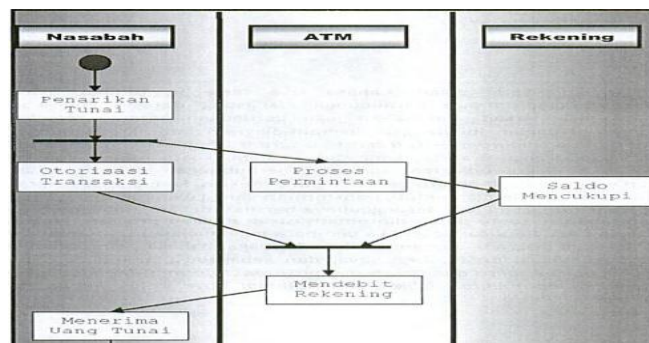
Menggambarkan transisi dan perubahan keadaan suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan class tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).



**Gambar.II.7. Statechart Diagram**  
 Sumber : (Adi Nugroho: 2009)

### 4. Activity Diagram

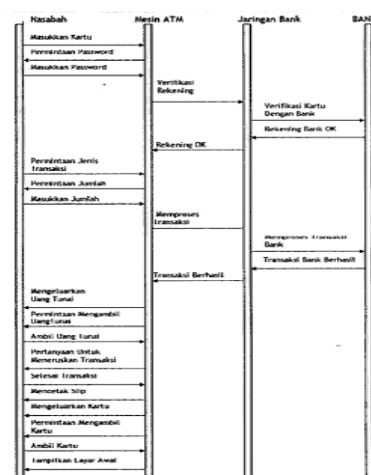
*Activity diagrams* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Dibawah ini salah satu contoh diagram *Activity* pada mesin ATM.



**Gambar.II.8. Activity Diagram**  
**Sumber : (Adi Nugroho: 2009)**

## 5. Sequence Diagram

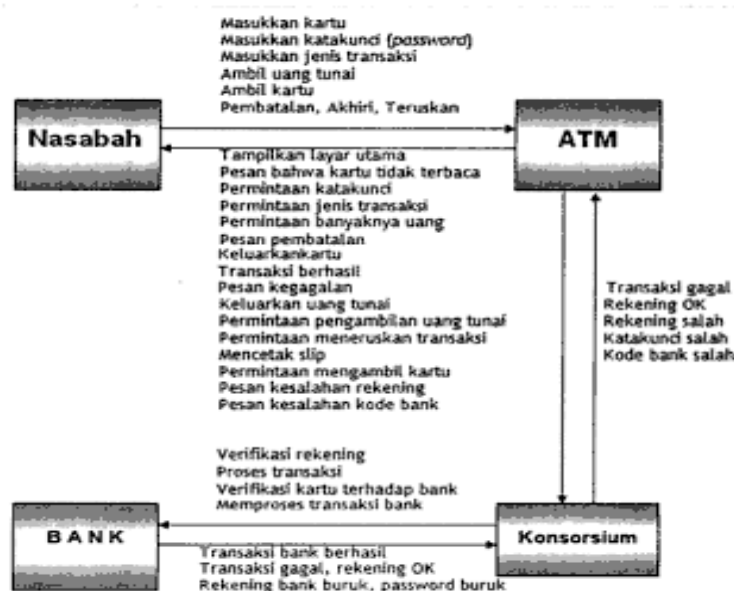
*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Contoh *Sequence Diagram*



**Gambar.II.9. Sequence Diagram**  
**Sumber : (Adi Nugroho : 2009)**

## 6. Collaboration Diagram

*Collaboration Diagram* adalah suatu diagram yang memperlihatkan/ menampilkan pengorganisasian interaksi yang terdapat disekitar objek (seperti halnya *sequence diagram*) dan hubungannya terhadap yang lainnya. *Collaboration Diagram* lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan/*message*. Dibawah ini salah satu contoh diagram *Activity* pada mesin ATM.



**Gambar.II.10. Collaboration Diagram**  
*Sumber : (Adi Nugroho: 2009)*

## II.10. Database

Database merupakan komponen terpenting dalam pembangunan sistem informasi, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. Database merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian

rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi. (Budi Sutedjo; 2006:99)

### **II.10.1. Hierarki Data Dalam Database**

Data dalam sebuah database disusun berdasarkan sistem hierarki yang unik, yaitu:

#### *1. Database*

Merupakan kumpulan file yang saling terkait satu sama lain, misalnya file data induk karyawan, file jabatan, file penggajian dan lain sebagainya. Kumpulan file yang tidak saling terkait satu sama lain tidak dapat disebut database, misalnya file data induk karyawan, file tamu undangan perkawinan, file barang retail pasar swalayan.

#### *2. File*

Yaitu kumpulan dari record yang saling terkait dan memiliki format field yang sama dan sejenis.

#### *3. Record*

Yaitu kumpulan field yang menggambarkan suatu unit data individu tertentu.

#### *4. Field*

Yaitu atribut dari *record* yang menunjukkan suatu item dari data seperti nama, alamat, dan lain sebagainya.

#### 5. *Byte*

Yaitu atribut dari *field* yang berupa huruf yang membentuk nilai dari sebuah *field*. Huruf tersebut dapat berupa numerik maupun abjad atau huruf tersebut.

#### 6. *Bit*

Yaitu bagian terkecil dari data secara keseluruhan, yaitu berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk *byte*.

### II.11. MySQL

Menurut Agus Saputra (2012 : 7) MySQL bukan termasuk bahasa pemrograman. MySQL merupakan salah satu database populer dan mendunia. Intinya, MySQL bekerja menggunakan SQL *Language* (*Structure Query Language*). Itu dapat diartikan bahwa MySQL merupakan standar penggunaan database di dunia untuk pengolahan data.

Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah SELECT (mengambil), INSERT (menambah), UPDATE (mengubah), dan DELETE (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah dan menghapus data.

Beberapa kelebihan yang dimiliki oleh MySQL, antara lain:

1. Bersifat *open source*, yang memiliki kemampuan untuk dapat dikembangkan lagi.
2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dunia dalam pengolahan data.

3. Super *performance* dan *reliable*, tidak bisa diragukan, pemrosesan databasenya sangat cepat dan stabil.
4. Sangat mudah dipelajari (*ease of use*).
5. Memiliki dukungan *support (group)* pengguna MySQL.
6. Mampu lintas *Platform*, dapat berjalan di berbagai sistem operasi.
7. *Multiuser*, di mana MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami konflik.

## II.12. Basis Data

*Database* (basis data) sering didefinisikan sebagai kumpulan data yang terkait. Secara teknis, yang berada dalam sebuah *database* adalah sekumpulan tabel atau objek lain (*indeks, view, dan lain-lain*). Tujuan utama pembuatan *database* adalah untuk memudahkan dalam mengakses data. Data dapat ditambahkan, diubah, dihapus, atau dibaca dengan relatif mudah dan cepat.

Sebuah tabel (atau kadang disebut relasi) berisi sejumlah baris dan kolom. Perpotongan baris dan kolom menyatakan sebuah data. Gambar II.11 menunjukkan contoh sebuah tabel. Pada tabel tersebut terdapat 4 buah kolom dan 5 buah baris. (Abdul Kadir; 2009:14)

Nip	Nama	Alamat	Kota
90023	Ali Hamdan	Jl.Turangga 23	Yogya
90024	Sardi Waskito	Jl.Karya 11/7	Yogya
90025	Ida Rohanda	Jl.Timang 98A	Sleman

91026	Rahmanda	Jl.Gatotkaca 2	Yogya
91027	Prabukusuma	Jl.Amarta 3	Magelang

The diagram shows a table with two rows and four columns. The first row contains the values '91026', 'Rahmanda', 'Jl.Gatotkaca 2', and 'Yogya'. The second row contains '91027', 'Prabukusuma', 'Jl.Amarta 3', and 'Magelang'. Below the table, there are two labels in rounded rectangles: 'Kolom' and 'Sebuah baris'. Arrows point from 'Kolom' to the first three columns of the table, and an arrow points from 'Sebuah baris' to the second row of the table.

**Gambar II.11. Contoh Tabel Pribadi**

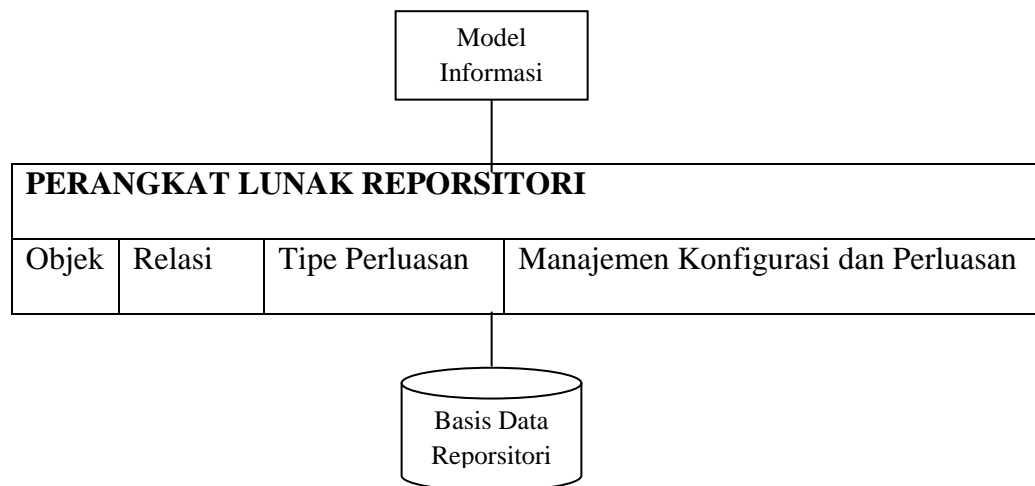
*Sumber : (Abdul Kadir ;2009:15)*

Saat ini tersedia banyak perangkat lunak yang ditujukan untuk mengelola *database*. Perangkat lunak seperti itu biasa dinamakan DBMS (*database management system*), *Access*, *MS SQL Server*, dan *MySQL* merupakan segelintir contoh produk pengelola *database*. Beberapa di antaranya berkelas *database server*, yaitu jenis yang secara aktif memantau permintaan menanggapi permintaan data. Adapun yang bukan termasuk *database server* adalah *Access*. (Abdul Kadir; 2009:15)

### II.12.1. Kamus Data

Kamus data pada prinsipnya merupakan metadata basis data yang berukuran relatif kecil. Salah satu bagian yang terintegrasi pada DBMS bertipe *relational* adalah kamus data (*Data Dictionary*), yang menyimpan metadata atau informasi-informasi mengenai basis data, termasuk nama-nama dan defenisi-defenisi untuk setiap tabel basis data. Kamus data biasanya merupakan bagian dari katalog sistem (*System Catalog*) yang dibentuk untuk setiap basis data. Katalog sistem mendeskripsikan semua objek basis data, termasuk data tentang tabel-tabel, seperti nama tabel, pembuat tabel, pengguna yang berhak menggunakannya, nama dan tipe data, kunci tamu (*Foreign key*) dan kunci primer

(*Primary key*), berkas *index*, dan sebagainya. Katalog sistem diciptakan oleh DBMS dan informasinya di tabel sistem, yang padanya dapat dilakukan permohonan seperti tabel data biasa, jika pengguna memiliki hak yang memadai.



**Gambar II.12. Komponen arsitektur sistem basisdata**  
*Sumber : (Adi Nugroho; 2011:530)*

Kamus data bersifat aktif atau pasif. Kamus data dikelola secara otomatis oleh perangkat lunak pengelola basis data. Sistem aktif selalu konsisten dengan struktur asli dan defenisi basis data karena dipelihara oleh sistem itu sendiri. Kamus data pasif dikelola oleh pengguna-pengguna sistem, dan dimodifikasi kapanpun struktur basis data berubah. Karena modifikasi ini harus dilakukan oleh pengguna secara manual, kamus data mungkin bukan merupakan struktur basis data saat ini. Bagaimanapun juga, kamus data pasif mungkin dipelihara sebagai basis data terpisah. Ini mungkin diinginkan selama tahap perancangan sehingga kamus data pasif memungkinkan pengembang tetap mandiri dari penggunaan DBMS lainnya. Kamus data pasif tidak dibatasi oleh informasi yang dapat dikenali oleh DBMS. Karena kamus data pasif dipelihara oleh pengguna, mereka

mungkin diperluas untuk mencakup informasi tentang organisasi yang tidak terkomputerisasi. (Adi Nugroho; 2011:530-531).

### **II.12.2. Normalisasi**

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

1. Memuat semua data penting yang dapat disediakan oleh basis data.
2. Memiliki *redundancy* data yang sesedikitnya 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 *insertion anomaly*, *deletion anomaly*, dan *update anomaly*. (Ema Utami; 2012:40)

#### **II.12.2.1 Tahap Normalisasi**

Tahapan normalisasi berdasarkan 2 (dua) prinsip, yaitu prinsip kebergantungan fungsional serta prinsip dekomposisi tanpa kehilangan informasi yang dibutuhkan, normalisasi dapat di pahami sebagai tahapan-tahapan yang masing-masing berhubungan dengan bentuk normal. Bentuk normal adalah keadaan relasi yang dihasilkan dengan menerapkan aturan sederhana berkaitan dengan konsep kebergantungan fungsional pada relasi yang bersangkutan. Maka akan digambarkan secara garis besar bentuk normalisasinya :

1. Bentuk Normal Pertama (1NF/ *First Normal Form*)

Bentuk normal pertama adalah suatu bentuk relasi di mana atribut bernilai banyak (*multivalued attribute*) telah dihilangkan sehingga kita akan menjumpai nilai tunggal (mungkin saja nilai *null*) pada perpotongan setiap baris dan kolom pada tabel.

2. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

Semua kebergantungan fungsional yang bersifat (*partial functional dependency*) telah dihilangkan.

3. Bentuk Normal Ketiga (3NF/ *Third Normal Form*)

Semua kebergantungan transitif (*transitive dependency*) telah dihilangkan.

4. Bentuk Normal *Boyce-Codd* (BCNF/ *Boyce-Codd Normal Form*)

Semua anomali yang tersisa dari hasil penyempurnaan kebergantungan fungsional sebelumnya telah dihilangkan.

5. Bentuk Normal Keempat (4NF/ *Fourth Normal Form*)

Semua Kebergantungan bernilai banyak telah dihilangkan.

6. Bentuk Normal Ketiga (5NF/ *Fifth Normal Form*)

Semua anomali yang tertinggal telah dihilangkan. (Adi Nugroho; 2011:199-200)

## **II.13. Entity Relationship Diagram - ERD**

Pada perancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antar data. Hubungan tersebut dapat dinyatakan dalam model bentuk E-R. Model ini adalah dasar yang penting dalam merancang

database. Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut, dan hubungan antar entitas. Namun, model ini tidak mencerminkan bentuk fisik yang nantinya akan disimpan dalam database melainkan hanya bersifat konseptual. (Abdul Kadir; 2009:30)

Entity Relationship (ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas.


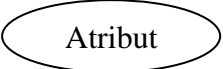


Entitas digambarkan dalam basis data dengan kumpulan atribut. Misalnya atribut NIM, nama, alamat, dan kota bisa menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

Atribut NIM digunakan untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terdapat dua mahasiswa dengan nama, alamat, dan kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa.

Relasi adalah hubungan antara beberapa entitas. Sebagai contoh, relasi menghubungkan mahasiswa dengan mata kuliah yang di ambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan relasi (*relationship set*). (Abdul Kadir; 2009:13)

Struktur logis (skema database) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut:

**Tabel II.2. Notasi ERD (*Entity Relationship Diagram*)**

Notasi	Keterangan
	Persegi panjang mewakili kumpulan entitas
	Elips mewakili atribut
	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

**Sumber : Janner Simarmata & Imam Prayudi (2006 : 59)**

## II.14. PHP

Menurut kamus komputer, PHP adalah bahasa pemrograman untuk dijalankan melalui halaman web, umumnya digunakan untuk mengolah informasi di internet. Sedangkan dalam pengertian lain PHP adalah singkatan dari PHP (*Hypertext Preprocessor*) yaitu bahasa pemrograman *web server-side* yang bersifat *open source* atau gratis. PHP merupakan *script* yang menyatu dengan HTML dan berada pada *server (server side HTML embedded scripting)*. (Rulianto Kurniawan; 2010:3)

### II.14.1. Kelebihan PHP

Banyak sekali kelebihan-kelebihan yang dimiliki PHP dibandingkan bahasa pemrograman yang lain, diantaranya:

1. Bisa membuat web menjadi dinamis.
2. PHP bersifat *open source*, sehingga dapat digunakan oleh siapa saja secara gratis.
3. Program atau aplikasi yang dibuat menggunakan PHP dapat *running* atau dijalankan di semua sistem operasi. Karena PHP berjalan secara *web base*, itu artinya semua sistem operasi yang memiliki *web browser* dapat menggunakan aplikasi ini, dan semua OS bahkan Handphone tentu saja selalu memiliki *web browser* dengan kata lain sangat *mobile* dan *fleksible*.
4. Aplikasi dengan PHP cukup cepat dibandingkan dengan ASP maupun java dalam berbagai aplikasi *web*.
5. Mendukung banyak paket database baik komersial maupun nonkomersial.
6. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang yang tidak melakukan sebuah kompilasi dalam penggunaannya.
7. *Web server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
8. Dalam sisi pengembangan lebih mudah.
9. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.

10. Berbagai *script* atau aplikasi siap pakai dan gratis telah tersedia dan dapat Anda gunakan kapanpun Anda mau.

#### **II.14.2. Kekurangan PHP**

Selain kelebihan PHP, PHP juga mempunyai beberapa kekurangan. Namun untuk masalah kekurangan, PHP dapat dibilang bahasa pemrograman yang jauh dari kekurangan. Namun disini akan ditulis beberapa kekurangan yang dimiliki oleh PHP.

1. Permasalahan yang sering terjadi pada *register-globals*.
2. Tidak mengenal *package*.
3. Jika tidak di *encoding*, maka kode PHP dapat dibaca semua orang. Untuk *encoding* Anda membutuhkan *Tool* dari *Zend* dan biaya *encoding* ini sangat mahal.
4. Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya.
5. PHP memiliki kelemahan *security*. Tertentu apabila programmer tidak jeli, atau berhati-hati dalam melakukan pemrograman dan kurang memperhatikan isu dan konfigurasi PHP.

#### **II.15. Macromedia Dreamweaver**

*Dreamweaver* adalah sebuah *editor* profesional untuk mendisain *web* secara visual dan mengolah situs atau halaman *web*. *Dreamweaver* merupakan *software* utama yang digunakan oleh *web desainer* maupun *web Programmer* dalam mengembangkan situs *web*, karena *Dreamweaver* mempunyai ruang kerja,

fasilitas dan kesempatan yang mampu meningkatkan produktivitas dan efektivitas dalam disain maupun membangun situs *web*. (Madcoms Madium; 2011:13)

Selain itu *Dreamweaver* juga digunakan sebagai *tool* untuk memadukan disain *web* yang dibuat dengan HTML dan melakukan *coding scripting* bahasa pemrograman *web* dalam membuat *web* menjadi lebih dinamis. (Master.com;2012:37)