

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Pengertian Sistem**

Istilah data dan informasi sering digunakan secara bergantian. Ada yang menyebut data, padahal informasi, sebaliknya ada yang mengatakan informasi, padahal data. Gordon B. Davis menjelaskan kaitannya dengan informasi dalam bentuk defenisi “informasi adalah data yang telah diproses kedalam suatu bentuk yang mempunyai arti bagi si penerima dan mempunyai nilai nyata dan terasa bagi keputusan saat itu atau keputusan mendatang”. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu di dalam dunia bisnis. Bisnis adalah perubahan dari suatu nilai yang disebut transaksi. Misalnya, penjualan adalah transaksi perubahan nilai barang menjadi nilai uang atau nilai piutang. Kesatuan nyata adalah berupa suatu objek yang nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi (Tata Sutabri : 2013 : 1).

#### **II.2. Data Dan Informasi**

Sistem informasi manajemen berhubungan dengan informasi. Berapa banyak informasi yang diberikan oleh sebuah sistem informasi ? belum ada metode untuk mengukur informasi dalam sebuah sistem untuk menghitung isinya. Informasi adalah sebuah istilah yang tepat dalam pemakaian umum. Informasi

dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi , dan lain sebagainya. Informasi ibarat darah yang mengalir dalam tubuh suatu organisasi sehingga informasi ini sangat penting di dalam suatu organisasi. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil, dan akhirnya mati (Tata Sutabri ; 2013 : 29).

### **II.3. Sistem Informasi Geografis**

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

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

1. Menjelaskan tentang lokasi atau letak
2. Menjelaskan kondisi ruang

3. Menjelaskan suatu kecendrungan (*trend*)
4. Menjelaskan tentang pola spasial (*spatial pattern*)
5. Pemodelan (Eko Budiyanto ; 2010 : 1).

*Data digital* geografis diorganisir menjadi dua bagian, yaitu Data Spasial dan Data Atribut/Tabular. Definisi dari kedua bagian tersebut adalah sebagai berikut :

1. *Data Spasial*

Merupakan kenampakan-kenampakan permukaan bumi, seperti : jalan, sungai, pemukiman, jenis penggunaan tanah, jenis tanah, dan lain-lain. Model Data Spasial dibedakan menjadi dua, yaitu : Model Data Vektor dan Model Data Raster.

2. *Data Atribut / Tabular*

Adalah yang menyimpan atribut dari kenampakan-kenampakan permukaan bumi tersebut. Misalnya, tanah yang memiliki *atribut* tekstur, kedalaman, struktur, pH, dan lain-lain. Model data tabular tersimpan kedalam bentuk baris (*record*) dan kolom (*field*) (Riyanto, dkk:2009 : 43-48).

Sistem Informasi Geografi (SIG) adalah sebuah alat bantu manajemen berupa informasi berbantuan komputer yang berkait erat dengan sistem pemetaan dan analisis terhadap segala sesuatu serta peristiwa – peristiwa yang terjadi di muka bumi. Teknologi SIG mengintegrasikan operasi pengolahan data berbasis database yang biasa digunakan saat ini, seperti pengambilan data berdasarkan kebutuhan, serta analisis statistik dengan menggunakan visualisasi yang khas serta

berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya. Dari definisi yang ada, 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 seperti penggunaan lahan, sumber daya alam, lingkungan transportasi, perencanaan fasilitas kota, dan pelayanan umum lainnya (Mokhamad Nurdiansyah, dkk : 3).

Sistem Informasi Geografis (*Geographic Information System*) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Atau dalam arti yang lebih sempit, adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut lokasinya, dalam sebuah *database* (Imat Ruhimat : 2).

#### **II.4. ArcView**

Kemampuan *Arcview* GIS pada berbagai serinya tidaklah diragukan lagi. *Arcview* GIS adalah software yang dikeluarkan oleh ESRI (*Environmental Systems Research Institute*). Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengolahan data spasial. Kemampuan grafis yang baik dan kemampuan teknis dalam pengolahan data spasial tersebut memberikan kekuatan secara nyata pada *Arcview* untuk melakukan analisis spasial. Kekuatan analisis inilah yang pada akhirnya menjadikan *Arcview* banyak diterapkan dalam berbagai

pekerjaan, seperti analisis pemasaran, perencanaan wilayah dan tata ruang, sistem informasi persis, pengendalian dampak lingkungan, bahkan untuk keperluan militer. Mengapa *Arcview* dapat memiliki keluwesan yang sedemikian hebat? Hal itu disebabkan oleh adanya dukungan dari skrip *Avenue*. Melalui *avenue* ini dapat dibentuk suatu “kemampuan baru” pada *Arcview*. Tentu saja hal ini membuat *Arcview* menjadi sangat luwes untuk diterapkan pada berbagai permasalahan spasial. *Avenue* dapat digunakan untuk “merombak” wajah *Arcview* sesuai kebutuhan penggunaanya (Eko Budiyanto : 2010 : 177).

## **II.5. MySQL**

Dalam bekerja dengan *database*, anda perlu memiliki pengetahuan yang cukup tentang konsep *client/server* dan tipe-tipe arsitektur aplikasi *database* yang ada. Dengan demikian, anda akan mengetahui dengan pasti tujuan dan fungsi (untuk apa) anda membuat dan mengolah *database*. Dimana *MySQL* adalah *software* ini digunakan untuk menyampaikan *database* yang akan digunakan (Budi Raharjo : 2011 : 15).

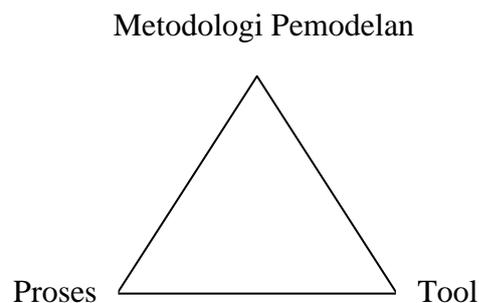
## **II.6. PHP**

Setelah anda merampungkan pembelajaran tentang *MySQL* pada bab-bab sebelumnya, pada bab 20 ini anda akan diajak untuk mengenal pembuatan aplikasi berbasis *web* yang menggunakan data dari *database* yang telah kita buat. Bahasa pemrograman yang akan kita gunakan untuk keperluan ini adalah *PHP*. Melalui bab ini, anda akan mengetahui cara mengakses data dari *database* kemudian

menampilkannya ke dalam halaman *web*. Untuk dapat mengkawinkan *database MySQL* dengan aplikasi yang ditulis menggunakan *PHP*. Dimana *PHP* adalah *software* ini digunakan untuk melakukan interpretasi dari kode *PHP* menjadi *HTML* sehingga hasilnya dapat ditampilkan di dalam *web browser* (Budi Raharjo : 2011 : 245)

## II.7. Pengertian UML

*Unified Modelling Language (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 (Yuni Sugiarti : 2013 : 34).



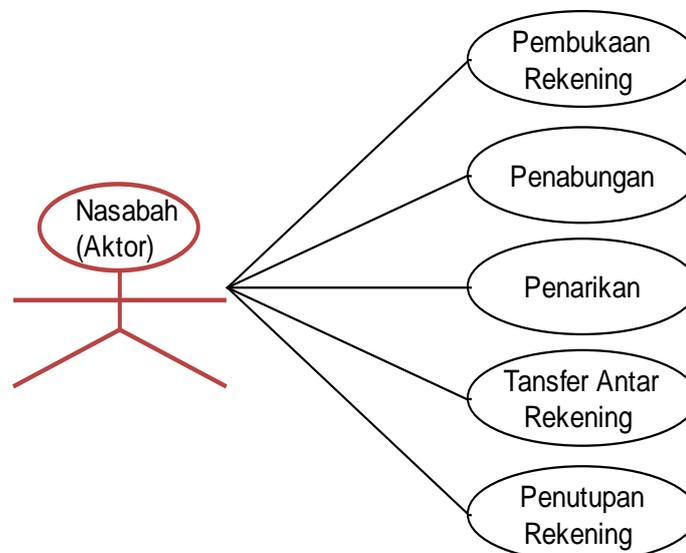
**Gambar II.1. The Triangle For Success**  
(Sumber : Yuni Sugiarti : 2013 ; 34)

### II.7.1. Use Case Diagram

Segala sesuatu yang secara akademis dikembangkan pada umumnya berawal dari suatu konsep. Demikian juga halnya dengan pengembangan sistem pada umumnya dikembangkan berdasarkan analisis kebutuhan. Analisis kebutuhan ini adalah tahap konseptualisasi, yaitu suatu tahap yang mengharuskan analis dan perancang sistem untuk berusaha tahu secara pasti mengenai hal yang

menjadi kebutuhan dan harapan pengguna sehingga kelak aplikasi yang dibuat memang akan digunakan oleh pengguna (*user*) serta akan memuaskan kebutuhan dan harapannya.

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *use case diagram* yang sesungguhnya merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya, *use case diagram* tidak hanya sangat penting pada saat analisis, tetapi juga sangat penting dalam tahap perancangan (*design*), untuk mencari kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian (*testing*). Saat akan mengembangkan *use case diagram*, hal yang pertama kali harus dilakukan adalah mengenali *actor* untuk sistem yang sedang dikembangkan. Dalam hal ini, ada beberapa karakteristik untuk para *actor*, yaitu *actor* yang ada di luar sistem yang sedang dikembangkan dan *actor* yang berinteraksi dengan sistem yang sedang dikembangkan (Adi Nugroho ; 2009 : 7)



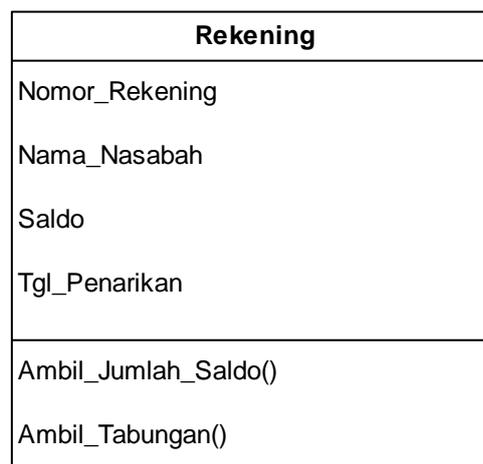
**Gambar II.2. Contoh Use Case Diagram**  
(Sumber : Adi Nugroho ; 2009 : 8)

### II.7.2. Class Diagram

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

- a. Atribut merupakan *varabel-variabel* yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

(Rosa A.S dan M. Shalahuddin : 2011 : 122).

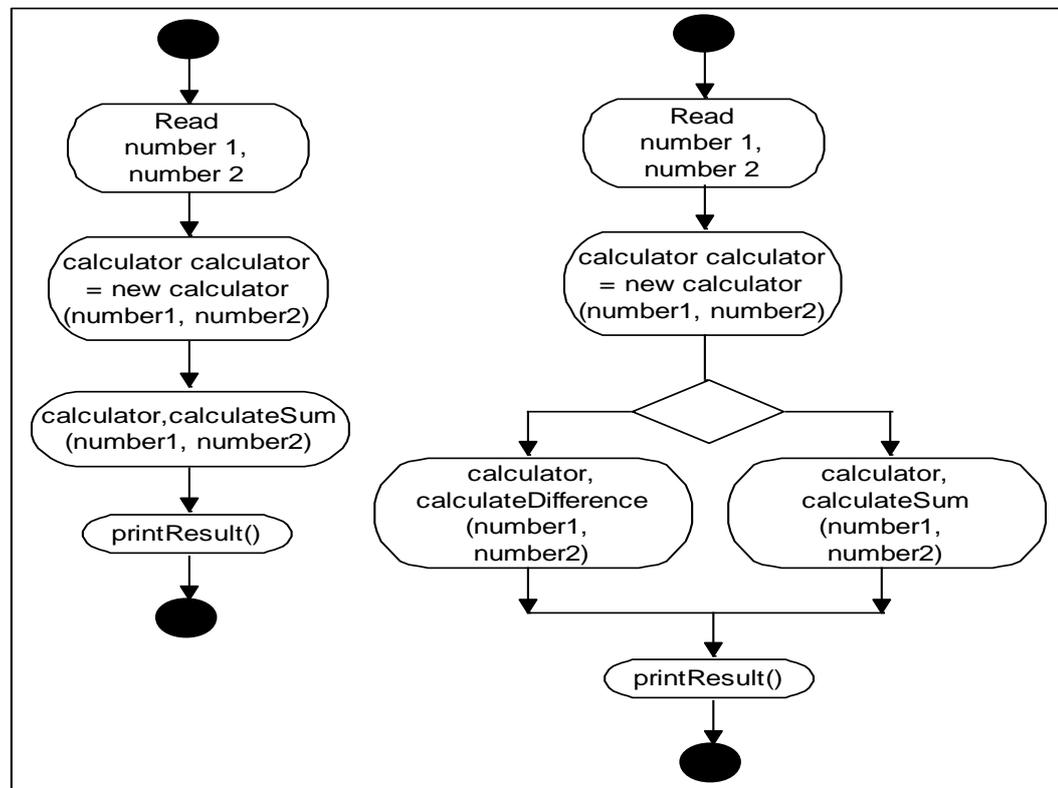


**Gambar II.3. Contoh Class Diagram**  
(Sumber : Adi Nugroho ; 2009: 61)

### II.7.3. Activity Diagram

Apakah langkah yang harus kita lakukan selanjutnya setelah kita membuat use case diagram ? use case diagram merupakan gambaran menyeluruh dan pada umumnya sangatlah tidak terperinci. Oleh karena itu, kita harus memperinci lagi perilaku sistem untuk masing-masing use case yang ada. Apa perkakas (tool) yang bisa kita gunakan ? jika kasus kita cukup sederhana, mungkin kita bisa menggunakan skenario seperti yang tercantum berikut,

sementara jika kasusnya cukup kompleks, kita mungkin bisa menggunakan activity diagram agar bisa mendapatkan gambaran yang lebih menyeluruh (Adi Nugroho ; 2009 : 10).



**Gambar II.4. Contoh Activity Diagram**  
(Sumber : Adi Nugroho ; 2009 : 74)

#### II.7.4. Sequence Diagram

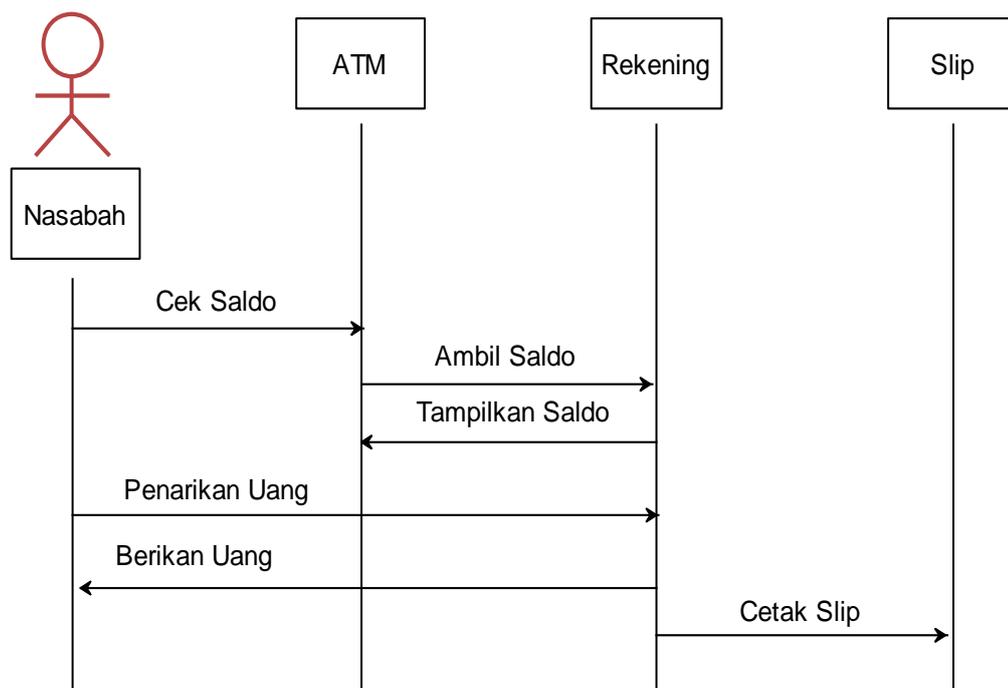
Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah diagram yang disusun berdasarkan urutan waktu. Kita membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan suatu aliran dari beberapa aliran di dalam *use case*.

Jadi dengan kata lain sekuensial diagram menunjukkan aliran fungsionalitas berdasarkan urutan waktu serta kejadian yang nantinya akan

menentukan metode/fungsi atribut masing-masing. Dimana fungsi-fungsi tersebut akan diterapkan pada suatu kelas/objek.

Perhatikan gambar II.5. dimana terlihat pengelompokkan *event-event* serta fungsi masing-masing atribut tersebut. Di dalam diagram terlihat jelas bagaimana aliran suatu proses kejadian dimana seorang nasabah yang akan melakukan transaksi dengan sebuah mesin ATM. Dari diagram tersebut kita mengetahui *event-event* yang terjadi, seperti : Nasabah memasukkan kartu ATM, Mesin ATM merespon dengan meminta *password* atau PIN, dan selanjutnya.

Kita dapat melihat setiap fungsi atribut dan *event-event* apa saja yang terjadi. Sehingga melalui diagram sekuensial ini kita dapat merancang suatu program aplikasi yang baik, sehingga dalam menghadapi sebuah kasus yang benar-benar kompleks diagram sekuensial ini sangat membantu.



**Gambar II.5. Contoh Sequence Diagram**  
(Sumber : Adi Nugroho ; 2009 : 102)

## II.8. Database

Dengan *Database*, data atau informasi dapat disimpan secara permanen. Informasi yang tadinya ada didalam *variabel*, akan segera hilang bersamaan dengan selesainya skrip *PHP* yang dieksekusi. Untuk itu diperlukan *database* untuk menyimpan informasi yang ingin dipertahankan saat eksekusi selesai. Misalnya informasi nama, alamat, tanggal lahir, dan lain-lain (Riyanto, dkk : 2009 : 306).

## II.9. Kamus Data Dan Normalisasi

Dalam suatu rancangan *database*, *data dictionary* digunakan untuk menjelaskan atau mendeskripsikan kolom-kolom pada masing-masing tabel yang akan dibuat dalam *database*. Deskripsi kolom yang dimaksud di sini meliputi tipe data, lebar karakter atau digit, serta keterangan tentang kunci relasi (Budi Raharjo : 2011 : 59)

**Tabel II.1. Tabel kategori**

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Kategori_id	INT	11	NOT NULL	Primary Key
Kategori_nama	VARCHAR	25		

**Tabel II.2. Tabel pengarang**

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Pengarang_id	CHAR	3	NOT NULL	Primary Key
Pengarang_nama	VARCHAR	30		

**Tabel II.3. Tabel penerbit**

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Penerbit_id	CHAR	4	NOT NULL	Primary Key
Penerbit_nama	VARCHAR	50		

**Tabel II.4. Tabel buku**

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Buku_isbn	CHAR	13	NOT NULL	Primary Key
Buku_judul	VARCHAR	75		
Penerbit_id	CHAR	4		
Buku_tglterbit	DATE	-		
Buku_jmlhalaman	INT	11		
Buku_deskripsi	TEXT	-		
Buku_harga	DECIMAL	10,0		

**Tabel II.5. Tabel link\_buku\_pengarang**

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Buku_isbn	CHAR	13	NOT NULL	Primary Key dan Forign Key
Pengarang_id	CHAR	3	NOT NULL	Primary Key dan Forign Key

**Tabel II.6. Tabel link\_buku\_kategori**

Nama Kolom	Tipe Data	Lebar	NULL?	Kunci
Buku_isbn	CHAR	13	NOT NULL	Primary Key dan Forign Key
kategori_id	CHAR	11	NOT NULL	Primary Key dan Forign Key

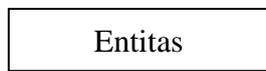
## II.10. Entity Relationship Diagram (ERD)

*Entity relationship diagram* merupakan salah satu alat bantu (berupa gambar) dalam *database relasional* yang berguna untuk menjelaskan hubungan atau relasi antartabel yang terdapat di dalam *database*. Dalam ERD kita juga dapat melihat daftar kolom yang menyusun masing-masing tabel. ERD inilah yang akan kita gunakan sebagai acuan untuk membahas materi dalam buku ini.

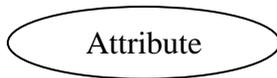
Berikut keterangan tanda yang digunakan :

1. 1-∞ menunjukkan relasi 1-ke-banyak
2. n-∞ menunjukkan relasi banyak-ke-banyak (Budi Raharjo : 2011 : 57)

Menurut Rosa A.S dan M. Shalahuddin (2011 : 60), Struktur *logis* (skema *database* dapat ditunjukkan secara *grafis* dengan ER yang dibentuk dari komponen-komponen berikut ini :



Persegi panjang mewakili kumpulan entitas.



Elips mewakili attribute



Belah ketupat mewakili relasi



Garis menghubungkan atribut dengan kumpulan entitas dengan relasi.