

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Tujuan dasar suatu sistem tergantung pada jenis sistem itu sendiri. Sebagai contoh, sistem peredaran darah manusia merupakan sistem biologi yang memiliki tujuan untuk mengedarkan darah yang mengandung oksigen dan sari makanan ke seluruh tubuh. Sedangkan sistem buatan manusia seperti sistem yang terdapat di sekolah, organisasi bisnis, atau instansi pemerintah juga mempunyai tujuan yang berbeda-beda. Organisasi bisnis biasanya memiliki tujuan yang lebih jelas (Anastasia Diana dan Lilis Setiawati : 2011 ; 2).

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

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 definisi “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 : 2012 : 1).

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 ; 2012 : 29).

### **II.3. Sistem Informasi Akuntansi**

Sistem informasi akuntansi (SIA) adalah sebuah sistem informasi yang menangani segala sesuatu yang berkenaan dengan akuntansi. Akuntansi sendiri sebenarnya adalah sebuah sistem informasi. Sistem informasi akuntansi (SIA) merupakan suatu kerangka pengkoordinasian sumber daya (data, materials, equipment, suppliers, personal, and funds) untuk mengkonversi input berupa data ekonomik menjadi keluaran berupa informasi keuangan yang digunakan untuk melaksanakan kegiatan suatu entitas dan menyediakan informasi akuntansi bagi pihak-pihak yang berkepentingan (Tata Sutabri : 2012 : 83).

#### **II.4. Microsoft Visual Basic**

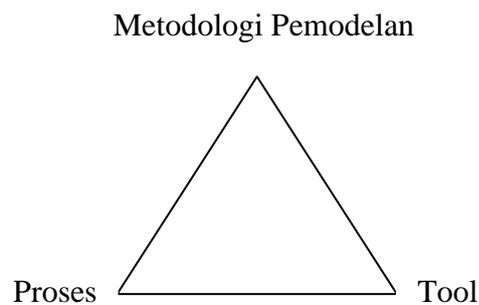
*Visual basic* merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi *windows*. *Visual basic 2008* merupakan aplikasi pemrograman yang menggunakan teknologi. *NET Framework 3.5*. Teknologi. *NET Framework 3.5* merupakan komponen *windows* yang terintegrasi serta mendukung pembuatan, penggunaan aplikasi, dan halaman *web*. Teknologi *.Net Framework 3.5* mempunyai 2 komponen utama, yaitu *CLR (Common Language Runtime)* dan *Class Library*. *CLR* digunakan untuk menjalankan aplikasi yang berbasis *.NET*, sedangkan *Library* adalah kelas pustaka atau perintah yang digunakan untuk membangun aplikasi (Wahana Komputer : 2010 : 2).

#### **II.5. Microsoft SQL Server**

Bahasa query merupakan bahasa khusus yang digunakan untuk melakukan manipulasi dan menanyakan pertanyaan (query) yang berhubungan dengan bahasa pemrograman, dimana bahasa query tidak memiliki kemampuan untuk menyelesaikan banyak masalah seperti bahasa pemrograman pada umumnya. Dalam pemrograman basis data, salah satu bahasa yang harus kita kuasai adalah SQL. SQL merupakan bahasa komputer standar yang digunakan untuk berkomunikasi dengan sistem manajemen basis data relasional (RDBMS) (Ema Utami dan Anggi Dwi Hartanto : 2012 : 63)

## II.6. Pengertian UML

Menurut Yuni Sugiarti (2013 : 34) *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.



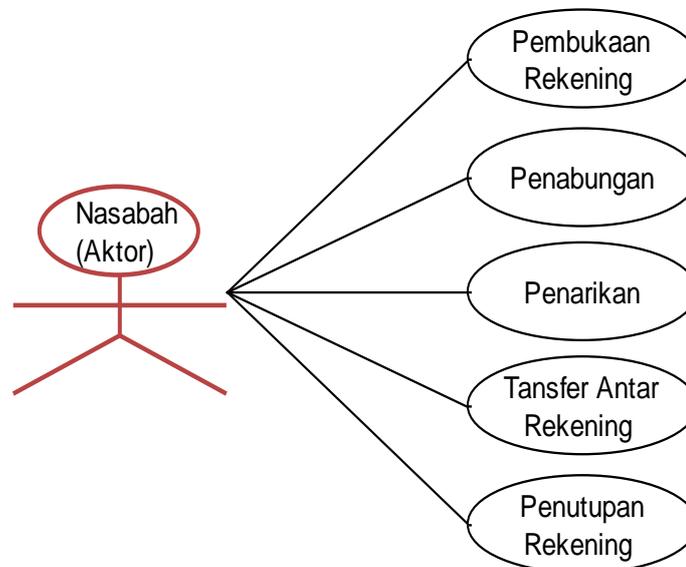
**Gambar II.1. The Triangle For Success**  
(Sumber : Yuni Sugiarti : 2013 ; 34)

### II.6.1. 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)

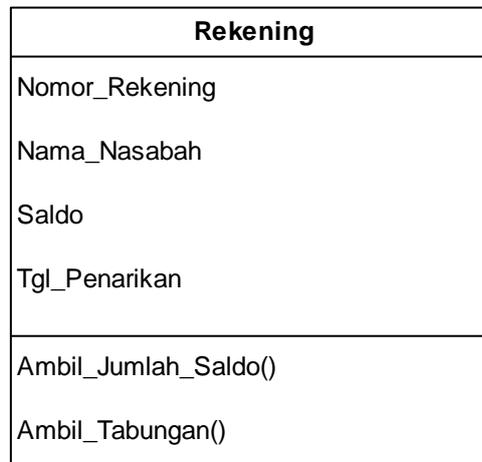


**Gambar II.2. Contoh Use Case Diagram**  
(Sumber : Adi Nugroho ; 2009 : 8)

### II.6.2. Class Diagram

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

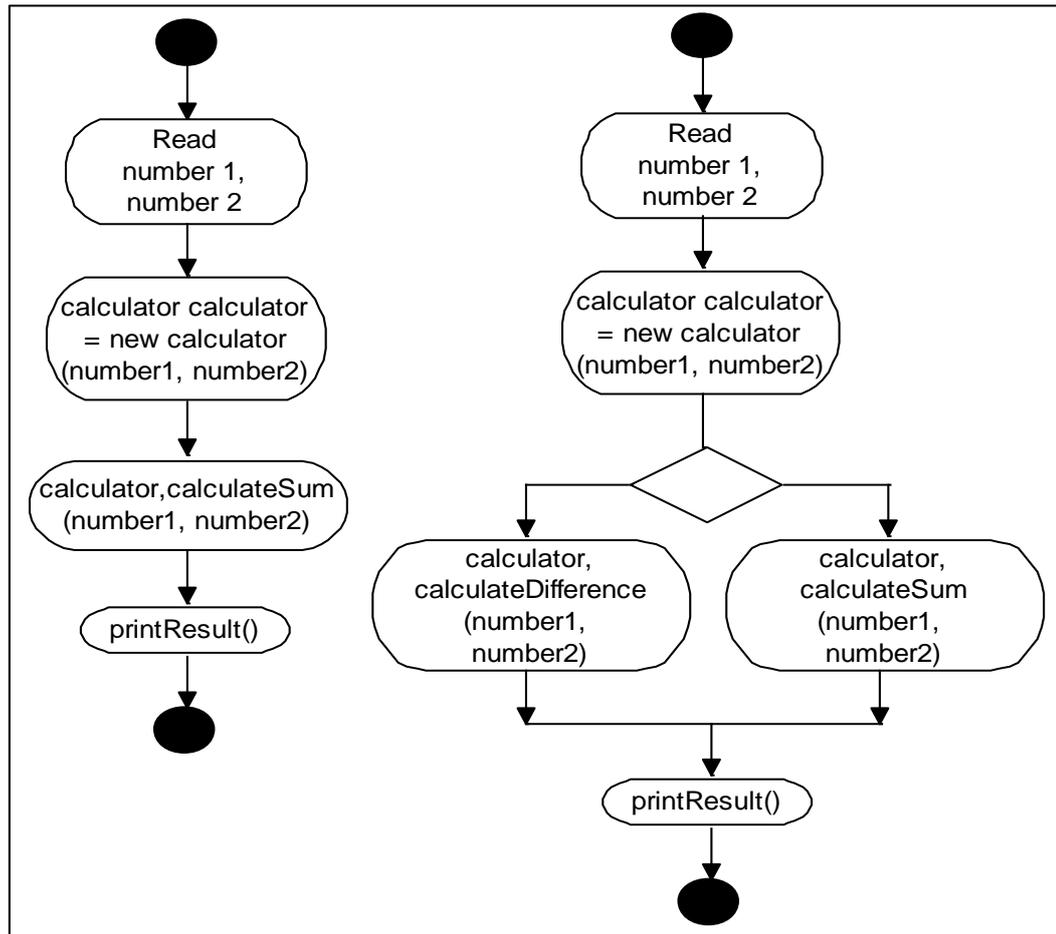
- a. Atribut merupakan *varabel-variabel* yang dimiliki oleh suatu kelas.
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.



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

### II.6.3. 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).

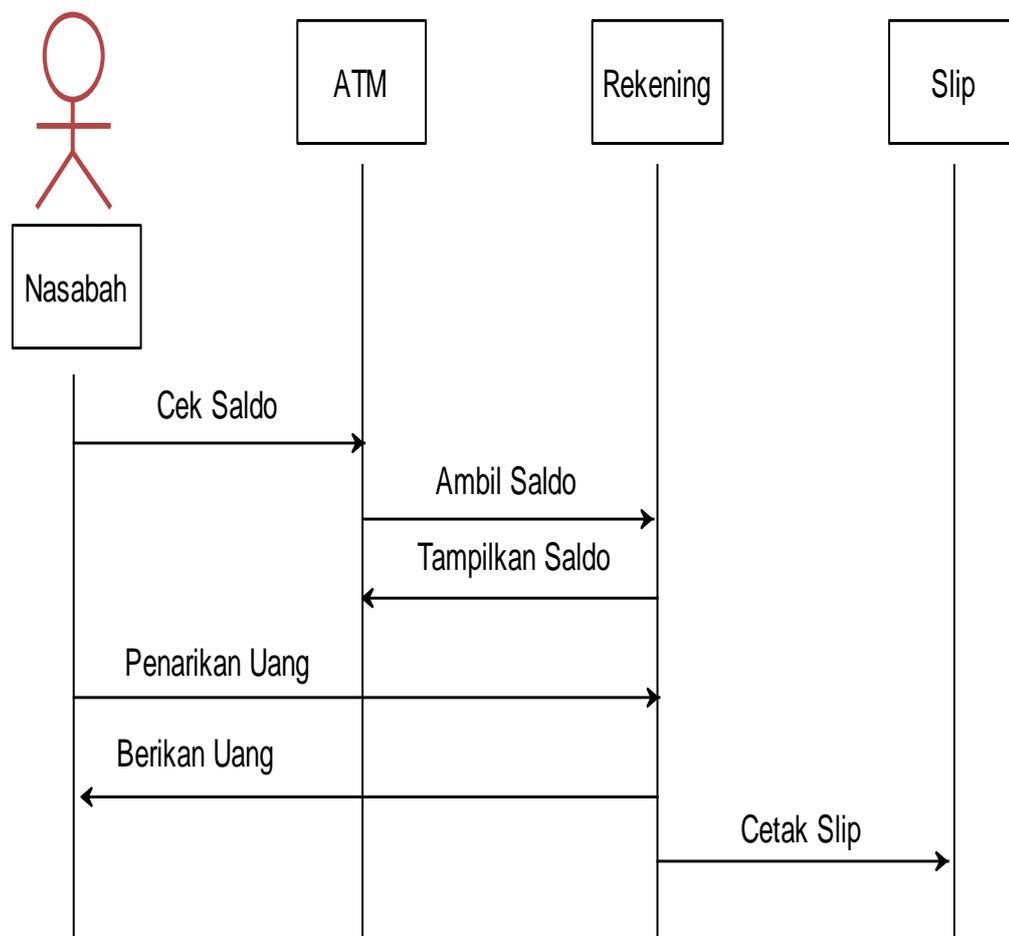


**Gambar II.4. Contoh Activity Diagram**  
(Sumber : Adi Nugroho ; 2009 : 74)

#### II.6.4. Sequence Diagram

Dalam termonologi penggunaan UML, kita bisa menggambarkan sequence diagram/collaboration diagram untuk menentukan metoda/fungsi yang kelak dapat diterapkan pada suatu kelas/objek. Dengan gambar tersebut, kita bisa melakukan pemetaan aliran event antarkelas/objek. Event-event itu dapat kita anggap sebagai metode/fungsi yang kelak dapat diterapkan dalam suatu kelas/objek. Kita tinggal melihat ke mana arah event terjadi. Sebagai contoh, event “masukkan kartu” mengalir dari nasabah ke mesin ATM sehingga event ini dapat dianggap sebagai metode untuk kelas nasabah, event “permintaan pin”

mengalir dari mesin ATM ke nasabah sehingga event ini dapat dianggap sebagai metoda untuk kelas mesin Atm, dan seterusnya. Dalam hal ini, tentunya tidak semua metoda yang dijumpai dalam skenario merupakan metoda yang bisa diimplementasikan dalam bentuk program komputer. Event “memasukkan kartu”, misalnya bersifat mekanis sehingga tidak perlu diimplementasikan dalam program komputer (kecuali jika event ini memerlukan program komputer untuk menggerakkan mesin ATM untuk menerima kartu).



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

## II.7. Database

Menurut Riyanto, dkk (2009 : 306-308), 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.

## II.8. Kamus Data Dan Normalisasi

Menurut Budi Raharjo (2011 : 59) 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.

**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

(Sumber : Budi Raharjo : 2011 : 59)

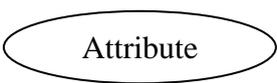
## II.9. Entity Relationship Diagram (ERD)

Menurut Budi Raharjo (2011 : 57) *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

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 :

**Tabel II.7. Komponen ERD**

Simbol	Keterangan
	Persegi panjang mewakili kumpulan entitas.
	Elips mewakili attribute
	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dengan relasi.

(Sumber : Rosa A.S dan M. Shalahuddin : 2011 : 60)

## II.10. Flat Rate

Menurut Kasmir (2008:138) Pembebanan bunga setiap bulan tetap dari jumlah pinjamannya, demikian pula pokok pinjamannya setiap bulan juga dibayar sama sehingga cicilan setiap bulan sama sampai kredit tersebut lunas.

Menghitung pokok pinjaman (PJ) per bulan sebagai berikut.

$$PJ = \frac{\text{Jumlah Pinjaman}}{\text{Jangka Waktu}} \Rightarrow PJ = \frac{\text{Rp.60.000.000,-}}{12 \text{ bulan}} = \text{Rp. 5.000.000,-}$$

Menhitung bunga (BG) per bulan adalah.

$$BG = \frac{\text{Bunga} \times \text{Nominal Pinjaman}}{12 \text{ Bulan}} \times 1$$

$$BG = \frac{24\% \times \text{Rp.60.000.000,-}}{12 \text{ Bulan}} \times 1 = \text{Rp.1.200.000,-}$$

### **II.11. Denda Jatuh Tempo**

Menurut Kasmir (2008:283) denda jatuh tempo adalah sangsi-sangsi yang diberikan pihak lessor kepada pihak lessee apabila lessee ingkar janji atau tidak memenuhi kewajibannya kepada pihak lessor sesuai perjanjian yang telah disepakati adalah sebagai berikut :

1. Berupa teguran lisan supaya segera melunasi
2. Jika teguran lisan tidak digubris, maka akan diberikan teguran tertulis;
3. Dikenakan denda sesuai perjanjian;
4. Penyitaan barang yang dipegang oleh lessee.