

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Menurut Anastasia Diana dan Lilis Setiawati (2011 : 2 )Pada bagian ini akan dibahas defenisi sistem informasi akuntansi dan istilah yang berkaitan dengan hal tersebut untuk membantu anda memahami seluruh materi dalam buku ini. Sistem merupakan serangkaian bagian saling berhubungan dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang juga saling tergantung dan bekerja sama untuk mencapai tujuan.

Menurut Tata Sutabri (2013 : 29) 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.

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

Menurut Tata Sutabri (2013 : 83) 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 ekonomi menjadi keluaran berupa informasi keuangan yang digunakan untuk melaksanakan kegiatan suatu entitas dan menyediakan informasi akuntansi bagi pihak-pihak yang berkepentingan.

### **II.4. Penggajian**

Menurut Soemarso S.R (2009 : 222) Upah dan gaji yang telah menjadi hak para pegawai tapi belum saatnya dibayar merupakan utang perusahaan. Utang ini serta beban yang berhubungan dengannya belum dicatat, sehingga perlu dibuatkan ayat jurnal penyesuaian. Upah dan gaji perusahaan dagang maju dibayar tiap hari Sabtu untuk masa seminggu sebelumnya.

### **II.5. Microsoft Visual Basic**

Menurut Wahana Komputer (2010 : 2) *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.

## **II.6. Microsoft SQL Server**

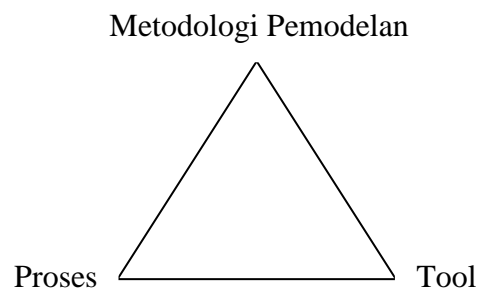
Menurut Ema Utami dan Anggi Dwi Hartanto (2012 : 63) Bahasa query merupakan bahasa khusus yang digunakan untuk melakukan manipulasi dan menanyakan pertanyaan (*query*) yang berhubungan dengan bahasa pemrograman, dimana bahasa query tidak memiliki kemampuan untuk menyelesaikan banyak masalah seperti bahasa 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).

## **II.7. Pengenalan Database**

Menurut Budi Raharjo (2011 : 245) 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*.

## II.8. 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)

Menurut Prabowo Pudjo Widodo dan Herlawati (2011 : 6-7), *UML* singkatan dari *Unified Modelling Language* yang berarti bahasa permodelan standar. *UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

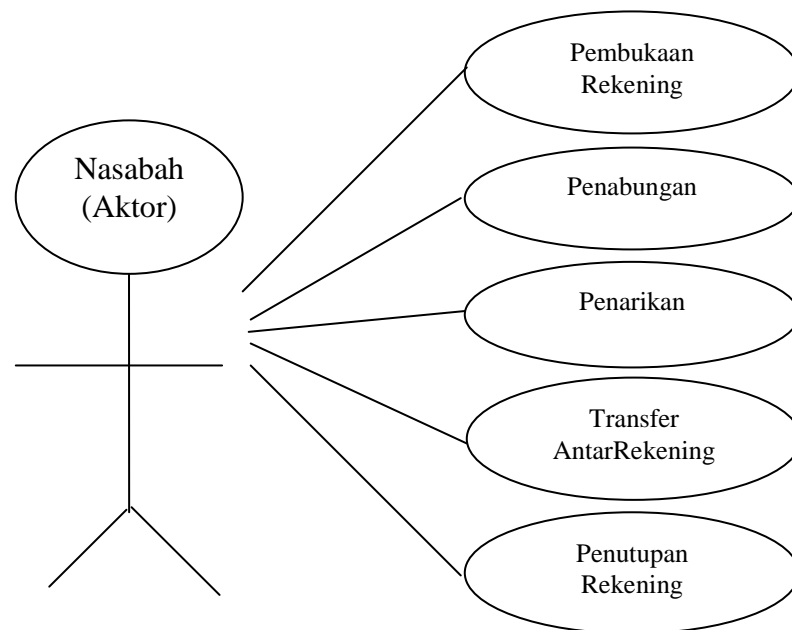
1. Merancang perangkat lunak
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

### II.8.1. Use Case Diagram

Menurut Adi Nugroho (2009 : 7) 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.



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

### II.8.2. Class Diagram

Menurut Adi Nugroho (2009 : 18) *Class* didefinisikan sebagai kumpulan/himpunan objek yang memiliki kesamaan dalam atribut/properti, perilaku (operasi), serta cara berhubungan dengan objek lain.

Selain itu, kita juga mendefinisikan objek sebagai konsep, abstraksi dari sesuatu dengan batas nyata, sehingga kita dapat menggambarkan secara sistematis. Pemahaman objek memiliki dua fungsi, yaitu :

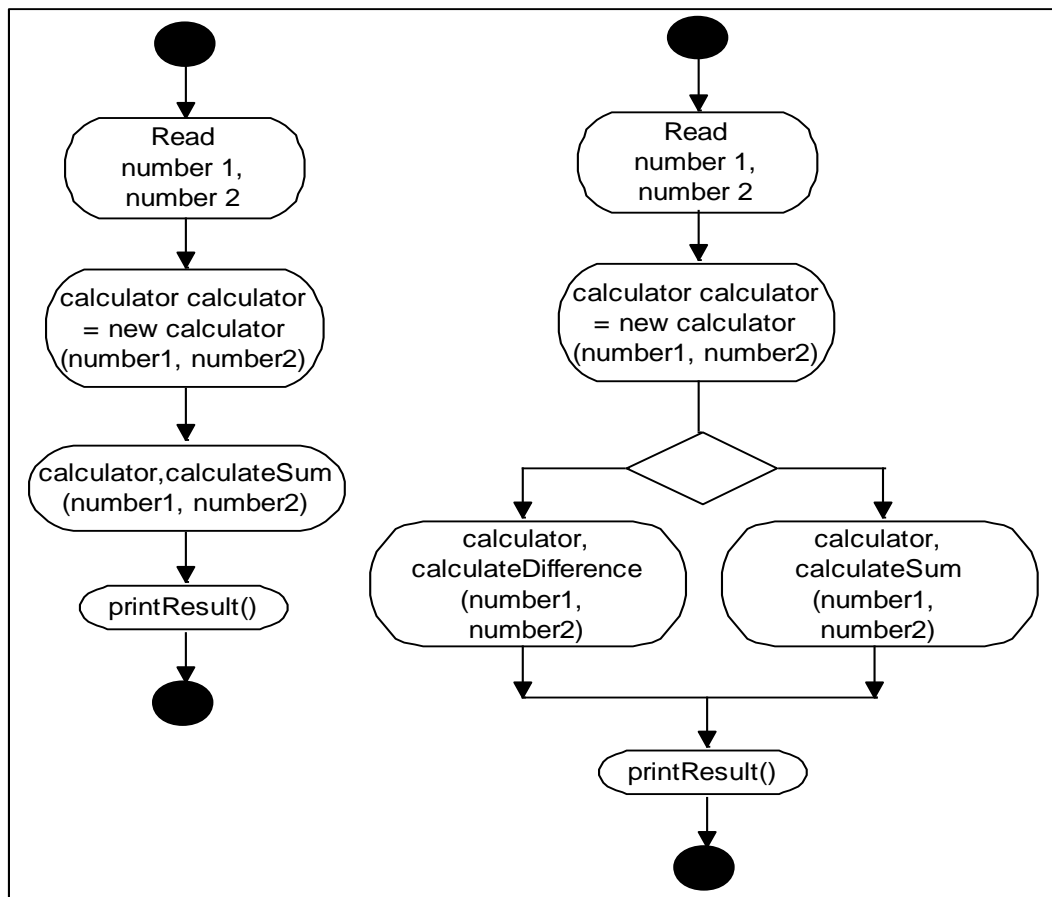
- a. Memudahkan untuk mempelajari secara seksama hal-hal yang ada di dunia nyata.
- b. Menyediakan suatu dasar yang kuat dalam implementasi ke dalam sistem terkomputerisasi.



**Gambar II.3. Contoh Class Diagram**  
(Sumber : Adi Nugroho ; 2009: 39)

### II.8.3. Activity Diagram

Menurut Adi Nugroho (2009 : 10) 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.



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

#### II.8.4. 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*.

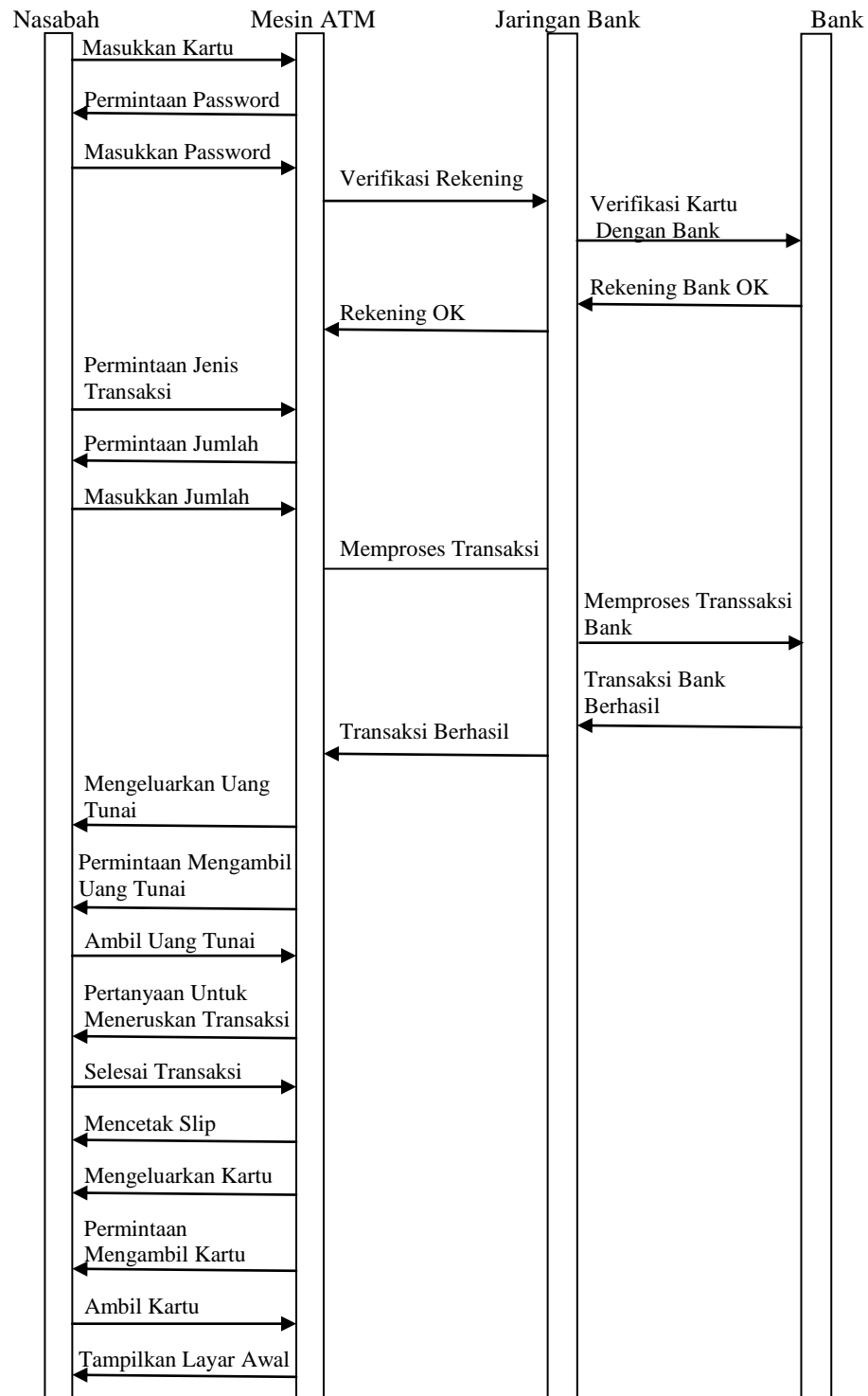
Jadi dengan kata lain sekuensial diagram menunjukkan aliran fungsionalitas berdasarkan urutan waktu serta kejadian yang nantinya akan



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 : 36)**

## II.9. Kamus Data

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.

## II.10. Normalisasi

Menurut Janner Simarmata (2007: 197), Proses normalisasi merupakan proses pengelompokan data ke dalam bentuk tabel atau relasi atau file untuk menyatakan entitas dan hubungan sehingga terwujud satu bentuk basis data yang mudah dimodifikasi.

Menurut Janner Simarmata dan Iman Paryudi (2006 : 79-84), terdapat beberapa langkah Normalisasi diantaranya :

### 1. Bentuk Normal Pertama (1NF)

Sebuah tabel relasional secara definisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah *atomik*. Ini berarti kolom-kolom tidak mempunyai nilai berulang.

### 2. Bentuk Normal Kedua (2NF)

Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama.

### 3. Bentuk Normal Ketiga (3NF)

Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama.

### 4. Bentuk Normal Boyce-Code (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Bentuk Normal Boyce-Code (BCNF) adalah versi 3NF yang lebih teliti dan berhubungan dengan tabel relasional yang mempunyai banyak kunci kandidat, kunci kandidat gabungan, dan kunci kandidat yang saling tumpang tindih.

## II.11. *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.  $1-\infty$  menunjukkan relasi 1-ke-banyak
2.  $n-\infty$  menunjukkan relasi banyak-ke-banyak

## **II.12. PPH 21**

Menurut Mardiasmo (2006 : 152) Ketentuan pasal 21 undang-undang pajak penghasilan mengatur tentang pembayaran pajak dalam tahun berjalan melalui pemotongan pajak atas penghasilan yang diterima atau diperoleh dari wajib pajak pribadi dalam negeri sehubungan dengan pekerjaan, jasa, dan kegiatan. Penerima penghasilan yang dipotong PPh 21 adalah :

1. Pejabat negara
2. Pegawai negeri sipil
3. Pegawai
4. Pegawai tetap
5. Pegawai dengan status wajib pajak luar negeri
6. Tenaga lepas
7. Penerima pensiun
8. Penerima honorarium
9. Penerima upah

Contoh :

Mardiasmo (2006 : 170) Hasan bekerja pada perusahaan PT. ABC dengan memperoleh gaji sebulan Rp. 2.000.000,00 dan membayar iuran pensiun sebesar Rp. 50.000,00. Hasan menikah dan mempunyai 1 anak.

Perhitungan PPh 21 :

Gaji sebulan

Pengurangan :

1. Biaya jabatan :

5 % x Rp. 2.000.000,00      Rp. 100.000,00

2. Dana Pensiun:                      Rp. 50.0000,00

Rp. 150.000,00

Penghasilan neto sebulan                      Rp. 1.850.000,00

Penghasilan neto setahun adalah

12 x Rp. 1.850.000,00                      Rp. 22.000.000,00

3. PTKP setahun

Untuk WP sendiri                      Rp. 13.200.000,00

Tambahan WP Kawin                      Rp. 1.200.000,00

Tambahan 1 Anak                      Rp. 1.200.000,00

Rp. 15.600.000,00

Penghasilan kena pajak setahun                      Rp 6.600.000,00

PPh Pasal 21 terutang :

5 % x Rp. 6.000.000,00      Rp. 330.000,00

PPh Pasal 21 sebulan :

Rp. 330.000,00 : 12                      Rp. 27.500,00