

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

Pada bagian ini akan dibahas definisi 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 (Anastasia Diana dan Lilis Setiawati:2011;2).

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.2. Pengertian 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 (Tata Sutabri : 2013 : 1).

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 : 2013 : 83)

II.4. Pengertian UML

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 :

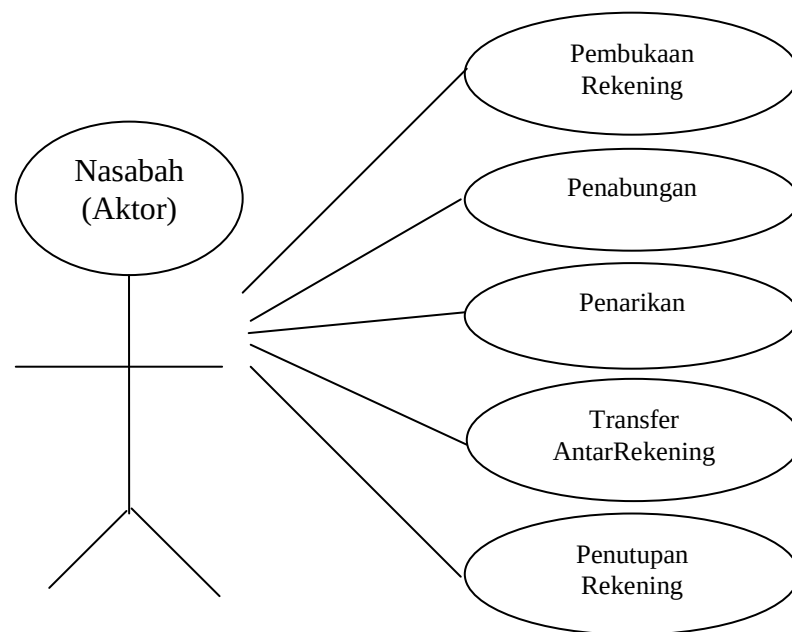
1. Merancang perangkat lunak
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

II.4.1. 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)



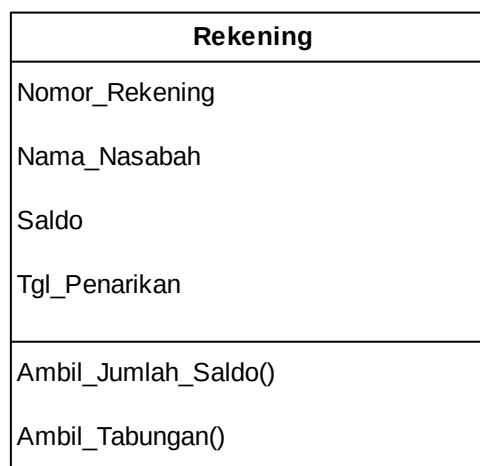
Gambar II.1. Contoh Use Case Diagram
(Adi Nugroho ; 2009 : 8)

II.4.2. Class Diagram

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

Selain itu, kita juga mendefenisikan 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 (Adi Nugroho ; 2009 :17).

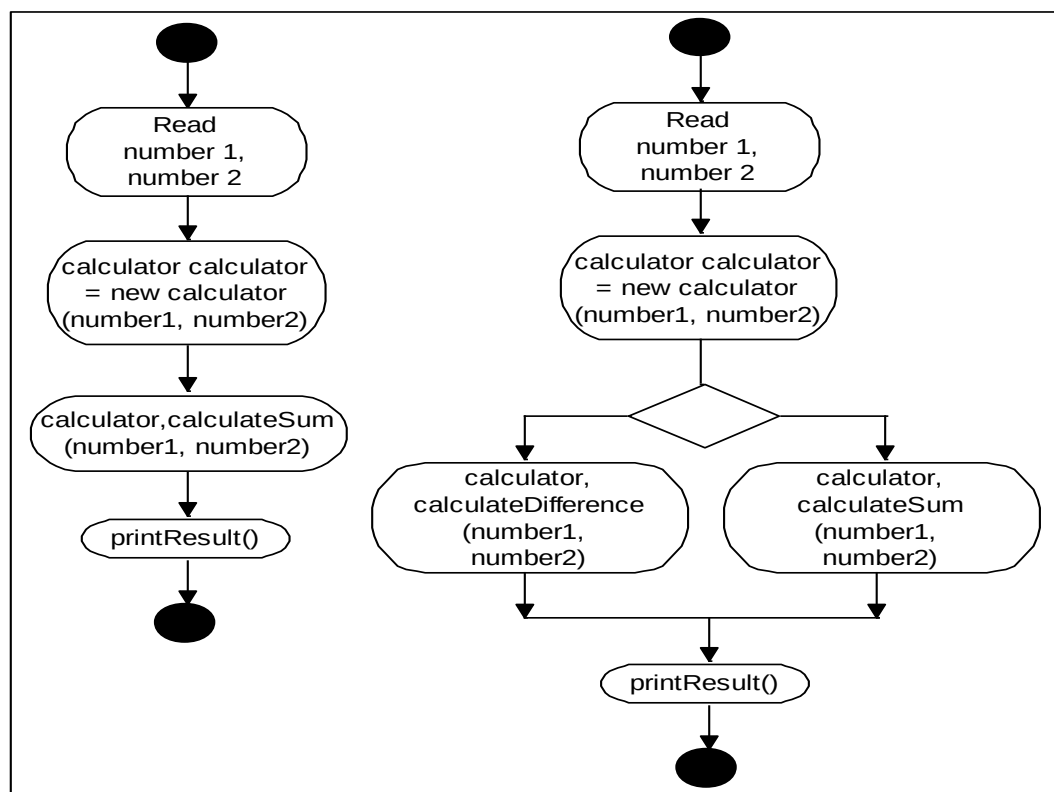


Gambar II.2. Contoh Class Diagram
(Nugroho ; 2009: 61)

II.4.3. 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.3. Contoh Activity Diagram
(Adi Nugroho ; 2009 : 74)

II.4.4. Sequence Diagram

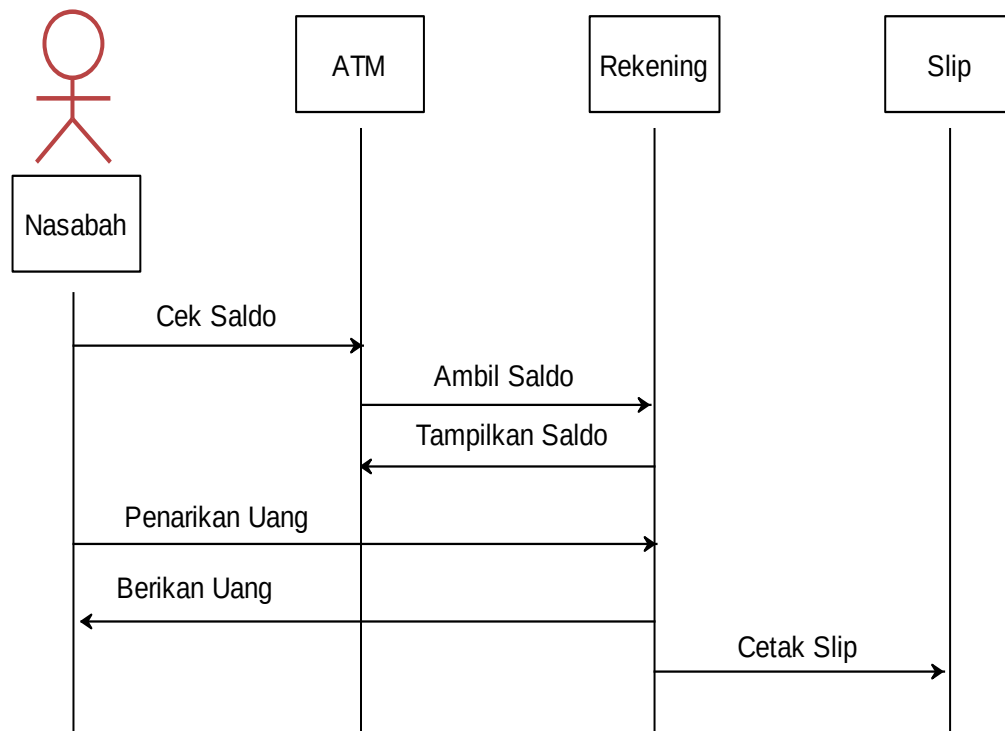
Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah

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.4. 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.4. Contoh Sequence Diagram
(Adi Nugroho ; 2009 : 102)

II.5. 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.6. 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.7. Basis Data

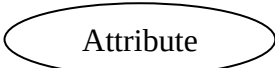
Menurut Janner Simarmata (2007 : 1), Kata “*basis data*” bisa digunakan untuk menguraikan segala sesuatu dari sekumpulan data tunggal, seperti daftar telepon. *Basis data* terdiri dari *file-file* fisik yang ditetapkan berdasarkan komputer saat menerapkan perangkat lunak basis data. Sedangkan menurut *Stephens* dan *Plew* (Dalam Janner Simarmata dan Iman Paryudi; 2006 : 1), Basis Data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah

data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan kedalam basis data, dimodifikasi, dan dihapus.

II.8. *Entity Relationship Diagram (ERD)*

Menurut Janer Simarmata dan Imam Prayudi (2006:59) Struktur yang mendasari suatu basisdata adalah model data yang merupakan kumpulan alat-alat konseptual untuk mendeskripsikan data, relasi data, data *semantik*, dan batasan konsistensi. Untuk mengilustrasikan konsep model data dapat disajikan dengan *entity relationship* model. *Entity relationship* mendeskripsikan rancangan basisdata pada tingkatan *logis*. *Entity relationship (ER)* data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut *entitas* dan hubungan antar obyek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain (Janer Simarmata dan Imam Prayudi ; 2006 : 59). Struktur *logis* (skema *database* dapat ditunjukan secara *grafis* dengan *ER* yang dibentuk dari komponen-komponen dapat dilihat pada tabel II.1. berikut ini :

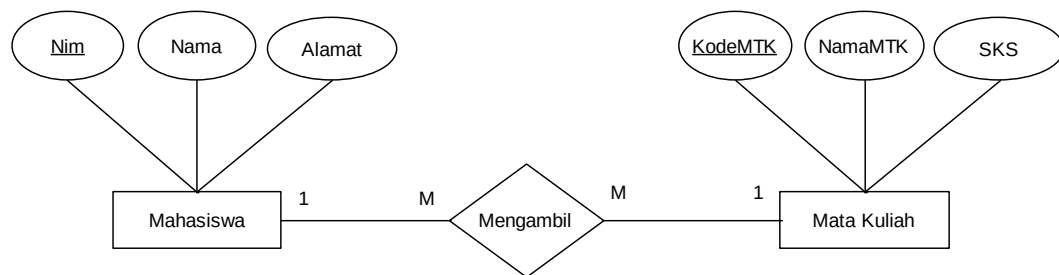
Tabel II.1. Komponen-komponen *Entity Relationship*

Simbol	Arti
	Persegi panjang mewakili kumpulan entitas.
	Elips mewakili attribute
	Belah ketupat mewakili relasi

_____	Garis menghubungkan atribut dengan kumpulan entitas dengan relasi.

(Janer Simarmata dan Imam Prayudi ; 2006 : 60)

Sebagai ilustrasi, bayangan anda mengambil bagian sistem basis data universitas yang terdiri dari mahasiswa dan mata kuliah. Gambar II.5. menunjukkan *ER Diagram* dari contoh. Diagram menunjukkan bahwa ada dua kumpulan *entitas*, yaitu mahasiswa dan mata kuliah, dan bahwa relasi mengambil mahasiswa dan mata kuliah.



Gambar II.5. Diagram ER

(Janer Simarmata dan Imam Prayudi ; 2006 : 60)