

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Dengan berbagai pendekatan, beragam pula istilah “sistem” didefinisikan. Sistem adalah suatu pengorganisasian yang saling berinteraksi, saling bergantung dan terintegrasi dalam kesatuan variabel atau komponen. Terdapat dua kelompok pendekatan sistem, yaitu menekankan pada prosedur dan komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkelompok dan bekerjasama untuk melakukan kegiatan pencapaian sasaran tertentu. Makna dari prosedur sendiri, yaitu urutan yang tepat dari tahapan-tahapan instruksi. Sedangkan pendekatan yang menekankan pada komponen mendefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. *“Serangkaian atau tatanan elemen-elemen yang diatur untuk mencapai tujuan yang ditentukan sebelumnya melalui pemrosesan informasi”* (Riyanto, dkk; 2009 : 22).

#### **II.2. Siklus Informasi**

Menurut Riyanto, dkk. (2009 : 24), Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui model tertentu menjadi informasi yang dapat dimanfaatkan oleh penerima dalam membuat keputusan dan melakukan tindakan, yang berarti

melakukan suatu tindakan lain yang akan membuat sejumlah data kembali. Data yang masih belum diolah akan disimpan dalam bentuk *database*. Data yang disimpan ini nantinya dapat diambil kembali untuk diolah kembali menjadi informasi. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model tertentu dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*information cycle*).

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

Organisasi tergantung pada sistem informasi untuk dapat berdaya saing. Informasi juga merupakan sumber daya, sama seperti pada pabrik dan peralatan. Produktivitas, sebagai faktor yang penting untuk mempertahankan daya saing perusahaan, dapat ditingkatkan dengan sistem informasi yang lebih baik. Akuntansi, sebagai suatu sistem informasi, mengidentifikasi, mengumpulkan, memproses dan mengkomunikasikan informasi ekonomi mengenai suatu entitas ke berbagai kelompok orang. Informasi merupakan suatu data yang diorganisasi yang dapat mendukung ketepatan pengambilan keputusan. Sistem merupakan sekumpulan sumber daya yang dapat terkait untuk mencapai suatu tujuan. SIA (Sistem Informasi Akuntansi) merupakan kumpulan sumber daya, seperti manusia dan peralatan, yang dirancang untuk mengubah data keuangan dan data lainnya ke dalam informasi. Informasi tersebut dikomunikasikan kepada para pembuat keputusan. Sistem informasi akuntansi melakukan hal tersebut dengan sistem manual atau melalui sistem terkomputerisasi. (George H. Bodnar;2006:3)

Analog dengan definisi sebelumnya, SIA (Sistem Informasi Akuntansi) adalah sistem berbasis computer yang dirancang untuk mentransformasi data akuntansi menjadi informasi. (George H. Bodnar; 2006: 8)

### **II.3.1. Pengembangan Sistem Akuntansi**

Dalam aktivitas pengembangan sistem, diharapkan akuntan dan auditor dapat bertindak secara profesional. Akuntansi dapat menjalankan aktivitas pengembangan sistem baik untuk perusahaan mereka sendiri atau untuk perusahaan lain, dalam hal mereka memiliki posisi sebagai konsultan. *Auditor eksternal* maupun *internal* berhadapan dengan aktivitas pengembangan sistem pada saat mereka mengevaluasi pengendalian sistem informasi sebagai bagian dari penugasan audit suatu perusahaan. (George H. Bodnar;2006:22)

### **II.3.2. Karakteristik Pengembangan Sistem Akuntansi**

Sebuah proyek pengembangan sistem biasanya terdiri dari tiga fase, yaitu analisa sistem, desain sistem, dan implementasi sistem. Analisa sistem melibatkan penyusunan solusi dan evaluasi solusi untuk menyelesaikan masalah sistem. Analisa sistem menekankan tujuan sistem secara keseluruhan. Dasar dari analisa sistem ini adalah timbal balik antar tujuan sistem. Tujuan umum analisa sistem, yaitu:

1. Untuk meningkatkan kualitas ekonomi.
2. Untuk meningkatkan pengendalian internal.
3. Untuk meminimalkan biaya, jika memungkinkan.

Desain sistem merupakan proses merancang secara rinci solusi yang telah dipilih dalam proses analisis. Desain sistem mencakup evaluasi efektivitas dan efisiensi alternatif rancangan sistem terkait dengan kebutuhan sistem secara keseluruhan. Implementasi sistem merupakan proses penerapan prosedur dan metode yang telah dirancang ke dalam operasi. Implementasi sistem mencakup pengujian solusi sebelum implementasi, dokumentasi, serta evaluasi sistem pada saat sistem tersebut mulai dioperasikan untuk memastikan bahwa sistem berfungsi sesuai dengan yang telah direncanakan. (George H. Bodnar;2006:23).

#### **II.4. Pengertian Aktiva Tetap**

Aktiva tetap adalah harta yang dimiliki oleh perusahaan yang dapat dipakai atau memberikan manfaat dalam jangka waktu yang lebih dari satu tahun. Aktiva tetap pada umumnya mempunyai nilai yang cukup tinggi dan dapat dipakai untuk perusahaan sendiri, sedangkan aktiva yang diperdagangkan tergolong persediaan barang dagang. Misalnya kendaraan, sepeda motor, dan lain-lain, kalau barang tersebut dipakai perusahaan untuk keperluan perusahaan termasuk aktiva tetap, tetapi kalau barang tersebut diperjualbelikan masuk persediaan. Aktiva ini ada dua macam yaitu aktiva tetap berwujud (*tangible fixed assets*) dan aktiva tetap tak berwujud (*intangible fixed assets*) (Bantu Tampubolon, dkk : 2009 ; 285).

## II.5. Penyusutan Aktiva Tetap

Aktiva tetap yang dipakai mempunyai masa pakai yang terbatas, oleh karena aktiva tetap yang dipakai selama beberapa periode jelas nilainya semakin menurun (menyusut), dan terbukti bila dijual nilainya semakin menurun. Oleh karena itu aktiva tetap dalam akuntansi nilai yang susut diperhitungkan biaya penyusutannya (beban penyusutan) pada setiap akhir periode akuntansi dalam penyusunan laporan keuangan. Sehingga nilai aktiva semakin menurun dalam pelaporannya atau dilaporkan harga perolehan dikurangi penyusutannya. Aktiva tetap yang nilainya tidak mengalami penurunan adalah tanah (land), maka tanah tidak dilakukan pengurangan atas penurunan nilai. Jadi dilaporkan dalam neraca sebesar harga perolehannya saja (Bantu Tampubolon, dkk : 2009 ; 288).

## II.6. Metode Satuan Produksi (Jam Jasa)

Menurut Bantu Tampubolon, dkk (2009 ; 296) Penentuan besarnya penyusutan besarnya metode satuan produksi tergantung dari besar kecilnya kapasitas produksi pada tahun yang bersangkutan, dengan terlebih dahulu menghitung beban penyusutan persatuan.

$$\text{Biaya penyusutan persatuan} = \frac{\text{Harga perolehan} - \text{Nilai residu}}{\text{Taksiran hasil produksi}}$$

$$\text{Biaya penyusutan persatuan} = \frac{650.000 - 50.000}{30.000 \text{ Satuan}} = \text{Rp. 20,-}$$

Tahun 1990	6.100 Satuan
Tahun 1990	6.300 Satuan
Tahun 1992	7.000 Satuan
Tahun 1993	6.500 Satuan
Tahun 1994	4.100 Satuan

**Tabel II.1. Daftar Tabel Penyusutan**

**Metode Satuan Produksi**

<b>Tahun</b>	<b>Perhitungan</b>	<b>Penyusutan</b>	<b>Akumulasi Penyusutan</b>	<b>Nilai Buku</b>
-	-	-	-	Rp. 650.000
1990	6.100 Satuan x Rp. 20,-	Rp. 122.000	Rp. 122.000	Rp. 528.000
1990	6.300 Satuan x Rp. 20,-	Rp. 126.000	Rp. 248.000	Rp. 402.000
1992	7.000 Satuan x Rp. 20,-	Rp. 140.600	Rp. 388.600	Rp. 262.000
1993	6.500 Satuan x Rp. 20,-	Rp. 130.000	Rp. 518.000	Rp. 132.000
1994	4.100 Satuan x Rp. 20,-	Rp. 82.000	Rp. 600.000	Rp. 50.000

**(Bantu Tampubolon, dkk. : 2009 ; 297)**

## **II.7. Pengertian UML**

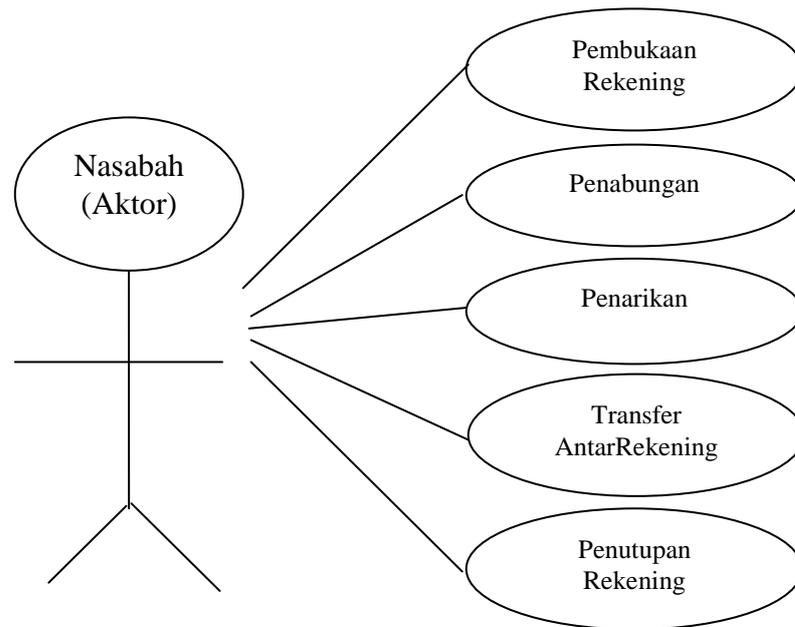
Menurut Prabowo Pudjo Widodo dan Herlawati (2011 : 6-7), *UML* singkatan dari *Unified Modelling Language* yang berarti bahasa permodelan standar. UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

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.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)



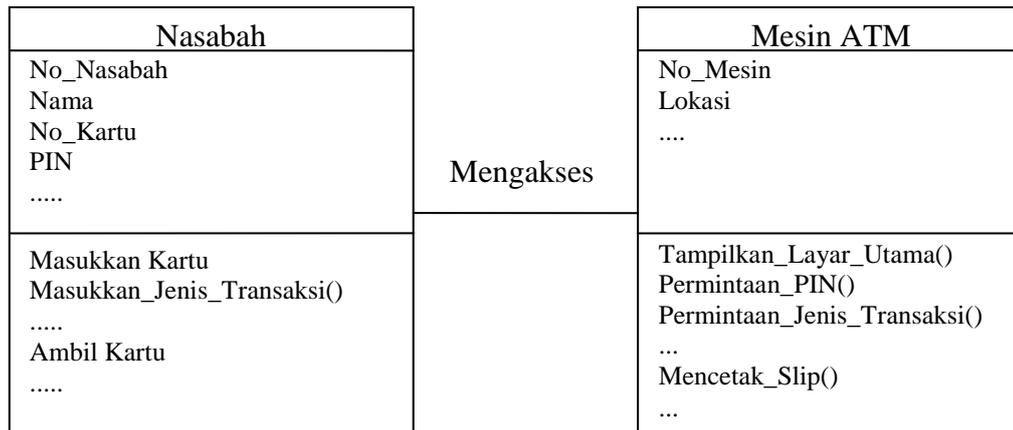
**Gambar II.1. Contoh Use Case Diagram  
(Adi Nugroho ; 2009 : 8)**

### II.7.2. 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 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 (Adi Nugroho ; 2009 :17).



**Gambar II.2. Contoh Class Diagram  
(Nugroho ; 2009: 39)**

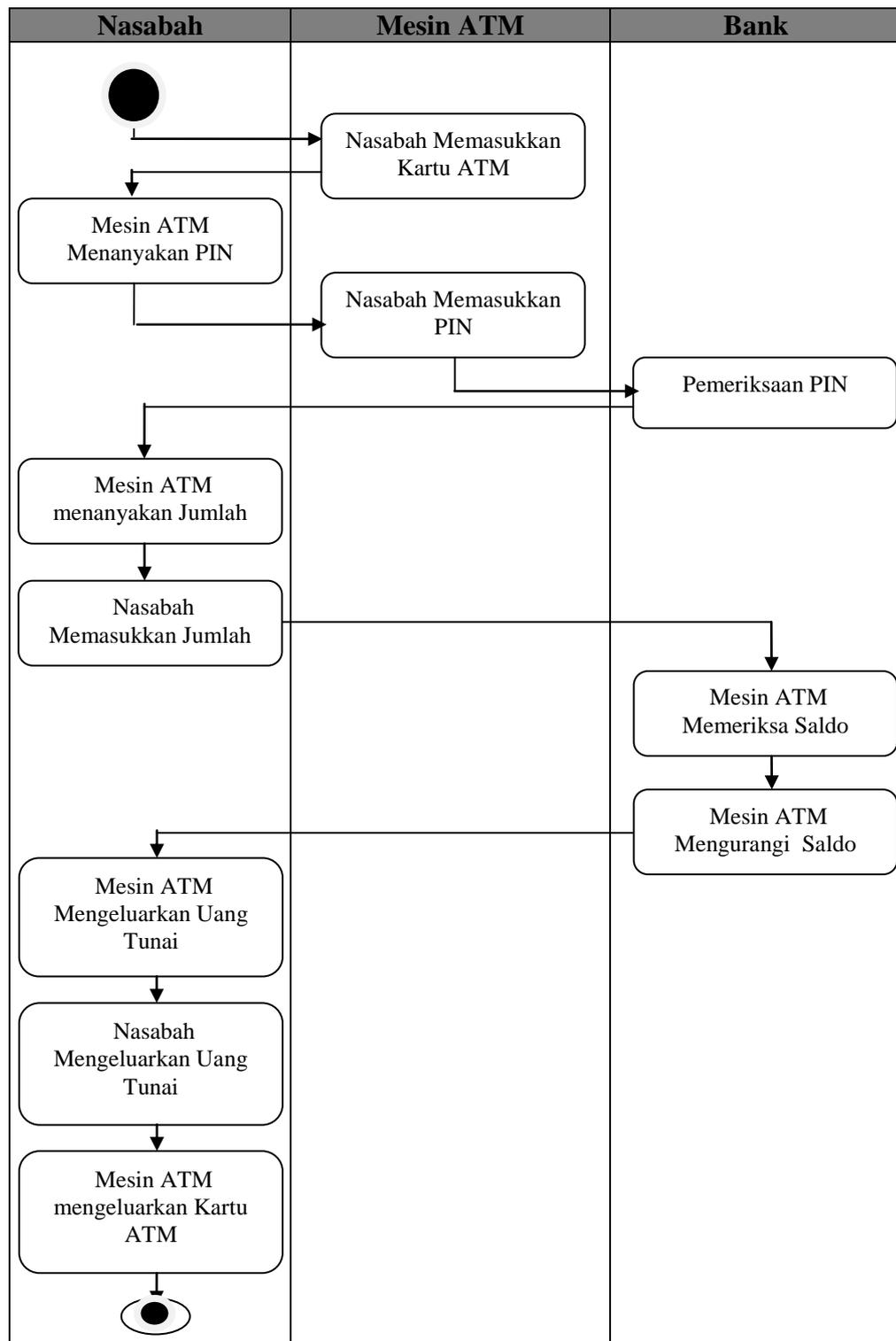
### II.7.3. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*). Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of event*) dalam *use case*.

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *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. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis *horizontal* atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu (Adi Nugroho ; 2009 : 13).



**Gambar II.3. Contoh Activity Diagram**  
(Adi Nugroho ; 2009 : 11)

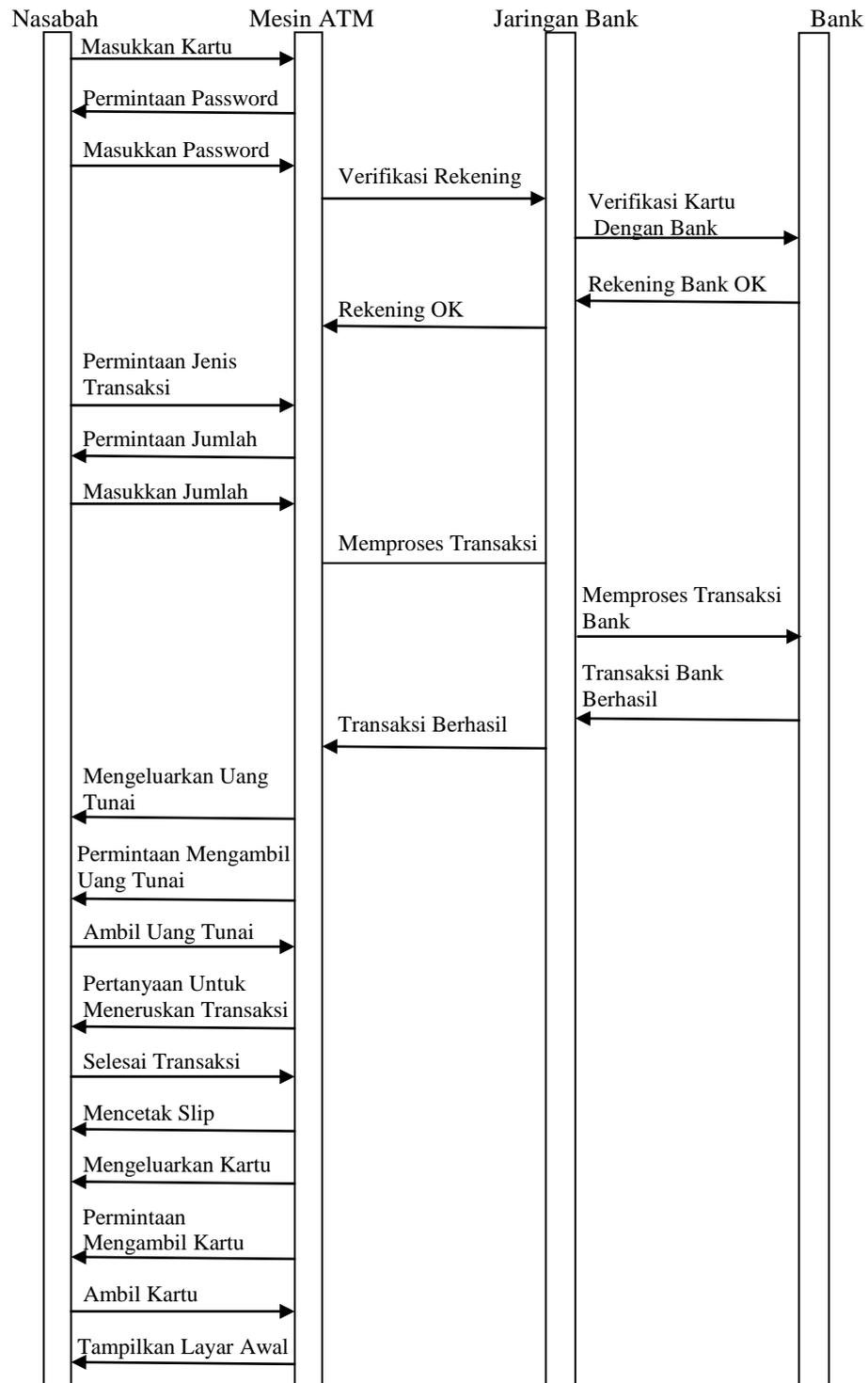
#### II.7.4. Sequence Diagram

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.6. 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 : 36)

## II.8. Desain Database

Menurut Yuniar Supardi (2008 : 9) *Desain database* merupakan pekerjaan yang penting dalam pembuatan atau pengembangan sistem, karena desain *database* akan mendapatkan susunan data atau *table* yang efektif dan efisien. Alat desain *database* yang populer ada dua, yaitu : ERD (*Entity Relationship Diagram*) dan Normalisasi. Jika memakai *Normalisasi* harus mendapatkan Data Dasar (Dokumen Dasar), sedangkan ERD tidak perlu. Dalam desain ERD terbagi dua tahapan yaitu: *Preliminary Design* (Desain Awal) dan *Final Design* (Desain Akhir). Tetapi disain Akhir dari ERD juga berisi Normalisasi.

## II.9. 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.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 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).

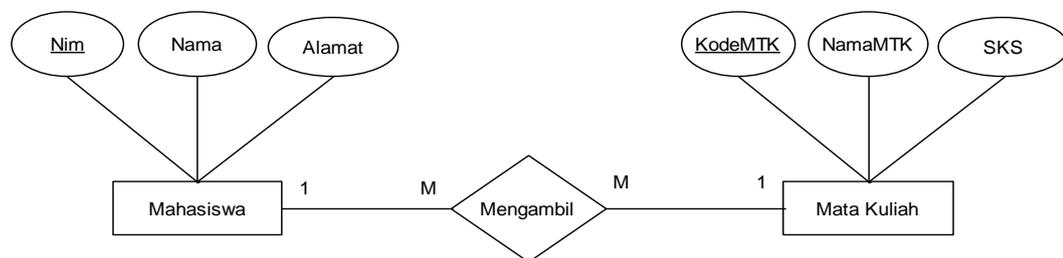
Struktur *logis* (skema *database* dapat ditunjukkan secara *grafis* dengan ER yang dibentuk dari komponen-komponen dapat dilihat pada tabel II.1. berikut ini :

**Tabel II.2. Komponen-komponen Entity Relationship**

Simbol	Arti
 Entitas	Persegi panjang mewakili kumpulan entitas.
 Attribute	Elips mewakili attribute
 Relasi	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**

(Sumber : Janer Simarmata dan Imam Prayudi ; 2006 : 60)