

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Menurut Riyanto, dkk (2009 : 22) 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”*.

#### **II.2. 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*).

Menurut Riyanto, dkk. (2009 : 23), Agar menjadi informasi yang berguna, data perlu diolah melalui sebuah siklus. Siklus ini disebut siklus pengolahan data (*data processing life cycle*).

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

Menurut Anastasia Diana dan Lilis Setiawati (2011 : 2) 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. 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. Analong dengan defenisi sebelumnya, SIA (Sistem Informasi Akuntansi) adalah sistem berbasis computer yang dirancang untuk mentransformasi data akuntansi menjadi informasi. (George H. Bodnar; 2007: 8)

#### II.4. Metode FIFO

Dalam metode masuk pertama keluar pertama (MPKP) atau FIFO, dinyatakan bahwa biaya yang melekat pada barang proses awal periode perhitungan secara terpisah dengan biaya produksi yang terjadi, dan dianggap selesai dan ditransfer pertama kali atau ditambahkan pada periode sekarang. Barang pertama dihasilkan pada periode sekarang adalah barang dalam proses awal periode (Mursyidi : 2008 ; 102).

Menurut Mursyidi (2008 ; 102 ).Persediaan yang masih ada :

10/01 masih ada 50 unit @ Rp. 110,- = Rp. 5.500,-

24/01 masih ada 100 unit @ Rp. 115,- = Rp. 11.500,-

Rp. 15.500,-

#### II.5. 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* Desain (Disain Awal) dan *Final Design* (Disain Akhir). Tetapi disain Akhir dari ERD juga berisi Normalisasi.

## II.6. Konsep Database

Menurut Edhy Sutanta (2011 : 25) Sebelum memperoleh definisi formal basis data, kita akan mencoba memahaminya secara sederhana terlebih dahulu. Istilah basis data tersusun atas dua suku kata, yaitu basis dan data (basis data = basis + data). Dalam sistem bilangan biner, kita dapat menuliskan beberapa contoh bilangan sebagai berikut.

0 → sama dengan 0 dalam sistem bilangan desimal

1 → sama dengan 1 dalam sistem bilangan desimal

10 → sama dengan 2 dalam sistem bilangan desimal

11 → sama dengan 3 dalam sistem bilangan desimal

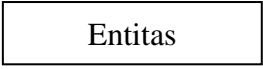
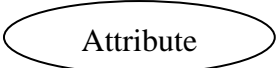


100 → sama dengan 4 dalam sistem bilangan desimal

## II.7. Entity Relationship Diagram

Menurut Edhy Sutanta (2011 : 91) *Entity Relationship Diagram/ER\_M* merupakan suatu model data yang dikembangkan berdasarkan objek. *ER\_M* digunakan untuk menjelaskan hubungan antara data dalam basis data kepada pengguna secara logik. *ER\_M* didasarkan pada suatu persepsi bahwa *real world* terdiri atas objek-objek dasar yang mempunyai hubungan/kerelasiaan antar objek-objek dasar tersebut. *ER\_M* digambarkan dalam bentuk diagram yang disebut dengan *ER (ER\_Diagram/ER\_D)*. Untuk menggambarkan *ER\_D* digunakan simbol-simbol grafis tertentu.

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. 1. Komponen ERD**

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

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

## II.8. Kamus 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. Basis data adalah suatu aplikasi terpisah yang menyimpan suatu koleksi data. Masing-masing basis data memiliki suatu API atau lebih yang lebih berbeda untuk menciptakan, mengakses, mencari, dan merplikasi data.

## II.9. 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. 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)

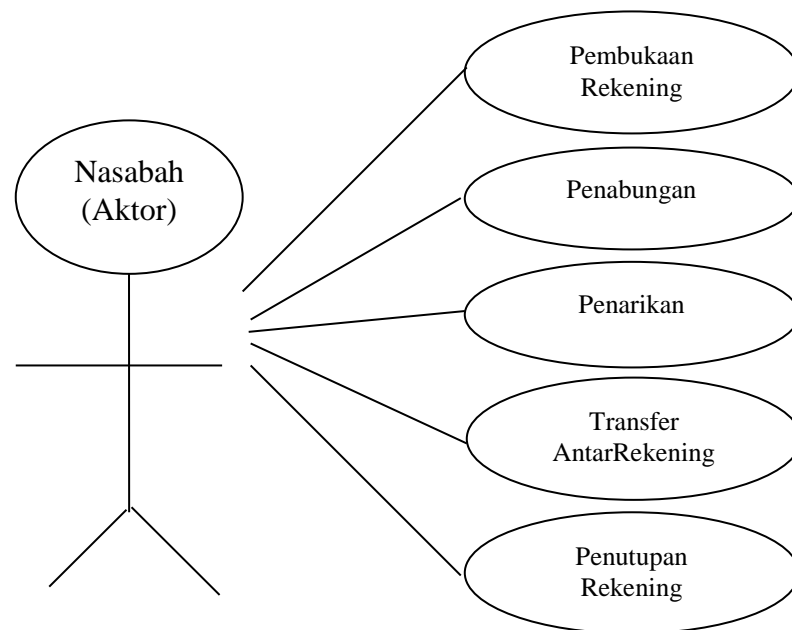
Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Bentuk

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.10. Unified Modelling Language**

### **II.10.1. Use Case Diagram**

Munurut Adi Nugroho (2009 : 7) 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. Adapun contoh *use case diagram* dapat dilihat pada gambar II.1. berikut ini :



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

### II.10.2. Class Diagram

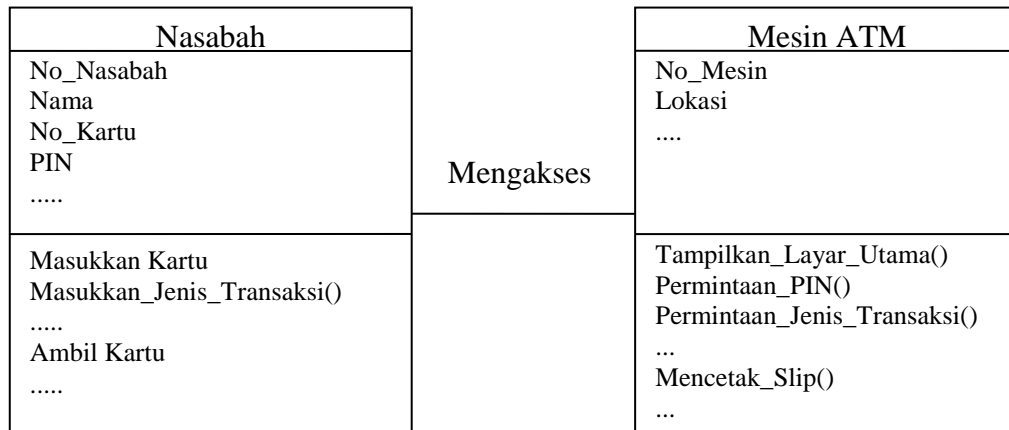
Menurut Adi Nugroho (2009 : 18) *Class* didefinisikan sebagai kumpulan/himpunan objek yang memiliki kesamaan dalam atribut/properti, perilaku (operasi), serta cara berhubungan dengan objek lain.

Menurut Adi Nugroho (2009 : 17) 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 :

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



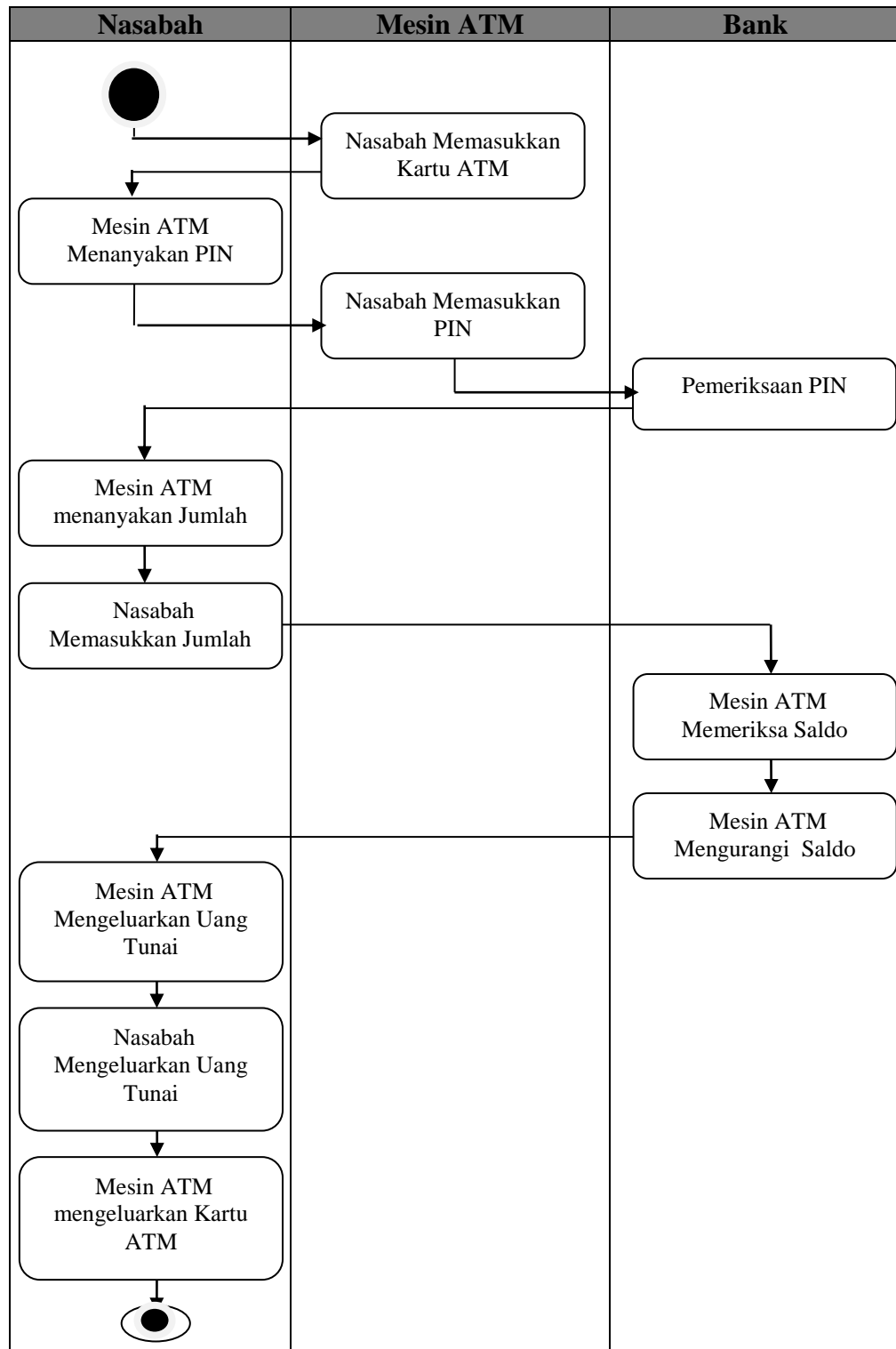
Adapun contoh *class diagram* dapat dilihat pada gambar II.2. berikut ini :



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

### II.10.3. 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. Adapun contoh *activity diagram* dapat dilihat pada gambar II.3. berikut ini :



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

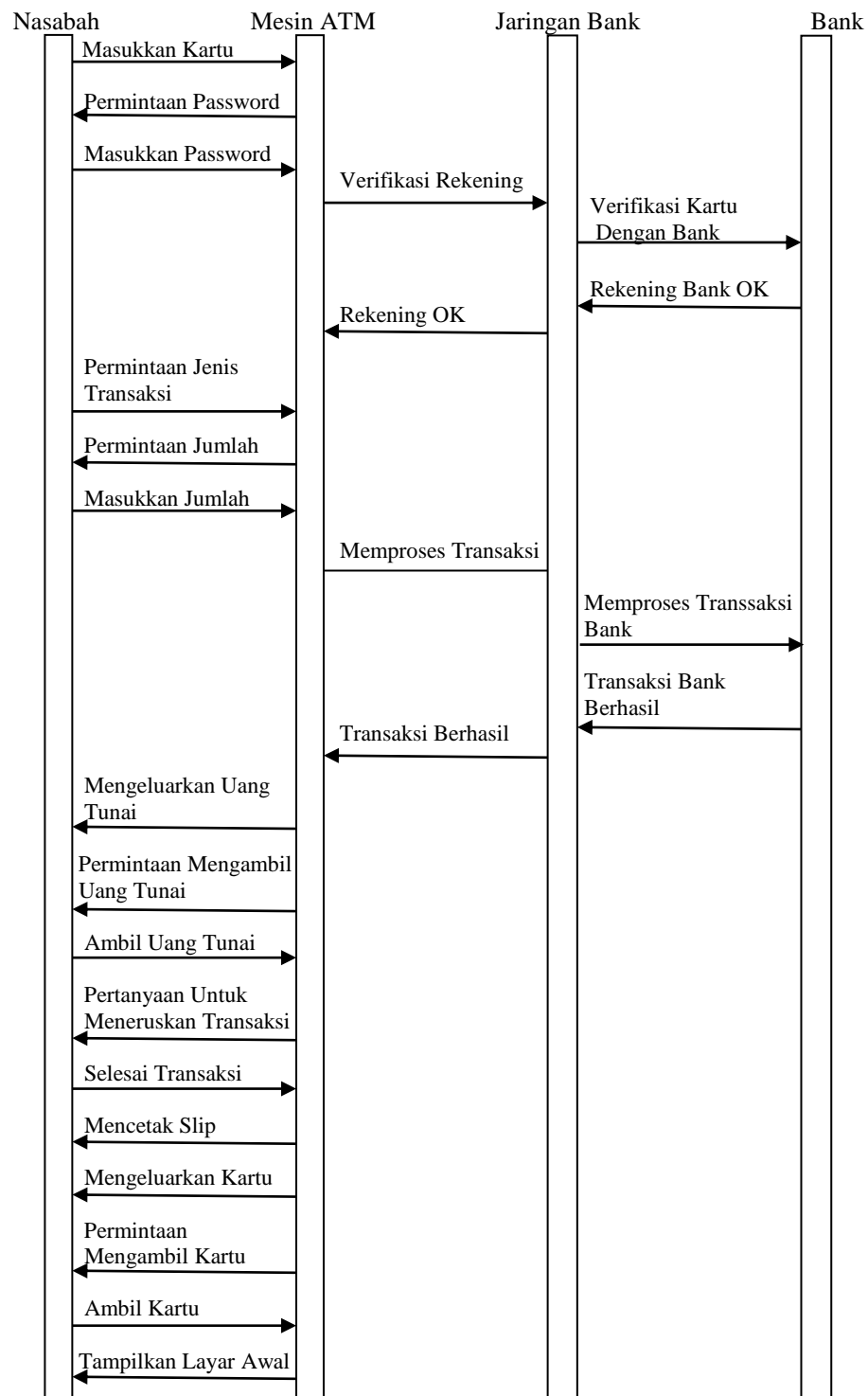
#### II.10.4. Sequence Diagram

Menurut Adi Nugroho (2009 : 34) 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. Adapun contoh *sequence diagram* dapat dilihat pada gambar II.4. berikut ini :



**Gambar II.4. Contoh Sequence Diagram  
(Adi Nugroho : 2009 : 36)**

### **II.11. 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.12. 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*).